SecHack365

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

GitHub repository QR

表現駆動コース4C
永井美輝/小野寺恵理子/片岡昴晴/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。

Before

研修やルールはあるが、
自分ごととして見ていない

結局、学んだものが身につかず
後から対応に追われる羽目に…

After

**研修前に脆弱なコードが
どのように攻撃されるかを体感**

セキュリティを自分ごととして
捉え、学習意欲が向上！

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。

バックエンド
- 認可不備・ロジックバイパス
- SQLインジェクション
- PW検証時のタイミング攻撃
- HMAC検証時のタイミング攻撃
- ユーザー検証時のタイミング攻撃

フロントエンド
- ハードコーディング
- XSS(クロスサイトスクリプティング)
- キャッシュ利用時のタイミング攻撃

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！

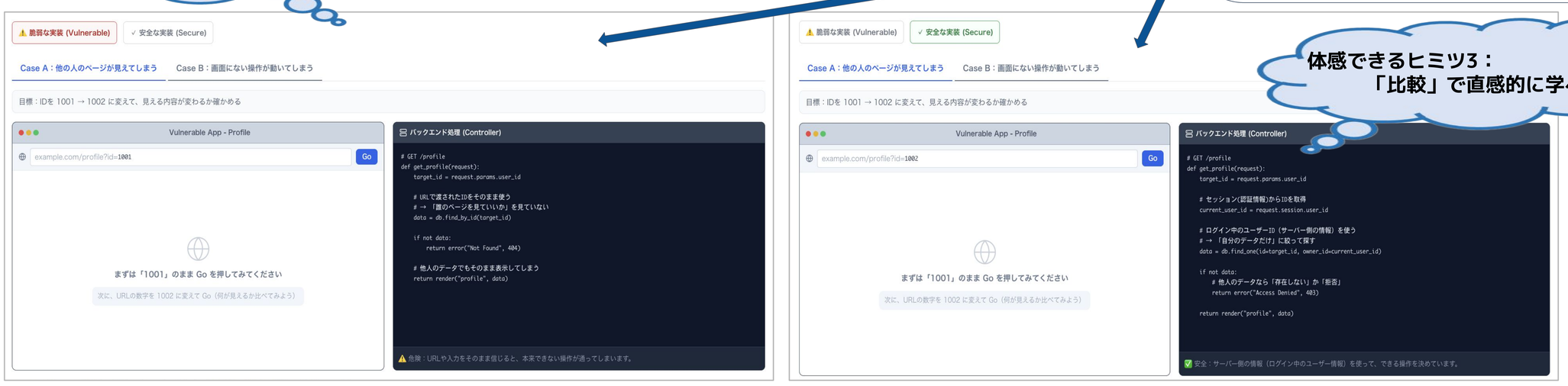
専門用語を前面に出さず、冒頭の疑問を噛み砕いていきます。
そして、実装次第で危なくなってしまうことを理解してもらいます。

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こ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2：
攻撃者視点を体験！

体感できるヒミツ3：
「比較」で直感的に学べる！

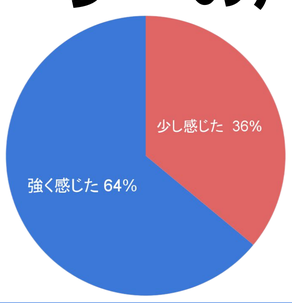


1章のまとめ

最後に冒頭の疑問の回答を復習もかねて確認します。

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。