Missions.

Missions

Reduce risk, cost and time-to-release with developer-centric interactive learning that makes secure coding second nature.

Secure Code Warrior Missions simulate real-world scenarios for developers to experience the impact of poor code, making them better at defending your code and brand reputation.

Just as pilots train in a flight simulator, learning to recognize and pre-empt risk, Missions helps developers increase their secure coding knowledge, and prevent poor coding practices before they occur – resulting in faster delivery of features with fewer security issues.

Learn how to turn your developers into security aces.

SECURE
CODE
WARRIOR

securecodewarrior.com

Empower your developers to become your first line of defense.

A defensive security posture

Missions provide contextual, interactive learning that beats other forms of training hands-down, allowing developers to hone transferable skills that are immediately relevant to real-life situations.

Interactive, developer-centric training

Missions are kind of like a flight simulator, for coders – a place they can pro-actively experience the impact of poor code practices and vulnerabilities, in a safe and secure environment.

In Missions, developers see both the live code and the front-end application on-screen at the same time. They see the impact of vulnerabilities and the impact of the code they write to fix them, in real-time.

This is as close to the real thing as you can get. No other developer training even comes close.

By taking on hyper-relevant missions based on real-world scenarios, developers gain a security-first mindset that helps prevent security issues in the first place.

Skills gained are transferable to other apps

With Missions developers learn by overcoming challenges in a pragmatic way. Applied learning develops skills that are transferable to other apps – unlike competitive offerings that are one-off, or restrictive linear flow.

Level-up once… then do it again

Training fatigue is prevented with progressively challenging missions, which build secure code muscle memory, step-by-step. Missions are deeply immersive, highly interactive, and engaging enough to interest senior developers.

These are skills that are not just learned once and forgotten, but earned and embedded. That makes Missions an incredibly innovative way to engage developers and enhance a security posture and culture that delivers quality software faster for customers.

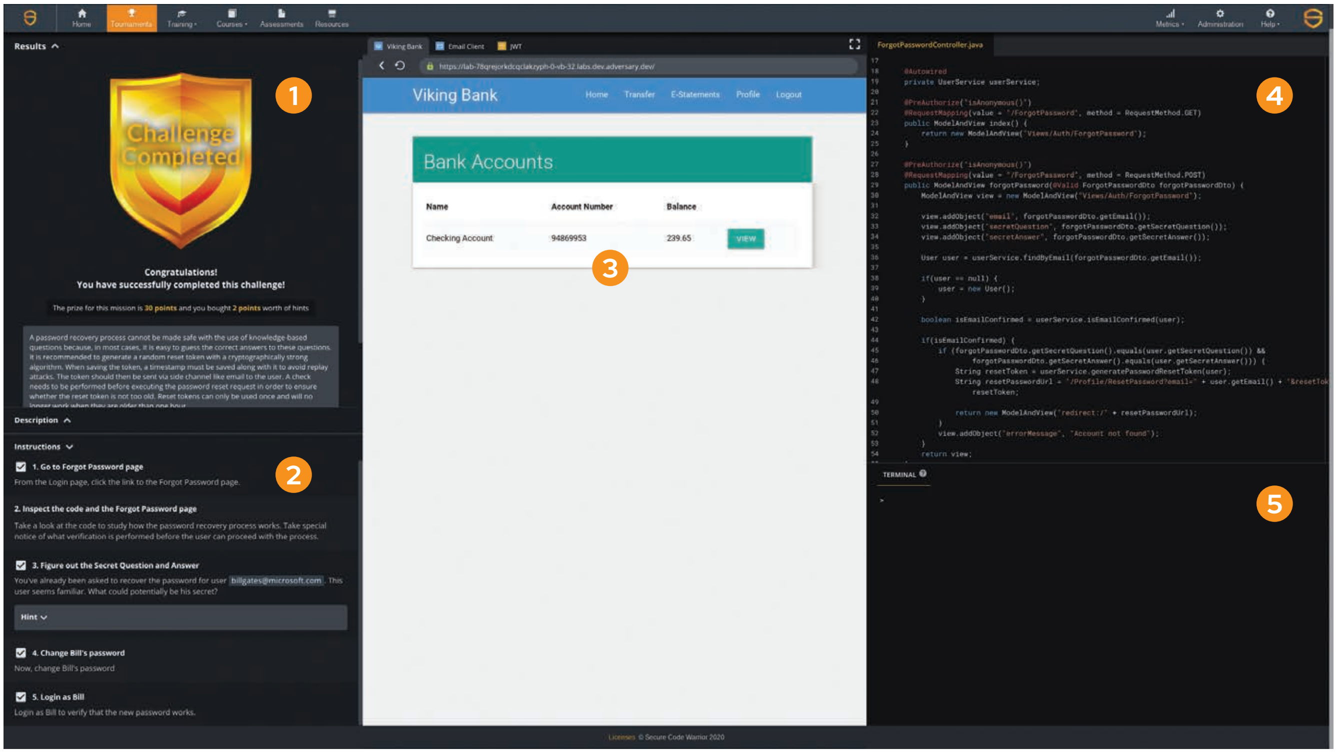
This approach has huge potential upside for your business

Missions offers 40 missions playable in your choice of language:framework targeting common vulnerabilities you see every day.

When security becomes second nature:

- Developers are more efficient and more skilled at defending your code and your brand reputation.
- AppSec and engineering teams' goals are unified.
- Code quality is enhanced and product is delivered faster.
- Customer trust and loyalty is reinforced.

Experience the impact of vulnerabilities and experiment with secure coding solutions, in real-time!



Congratulations!
You have successfully completed this challenge!

The prize for this mission is 30 points and you bought 2 points worth of hints

A password recovery process cannot be made safe with the use of knowledge-based questions because, in most cases, it is easy to guess the correct answers to these questions. It is recommended to generate a random reset token with a cryptographically strong algorithm. When saving the token, a timestamp must be saved along with it to avoid replay attacks. The token should then be sent via side channel like email to the user. A check needs to be performed before executing the password reset request in order to ensure whether the reset token is not too old. Reset tokens can only be used once and will no longer work when they are older than one hour.

Description

Instructions

☑ 1. Go to Forgot Password page

From the Login page, click the link to the Forgot Password page.

2. Inspect the code and the Forgot Password page

Take a look at the code to study how the password recovery process works. Take special notice of what verification is performed before the user can proceed with the process.

☑ 3. Figure out the Secret Question and Answer

You've already been asked to recover the password for user billgates@microsoft.com. This user seems familiar. What could potentially be his secret?

☑ 4. Change Bill's password

Now, change Bill's password

☑ 5. Login as Bill

Login as Bill to verify that the new password works.

https://lab-78qrejorkdcqclakzyph-0-vb-32.labs.dev.adversary.dev/

Bank Accounts

Name	Account Number	Balance	
Checking Account	94869953	239.65	VIEW

```
    @Autowired
    private UserService userService;

    @PreAuthorize("isAnonymous()")
    @RequestMapping(value = "/ForgotPassword", method = RequestMethod.GET)
    public ModelAndView index() {
        return new ModelAndView("Views/Auth/ForgotPassword");
    }

    @PreAuthorize("isAnonymous()")
    @RequestMapping(value = "/ForgotPassword", method = RequestMethod.POST)
    public ModelAndView forgotPassword(@Valid ForgotPasswordDto forgotPasswordDto) {
        ModelAndView view = new ModelAndView("Views/Auth/ForgotPassword");

        view.addObject("email", forgotPasswordDto.getEmail());
        view.addObject("secretQuestion", forgotPasswordDto.getSecretQuestion());
        view.addObject("secretAnswer", forgotPasswordDto.getSecretAnswer());

        User user = userService.findByEmail(forgotPasswordDto.getEmail());

        if(user == null) {
            user = new User();
        }

        boolean isEmailConfirmed = userService.isEmailConfirmed(user);

        if(isEmailConfirmed) {
            if (forgotPasswordDto.getSecretQuestion().equals(user.getSecretQuestion()) &&
                    forgotPasswordDto.getSecretAnswer().equals(user.getSecretAnswer())) {
                String resetToken = userService.generatePasswordResetToken(user);
                String resetPasswordUrl = "/Profile/ResetPassword?email=" + user.getEmail() + "&resetTok
                    resetToken;

                return new ModelAndView("redirect:/" + resetPasswordUrl);
            }
            view.addObject("errorMessage", "Account not found");
        }
        return view;
```

```
>
```

Licenses © Secure Code Warrior 2020

1.

Description of mission and results.

2.

Instructions provide detailed steps, and potential hints available to complete the mission.

3.

Front-end application to simulate the end-user experience.

4.

Interact with the application source code in real-time and see the impact.

5.

Terminal console to view system logs and identify potential issues.

Why Secure Code Warrior?

- **Hands-on, developer-focussed training:** Contextual, interactive training that grows the developers' skills and knowledge.
- **Extensive content library:** Over 50 language:framework-specific categories, including; Web, Mobile, IaC, API, Backend/Frontend.
- **Tournaments:** Gamified training that makes learning fun and keeps developers engaged.
- **Assessments** to benchmark and measure secure coding competency within your team.
- **Enterprise all-in-one platform:** Robust reporting and data insights to guide and target your program. Enterprise API to easily integrate with your line-of-business systems.
- **Customer success and support:** Assigned Customer Success Manager to help drive program goals.

Try an online demo:

Take off on our public mission and learn how GitHub was exposed

securecodewarrior.com/fly-a-test-mission

SECURE
CODE
WARRIOR