seemplicity eBook

3 THINGS THAT

Slow Your Time to Remediation

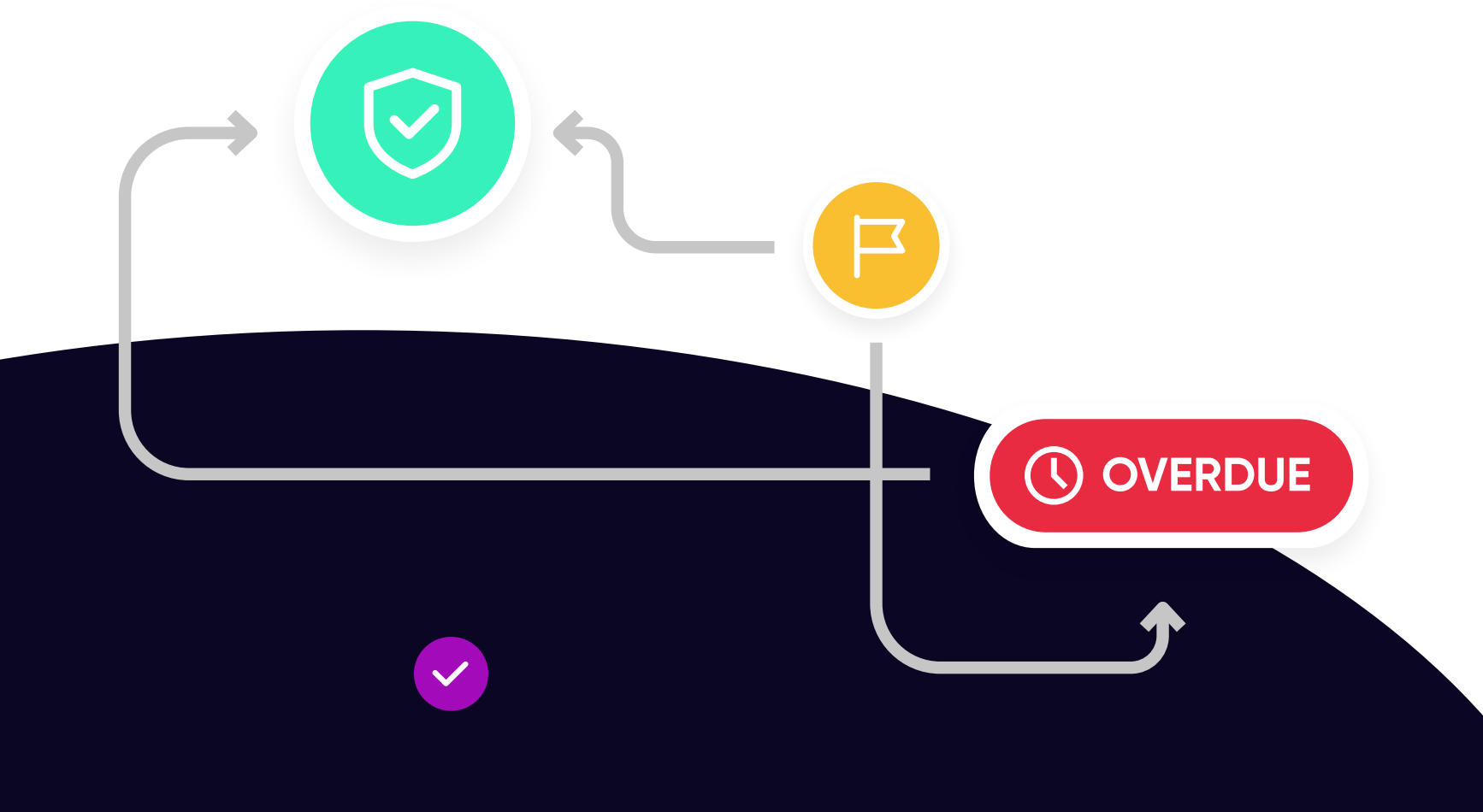


TABLE OF CONTENTS

3-4
Introduction

4-5
The Time-to-Remediation challenge

6
Case Study: Distributing First Reduces Time to Remediation

7-8
Bottlenecks and Barriers in Remediation

Findings in Bulk

Follow-Up

Non-Native Workarounds

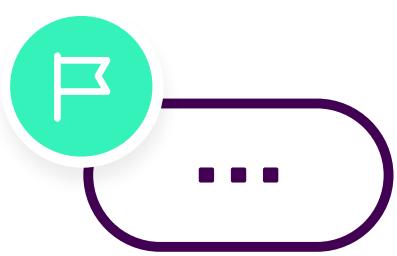
9
Distribute First, Prioritize Second

10
About Seemplicity

There are more cyber threats and security vulnerabilities today than ever before,

making it almost impossible for any company to address them all. There simply aren't enough hours in the day to patch every exposure, and never enough qualified team members—especially given the severe skills shortage in the market. That's why cybersecurity today is all about leveraging the resources you have to fix the vulnerabilities that are most likely to have the most significant business impact as quickly as possible. However, with so many vulnerabilities on the to-do list, that isn't always an easy task.

One of the key challenges that security teams face is that although they are responsible for finding and identifying vulnerabilities, in most cases, they're not responsible for fixing the problems they find. Instead, after they identify a problem, security teams need to assign the problem to the right person or team for remediation. That makes security very different than a field like software development, where the team responsible for identifying a problem is also the team responsible for remediation.



The challenge is compounded by the fact that security vulnerabilities often end up being multifaceted and can't be solved with a single solution or process. In fact, security teams generally function in a "many-to-many" fashion, using several tools and working with many teams within the organization. In order to distribute the workload and prioritize effectively, they have to be familiar with all IT assets, know what operating systems are running, what software is installed and being used, who has access to each system, what the data flow is, what ports/protocols/services are being used, the business criticality of each system, and where systems reside in the network.

Even if security teams can stay on top of all that information, they don't necessarily know who should fix which vulnerability, and figuring it out is time-consuming. When there are thousands of findings, even if each operation takes only half an hour, it adds up quickly. It's also difficult to preserve the knowledge and maintain it within teams.

The Time-to-Remediation Challenge

Time to remediation is a critical metric in managing that challenge—it's a direct expression of exposure to vulnerability risk. Yet time to remediation is complex, and includes more than the net amount of time it takes to fix a given problem. To better understand the process let's look at the key stages of time to remediation, and where delays are likely to occur.

STEP 1

Identify

Owned by the Security Team

This stage is all about identifying existing security weaknesses. Whether it's a misconfiguration, an application vulnerability, or compromised credentials, there are numerous tools available to scan your environment and detect your vulnerabilities.

STEP 2

Prepare for Remediation

Owned by the Security Team

This step focuses on getting the problem in front of someone who can fix it. First, the overwhelmed security team has to see the problem–something that often takes time. Then the security team has to confirm that there are no duplications, add the findings to the to-do list, prioritize findings across different security programs, pair each finding to a fixer and open a task for it in the relevant ticketing system. Weeding through the alerts takes quite a lot of time, as security teams have to switch between different technology consoles to put the pieces together. Unlike the previous stage, this stage is usually managed manually, leading to significant delays and bottlenecks.

STEP 3

Remediate

Executed by Security's Counterpart

This step includes the actual fix—the time taken to patch or resolve the vulnerability. Unlike the previous step, it is generally conducted outside the security team by developers, DevOps, or IT teams.

STEP 4

Verify

Owned by the Security Team

This step includes the time needed to ensure that the task has been completed and the problem has been fixed. It also encompasses reporting and SLA metrics, which are not necessarily conducted on a problem-by-problem level.

Time-to-remediation encompasses the entire cycle, all four stages. Surprisingly, the second step is usually done manually, and our research indicates that getting the problem to the right person for remediation often takes more time than the fix itself. One of our customers, a fintech cloud-native company, reported that, on average, the second stage took 70 days, while the average time for stage 3, the fix time, was only five days.

Reducing the "find time", the time it takes to get a vulnerability to someone who can actually fix it and act upon it should be a key goal for security teams.

CASE STUDY

Distributing First Reduces Time to Remediation

An **online marketplace organization** had **250 critical security findings** in one report. Our analysis found that the remediation of those findings was allocated to **20 different teams** in the organization. 17 teams had 3-5 findings each, and **three teams had 50+ each.**

If the security team had been working traditionally, they would have started by allocating the top ten organizational priorities, regardless of the team they were assigned to. Then, they would have stopped, without understanding how those tasks were distributed across the organization. That's because security teams generally assess prioritization on an organizational rather than at a team level.

A **distribution-first strategy** allowed the 17 teams to get working on their short backlog because they could fit it into their sprints continuously. **The security team could then focus their prioritization efforts on the three teams** that had a bigger backlog, rather than being weighed down by the entire list. **The bottleneck was minimized**, and the issues were addressed quickly and more effectively than they would have been using other workflow methods. The security team could also see where the bottlenecks were occurring and where more attention was needed in terms of training and other resources.

Bottlenecks and Barriers in Remediation

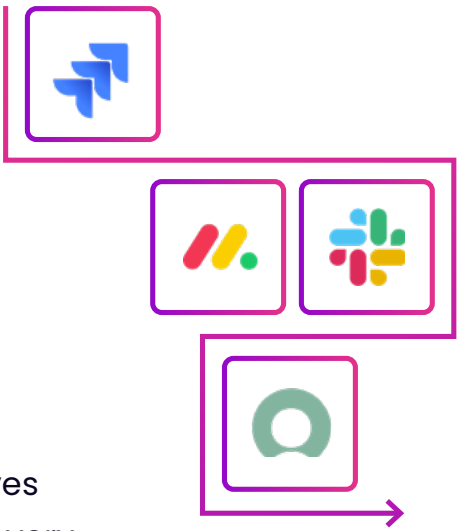
Findings in Bulk

Security scanners often generate long lists of findings making them difficult to process. When people see a list with hundreds of items, it can be overwhelming. They need to stop to review the list—a process that can take days, or even weeks—before they can act on the tasks at hand. Multiply that by the number of tools and you get multiple lists of findings, further slowing down the process. That type of workflow doesn't integrate well with the sprint methodology used in development contributing to the friction between the processes.

Follow-Up

Even after the problem has been assigned, the security team has to follow up to make sure that it was actually fixed and collect data on different metrics for reporting purposes. Since there are many teams fixing vulnerabilities, each using different tools, the tracking and verification process also slows down security teams and delays them from starting on the next back of issues because they are not sure if the existing batch is complete. Security spends significant time following up with various teams using different tools and generating metrics manually, which takes them away from the mission-critical tasks in stage 2 of the remediation process.

Non-Native Workarounds

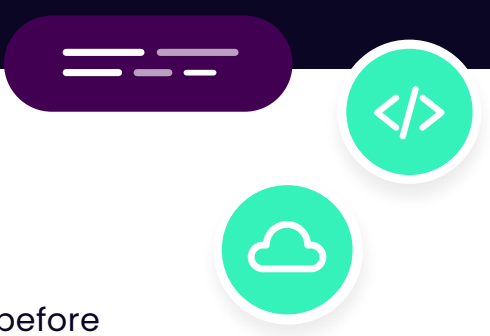


Common project management platforms from other domains, like Salesforce and JIRA weren't designed for security needs. For example, JIRA has a priority function. However, it doesn't include risk, which is a defining metric for security, of which priority is a derivative. Therefore, when managing security processes on JIRA, risk metrics need to be added into each project which is cumbersome and time-consuming.

Security teams often try to develop these capacities themselves or build them on top of JIRA. Yet that requires a developer for every change, and security teams don't usually include developers. New tools that need to be integrated are not native and problems are likely to occur. For example, a company tried to generate JIRA reports for resolved tasks. In some, the status was defined as a non-capitalized "done" and others as "Done". When it came time to generate a report, the data between the two options could not be automatically integrated, causing an unnecessary delay. And the more tools that are adopted, the more problems of this sort are likely to occur.

Distribute First, Prioritize Second

Time to remediation shouldn't be delayed by a prioritization backlog. Security teams need an effective method to get vulnerabilities to the people who can fix them without delay. Seemplicity was created to achieve that goal.



Seemplicity accelerates time to remediation

using an empiric, scientific strategy. It automates the process before and after the fix, stages 2 and 4 in the remediation process, providing a single point of truth for security reporting available at any time. Tasks are automatically sent out to teams for remediation in small, easy-to-manage batches, eliminating peaks of activity and the need for escalation. It is integrated with JIRA, and reporting can easily be tracked on the platform.

Seemplicity is unique because it addresses the entire security stack,

unlike other aggregation tools that only solve part of the problem and create more noise than clarity. It goes beyond process automation, which focuses on the automation of individual business-critical tasks. Instead, it uses process orchestration to manage the automation lifecycle end-to-end, across various teams and systems, unifying multiple individual tasks into one smart unit and automating hand-offs between teams and tools.

Security teams that effectively leverage security orchestration and automation using a platform like Seemplicity are able to

spend less time manually connecting the dots between fragmented security findings, siloed teams, and distributed tracking systems

and can focus on the tough problems that really need the human touch for investigation, mitigation, and remediation. It's a smarter way to use a limited set of resources to manage an ever-growing threat.

About seemplicity

Seemplicity revolutionizes the way security teams drive risk down access the organization by orchestrating, automating and scaling all risk reduction workflows in one workspace. With their very own dedicated workflow platform, security teams are empowered to turn risk reduction into a self-service process that can be easily consumed by developers, DevOps and IT across the organization, in a simple, effective and collaborative manner that ultimately accelerates time-to-remediation and improves the overall security posture of the organization.

Would You Like to Learn More About Our Platform?

Book a Demo so we can introduce you to our platform.