

安全なウェブサイトのために

「安全なウェブサイトの作り方」シリーズ

<https://www.ipa.go.jp/security/vuln/websecurity.html>

脆弱性を作りこまない開発

脆弱性対策

1.1 SQL インジェクション

データベースと連携したウェブアプリケーションの多くは、利用者からの入力情報を基に SQL 文(データベースへの命令文)を組み立てています。ここで、SQL 文の組み立て方に問題がある場合、攻撃によってデータベースの不正利用をまねく可能性があります。このような問題を「SQL インジェクションの脆弱性」と呼び、問題を悪化した攻撃を、「SQL インジェクション攻撃」と呼びます。

SQL インジェクションの脆弱性がある場合、悪意あるアクセスにより、データベースの不正利用をまねく可能性があります。

悪意のある人: データベースへの命令文を悪用する入力値を送信

ウェブアプリ: データベースへの命令文を送信

データベース: 悪意ある SQL インジェクションの脆弱性があるウェブアプリケーション

失敗修正例

■ PHP による登録情報編集機能

【脆弱な実装】

下記は、会員制ウェブサイトにおける、ユーザの会員登録情報を変更する機能の、典型的な実装例です。ここでは、ユーザが住所を東京都から大阪府に変更するときの操作を例にしています。

画面1: http://00example.jp/ 会員登録情報編集

画面2: http://00example.jp/ 会員登録情報の編集

画面3: http://00example.jp/ 会員登録情報の編集確認

画面4: http://00example.jp/ 会員登録情報の編集完了

実態調査

5. DBMS製品の実態調査

5.1. 調査内容

前項までの検討を踏まえ、ウェブアプリケーション開発で実際に使用されること多い DBMS とプログラム言語の組み合わせを対象に、以下の点について実態を調査しました。

- ・プレースホルダの実装は動的プレースホルダ(準備された文)か、静的プレースホルダか
- ・quote メソッドは正しく処理されるか
- ・文字エンコーディングの扱い

5.2. Java + Oracle

Java から Oracle を呼び出すには通常、JDBC を使用します。Oracle 用 JDBC は数種類提供されていますが、この調査では、Oracle が提供する ojdbc.jar を用い、データベースの文字エンコーディングを UTF-8 に設定しました(付録 A.4 参照)。

【調査結果】

項目	調査結果
プレースホルダの実装	静的プレースホルダのみ
動的プレースホルダのエスケープ処理	調査対象(動的プレースホルダは提供されていない)
quote メソッドの処理	調査対象(quote メソッドは提供されていない)
文字エンコーディングの扱い	DB 接続には UTF-8 が使用される

「安全なウェブサイトの作り方」では

- ・11種類の脆弱性の説明とその対策を説明
- ・さらに、ウェブサイトの運用面からウェブサイト全体の安全性を向上させるための方策を説明
- ・また、ウェブサイトにおけるパスワードの運用方法を説明

「安全なSQLの呼び出し方」では

- ・SQLインジェクションの脆弱性を作りこまないために、実態調査を踏まえて具体的に安全なSQLの呼び出し方を説明。



開発したウェブサイトに対して、基本的な脆弱性対策ができていいるかどうかを、必要かつ最小限の項目で診断。

対策できていない点を見つける

診断基準

2.2. 危険度基準

各脆弱性に行方している危険度のレベルは、以下の基準に則って提示しました。

危険度	説明
高危険度	被害者ユーザの関与がなくても攻撃者が遠隔アプリケーションに対して攻撃可能である脆弱な脆弱性。攻撃を受けると、大量の情報漏洩や改ざんの被害を生じることがある。
危険度「中」	攻撃成功には被害者ユーザの関与(攻撃者のリンクをクリックする等)が必要である脆弱な脆弱性。若しくは脆弱な脆弱性であっても大量の情報漏洩や改ざんにはつながりにくいもの。
危険度「低」	攻撃成功の確率が低い若しくは攻撃が成功しても被害が軽微であると考えられる脆弱性。ただし、確率は低いものの被害に遭う可能性はある。

2.3. 総合判定基準

脆弱性が発見された場合、「総合判定用表」が

- ・「要治療・精密検査」
- ・「要し支えない」
- ・「異常は検出されなかった」

のいずれかとなります。いずれの「総合判定用表」になるかは、以下の基準に基づきます。

総合判定用表	要治療・精密検査
説明	危険度が「高」又は「中」の、明らかに危険な脆弱性が検出された。ウェブアプリケーションの改修等の措置を講じる必要がある。また、検査箇所以外にも危険な脆弱性が発見される可能性がある。
基準	脆弱性(AI~(M)の13項目のうち、1つでも脆弱性が発見された場合。ただし、「要し支えない」の判定基準にある脆弱性以外のもの(危険性が低い脆弱性)
総合判定用表	要し支えない
説明	今回の診断では危険度が「低」の脆弱性のみが検出された。現状ですべて被害に及ぶ可能性は低く、運用上は要し支えないと判断されるが、本件は注意が必要であり設置しない方がよい。
基準	下記4つの脆弱性 (E)、(J)、(K)、(M)のみが検出された場合。 (E) デレンクドリスティング (P.12 参照) ただし、重要な情報(個人情報や機密情報)が検出された場合は、既に危険な状態といふことから、「要治療・精密検査」とします。 (J) 認証 (P.16 参照) ただし、検出パターン1かつ2が検出された場合もしくは検出パターン5が検出された場合は、危険度が高いので「要治療・精密検査」とします。 (K) セッション管理の不備 (P.18 参照)

「ウェブ健康診断」を使用すれば、ウェブアプリケーションで基本的な脆弱性対策ができていいるかどうかを確認できます。13種類の脆弱性について、具体的な検出パターンと、脆弱性有無の判定基準を記載しています。これまで、地方公共団体 1,200 団体の診断に活用(LASDEC)された実績があります。

検査項目

検出パターン	脆弱性有無の判定基準	異常 脆弱性有無の判定基準(その他)	対象製品(脆弱)
1 「」(シングルクォート)の1つ	エラーに成る	レバンスにDBMS等が提供するエラーメッセージ(例: SQLException, query failed)が検出された場合にエラーが発生したと判定します(要し支えない)。	DB アクセス
2 「'」(アポストロフ)の1つ	エラーに成る	同1と同様にエラーメッセージ(例: SQLException, query failed)が検出された場合にエラーが発生したと判定します(要し支えない)。	
3 「'」(アポストロフ)の2つ	エラーに成る	同1と同様にエラーメッセージ(例: SQLException, query failed)が検出された場合にエラーが発生したと判定します(要し支えない)。	

※ ① DBMS のエラーメッセージの検出は以下の通りです。
 ・ DBMS の名前(Oracle, Microsoft SQL Server, IBM DB2, MySQL, PostgreSQL)の全て又は一部が検出される。
 ・ SQL の一部が検出されている。
 ・ シングルクォートが検出されていない等、SQL の構文上の問題が検出されている。
 ・ 他のエラーメッセージとは異なるに脆弱なメッセージ、例えば、通常のエラーメッセージが検出されるのに、異常のメッセージになっている等。

② (A) SQL インジェクションの脆弱性ありと判定された場合は「安全なウェブサイトの作り方」中の「1.1 SQL インジェクション」を参考に、脆弱性を修正してください。

ウェブ健康診断仕様

「安全なウェブサイトの作り方」別冊

注意事項
本診断は検査パターンを絞り込んだ診断ですので、安全宣言には繋がりません。



IPA 独立行政法人情報処理推進機構
セキュリティセンター

2012年12月

※本診断は、検査パターンを絞り込んだ診断ですので、安全宣言には繋がりません。