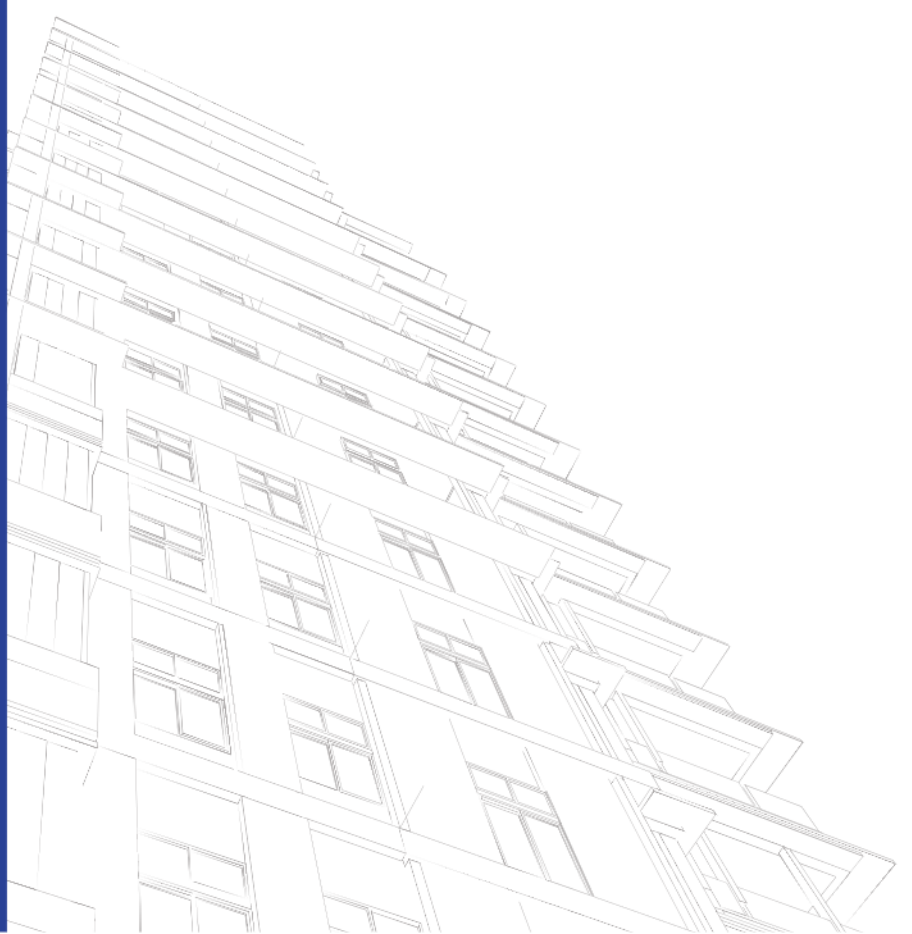
KENNA katalyst

THE ESSENTIAL GUIDE TO RISK-BASED VULNERABILITY MANAGEMENT

Six Steps to Setting Up Your Own Program



by KENNA Security

THE ESSENTIAL GUIDE TO RISK-BASED VULNERABILITY MANAGEMENT

SIX STEPS TO SETTING UP YOUR OWN PROGRAM

What is the Kenna Katalyst workshop? Kenna Katalyst was born from the inherent "gap" that exists in our IT and security communities as a way to create a meeting place where professionals can get actionable tips, industry best practices, and useful hands-on knowledge, as well as network with their peers. We wanted to provide the "katalyst" that helps attendees take their risk-based vulnerability management practice to the next level—free of vendor pitches. Over the next several months we will be taking these workshops on a tour across the country so be on the lookout for one in your community.

Introduction to Risk-Based Vulnerability Management 4

01 Get the Right Team and Skill Sets 9

02 Tools Needed for Vulnerability Management 15

03 Know Your Vulnerabilities and the Threat Landscape 19

04 Understand Your Assets, Applications, and the Risk Tolerance of Your Business 24

05 Bring It All Together, Measure and Evaluate 26

06 Communicate, Remediate, and Report 30

Introduction to Risk-Based Vulnerability Management

We all know that vulnerability management is difficult. But why? And how can you make it easier on everyone—security, IT, and DevOps?

This guide presents a solution to make vulnerability management, well, more manageable by helping teams focus on what matters most. To spend their time wisely and efficiently. And to have real, measurable impact to show for their effort.

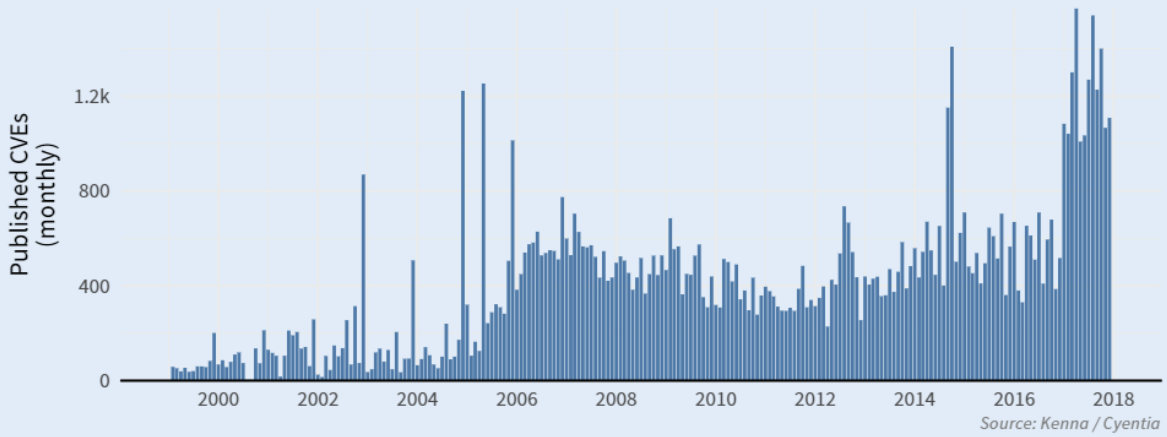
The key here is adopting a risk-based approach to vulnerability management. In the following we'll walk you through six key steps to get you started with a risk-based vulnerability management program. But first, let's look at some of the challenges of traditional approaches, which helps explain why vulnerability management is often so difficult.

The Number of Vulnerabilities is Overwhelming

The average enterprise has millions of vulnerabilities. And the number of vulnerabilities found each year just keeps growing. (Figure 1) This presents a classic dilemma for most organizations: a never-ending pipeline of work and not enough people to do it all.

FIGURE 1

Volume of published CVEs from 1999 through 2017



Source: Kenna / Cyentia

When this vast quantity is combined with the well-acknowledged cybersecurity staffing shortages, and the reality that an average organization can only remediate one out of every ten vulnerabilities in their environment in any given month, the task quickly becomes overwhelming, and the ability to ever get ahead likely seems hopeless.

Another challenge organizations often have with vulnerability management is that security goals can be at odds with business goals. IT and DevOps teams have their "day jobs" frequently tied to revenue-generating or mission-critical applications and services. Simply handing over a list of 10,000 vulnerabilities with no context is demotivating at best. In many cases, the asset owners are burdened with figuring out which vulnerabilities are their responsibility based on the device or application, then determining a timeline for remediation based on whether the vulnerability is considered a high, medium, or low priority according to its Common Vulnerability Scoring System (CVSS) score, the scoring provided by the vulnerability scanner, or the results of the application security test. It's a tedious process often performed manually

and with no real insights into the level of risk the vulnerability poses to the organization. What's needed is context so the asset owners understand why fixing a particular vulnerability quickly is important, and also why some vulnerabilities can be addressed later, when they have time to fit remediation into their normal work schedule.

The fact is any of these countless vulnerabilities has the potential of being exploited with some level of impact—maybe high, maybe low. But one thing is certain: no one can fix them all. Therefore, knowing which vulnerabilities pose a genuine risk of being exploited, and which ones do not, is essential.

Fortunately, there's data available to help. And because of that, you don't need to fix them all, or even most of them—at least not today. The reality is that, statistically, only two to five percent of vulnerabilities (specifically, CVEs) are actively exploited in the wild. This means that your security and IT teams don't actually need to focus on all of your vulnerabilities, they can de-prioritize most of them to focus on just those that pose the most risk.

So the real question is how do you identify these riskiest of vulnerabilities so that you're fixing the right vulnerabilities—at the right time? The answer: risk-based vulnerability management.

Focusing on Risk is Key

Chances are if you're reading this guide you already have a general understanding of the value of using risk as the basis of your vulnerability management strategy. However, we'd be remiss if we didn't take you through a few of the more common vulnerability management strategies, and explain why your strategy should focus on risk.

7 Reasons Why Risk-Based is Superior to Other Vulnerability Management Strategies

1. You can't remediate everything, and don't need to, but can remediate the riskiest stuff
2. Not all assets or applications are created equal
3. You can get the full picture from multiple feeds and tools
4. You can take advantage of automation
5. You get the ability to report to management on something they actually care about—risk
6. It avoids being distracted by the hype around high-profile breaches
7. A risk-based approach increases collaboration among all teams

The Research

Throughout this guide we will be citing numbers and other statistics to really give you, the reader, a fuller picture of the state of vulnerability management and any findings that give insight into best practices. Unless a citation is specifically referenced, all of that data is the result of an ongoing research initiative between Kenna Security and the Cyentia Institute. This research currently encompasses four volumes of results and spans the review of billions of vulnerabilities across hundreds of organizations, and, more recently, the survey results from ~100 Kenna customers.

All four reports can be found here: ***www.kennasecurity.com/research/prioritization-to-prediction-reports***

A Wing and A Prayer No Longer Flies

There was a time when organizations would tackle vulnerability management with a program that sat somewhere along the "...spectrum of intuition to alchemy." Many used, and still use, the prevailing wisdom of patching anything above a certain threshold, such as CVSS scores of seven or above. Some continue to use huge spreadsheets of vulnerabilities that they then sort based on intuition, the newsworthiness of a vulnerability, or the number of assets affected. Unfortunately, none of these are effective, much less efficient, strategies.

Why You Should Look Beyond CVSS

As mentioned, patching vulnerabilities that have a CVSS score of a certain range is a very common way to sort and prioritize which vulnerabilities to fix first. However, there are some inherent problems with ranking vulnerabilities based on CVSS scores. First, it's a static scoring method. Most of the vulnerabilities in MITRE's Common Vulnerabilities and Exposures (CVE) list receive a CVSS score within a few weeks of their discovery, before any exploits are written against them, so they're scored based on the initial assessment of their potential to be exploited, and then rarely—if ever—updated.

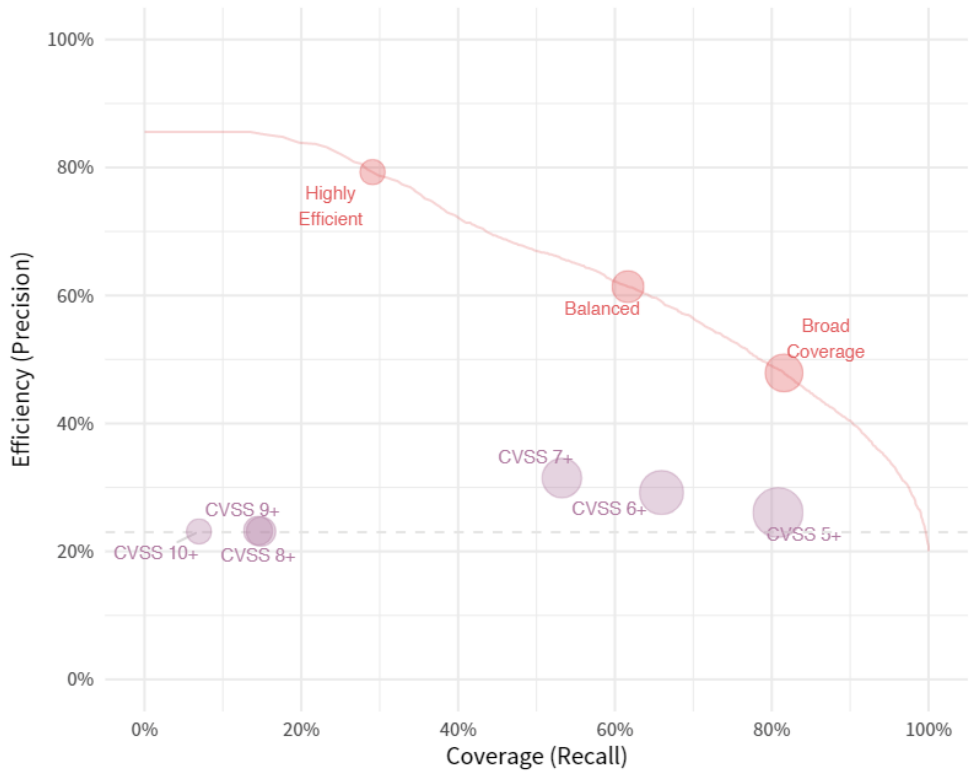
Another major issue with CVSS is that it lacks any degree of context. That is, the scores are developed based on the code itself, and its potential for being exploited. No consideration is given to the prevalence of the vulnerability in actual network environments, whether or not exploits are published against them, or any other contextual information required for the security analyst to truly understand the level of risk. If the initial assessment of a particular CVE determines that it's a low risk, it's assigned a correspondingly low CVSS score. Even if months later dozens of exploits are published against it, and some of those exploits even lead to hundreds of successful security events, the CVSS score will remain at its initial low level.

Comparing Vulnerability Management Strategies

Using a risk-based vulnerability management strategy built on a predictive data science model provides a much better way to prioritize which vulnerabilities to address, and in what order, than simply basing remediation on CVSS scores. As shown in Figure 2, the predictive model vastly outperforms CVSS (Note this data is taken from research completed by our team, in conjunction with the Cyentia Institute, as mentioned earlier.)

FIGURE 2

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



Source: Kenna / Cyentia

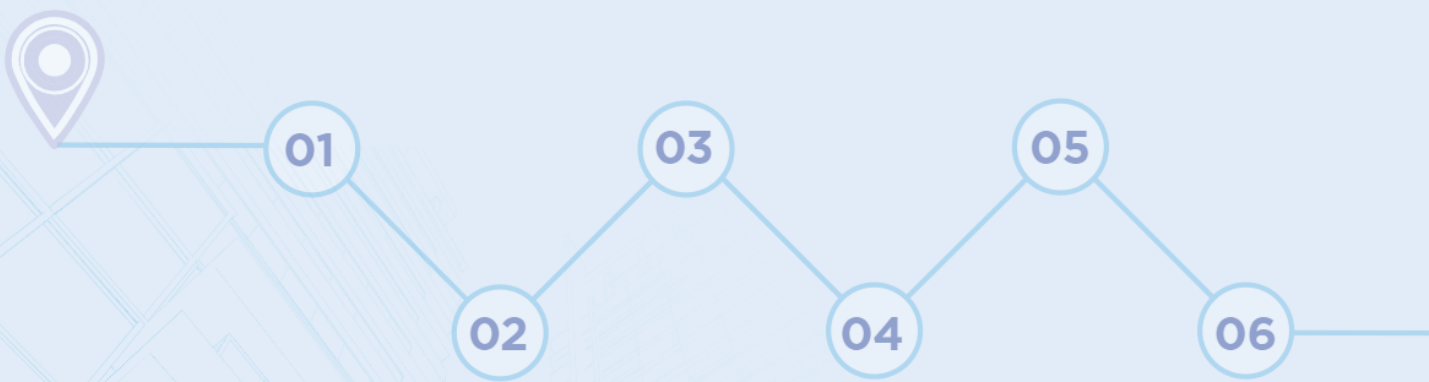
In this graph, the top line represents a predictive model, and the bubbles below represent the CVSS scores as noted. For our purposes, "Coverage" is the equivalent of how many vulnerabilities are remediated and "Efficiency" is how many of those vulnerabilities that were remediated should have been remediated; i.e, how many of those vulnerabilities would have been exploited.

Compared to a strategy of remediating all CVEs with a CVSS score of 7 or more, the predictive model achieves twice the efficiency (61% vs.31%), half the effort (19K vs.37K CVEs), and one-third the false positives (7K vs.25K CVEs) at a better level of coverage (62% vs.53%).

Where to Go from Here

So what does it take to get to a strategy that focuses your remediation efforts on the highest-risk vulnerabilities first, using this high-performing predictive model? You need the right teams. The right tools. The right data. The right data science. And you need to bring it all together in the right way. In this document we provide guidance gleaned from a number of our own security experts with years of experience in vulnerability management and risk-based vulnerability management.

Let's Get Started



01
Get The Right Team and Skill Sets

Vulnerability management is a team sport. To be a winning team, you'll need a roster of qualified people with the right skills. We fully acknowledge that the structure of the team or teams responsible for remediating vulnerabilities and the associated skills will be unique to your situation. However, there are some universal truths we have found in our time helping teams in this area.

In this step, we provide tips for organizing your teams and advice on the skill sets required to produce the best results when tracking down and remediating vulnerabilities on infrastructure assets and within applications. We do so, however, recognizing the growing workforce shortage in cybersecurity—and we will also propose ways to potentially tackle that thorny issue.

While we cannot give you any suggestions of team size—and, indeed, the size of the team involved in vulnerability remediation will depend entirely on the dictates of your organization—our research recently found that team size did not necessarily correlate to any performance gains or losses. However, the same research suggests that organizations should have clear separation of duties: security teams who find and rate the vulnerabilities requiring remediation, and the IT (or development) teams responsible for fixing the vulnerabilities. Following this practice, according to our research, leads to a mean time to remediation (MTTR) that is a month and a half shorter. (Figure 3)

FIGURE 3

Test comparing VM team structure with remediation velocity

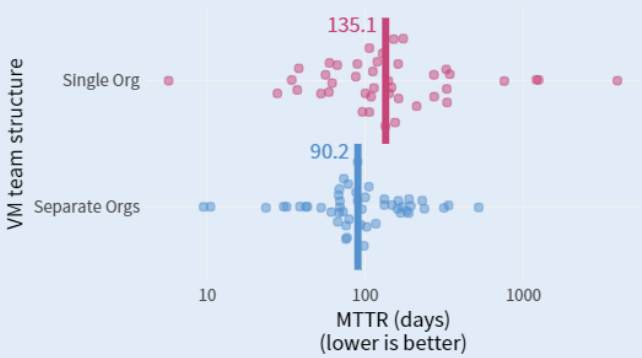
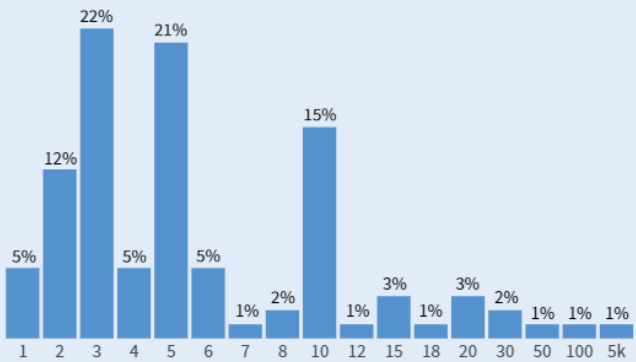


FIGURE 4

Participants in remediatioon

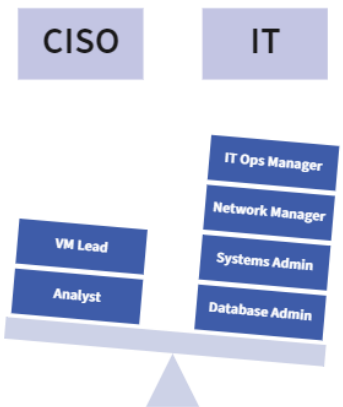


As you might expect, we'll suggest separate teams in the following. We also separate out our guidance into infrastructure management and application security management given the differences between the two.

Today's Security Executive

Larger organizations will very often have a chief information security officer (CISO). In fact, Cybersecurity Ventures predicts that 100 percent of Fortune 500 and Global 2000 companies will have a CISO or equivalent position by 2021, which is up from 70 percent in 2018[1]. If the title is not CISO, then there is an executive overseeing the digital security of the organization, be it the VP of security, chief security officer (CSO), or CISO. For ease of reading we'll assume that the security executive at the top of the organization is a CISO, but feel free to insert whatever title fits your organization.

Today's CISO has more on his/her shoulders than ever before. Given the increased awareness of the importance of cybersecurity at the highest levels of the business, the CISO at a larger organization has multipronged responsibilities. One is having the technical chops for items such as managing vulnerabilities and cyber risk, protecting from cyber threats (and mitigating the impact of same), managing security technologies, and overseeing compliance, budgeting and strategy. But they also represent the diplomatic interests of the security organization, both within the larger business and among executive management. There is a high likelihood the CISO will get asked to present either to the CIO or the executive staff, and even the board. We'll give some guidance for having these conversations in step 6, and we'll also talk more about what that reporting might look like for vulnerability management.



Infrastructure Vulnerability Management

Under the CISO you'll have the vulnerability management security function. The roster on the team hopefully consists of a vulnerability management director or lead and then an analyst or analysts (titles will vary from organization to organization). Teams are responsible for identifying, tracking, and assessing vulnerabilities that may exist across hundreds, thousands, and even tens of thousands of assets across the enterprise. This includes every piece of hardware and software—computer systems, storage, networking—in your data center(s) and across all remote locations.

The vulnerability management team must understand the threat landscape:

- Does exploit code exist for the vulnerability?
- What is the likelihood of this vulnerability being exploited?
- Is there evidence of it being exploited in the wild?
- What is the impact on the business if the vulnerability is exploited?

All of these factors combined enable the team to assess the risk of a vulnerability and prioritize it for remediation.

Typical Vulnerability Assessment Process

"In order to make an assessment about a vulnerability or finding, a security analyst will typically visit 5 – 10 sites, read about the vulnerability, look through threat intelligence reports, look for signature triggers or exploits in whatever data is available (open or closed source), consider the technical details of the vulnerability, and make an assessment." [Multiply this by every vulnerability in your enterprise]

Infrastructure Vulnerability Remediation

As mentioned, the team tasked with remediating infrastructure vulnerabilities is usually separate from the team assessing them, and is usually part of the IT organization. For this reason, it is important to have a universal language around risk and processes that can easily be shared across both the security and the IT organizations.

The remediation team may include system and database administrators, IT operations staff, network managers, etc. There are generally many more people on the remediation side than on the assessment side. However, security is not the main task of IT, it's only a small subset of their overall job, so the more targeted and impactful security can make their efforts, the better. Even so, as mentioned earlier, typically an organization has the capacity to address only about one in 10 vulnerabilities in a month.

Depending on their technology stack, some organizations have well-established remediation cycles for certain elements of their technology—for example, after Microsoft vulnerabilities are released on Patch Tuesday, then an organization may have automated systems to push those patches. This makes it relatively easy to patch those elements of the infrastructure. In addition, the majority of teams have set service-level agreements (SLAs) or deadlines for remediation. Here an example is, say, all high-priority vulnerabilities must be remediated within a month. Our research found that teams with defined SLAs (rather than ad hoc or non-existent schedules) are the ones addressing vulnerabilities more quickly. You'll see in Figure 6 that those with defined SLAs have a lower area under the curve (AUC), a term from the Prioritization to Prediction research reports that indicates they remediate vulnerabilities faster. Interestingly, we also found that the specifics of the SLA matter less than just having them.

Median Time to Remediation

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.

[Source: "Prioritization to Prediction, Volume 2: Getting Real About Remediation," Kenna Security & the Cyentia Institute, 2019]

FIGURE 5

Are remediation deadlines defined?

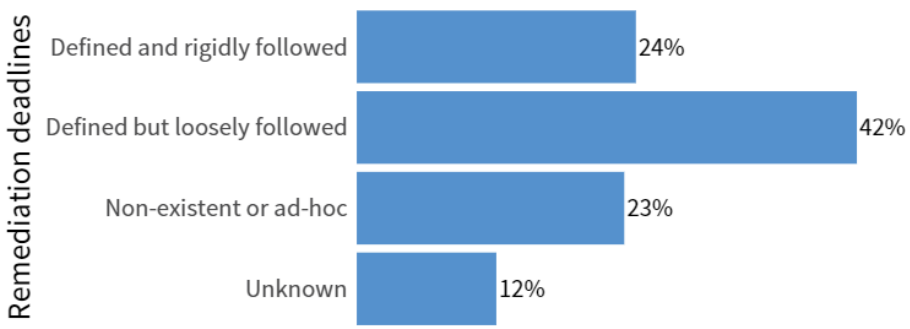
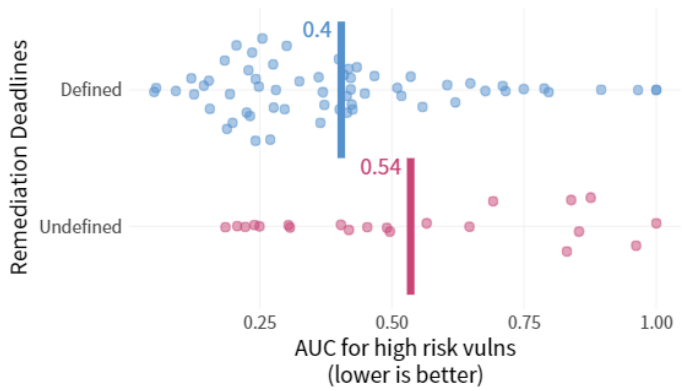


FIGURE 6

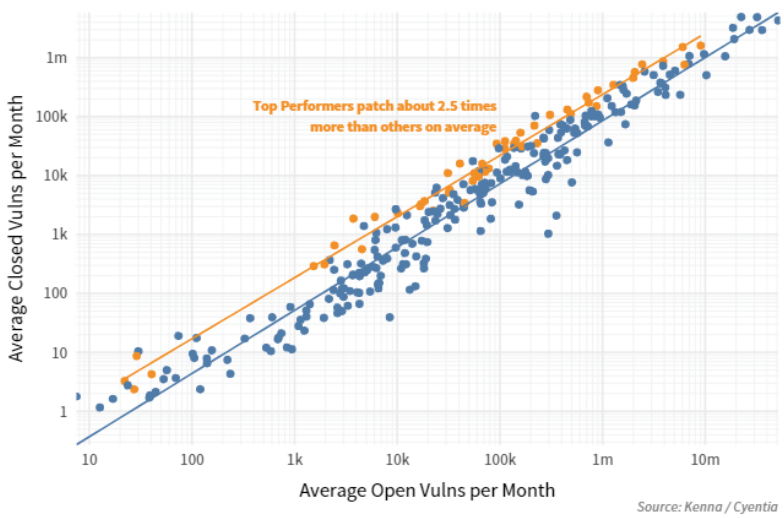
Tests comparing defined and undefined SLAs on remediation velocity



Still, even with defined deadlines, the vast majority of organizations struggle to keep up with an ever-growing volume of vulnerabilities. Even vulnerabilities identified as high-risk—very likely to be exploited and having potentially damaging impact—may take a team 90 days or more to fix. Often, under such circumstances, the team may have to implement a mitigating control, such as a new intrusion rule on the firewall, until full remediation is completed. If that feels like lots of bad news, the good news is, firms that focus on risk to guide their remediation efforts can get ahead of the number of net new high-risk vulnerabilities that pop up in their environment over time. That's what this guide is all about helping you achieve. As you can see in Figure 7, While organizations by and large can only remediate 1 in 10 vulnerabilities in a month, the top performers exhibit 2.5 times the remediation capacity of others.

FIGURE 7

Ratio of open to closed vulnerabilities per month for top performers vs. other firms

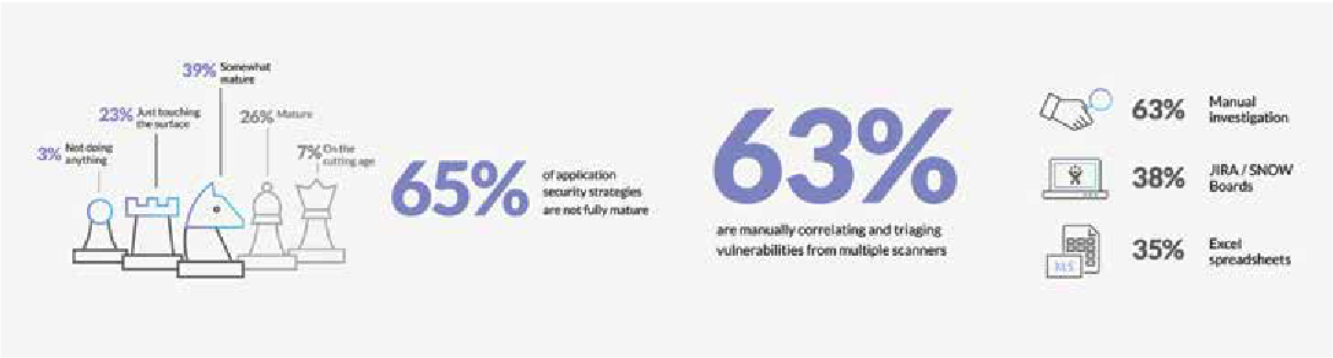


Application Vulnerability Management

Now let's talk application security (AppSec). If your AppSec vulnerability management efforts are still in their infancy, don't worry, you're not alone.

FIGURE 8

Cybersecurity Insiders found most AppSec programs are not fully mature and still manually investigate vulnerabilities[2]



You may still be building your team, and as you do, here is some basic guidance. Application vulnerabilities are typically managed by the AppSec team, a different part of the larger security organization that generally also reports up to the CISO. Much like the infrastructure team, as part of their overall responsibility, the AppSec team is responsible for gathering data on vulnerabilities, assessing them, and then prioritizing them for remediation. To do this, they need to work hand-in-hand with the development team or teams that create the applications because the developers are typically the ones who will remediate those vulnerabilities.

The good news for AppSec teams is that there are fewer vulnerabilities than in the infrastructure. But application vulnerabilities will take longer to fix given they will require the development team to come up with a custom fix. Because development teams or DevOps teams, depending on your organization, are laser focused on developing applications, usually ones that impact the bottom line, it's vitally important that they also be laser focused only on remediating those vulnerabilities that reduce the most application risk.

The application security team must know whether an application vulnerability:

- Is being actively exploited
- Affects the most critical applications (especially those with sensitive data)
- Occurs prior to login
- Affects important apps that don't require login

In addition, AppSec teams have the unique challenge of having to bring together the data on vulnerabilities from many different tools used throughout the application development lifecycle. These tools often sit in and roll into different areas of responsibility, yet spit out similar results. In the following step we present more on the tools themselves. It is also important to employ a solution that will correlate, normalize, and de-duplicate the information from these myriad tools to prioritize vulnerabilities based on risk.

Application Vulnerability Remediation

Again, similar to infrastructure vulnerability management, application vulnerabilities go to a separate team for remediation, in this case, the developers who created the applications. It's important to understand that your dev teams can't and shouldn't just try to fix everything up front, as every moment spent on security is time spent away from their core function of getting business-critical applications out the door. This is especially true given that the time for an application vulnerability to be exploited keeps getting smaller and smaller—as shared with us by our partner Sonatype, we're now at an average of just three days[3]. With only three days before an exploit, it's unlikely you'll execute a patch in time. That's why automation is key, where the information about the vulnerability and its fix have already been pulled into the tool and quickly distributed between AppSec and dev.

Waterfall and Agile

Application security may be handled differently depending on the development methodology practiced by the organization. For example, in a traditional waterfall development process that may release software updates only once or twice per year, the opportunity to identify and assess vulnerabilities can be more limited. In waterfall, security is at the end of the process, often after release—either delaying release or leaving additional opportunities for exploitation given that vulnerabilities are then fixed after the applications are live.

Iterative and incremental development processes, such as Agile, open opportunities for more frequent remediation. However, building remediation practices into established development cycles can be challenging due to the potential impact on release schedules. Agile development can produce software releases daily and even multiple times per day. The challenge is integrating testing tools into the rapid release cycles and having sufficient security checks to stop deployment of software that isn't secure and yet not standing in the way of the benefits of the fast release cycle.

Special Section: Tackling the Cybersecurity Workforce Shortage

By now we've all heard that there are not nearly enough skilled security practitioners to meet the demand. This is a thorny issue that many are trying to propose solutions for. The true solutions are beyond the scope of this guide, but if your team is struggling to keep up, there are a few things that we'd like to share. First, our research has uncovered that well-funded teams remediate vulnerabilities significantly faster than those who have less-adequate budgets. Hopefully, this will help you request additional resources. In addition, there are a number of vendors out there who can help amplify your team.

02
Tools Needed for Vulnerability Management

To do any job the right way, you need proper tools. Vulnerability management is no different. However, the tools vary depending on whether you're focused on infrastructure or application vulnerability management. In this step, we break down the basic tools you need, and explain how a risk-based vulnerability management solution works in conjunction with these tools.

First: Find The Vulnerabilities

Infrastructure vulnerability scanners

Vulnerability management starts with scanning your infrastructure assets—computers, storage systems, networks, operating environments—to uncover known weaknesses. You may choose to run the scanner on premises or opt for a software-as-a-service solution. Either way, you need at least one vulnerability scanner that can scale to your enterprise.

Vulnerability scanners have become basically interchangeable, and we do not recommend one vendor over another. All asset scanners perform essentially the same function. However, some organizations choose to run more than one scanner as a cross-check in case one misses a vulnerability that the other vendor's scanner could catch.

Application Security Tools

Uncovering vulnerabilities in applications requires a separate set of tools. To ensure you evaluate the widest breadth of vulnerabilities, if possible, we recommend the following set of tools.

01 SAST, DAST & IAST

An application security tool set generally starts with **static application security testing (SAST)** to analyze applications at the code level for vulnerabilities. SAST tools are an essential part of any AppSec program, and all pretty equal today, so we don't have any recommendations on how to choose one over another. But note that SAST is notoriously slow and known for having a high level of false positives. Therefore, many development organizations perform **dynamic application security testing (DAST)**, which scans a running application. Also pretty equivalent across vendors, DAST can uncover vulnerabilities that SAST may miss, but it comes late in the development process—typically post-production. DAST also only assesses a small percentage of the total attack surface. This is why we advocate using both SAST and DAST scanners.

Somewhere between SAST and DAST you'll find **interactive application security testing (IAST)**, a relative newcomer. This tool sits inline with the application, watches traffic, and identifies vulnerabilities. IAST has fewer false positives and more false negatives than SAST, but is also harder to set up. Compared to DAST, IAST finds vulnerabilities earlier in the software development lifecycle, but it has more false positives and fewer false negatives. We still recommend both SAST and DAST, but IAST could be an interesting compromise between the two.

Build Out a Robust Set of Tools

While SAST, DAST and IAST are a great place to start when building an application security tool set, to really understand the application vulnerabilities in your environment, there are several more layers to add.

02 Pen (penetration) Testing

Pen (penetration) testing is standard practice in most development organizations. It is an effective way to uncover application vulnerabilities that other security tools may have missed, especially because it simulates a real-world cyberattack performed by professionals. How often you perform a pen test can vary based on your company size and budget, and could be dictated by regulatory compliance requirements. Regardless, it is a critical component to ensuring application security.

03 Bug Bounty

Bug bounty programs are the new kids on the block. A bug bounty program compensates individuals for finding bugs and reporting them to the organization. The idea is to uncover vulnerabilities before they become known to the general public, where they would run a much greater risk of being exploited. The number of bug bounty programs continues to grow rapidly, with many bug hunters uncovering vulnerabilities that cannot easily be found using automated tools. Bug bounties are often used by organizations to discover flaws in their external web applications and systems and even in products such as Internet of Things (IoT) devices.

04 SCA

Given the widespread use of open source software—comprising 80–90 percent of most custom applications according to some studies[4]—you will also want to include **software composition analysis (SCA)** in your AppSec toolbox. An SCA tool breaks down open source into its prime components, identifies each component in use, and then runs a check of each component against a database of known vulnerabilities. It's important to note, however, that SCA tools are only as good as the database they link to. Therefore, we recommend choosing an SCA tool that reaches far beyond the information available in public databases to the entire universe of open source vulnerabilities, and leverages AI and machine learning to draw actionable intelligence from those sources.

05 What About Scanning Containers?

Many organizations today are moving to containers as an efficient way to develop and deploy microservices using container engines like Docker, CoreOS, Joyent, Rancher, and others. Containers, as the name implies, are self-contained components of application logic, and may contain one or more microservices. While containers are "sealed up" units of application components, they are not impenetrable, and present a rich attack surface for cybercriminals.

Especially as companies move more workloads into the cloud, they are taking advantage of the efficiencies and agility offered by containers and microservices. Interestingly, as the use of containers has grown, the number of vulnerabilities within them has also grown substantially. In fact, Kenna's own research found that one-fifth of the most-used Docker containers have at least one critical vulnerability[5]. This would warrant immediate attention by any diligent security team. The key is finding the vulnerability in the first place.

However, traditional asset scanners cannot scan containers—a specific container scanner is needed. Good news: some of the leading asset vulnerability scanner vendors now also offer container scanners.

What to look for in an SCA tool

- ☐ Verifies all embedded dependencies in deployed apps
- ☐ Examines fingerprints, not just file names and package manifests
- ☐ Draws from complete universe of open source vulnerabilities
- ☐ Uses machine learning to derive actionable intelligence from vulnerability

Application Security Tools Checklist

- ☐ SAST
- ☐ DAST
- ☐ IAST
- ☐ Pen Test
- ☐ Bug Bounty
- ☐ SCA
- ☐ Container Scanner

Next: Figure Out Your Assets

Configuration Management Database

Another essential tool for any vulnerability management program is a configuration management database (CMDB). A CMDB provides detailed information about all the hardware and software assets in your organization, such as configuration, ownership, network security zone, etc. The CMDB is crucial for determining how critical an asset vulnerability is based on where it fits in the business and its relationship with other assets. This information is fundamental to establishing a risk-based vulnerability management program.

Add Threat and Exploit Intelligence

Real-time threat and exploit feeds are essential to understanding what is currently being exploited, and to what degree. This is important information that allows security the insight needed to factor attacker behavior into their prioritization. And let's not forget intelligence on zero day exploits. That is another area of valuable insight. We will discuss what you need to understand about the threat landscape in detail in the next step.

Then: Prioritititize Remediation

Now that you have the data, you'll need to make sense of it. This is where you'll want a vulnerability management platform that correlates all of that data, and then tells you which vulnerabilities pose the greatest risk and should be remediated first.

Disseminate the Intelligence

Knowing which vulnerabilities require immediate remediation is just the first step of the process, you will then feed this information into your remediation workflow. This is typically achieved by using a ticketing system to send tickets to IT, then track the remediation process through to completion. In this step of the process, you should disseminate the intelligence security has gathered on the vulnerability to IT, so IT will understand what to fix, how to fix it, and why it's a priority. This will not only facilitate better teamwork and collaboration between the two groups, but will save IT the hours of time required to re-investigate and validate the vulnerability, search for the solution, etc. Once the ticket is closed, you will want to rescan to ensure the vulnerability is truly resolved.

03
Know Your Vulnerabilities and the Threat Landscape

In the last step, we discussed the tools you need to identify where vulnerabilities exist in your infrastructure and applications. In this step, we broaden the view: What vulnerabilities have been identified in the wild that may not have shown up yet in your organization? Which of those vulnerabilities have been exploited to date? What is the larger threat landscape? Answering these questions provides another vital data set and context for determining the risk of vulnerabilities within your organization.

The fact is only a very small percentage of published vulnerabilities (specifically CVEs)—**only two to five percent**—are actually exploited in the wild. It is important to understand what attackers are doing because that determines the likelihood of a vulnerability being exploited. This is key to operationalizing risk-based vulnerability management.

How do you find out what attackers are doing? Threat intelligence can be obtained through internal sources such as your intrusion detection/protection system (IDS/IPS), antivirus software, and local honeypots. In addition, external sources are essential to gaining a complete picture of the threat landscape. These include threat feeds, threat exchanges, and online chatter.

You also need a complete understanding of the assets and known vulnerabilities in your organization so you can cross-reference them with the threat intelligence. This will help you determine the impact of a potential exploit—another key factor in determining risk. As discussed in step 2, you have data from your vulnerability scanners and CMDB. But this information is often only part of the story. To get a complete picture of how important your assets are, you should pull in data from your identity and access management (IAM) infrastructure, have discussions with your finance people, and understand which teams are plugging into public clouds. There are many sources, and that is one of the big challenges.

To get started cross-referencing the threat landscape with the knowledge you have of your assets and vulnerabilities, you can use tools such as these developed by MITRE: CVE lists, CPE (common platform enumeration) and CWE (common weakness enumeration) information, as well as ATT&CK, which provides a taxonomy around attacker behavior. This is useful to cross-reference what is possible on an asset with what is actually happening in the broader threat landscape.

Sources of Useful Intelligence

(if you can, we suggest that you use all of these sources)

- ☐ Open source intelligence (intelligence feeds)
- ☐ Intrusion detection systems
- ☐ File-ordered antivirus analysis APIs
- ☐ Cloud-based honeypots
- ☐ Local honeypots
- ☐ Endpoint protection systems
- ☐ Zero-day threat intelligence

Zero-day Vulnerabilities Present A Special Challenge

A zero-day vulnerability is simply a vulnerability that has not been released to the public. Clever attackers find these unpublished vulnerabilities and exploit them before a patch is available. Mitigating this threat requires an advanced intelligence engine to identify a zero-day so you can patch it before it is exploited.

Important Note: gathering all this information is not a matter of "once and done." You then have to repeat this process continuously, because the threat landscape shifts all the time.

The Case For Prioritization

As noted earlier, there are many sources of vulnerability and threat information, which means there is even more actual information you will then have to sort out—far more than any organization could realistically handle. This makes a strong case for prioritization.

Consider this: there are over 110,000 published CVEs, but just over one-fifth of them have known exploits. Moreover, as highlighted above, less than 2 percent of those CVEs are ever exploited in the wild. Therefore, it makes logical sense to focus remediation efforts on those vulnerabilities *most likely to be exploited* rather than expending resources on all the others.

However, there are still other factors to consider. Say one of those "high-priority" vulnerabilities—one that has been exploited in the wild—exists on a non-critical system within your organization. The impact to the business of that vuln being exploited would be very low. So, is it truly a high priority for you? Or suppose you prioritize remediation on only those vulnerabilities that have published and observed exploits in the wild, delaying those that just have exploit code publicly released. What are the chances an attacker decides to execute that exploit code in your organization? Our research has shown that when exploit code is published, the odds are seven times higher that we will see exploitation in the wild than without published exploit code.

Ultimately, what we are asking is this: "What is the true risk of a vulnerability to your organization?" Answering that question is extraordinarily complex.

The Vulnerability Lifecycle

Creation – A vulnerability is created when flawed code is written and released in a vulnerable state

Discovery – Someone learns that a vulnerability exists

Disclosure – A vulnerability has been reported to CVE Numbering Authorities (CNA) and assigned a CVE ID

(Note: other sources of vulnerability disclosures exist)

Exploit Code – Proof-of-concept or working code is published for exploiting a vulnerability

Exploitation – Attacks targeting a vulnerability in the wild

Detection Signature – Tools with sensors identify exploitation in the wild based on exploit signatures

Four Key Factors to Consider When Prioritizing Remediation of a Vulnerability

Key consideration #1: Is the vulnerability useful (remote code execution (RCE), in general)?
This is what attackers dream of. If you have a vulnerability that allows an attacker to execute malicious code remotely, they can utilize a variety of different payloads, making the vulnerability something that's useful in many contexts, from wreaking havoc by wiping a disk to gaining an administrator account and pivoting. RCE is the best, in the attacker's eyes.

Key consideration #2: Does the vulnerability have an exploit in a widely used toolkit or Exploit Database?
Pen testing tools such as Metasploit and the Exploit Database can certainly be an asset in helping you determine your critical vulnerabilities. However, attackers can also use both to find—and exploit—weaknesses in your organization's defenses. Therefore, any vulnerability identified with a Metasploit module or identified in the Exploit Database should be a top priority for remediation. Also consider blackhat exploit kits. These are almost always used for malicious intent and should, therefore weigh heavily in how you prioritize remediation.

Key consideration #3: Is the vulnerability widely available (prevalent)?
Attackers are pretty darn lazy. If given the opportunity to use a known (and well-tested) exploit, they will. Vulnerabilities that are extremely prevalent (think: Java, Office, Reader or Flash as examples) are simply more likely to be used by attackers.

Key consideration #4: Is the vulnerability seeing active exploitation?
Check your threat feeds and determine if the vulnerability is something that is either being exploited as seen on public feeds or through your risk-based vulnerability vendor's internal database.

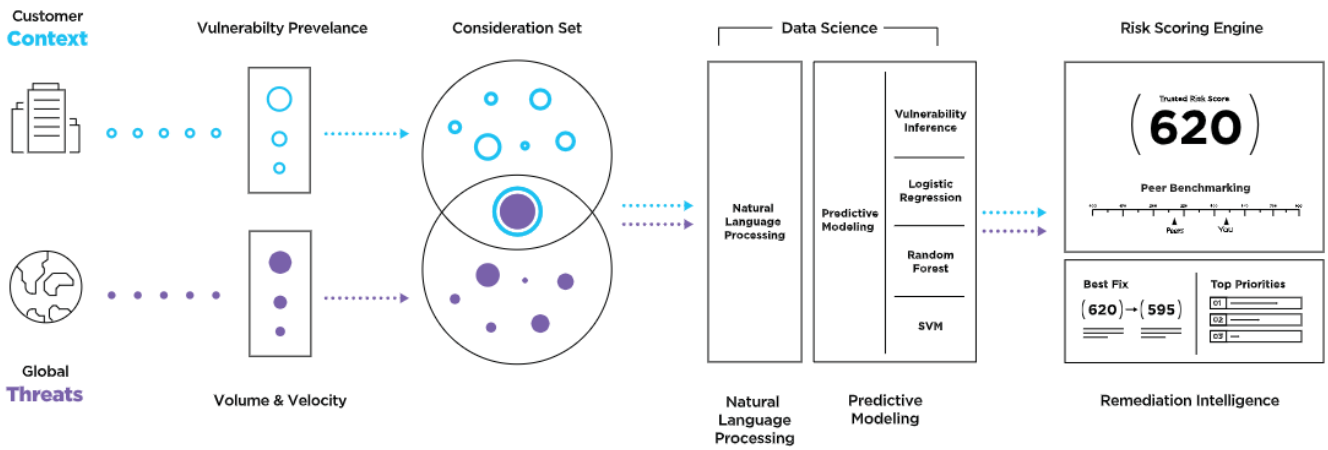
How to Analyze All the Threat and Exploit Data

Once you collect detailed information on your vulnerabilities and have established feeds for threat intelligence, the hard work of analyzing the information begins. From all that data, how do you determine which vulnerabilities present a genuine, clear and present danger to your organization? That is, which ones pose the greatest risk, and therefore, should be the highest priority for remediation? This gets to the essence of a risk-based vulnerability management program.

Large enterprises may have the resources to staff threat intelligence teams to do the necessary data analysis. But most organizations will not have this capability in-house, which makes a strong argument for implementing a tool that can automate the process. The Kenna Security Platform, for example, analyzes billions of pieces of threat and exploit data (see list on page 23), providing data-supported insights into the potential impact of each vulnerability. This includes the volume and velocity of attacker activity, as well as the criticality of each threat given your specific environment.

In addition, Kenna correlates all the security data from the internal and external sources to determine the relevant consideration set for your organization. We then run through our proven data science algorithm to deliver a risk score for every vulnerability; and our remediation intelligence determines which vulnerabilities pose the greatest risk to the organization and whose remediation will have the maximum impact. (See Figure 9)

FIGURE 9



Technology like this provides an automated way to continuously keep up with constantly evolving threat and exploit data, and to put it into an easily digestible context that helps you understand vulnerability risk. From this context, you then gain actionable security intelligence for guiding remediation prioritization and resource allocation.

What Kenna Security Analyzes

Kenna's Knowledgebase

- 15+ billion security events
- Global attack telemetry
- Remediation intelligence
- 3+ billion managed vulnerabilities

Threat Intelligence

- AlienVault OTX
- AlienVault Reputation
- Secureworks CTU
- Emerging Threats
- ReversingLabs
- Exodus Intelligence
- Sans Internet Storm Centre
- X-Force Exchange
- Other open and closed sources

Exploit Intelligence:

- Metasploit
- Exploit DB
- ReversingLabs
- Secureworks CTU
- Black Hat Kits on rotation (AlphaPack, Blackhole, Phoenix, more)
- Proofpoint
- Exploit DB
- Other open and closed sources

Internal Security Data Sources

- Any vulnerability scanner
- Asset- and network-specific data from configuration management database (CMDB) tools
- Penetration testing
- Bug bounty programs
- Static application testing
- Dynamic application testing
- Open source tools
- Custom data sources in JSON format

04
Understand Your Assets, Applications, and the Risk Tolerance of Your Business

To know your assets and applications is to truly know your risk. But how well does any organization actually know the details of their assets and applications to determine criticality, and thus, the risk they pose if compromised? Answering this question is an important step in establishing an effective risk-based vulnerability management program.

We touched on this subject in step 2 when referencing CMDBs. A CMDB is an essential tool for managing information about all the hardware and software assets in your organization. But a CMDB is only as useful as the details it contains. You need a complete and detailed asset inventory to realize the full value of a CMDB. That can be a daunting task, so it helps to break it down.

What Assets Do You Have?

First and foremost, you need to know what you have. There are automated tools to help with this discovery task, which will scan your organization to identify assets like servers, workstations, virtual machines, storage arrays, and network devices. You can also use an automated discovery process to understand your DNS and WHOIS records, identify IP address ranges and domains you own, as well as the ports, applications and services running on your assets. Some tools are specific to identifying internet-connected assets, and you will also need to identify assets in public cloud environments. Most of these tools are available as either cloud services or for download to run on premises.

You also need to understand how all of the components fit together. For example, looking at a web application your organization has built, you'll need to understand the source code written and the risk of same, the open source libraries being used and the risk of those, the risk of all of the infrastructure used to serve up that application, and take it all in context of the metadata noted. To really understand risk of an application, you'll need the context of all of those parts to understand it all.

The bottom line is the more information you have about your assets and applications, the more specific you can be when defining your risk tolerance levels.

Asset Metadata You Need To Determine

- Who owns the asset?
- What is the asset's function, and how is it used?
- Based on the asset's function, which is the most important: confidentiality, integrity, or availability?
- What's the impact of losing the confidentiality, integrity, or availability of the asset?

How Business Context Makes A Difference

Consider the following two applications. The first application, used internally for HR, contains sensitive information such as social security numbers, health care information, etc. The second application is a public-facing application that only contains public data. Without evaluating the size of the user base, the internal application processing sensitive data seemingly outweighs the risk of the public application. However, if the public application exposes each of its 100 million users to a persistent cross-site scripting vulnerability, you might think differently.

Know the Risk Tolerance of Your Business

To effectively determine what level of criticality and risk acceptance to assign an asset, you also need to understand the overall risk tolerance of your organization. The risk tolerance of a financial services company will be different from that of a grocery store chain. Risk will be viewed quite differently within a highly regulated industry than in a non-regulated one.

How do you begin to gauge where your organization stands? You can start by looking at other risk assessment activities within your organization. Find out if there is a risk manager for the business and collaborate with that risk management team. What is their methodology for assessing risk, and process for engaging with executive leadership to codify the organization's risk posture? You can then leverage this model for your vulnerability management program.

Deciding What Level of Risk to Accept

Risk tolerance comes into play when determining which vulnerabilities to remediate, and in what priority. For example, suppose you have a low-level vulnerability on an asset that poses little risk to the organization. It would be reasonable to deem that vulnerability a low priority for remediation. In other words, the risk of that vulnerability may be acceptable until the next patch cycle comes around.

How about a mid-level vulnerability? If it is on a low-risk asset, that vuln may not be a priority for immediate remediation. If it's on a high-risk asset with potential financial impact should it be exploited, then it may become a top priority. But does the risk outweigh the time and effort required to remediate the vulnerability? Or will remediation be so costly to the business that the risk is tolerated?

Critical vulnerabilities are usually prioritized high for prompt remediation. However, there can always be exceptions. Again, perhaps the cost to the business would be too great in terms of operational disruption. Instead of remediating the vulnerability, other risk mitigation measures are put in place. These are key tradeoff questions.

Some useful metrics to consider when determining the risk and the tradeoff:

- System susceptibility: value to attackers and vulnerabilities
- Time to compromise: the time it would take attackers to compromise any of the key controls for a given set of assets and applications
- Threat accessibility: access points and attack surface
- Threat capability: tools, resources, and techniques

The fundamental question in all of these circumstances is, "Who is empowered to accept the risk?" For low-level risks, that person may be a line-of-business owner. Mid-level risk might be a decision for the security team, but only with clear justification. Making such decisions requires all relevant stakeholders to be involved from the beginning of your vulnerability management program. It's important to understand each stakeholder's perspective and concerns. And it's essential for all stakeholders to buy into the program for it to be effective.

Considerations for Assessing Business Risk Tolerance

- Your organization's broad metadata including industry vertical, size, and geography
- The valuable and/or regulated information your applications or assets are processing
- How many people use an application
- The critical controls in place to protect confidentiality, integrity, and availability of key assets
- Your adversaries and their capabilities

05
Bring It All Together, Measure and Evaluate

In prior steps we established the information you will need to determine the risk of certain vulnerabilities on specific assets or applications. This provides the foundation for prioritizing remediation efforts, and a way to measure the effectiveness of remediation in terms of risk reduction in a tangible, quantitative way. Now you will need a way to pull all that data together, measure and evaluate it, and come up with a prioritized list of vulnerabilities for remediation—based on risk.

Leverage Machine Learning

Consider the expanse of data we've been discussing. You have reams of scanner and sensor data. Scores of vulnerabilities returned by SAST, DAST, IAST, SCA, and other application security tools. Vast amounts of threat intelligence. And all of this data must be correlated with yet more security data (billions of pieces, in fact) to provide the context necessary to determine the specific level of risk for each asset/application. This is no task for humans, even a team of highly skilled analysts. It requires leveraging modern technology, including automation and machine learning. Automation can perform in seconds what it would take human beings hours, or even days, to complete. In addition, automation can also scale to well beyond what a team of human beings could ever do.

By leveraging machine learning and automation, security leaders can arm their teams with the actionable intelligence they need to maximize their efficiency and effectiveness. And if implemented strategically, the entire organization can leverage the power of data-driven decision making to manage cyber risk and deliver measurable results across the organization.

Automate Risk Analysis

How do you automate risk analysis? You need to start with good data, as we have presented in this document. Then you need to create sophisticated data science algorithms—if you have the talent in-house. Few organizations have the internal resources to develop their own algorithms, let alone the data science expertise to apply machine learning. Moreover, the output of this exercise must be relatable to stakeholders in order to gain buy-in and ensure they adhere to your organization's remediation requirements.

Another approach is to use existing algorithms already proven to be effective in evaluating and predicting risk. For example, Kenna Security employed techniques from information retrieval, epidemiology, financial portfolios, and machine learning to create an algorithm that has high predictive power, but can also be understood by all stakeholders.

The Exploit Prediction Scoring System (EPSS)

One existing algorithm you can use to assist in your risk analysis is the EPSS. For teams who don't work with a vendor such as Kenna Security, and don't have a team of data scientists, and yet still wish to better predict whether a vulnerability will be exploited within the 12 months following its public disclosure, our very own Chief Data Scientist Michael Roytman joined forces with Jay Jacobs, Sasha Romanosky, Ben Edwards, and Idris Adjerid to create an open framework.

EPSS can be found here:
https://www.kennaresearch.com/tools/epss-calculator/

The paper laying out the framework can be found here:
https://arxiv.org/abs/1908.04856

Measuring Risk

To make the results of your sophisticated data science relatable to stakeholders, you will need a "metric" to measure and clearly communicate the risk of individual vulnerabilities. It must be simple, understandable, and repeatable so that your organization can monitor historical trends. The metric must take into account the number of instances of the vulnerability in the environment and the potential severity were it to be exploited, the exploitability of new vulnerabilities in the wild, current exploits, and real-time attacker activity.

Furthermore, this metric—or, in Kenna parlance, risk score—should serve as an actionable measurement that guides remediation efforts and resource allocation. As such, it can't be manually calculated using spreadsheets and CVSS scores. An objective risk score must be generated automatically (and revisited at regular intervals to reflect changes in the threat environment) to serve as a single source of truth for all teams, and ensure that everyone is focusing on the right thing at the right time.

But just having a risk score is not enough. Given the millions of vulnerabilities the average organization has, you will likely still have way more vulnerabilities with the same, critical, score than your team can remediate. The real value comes when you can pair that score with remediation intelligence that leverages data science. This will enable you to granularly prioritize your remediation efforts based on what will have the greatest impact on your overall risk score for the least amount of effort. Remediation intelligence can tell you which specific vulnerability to fix first for any asset or group of assets. This benefits your teams by maximizing their efficiency and effectiveness while reducing the greatest amount of risk to the organization.

To do this prioritization you could try to create your own system, but we recommend using a third-party solution that's well established and proven to work. Here are some considerations to ask vendors when evaluating their vulnerability scoring system:

1. What are they basing that score on? If they're using "machine learning and data science," can they explain to you what that looks like? What's the process?
2. Does it employ full context by leveraging real-world, real-time risk assessments and additional security information from throughout your environment?
3. Is the score the primary value-add, or is the score just the first step in a larger set of intelligence that directs you on how best to use your limited security and IT resources to reduce the most risk in the shortest amount of time?

Data Science for Risk-Based Vulnerability Management

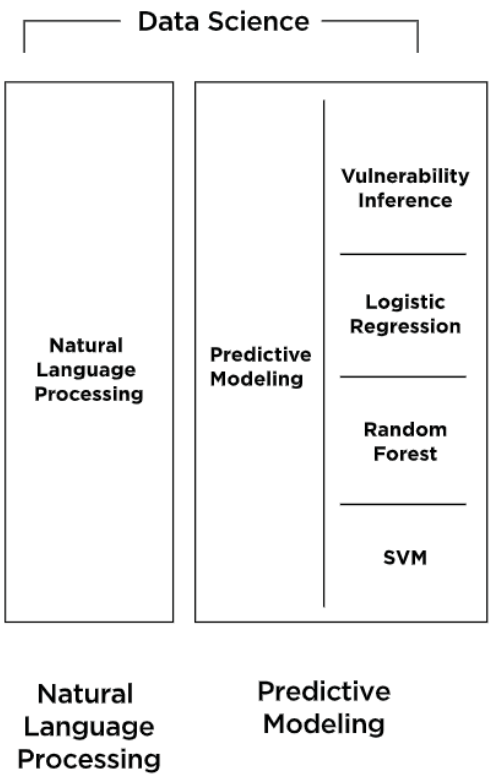
We showed you in Figure 2 the power of a predictive model over CVSS scores as a method to prioritize vulnerabilities. Now, to illustrate, let's walk through what to look for in a predictive model for evaluating and prioritizing vulnerabilities using data science.

First, every predictive model should be able to:

1. Ingest, aggregate, and process tens of billions of pieces of data from multiple sources
2. Automate the analysis of this data using a proven data science algorithm
3. Quickly deliver an accurate, granular, and quantifiable risk score for every vulnerability, asset, and group of assets

FIGURE 10

Data Science Framework



We recommend that the data science techniques employed should include machine learning, natural language processing, and predictive modeling to assess, prioritize, and even predict risk. (Figure 10)

Natural language processing investigates social media sites, the dark web, and other places where vulnerabilities are discussed, and extracts the language associated with vulnerabilities to assist in risk assessment. Natural language processing is also used to help score vulnerabilities that do not have a Common Vulnerability Scoring System (CVSS) score by analyzing various text key words and phrases that are shown to be high indicators of risk.

Predictive modeling calculates the risk of a vulnerability as soon as it is revealed—even before an exploit can be built. Predictive modeling should be able to forecast the weaponization of new vulnerabilities with a high degree of accuracy, and then prioritize remediation based on the risk of exploitation.

We recommend further analyzing the data using additional predictive technologies, such as SVM (support-vector machines), random forest, logistic regression, and vulnerability inference.

By harnessing the power of data science, security and IT teams gain the insights needed to focus remediation on vulnerabilities that truly pose imminent risk to the business. Moreover, data science provides security and IT teams with a common language—the language of risk—to gain buy-in from all stakeholders on a remediation strategy, to monitor and enforce remediation compliance, and measure the impact of those efforts through a demonstrable, quantified reduction in risk.

The Language of Risk

Effectively implementing a risk-based vulnerability management program requires cross-functional teams to work together with common goals, common tools, and a common language around risk.

As discussed at the beginning of this guide, security teams are charged with identifying vulnerabilities, but IT and application development teams are responsible for remediating them. Traditionally, infrastructure vulnerabilities are often prioritized based on CVSS scores, and remediation measured by reduction in vulnerability counts or the number of assets affected. This changes completely with a risk-based approach to vulnerability management.

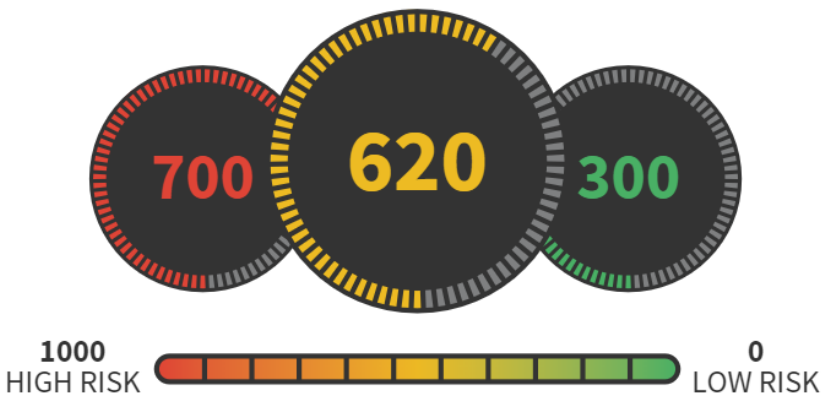
Actionable Risk Scoring

The language of risk can be effectively expressed in terms of a risk score for each unique vulnerability. This, in concert with asset criticality scores, then can determine an actionable risk score for each asset and group of assets.

A risk score provides an easy way for both security and IT/development teams to see risk in a common way. With accurate and quantifiable risk scores, your organization will understand your current risk posture and—more importantly—the actions you can take to reduce the greatest amount of risk.

FIGURE 11

Examples of the Kenna Risk Score



Operationalizing Risk Management

To affect measurable change using these data-driven risk scores, you should operationalize risk management by integrating it into existing processes and workflows wherever possible. For example, you could feed prioritized risks into your development team's bug tracking system. Similarly, consider weaving vulnerability intelligence and remediation guidance into your operations team's existing service desk, ticketing systems, change management platforms, and workflows. This would provide security and IT teams with a common source of actionable intelligence, helping IT to know what to fix, how to fix it, and why the fix is a priority. It also helps promote close collaboration between IT and security teams, fostering a common mission of optimizing the organization's risk reduction as quickly and efficiently as possible. And since IT would have access to the same intelligence as the security team, they could save the hours that would normally be spent on each vulnerability to re-investigate, validate, and determine how to remediate.

06
Communicate, Remediate, and Report

As we've seen, bringing together the right information with data science produces invaluable insights into vulnerability risk, enabling security teams to prioritize remediation efforts. To have an effective remediation strategy, however, you must establish clear lines of communication between security and those teams responsible for the fixes.

Communicating Between Security and IT

Risk can only be effectively mitigated when the whole organization works together toward the same goal. That means remediation tasks are automatically assigned to the right team, and those team members understand why they are being asked to remediate particular vulnerabilities, have bought into the tasks set before them, and receive the information they need to address the risk. Everyone, at all times, should understand where the organization stands in terms of risk.

Traditionally, vulnerability scanners produce a report with many thousands of vulnerabilities affecting hundreds of assets across an organization. You would then have to go into a ticketing system like ServiceNow and manually create individual tickets, enter information on the vulnerability, which assets are impacted, note where to find a patch, etc. However, if the information on the vulnerability and its patch are not entered—which is all too common— IT will then have to re-investigate and validate each vulnerability, a tedious job causing significant delays in the remediation process.

Tight integration between a risk-based vulnerability management platform and your ticketing system, as described in Step 5, resolves this tedious process. In addition, providing owners of assets or asset groups with their own dashboards and risk meters (or other view into asset risk) integrated with remediation policies puts both security and remediation teams on the same page. These technology solutions enable IT, development and DevOps teams to easily see what needs to be fixed, how to fix it, why it's a priority, and within what timeframe based on risk levels. But for the process to be truly optimal, there must also be policies and remediation service-level agreements (SLAs). We discussed a little about SLAs earlier in this guide, and will add additional detail here as it is an important aspect of the remediation process.

Ultimately, communicating risk and prioritizing remediation efforts requires that all players are bought into the program, and processes are in place with agreed-upon SLAs.

Remediation Strategies

Remediation strategies will vary based on the overall risk tolerance of your enterprise (e.g., highly regulated companies will likely have a lower risk tolerance than non-regulated), as well as risk tolerance within individual departments (e.g., revenue-generating organizations and those managing personally identifiable information may be more sensitive than, say, the marketing organization). However, such assessments are business-specific. These risk determinations must then be correlated with the risk levels of each vulnerability and the criticality of the asset or application, as well as the work effort and business impact of the remediation process itself. It's a tall order.

Remediation typically falls into three types:

Auto-updates

Auto-updates require minimal effort and can usually be handled quite easily by individual asset owners or end users. An example would be Google Chrome, which can download updates in the background and requires nothing more than a restart. It basically patches itself, thus resolving any open vulnerabilities.

Patch Management Tools

There are a variety of patch management tools your organization may choose, depending on the assets you're managing. For example, a Windows shop would likely have a tool like Microsoft SCCM or WSUS to package up the latest updates and push the patch out to all Windows machines simultaneously. Most organizations follow a specified cadence; e.g., "Patch Tuesday" is the day to run Microsoft patches. Similar tools may also be used for other operating environments such as Linux and Unix. When performed on a consistent basis, this approach keeps an entire group of assets up to date and resolves any vulnerabilities identified between patches.

Manual Updates

Manual updates present the greatest challenge in any vulnerability management program. These types of updates can be complex and time-consuming depending on where the vulnerability lies and any cascading dependencies. For example, suppose your organization relies on an application using an older version of Java. The only way to remediate the application vulnerability is to also upgrade Java. However, updating Java will negatively affect other applications running on the same system. So those applications also need to be updated, which could mean developers changing the source code of the applications that you've written using Java because it doesn't support the most current version of Java. This scenario requires additional decision-making in terms of prioritization.

Interestingly, we did find that the way organizations choose to update systems has some bearing on remediation performance. Auto-updates do not seem to have a positive or negative impact, but reliance on patch management tools definitely helps remediation performance, and use of manual tools negatively impacts coverage. We found that those who deployed patches using patch management tools averaged a 20% higher successful close rate on high-risk vulnerabilities, so we highly recommend the use of centralized patch management tools.

FIGURE 12

Methods for deploying system updates

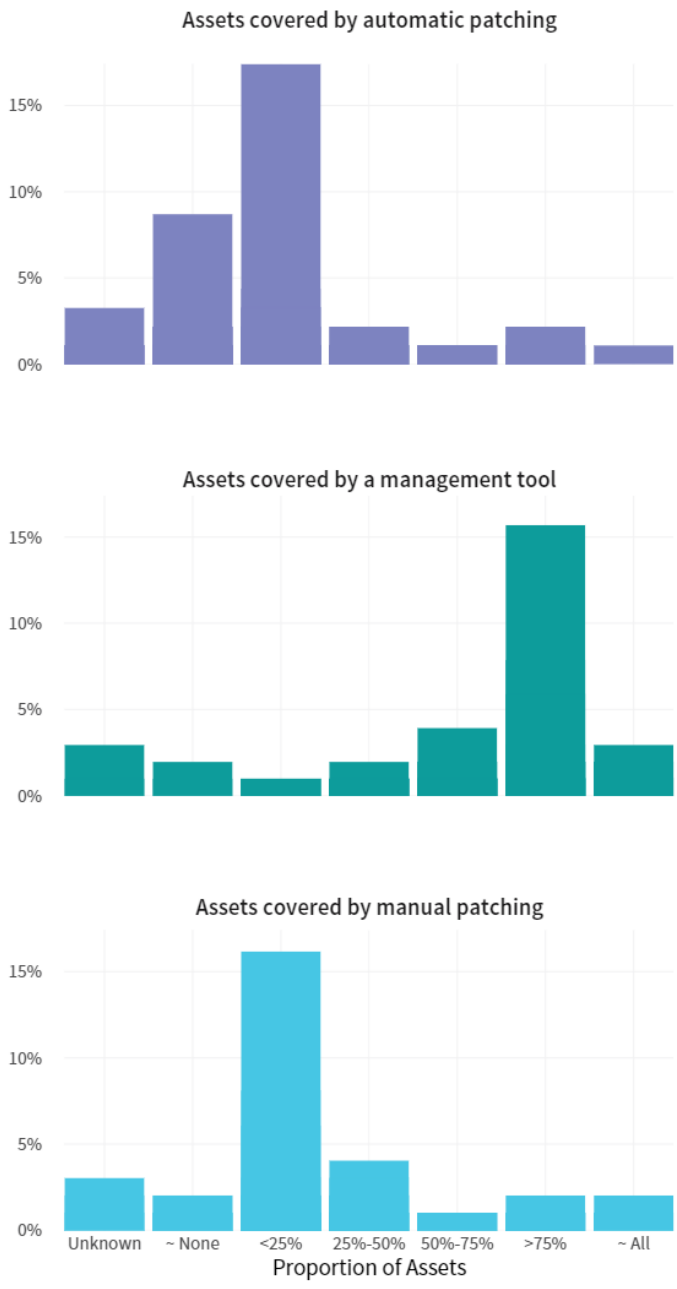
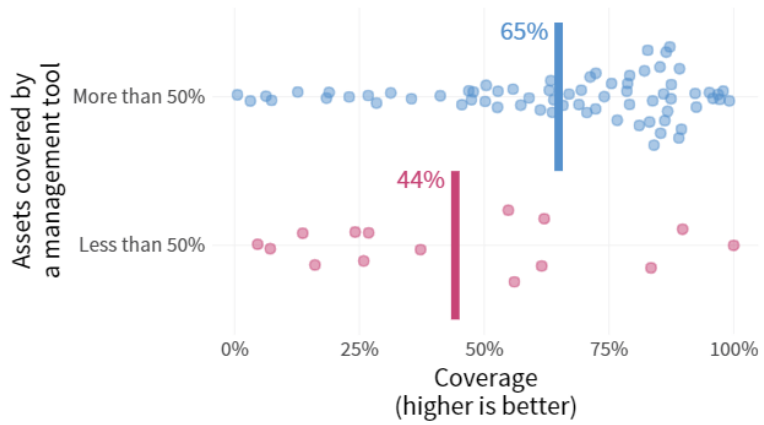


FIGURE 13

The effect of patch management tools on remediation coverage



Remediation Decision Considerations

Factoring in risk levels, asset criticality, and remediation work effort (including business impact of the work effort) informs certain decisions about when and how to address a vulnerability. Performing a manual update as described above could take months, disrupting operations to an unacceptable degree. This forces a difficult decision: to accept the risk or not. If you accept the risk, what compensating measures can you put in place to at least mitigate the risk? Perhaps the decision is to add new rules to the firewall blocking certain traffic that could exploit the vulnerability, and otherwise live with the risk.

Weighing risk versus effort will help guide important decisions about how to approach remediation.

FIGURE 14

Remediation Decision Matrix

High Risk – High Effort	High Risk – Medium Effort	High Risk – Low Effort
Medium Risk – High Effort	Medium Risk – Medium Effort	Medium Risk – Low Effort
Low Risk – High Effort	Low Risk – Medium Effort	Low Risk – Low Effort

Referring to the Remediation Decision Matrix, a high-risk vulnerability that requires a high degree of effort to remediate may point to mitigation as mentioned above due to the disruption caused by the remediation effort. In stark contrast, a high-risk vulnerability with an easy fix is probably at the top of your priority list for immediate attention. However, a low-risk vulnerability with a disruptive and time-consuming fix might simply be accepted until a larger-scale infrastructure modernization plan is put into place.

Each variable of risk/effort can inform your remediation strategy. For example, medium-risk vulnerabilities with a modest remediation effort may lead you to look for more robust patch management tools that can streamline the process. Or, if you have not implemented a routine patching policy, this analysis could be the springboard for doing so, and thus having a way to regularly remediate a large cross-section of vulnerabilities in a regular cadence.

Establishing Remediation Deadlines

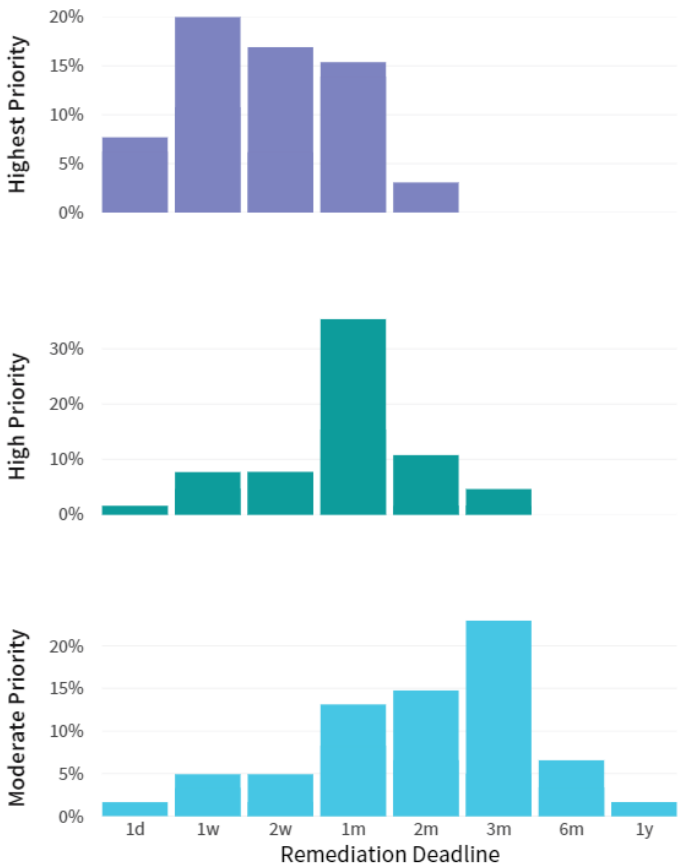
One of the most important factors in ensuring an effective vulnerability management program is establishing deadlines, or service-level agreements (SLAs), for how long it should take to remediate a given vulnerability. Due dates, or SLAs, may be established on a variety of criteria such as severity of the vulnerability, asset criticality, organizational risk tolerance, etc. Every organization will have its own set of factors on which to base the SLA. Regardless of how you determine your SLAs, it is important that you have some kind of policy in place.

In our research across hundreds of organizations, we found that about two-thirds have defined deadlines and follow them in some capacity. The remainder either handle things on a case-by-case basis or aren't sure if SLAs exist (meaning they probably do not). Of those organizations with established SLAs, most expected the highest-priority vulnerabilities to be remediated within one week, high priorities within a month, and moderates within three months. But, of course, these timeframes can vary widely across organizations.

Regardless of the specific SLA timeframes, as noted earlier in this guide, our analysis showed that while the timeframe of the SLA doesn't necessarily relate to a performance improvement, companies with defined SLAs address vulnerabilities more quickly than those with nonexistent or ad-hoc schedules.

FIGURE 15

Remediation deadlines by priority level



Reporting On Vulnerability Management Performance

With your remediation strategy, weighing risk versus effort is a key consideration, and this will also impact how you then measure and report back on the success of your approach. Given the potential impact of a cybersecurity breach on the business, this is often a board-level conversation and a very important component of your vulnerability management program.

You need a clear and accurate way of assessing your performance, which we find can be broken down into four key areas:

- **Coverage** – Measures the completeness of remediation efforts. What percentage of exploited or high-risk vulnerabilities were actually remediated?
- **Efficiency** – Measures the precision of remediation efforts. What percentage of vulnerabilities remediated were actually high risk?
- **Velocity** – Measures how long the remediation process takes.
- **Capacity** – Some vulnerability management programs fix fast within a limited scope, while others may take a long view, working on the full range of vulnerabilities over time. This is what we refer to as remediation capacity.

Remediation Coverage and Efficiency

Our research has shown that successful vulnerability management balances two simultaneous 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.

A quick recap:

Balancing these two goals is not easy. Every organization has different needs. In terms of coverage, we have found that most organizations using a risk-based vulnerability management strategy are addressing the large majority of their highest-risk vulnerabilities. (Figure 16). That's not surprising. However, many organizations struggle with efficiency. This is understandable given that the cost of leaving a vulnerability open only to then later be exploited will generally be higher than the cost of fixing it "just in case." But there's another reason as well—patching is inherently inefficient.

Guidance for Reporting to the Board

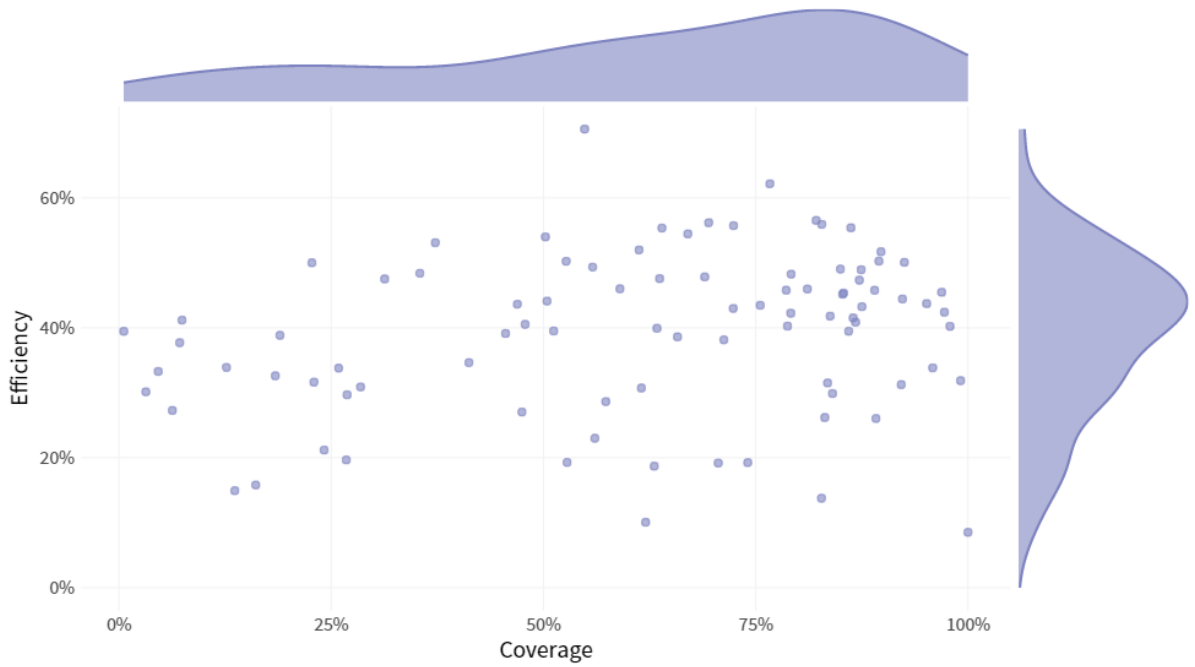
- Know your audience: leave the technical minutia at home
- Speak their language: the language of finance and risk
- Bring the right facts: don't just show lists of vulnerabilities closed
- Tell a compelling story: tell them about risk, resources, and tradeoffs
- Anticipate questions on your org's performance relative to others; you can use the following benchmarks:
  - Industry-specific Information Sharing and Analysis Centers (ISACs)
  - Factor Analysis of Information Risk (FAIR)
  - Conversations with other CISOs, even competitors

Read full article: *https://www.scmagazine.com/home/opinions/how-cisos-can-tell-a-better-security-story-to-their-board/*

Among the hundreds of thousands of patches we studied, about half fix multiple 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. Your efficiency metric reflects that penalty even though you really didn't explicitly choose to prioritize the other four.

FIGURE 16

Remediation coverage and efficiency metrics across companies



Remediation Velocity

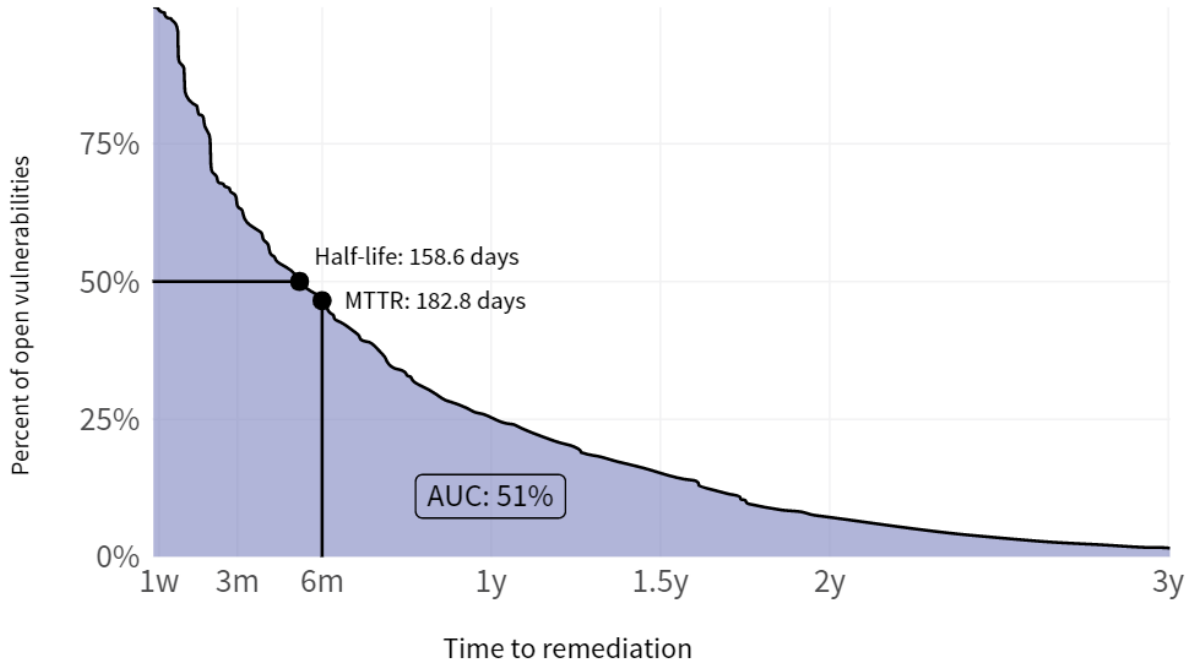
It's useful to understand how long the remediation process takes—this is a measure of velocity. For example, assume an organization observes 100 open vulnerabilities today (day zero) and manages to fix 10 of them on the same day, leaving 90 to live another day. The vulnerability survival rate on day zero would be 90% with a 10% remediation rate. As time passes and vulnerabilities continue to be fixed, that proportion will continue to change.

We tracked this metric across all vulnerabilities in multiple companies (Figure 17). The graph shows that the overall half-life of a vulnerability is 159 days, with many vulnerabilities remaining open beyond one year.

Remediation timeframes will vary substantially across organizations, which is why velocity is an important metric to consider: It represents both a directional and a speed aspect. However, bear in mind that it's not necessarily bad to not close all vulnerabilities quickly. In fact, it would be wasteful to close all of them as fast as possible. The more important question is how quickly are you able to fix the vulnerabilities that really matter. That's why higher remediation velocity typically reflects lower efficiency, indicating a tradeoff similar to that between coverage and efficiency.

FIGURE 17

Survival analysis curve for vulnerability remediation time



Remediation Capacity

Some organizations may attack a select cross-section of vulnerabilities with surgical force; others may take a long view, working on the full range of vulnerabilities over time. This is what we refer to as remediation capacity. With remediation capacity we look at two primary metrics:

- Mean Monthly Close Rate (MMCR): Measures the proportion of all open vulnerabilities a firm can close within a given timeframe.
- Vulnerability debt: Measures the net surplus or deficit of open high-risk vulnerabilities in the environment over time.

Key Metrics For Survival Analysis

Area Under the Curve (AUC) – Measures the area under the survival curve representing "live" (open) vulnerabilities. A lower AUC means higher velocity.

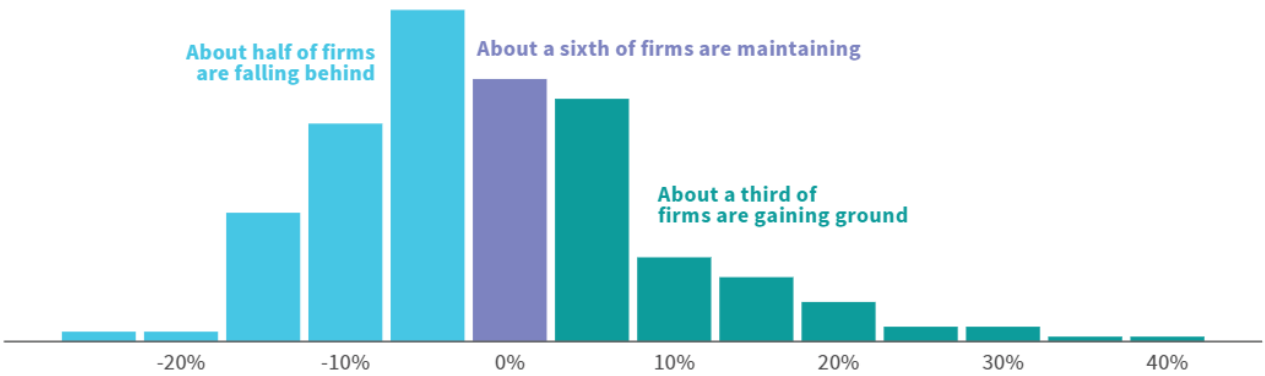
Mean Time To Remediation (MTTR) – The average amount of time it takes to close vulnerabilities.

Vulnerability half-life – The time required to close exactly 50% of open vulnerabilities.

MMCR is a measure of raw remediation capacity, and in our research, we've found that organizations remediate about one out of every 10 vulnerabilities in their environment within a given month. The question then is whether organizations are at least staying ahead of those vulnerabilities that represent the highest risk. That is, what is your vulnerability debt? Here, our research shows some encouraging signs. While on average about half of companies fall behind, one in six are keeping up, and one in three are actually gaining some positive ground. (Figure 18)

FIGURE 18

Net remediation capacity for high-risk vulnerabilities



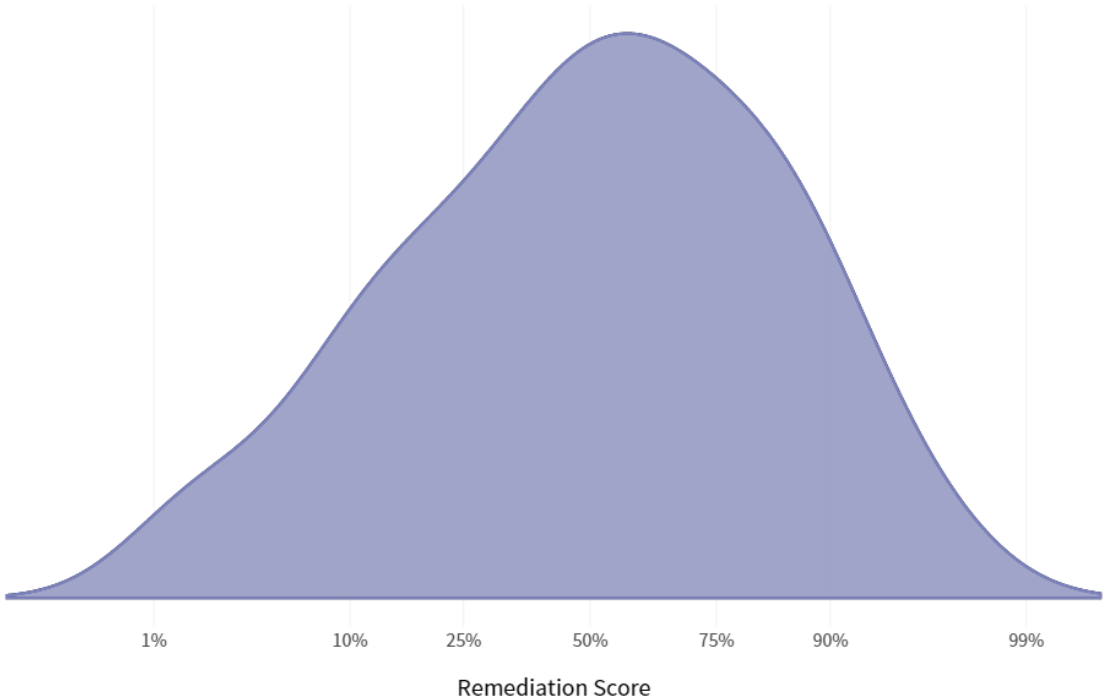
Overall Remediation Performance

Each of the above metrics is important for assessing and reporting on your vulnerability management program. However, no single metric tells the whole story. Therefore, in our research we used a proven mathematical method (Principal Components Analysis) for creating a credit score-like measure of overall vulnerability remediation performance. You can apply this method to capacity and velocity metrics first to get scores in those categories, and then combine everything to get a single-value holistic view of remediation.

Measuring vulnerability remediation performance—through particular metrics or an overall score—is a useful exercise for vulnerability management programs. Once you understand where your organization stands, you can then plot a course to improve performance based on your individual organizational characteristics and practices. In a review of approximately 100 Kenna customers utilizing risk-based vulnerability management, Figure 19 shows where their scores netted out.

FIGURE 19

Distribution of overall remediation performance scores across firms



We hope that this guide has provided you with some useful information that you can use to set up or refine your very own risk-based vulnerability management program.

Endnotes:

[1] 2019 Cybersecurity Almanac: 100 Facts, Figures, Predictions and Statistics; Cybersecurity Ventures (in collaboration with Cisco); February 6, 2019; *https://cybersecurityventures.com/cybersecurity-almanac-2019/*

[2] 2018 Application Security Report, Cybersecurity Insiders (in collaboration with Kenna Security), *https://resources.kennasecurity.com/research-reports/2018-application-security-report-kenna-final*

[3] Successful Application Security Strategies Webinar; Jonathan Cran and Tyler Shields; May 8, 2019; *https://www.brighttalk.com/webcast/13229/354868*

[4] Successful Application Security Strategies Webinar; Jonathan Cran and Tyler Shields; May 8, 2019; *https://www.brighttalk.com/webcast/13229/354868*

[5] One-fifth of the most-used Docker containers have at least one critical vulnerability; Jerry Gamblin; June 25, 2019; *https://www.kennasecurity.com/blog/one-fifth-of-the-most-used-docker-containers-have-at-least-one-critical-vulnerability/*

KENNA katalyst

THE RISK-BASED VULNERABILITY WORKSHOP

What's Inside...

- Key findings from research into 100 organizations.
- People and skill sets needed to implement RBVM.
- Step-by-step guide to achieving the results you want.
- Key technology investments for optimal RBVM.

About KENNA Security

With Kenna Security, companies efficiently manage the right level of risk for their business. Our Modern Vulnerability Management model eliminates the friction between Security and IT teams about what to patch, providing clear prioritization based on real-time threat intelligence and guidance applied to each customer's unique environment across infrastructure, applications and IoT.

Find out more at www.kennasecurity.com