



## Security Journey Case Study

# HackerOne Transforms Secure Coding from Compliance Requirement to Engineering Culture

## Executive Summary

At HackerOne, security is the business. That created a different kind of expectation internally: secure coding education could not just exist to satisfy compliance. It needed to actively make engineers better.

The reality, however, was more familiar than expected. Training was in place and technically complete, but engagement was inconsistent and often driven by reminders and deadlines rather than genuine learning.

By working with Security Journey, HackerOne reframed security education as a continuous, developer-centered experience. The shift focused less on enforcement and more on participation, visibility, and reinforcement.

Over time, that change moved training from something engineers had to do into something they engaged with, discussed, and competed around. The program scaled across approximately 100 engineers and became part of onboarding, team culture, and how HackerOne defines its own standard for security.

**We were compliant, but that doesn't mean we were stronger."**

– Martzen Haagsma, Security Engineer

### THE CUSTOMER

**HackerOne, Inc.**

Cybersecurity Platform

Continuous Threat Exposure Management

### CUSTOMER ROLE

**Martzen Haagsma**

Security Engineer



# hackerone

## AT A GLANCE



### CHALLENGE

Training was compliant, but not engaging or meaningfully connected to everyday work.



### SOLUTION

Reframed security education with clear structure, visibility, and developer-driven energy.



### IMPACT

Stronger engagement, more consistent participation, and a culture that celebrates security learning.



### SCALE

Program adopted across ~100 engineers and embedded into onboarding and team culture.



## The Starting Point: “Compliant, But Not Stronger”

When Martzen Haagsma took a closer look at security training, the program was functioning on paper but falling short in practice.

Developers were completing required modules, but the experience felt disconnected from their day-to-day work. Regular notifications drove activity, but also frustration. Over time, engineers became more skilled at managing the system than learning from it.

---

As Martzen puts it: **“We were compliant, but that doesn’t mean we were stronger.”**

---

That distinction became the catalyst for change. At a company built on identifying vulnerabilities, simply meeting requirements was not enough. The expectation was to lead by example, internally as well as externally.

## Reframing the Problem: From Obligation to Ownership

Rather than increase pressure or enforcement, Martzen took a different approach. She reframed the core problem.

Security education was not failing because it was mandatory. It was failing because it did not feel meaningful, visible, or connected.

The goal shifted from driving completion to building engagement and momentum.

This meant redefining success:

- Not whether training was completed
- But whether people came back
- Whether they talked about it
- Whether it changed how they thought about security

---

**“Security education is the first layer of everything we do.  
If that layer doesn’t work, the rest doesn’t matter.”**

---

## Building the Program: Structure, Then Energy

The first step was to establish a clear and credible baseline. Using Security Journey, HackerOne defined a mandatory set of foundational learning paths across web, API, and emerging AI security topics. This created a shared starting point across the engineering organization and aligned expectations at scale.

Importantly, that baseline was positioned as a foundation, not a finish line. Engineers were encouraged to continue into additional paths based on interest and role.

With structure in place, the focus shifted to something far more important: **energy**.



## The Inflection Point: Making Learning Visible

The biggest breakthrough did not come from changing content. It came from changing how learning showed up inside the organization.

Using Security Journey's API, Martzen built a simple Slack workflow that announced when someone completed training. What began as a small experiment quickly shifted behavior.

---

**"The moment it showed up in Slack, everything changed."**

---

Instead of silent completion, progress became visible. Engineers started congratulating each other. Curiosity replaced resistance. Participation became social.

In some cases, it even became competitive.

---

**"People wanted to be on the leaderboard. They wanted to beat their manager."**

---

This moment marked a clear turning point. Training was no longer something happening in the background. It became something the organization could see, celebrate, and build momentum around.

## Scaling the Impact: Making It Stick

With engagement improving, HackerOne focused on making the program sustainable.

Security Journey was embedded into onboarding so that new engineers encountered it early, before forming negative assumptions. This helped normalize training as part of the engineering experience rather than an external requirement.

At the same time, reporting and tracking shifted away from simple completion metrics. Martzen focused on signals that reflected real engagement, including how many people resumed training after pausing.

---

**"Someone coming back and doing one more module mattered just as much as finishing everything."**

---

This shift reinforced the core philosophy: consistency and participation matter more than perfect completion rates.



## What Changed: Observable Outcomes

The impact of the program showed up in clear and consistent ways across the organization.

Training activity shifted from being driven by reminders to being driven by internal motivation. Instead of clusters of activity near deadlines, participation became steadier over time.

Engineers who had previously disengaged began returning to the platform. Conversations about training and security became more common in shared channels. Managers found it easier to support time spent learning because the value was more visible.

Perhaps most importantly, security education began to feel like part of HackerOne's identity.

---

**“The first sign this worked wasn’t a report. It was engineers congratulating each other.”**

---

That sentiment was reinforced by one piece of feedback that stood out: **“This looks like us.”**

---

For a company that sets the standard for others, that alignment mattered as much as any metric.

## Looking Ahead: Security Education in an AI-First World

As HackerOne continues to evolve, the role of education is expanding alongside it.

AI is now part of the company's operating model, and security education is adapting to match. Foundational training includes AI and LLM topics, and there is growing interest in tailoring learning experiences through assessments and role-specific paths.

The underlying philosophy remains the same.

---

**“Education is still the first layer, especially in an AI world.”**

---

Tools will continue to change. The need for shared understanding will not.

## Partnership with Security Journey

Security Journey played a key role not just in delivering content, but in enabling flexibility.

The platform allowed HackerOne to build custom workflows, experiment with engagement strategies, and adapt the program over time. Equally important was the hands-on support provided throughout the process.

---

**“I didn’t just get a platform. I got a partner who helped me make this work.”**

---



## Advice to Other Organizations

Martzen's perspective is direct and grounded in experience:

**“Just do it. Security education isn’t optional. But it also shouldn’t stop at compliance. Make it engaging, make it visible, and give people room to go deeper.”**

### Key Takeaways



- Security education drives impact when it changes behavior, not just completion
- Visibility and positive reinforcement are stronger than enforcement
- Embedding learning into onboarding and culture creates long-term adoption
- Engagement is best measured through participation, not just metrics