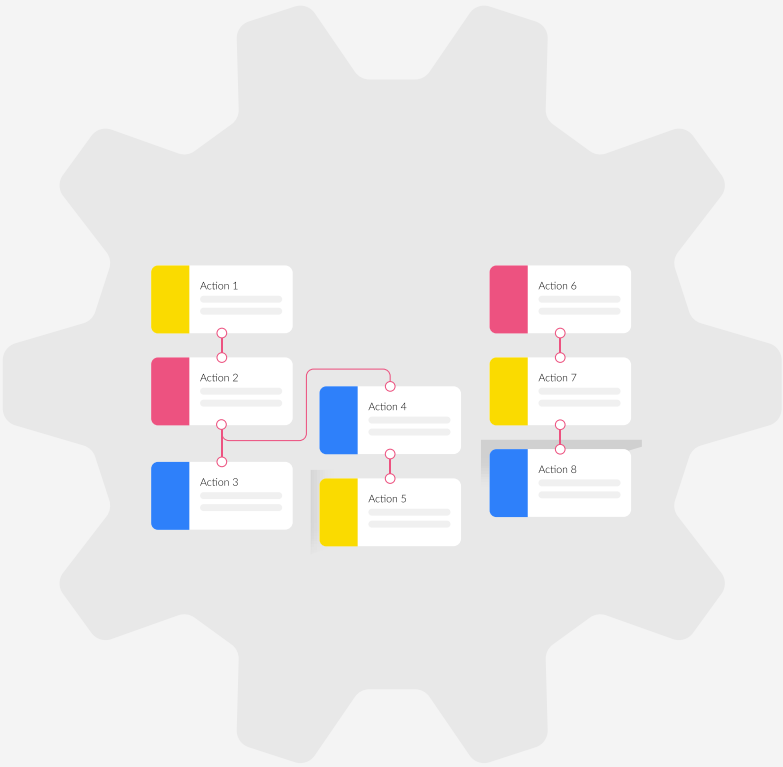
The Transformative Effect of No-Code Security Automation



torq

Contents

1

Introduction

Everyone loves automation. Whether you're fully automating a single task or building human-in-the-loop automations that streamline complex processes, automation saves time, scales operations, and allows teams to focus their creative energies where they can do the most good – as opposed to spending time on tedious manual tasks.

At least, that's what automation *should* do. If you look at how the story of automation has played out in different domains of the IT industry over the past decade, you realize that automation doesn't always live up to its promise.

On the one hand, consider software development teams. Over the past ten years, developers have transformed the way they work by adopting CI/CD automation. Source Code Management tools, CI servers, release automation suites, and the like have ushered in unprecedented degrees of efficiency, making it possible for teams to deploy new application releases multiple times per week instead of once or twice a year, as was the norm prior to widespread adoption of development automation tools.

On the other hand are security operations teams. These teams have added automation tools – such as SOARs, SIEMs, and incident management platforms – to their toolboxes over the past decade. And yet, SecOps remains plagued by operational inefficiencies. The struggle to integrate diverse sets of security tools, combined with the complex configurations that most of those tools require in order to perform their core functions, means that security operations often take longer and require more engineering resources today than they did ten years ago.

That's exactly the opposite position that SecOps teams should be in as they face pressure to manage threats that are constantly increasing in scope and complexity. To survive in the world of modern security, SecOps engineers need automation tools that actually automate security operations as a whole, rather than just individual components of it.

2

Squaring the Circle with No-Code Security Automation

The good news is that there's a solution to this conundrum: no-code security automation. With no-code automation frameworks designed for security operations, security teams can quickly and easily implement the automations they need to take full advantage of the complex tool sets at their disposal and integrate their tools seamlessly with the broader IT operations tooling stack.

In this way, no-code security automation finally unlocks for security teams the same efficiencies that CI/CD automation has brought to development over the past decade.

To prove the point, this eBook walks through the automation shortcomings that security teams typically face today, then explains the role of no-code security automation in addressing them. As we'll see, true, effective automation is within reach of security teams today – they just need a code-free way to leverage the power of the security monitoring, alerting, and analytics tools on which they depend.

3

The Paradox of Security Automation

Again, modern security teams have no shortage of tools that promise to automate security operations, such as SOARs and SIEMs. But the typical team faces two main challenges in unlocking the full potential of these tools:

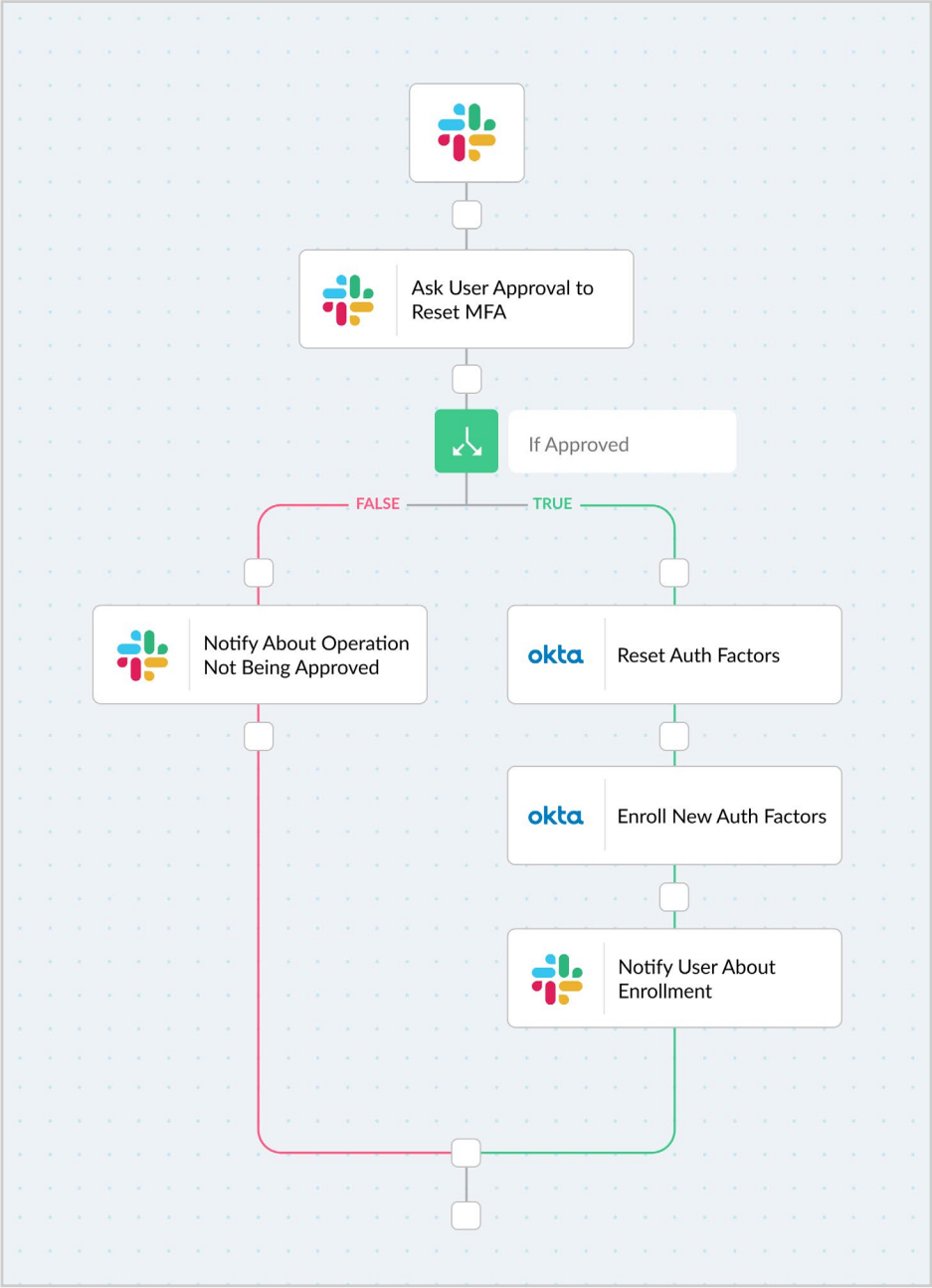
- **Lack of integrations:** In many cases, the various security tools that teams rely on don't automatically integrate with each other. Integration with external tools is certainly possible in most cases, but actually implementing it requires complex configurations that must be set up manually. As a result, getting data out of your SIEM or SOAR and into alert management systems, incident response platforms, and so on requires extra effort, which undercuts the efficiency that these tools are supposed to unlock.

- **Configuration complexity:** Setting aside the issue of integrations, most modern security tools require complex configurations just to perform their core jobs, let alone connect to other tools. Engineers must customize threat detection rules, write tool deployment scripts, use APIs to connect analytics engines to threat intelligence databases, and so on. Here again, this is all work that must be performed manually. Even worse, engineers may need to update configurations whenever they modify their environment, detect a new type of threat, and so on.

Addressing both of these challenges has traditionally required security engineers to moonlight as developers. They've had to write their own code to integrate and configure disparate tools.

As a result, all of the tools that are supposed to make security operations faster, more efficient, and more scalable have had precisely the opposite effect in certain key respects. They have caused security engineers to spend more time managing tools and less time on actual security operations. They have required SecOps teams to master domains that are ancillary to security operations, like coding. They have made security operations slower, and created barriers that make it more difficult to stay up-to-speed with constantly evolving security threats by updating tool configurations to detect new threats.

In short, security automation has proven to be mostly an illusion. Individually, security tools can automate operations like monitoring and analytics. But collectively, achieving real automation across a complex security toolchain requires so much manual effort, and so many special coding skills, that unlocking the full value of modern security tooling remains a struggle for many SecOps teams.



An actual workflow from Torq, shown as steps that security creators can drag and drop into the Workflow Designer module.

4

The Promise of No-Code Security Automation

No-Code security automation solves these challenges by empowering SecOps teams to manage security tools and operations without having to write a single line of code.

The premise of no-code security automation is elegantly simple: using prebuilt templates and a simple, drag-and-drop interface, engineers can quickly integrate disparate tools and configure management policies in order to build out complete workflows. In other words, without writing a single line of code, they can set up and deploy complex workflows that would otherwise take skilled developers hours or days to implement.

The benefits of adopting a no-code automation strategy for security automation are numerous.

Make the Most of Complex Tools

Security tools often require an unusual level of configuration and customization in order to deliver their full value.

For instance, a SOAR platform may offer a few out-of-the-box configurations that suffice for addressing generic threat detection use cases. But if you want to perform active threat hunting or contextualize threat analysis based on proprietary threat intelligence data, you'll need to configure your SOAR to do so. Likewise, alerting tools often deliver little value when alerts fire based on default thresholds or conditions. You'll need to customize them to avoid alert fatigue while still making sure that the important notifications reach your team.

Historically, these types of customizations required significant time and resources. Security teams had to choose between spending those resources in order to get the most value out of their tools or living with imperfect configurations that undercut the tools' performance.

No-Code security automation eliminates the need to make this choice. When anyone can customize complex workflows in minutes – without having to know special configuration languages or pore over hundreds of lines of policy files – even the smallest, most resource-strapped SecOps team can optimize its tools.

Let SecOps be SecOps

Businesses hire SecOps engineers and security analysts to do one main thing – secure their IT assets.

And yet, in a world where managing security tools requires complex coding skills, SecOps teams often spend much of their time writing the code necessary to manage their tools. In other words, they end up being de facto developers – or, alternatively, they burden the actual development team by leaning on developers to help them write the code they need to configure their tools and workflows.

With no-code security automation, security engineers can return to being security engineers. They don't need to code, and they don't need to ask others to code for them. Instead, they can implement the configurations they need rapidly and painlessly, then get to work performing their core jobs.

Make DevSecOps Work in Practice

Security teams don't exist in a silo – especially not in today's DevSecOps-centric world, where more and more businesses prioritize collaboration between development, security, and IT operations teams.

That collaboration is difficult to achieve, however, if security operations are based on arcane configuration files that only engineers with specialized skills can read, let alone write. It's hard to maintain across-the-board visibility or a sense of collective ownership over security when only a handful of engineers can understand how security tools fit together.

This problem disappears in a business that leverages no-code security automation. When security rules and workflows can be visualized on dashboards, every stakeholder can quickly understand the state of security operations.

Integrate with IT

Along similar lines, in some cases, security tools need to integrate closely with IT operations tools in order to deliver their full value. Alerts from a SOAR may need to be fed into incident management platforms or ticketing systems, for instance.

Traditionally, building those integrations required extensive resources. As a result, it was difficult to achieve seamless integration between security tools and the broader IT tool set, and security didn't always receive the priority it deserved within IT operations.

But with no-code security automation, anyone can set up the integrations necessary to ensure that security priorities are reflected within IT systems. In turn, security becomes an integral part of broader IT processes rather than remaining stuck in its own tools and its own workflows.

Do More with Smaller Teams

Last but not least, no-code security automation means that SecOps teams can address more threats at greater scale.

That's important, of course, because we live in a world where each year sets new records for the number of cyber threats businesses encounter, and where the size and complexity of IT environments are constantly increasing.

Under these conditions, SecOps teams don't benefit from no-code automation simply because it's a convenience. They need it because it's the only way they can remain productive as they face never-ending demands on their time.

By removing the need to devote enormous resources to tool configuration and management, no-code automation frees security teams to do what they do best and what drives real business value – detect and mitigate threats.

5

We've Seen This Movie Before

The efficiency and scalability that security teams stand to gain from no-code automation is not novel. It's what development teams have been enjoying for years, thanks to the widespread adoption of CI/CD automation tools.

Prior to the advent of the DevOps mindset and the tools that enable it, developers had to tune each of their development tools manually. Compilers required endless tweaking in order to build applications. Developers could spend weeks setting up the optimal dev/test environments. Integrating code from multiple developers into a single codebase was a messy, time-consuming affair. And actually deploying applications required carefully moving around binaries and configuration data while crossing fingers that everything actually ended up where it was supposed to.

Fast forward to the present, and CI/CD automation has changed all of this. Developers can now take full advantage of well-integrated CI/CD suites that automatically integrate, build, and deploy code. No matter which type of application they are building, or where they are deploying it, they can focus on their work – coding – and let CI/CD tools handle the tedious task of managing their code.

For SecOps teams, a similar world is possible. But again, it requires more than security tools that, on their own, are difficult and time-consuming to integrate into effective SecOps workflows. Instead, security teams need the same easy-to-implement, easy-to-manage automations that have transformed software development over the past decade.

6

No-Code Security Automation with Torq

Torq provides these solutions. By allowing any engineer to define security workflows visually, without writing a line of code, Torq provides practitioners of all skill levels with the tool they need to achieve full automation for complex security operations. There are no scripts to manage, no REST APIs to memorize, no lengthy training. Get started in minutes, not weeks.

Code-free automation doesn't mean, by the way, that code is left out of the picture. With Torq, every workflow you create is stored as code under version control. But you don't need to use code to write, update, or deploy the workflows. Torq gives you code when you need it, and a code-free experience when you don't.

See Torq in action for yourself by **requesting a demo**.

Torq is a no-code automation and orchestration platform for security and operations. We empower frontline security teams in their journey to becoming more efficient by allowing them to automate processes using our easy workflow builder, limitless integrations, and numerous prebuilt templates.

Built as an enterprise-grade software-as-a-service, Torq can be adopted with ease, delivering results within minutes, unlike traditional security automation solutions that require weeks or months of investment prior to providing value.