



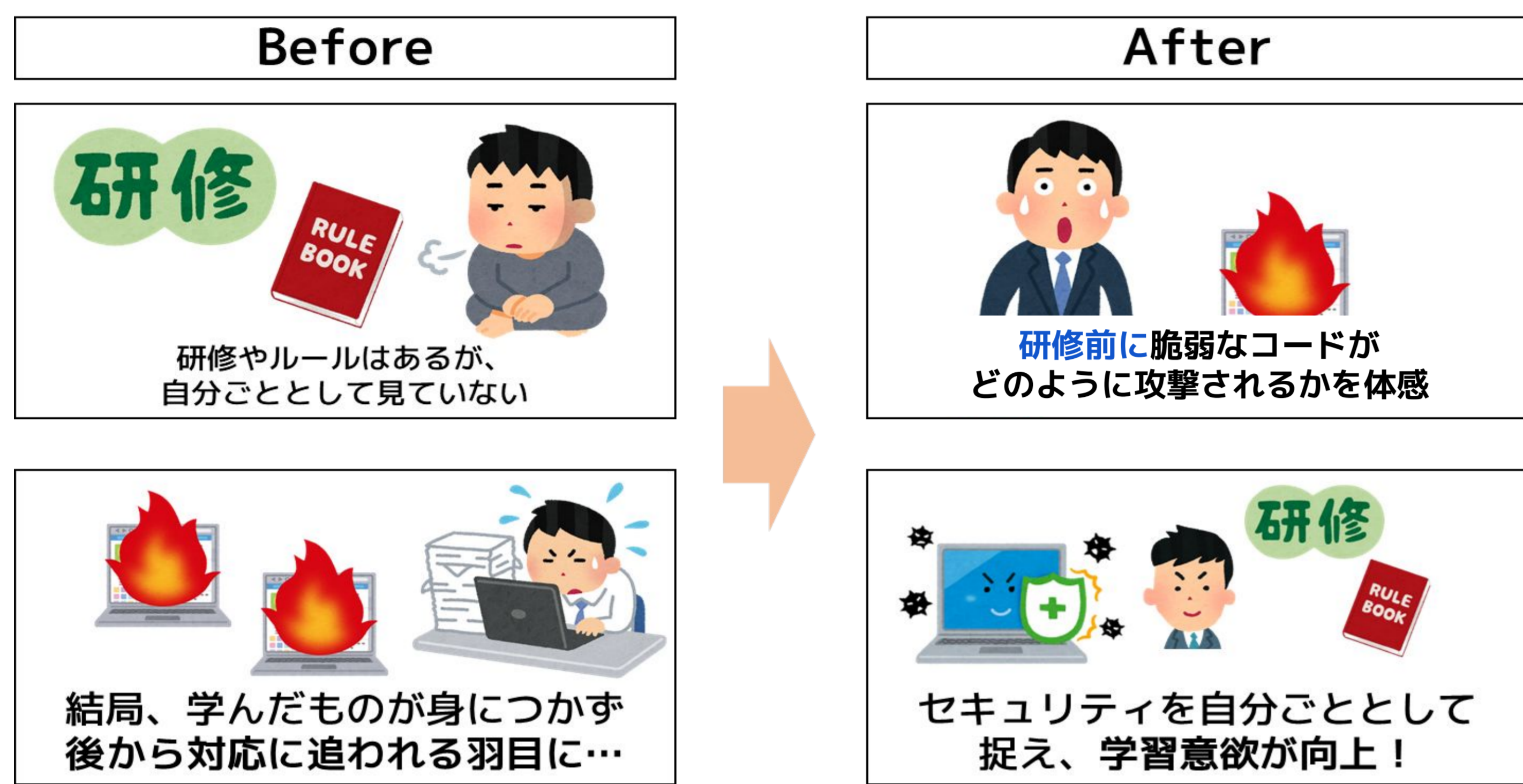
GitHub repository QR

Webエンジニアのためのセキュリティ“体感”ツール まもコード

表現駆動コース4C
永井美輝/小野寺恵理子/片岡昂晴/佐藤心

まもコードとは

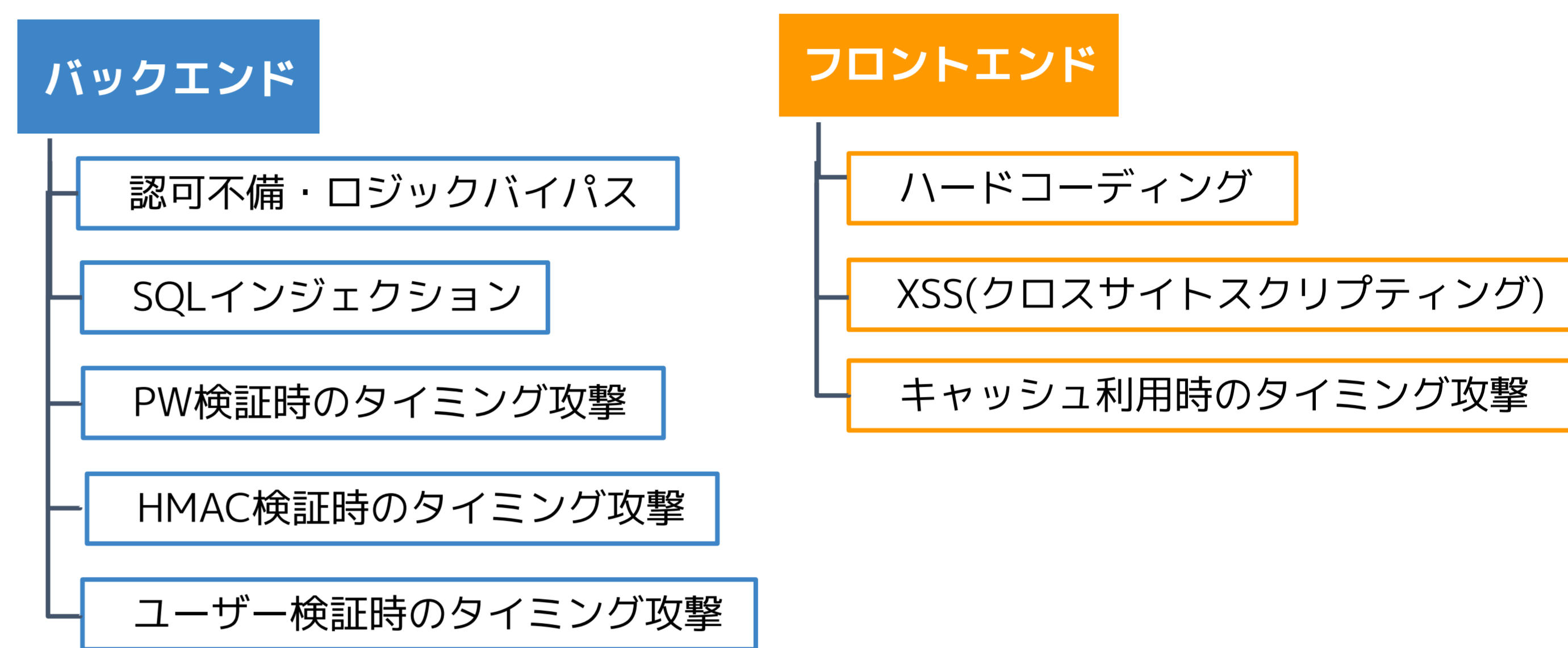
研修前に、学習者が**実際の攻撃者の視点**に立ち、脆弱なコードを実装するとどのように攻撃されるかを**体感**してもらうためのツールです。**セキュリティを自分ごととして捉え、学習意欲の向上**を目的としています。さらに、単に「危険である」と示すだけでなく、**どのように修正すれば安全になるのか、適切な実装とは何か**まで学べる構成にしています。



学習内容

まもコードは、**Webエンジニア**が実務に近い視点で学べるよう、バックエンドとフロントエンドに分けた構成になっています。前提知識は必要ありません。テーマは、XSSやSQLインジェクションなど実際のWebサービスでよく見られる脆弱性から、初学者には直感的に分かりにくい実装上の問題まで幅広く扱っています。

よくある脆弱性に加え、実装に踏み込んだテーマも含めることで、開発現場で求められる「判断の難しさ」を体験できる構成にしています。



学べることの例

(認可不備・ロジックバイパス)

体感できるヒミツ1：
構えずに学べる設計！

専門用語を前面に出さず、冒頭の疑問を噛み砕いていきます。そして、実装次第で危なくなってしまうことを理解してもらいます。

体感できるヒミツ2：
攻撃者視点を体験！

冒頭に**質問形式**で学習者の疑問を提示します。

脆弱な実装と安全な実装の**実際のコード**を見ながら、**デモで挙動の違い**を体験します。コードの違いだけでなく、動かしたときに何が起るのかを確認することで、**どのように修正すれば安全になるのか、適切な実装とは何か**、を知識ではなく「**体感**」として捉えられるようにしています。

体感できるヒミツ3：
「比較」で直感的に学べる！

最後に冒頭の疑問の回答を復習もかねて確認します。

1章の見どころ

- URLを少し変えるだけで、別の操作ができてしまうことがある？
- 「見えない操作」を勝手に実行できる、何が起きる？
- たった1つの確認を足すだけで、危なさはどう変わる？

→ 実際に操作しながら、違いを体験します

なぜ「URLを変えるだけ」で危なくなるの？

Webアプリのバックエンドでは、「ログインしているかどうか」だけを確認して処理を進めようと、思わぬ問題につながる場合があります。実はもう一つ、「この人が、この操作をしていいか」を確認することがとても重要です。ここが抜けていると、URLを少し変えただけで本来は許されていない操作が実行できてしまうことがあります。

まず押さえたい「2段階の確認」

バックエンドでは、リクエストを受けたときに順番に確認すべきことがあります。

① 誰からのリクエスト？

「このリクエストは、ログイン済みのユーザーから来ているか？」を確認します。まず本人かどうかを確認する段階です。

② その操作をしていい？

「このユーザーが、このデータや操作に触っていいか？」を確認します。本人でも、できる操作は限られることに注意します。

よくある誤解

「画面にボタンがないから実行できないはず」と思いがちですが、URLを直接指定されると、バックエンドの処理はそのまま進みます。そのため、バックエンド側で毎回チェックする設計が必要になります。

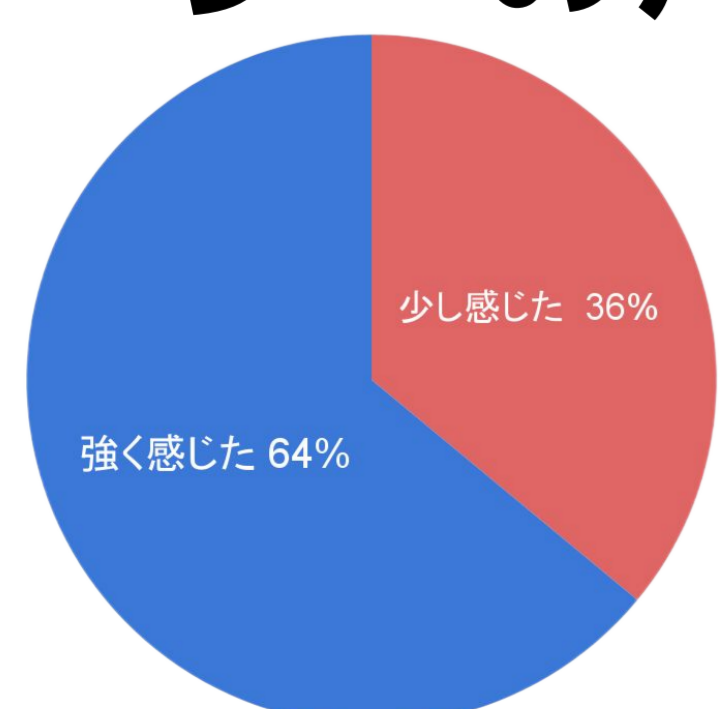
1章のまとめ

この章のはじめに投げた問い、答えはこうでした

- Q. URLを少し変えるだけで、別の操作ができてしまうことがある？
- あります。URLやパラメータをそのまま変えると、想定外のデータや操作が働いてしまいます。
- Q. 「見えない操作」を勝手に実行できると、何が起きる？
- 画面にボタンがなくても、削除や更新などの重要な操作が働いてしまうことがあります。
- Q. たった1つの確認を足すだけで、危なさはどう変わる？
- 「この人がこの操作をしていいか」を毎回確認するだけで、通ってはいけないリクエストを止められます。

上記の流れをテーマごとに体験することで、セキュリティを「知識」ではなく、**自分の実装に関わる問題**として捉えられるようになります。

ユーザーの声



セキュリティを学ぶ重要性をほとんど感じていない社会人および大学生のWebエンジニア、合計11人にまもコードを体験してもらい、アンケートを実施しました。「セキュリティを自分の実装の問題として感じるようになりましたか？」という質問に対して、「**強く感じた**」と答えた方が**64%**、「**少し感じた**」と答えた方が**36%**という結果になりました。これは、知識を増やしたことによる変化ではなく、**攻撃の仕組みと適切な修正を先に体感したこと**で、「**なぜ学ぶのか**」「**自分はどう実装すべきか**」が具体的に結びついた結果だと考えています。

また、「UIが見やすい」「コードを比較できる構成が分かりやすい」「専門用語も図で説明されていて理解しやすい」「ハードルをあまり感じることなく楽しく学べた」といった意見も得られました。



今後の展望

SecHack365 Returnsにてセキュリティ研修の前に行うツールとしてより効果的なテーマ選びや難易度設計を目標に開発を進めて行こうと考えています。また、フィードバックやアンケート結果を参考に改良していく予定です。