情報ネットワークⅡ（情213）
【第13回】
Webセキュリティ（2）
（教科書：第8章）

担当教員：長田智和

E-Mail: nagayan@ie.u-ryukyu.ac.jp

URL: http://n-lab.info/

（講義日：2018年1月18日）

第8章：Webセキュリティ（2）

8.4 SQLインジェクションとその対策

- 8.4.1 SQLインジェクションとは
  - 不正なSQL文を実行させることで、Webアプリケーションサーバーやデータベースを不正に操作する攻撃方法
  - SQLインジェクションが及ぼす脅威
    - 機密情報の漏洩
    - 重要情報の改竄
    - 認証処理のバイパス（不正な迂回）
- （教科書p.226の図8.17を参照）

8.4 SQLインジェクションとその対策

- 8.4.2 SQLインジェクションの仕組み
  - 正常動作の例（教科書で説明）
  - 攻撃を受ける動作の例
    - 不正なSQL文を構成させて、開発者が意図しない動作を実行させる。（教科書の例では、DBのuserから全ての情報が取得されてしまう）
- （教科書p.227-228の図8.18-8.19を参照）

8.4 SQLインジェクションとその対策

- 8.4.3 SQLインジェクションの対策
  - バインドメカニズムの利用
    - 解析済みの安全なSQL文を予めDBMS側に送信しておき、利用者からの入力値をDBMSに送信し、ブレースホルダにバインドしてSQL文を実行する方法
  - エスケープ処理の実施
    - XSS対策と同様なエスケープ処理。ただし、処理が煩雑であるため、バインドメカニズムの利用が推奨
- （教科書p.229-230の図8.20-8.21を参照）

8.5 CSRF攻撃とその対策

- 8.5.1 CSRF（Cress Site Request Forgeries）攻撃とは
  - 攻撃者が用意したWebサイトの閲覧中に、利用者の意図に関係なく、別のWebサイトで何らかの操作を行わせる攻撃
- 8.5.2 CSRF攻撃の仕組み
  - 正常動作の例（教科書で説明）
  - 攻撃を受ける動作の例（教科書で説明）
- （教科書p.231-232の図8.22-24を参照）

8.5 CSRF攻撃とその対策

- 8.5.3 CSRFの対策
  - CSRF攻撃の対策１
    - リクエストに秘密情報を埋め込むことで、秘密情報がないリクエストは正規のリクエストとはみなさない。
  - CSRF攻撃の対策２
    - 重要な操作の際は、操作の確認やパスワードの再入力を利用者に求める。
  - CSRF攻撃の対策３
    - Refererヘッダを確認する。（但し、Refererヘッダを返さないブラウザやプロキシを利用している場合は効果がない）
- （教科書p.232-234の図8.25-8.27を参照）

8.6 Web2.0技術のセキュリティ

- Web2.0とは
  - 2000年代半ば頃から、それまでのWeb技術に対して格段に進化したWeb技術をWeb2.0技術と呼ぶ。
  - Web2.0は具体的な技術名ではなく、古いWeb技術に対して相対的に新しいWeb技術を指す。
- 8.6.1 Web2.0のセキュリティ
  - Webサーバーやその OSだけではなく、クライアントやその上で動作するWebブラウザ、Web APIなど、Webセキュリティの要素技術は複雑化し、範囲は広がっている。
- （教科書p.235の図8.28を参照）

8.6 Web2.0技術のセキュリティ

- AjaxとHTML5
  - Ajaxは対話型Webアプリケーションの実装形態
    - Webブラウザに実装されている機能を使い、Webページのリロードを伴わずにWebアプリケーションサーバーとデータ通信を行い、処理を進めることができる。
    - AjaxはJavaScript, XML, CSS, DOM（Document Object Model）など、従来からある要素技術をベースとした集合体
  - DOM（Document Object Model）
    - プログラム（スクリプト）からWebページの文書内容や構造、スタイルに動的にアクセス・更新できる、プラットフォームに依存しないインターフェース。

8.6 Web2.0技術のセキュリティ

- 8.6.2 DOM Based XSS
  - DOMを通じたHTML操作の結果、意図しないスクリプトが実行されたり、それを許す脆弱性のこと。
  - Webアプリケーションサーバーと連携して、Webページを動的に更新する機能で、この脆弱性が入り込む。
- DOMによる動的なブラウザの表示情報更新
  - Webブラウザ上で実行されるJavaScriptの脆弱性を悪用
  - 注意すべき関数を用いる際はエスケープ処理が必要

    （注意すべき関数は教科書を参照）

8.6 Web2.0技術のセキュリティ

- 8.6.2 DOM Based XSS
  - eval()を用いた動的なスクリプトの実行
    - eval()は与えられた文字列をJavaScriptのコードと解釈して実行できる。
    - Webアプリケーションサーバーからのレスポンスに不正なコードが入っていた場合、適切なエスケープ処理が行われていないと、そのコードが実行されてしまう。

8.6 Web2.0技術のセキュリティ

- 8.6.3 同一生成元ポリシー
  - XHR（XMLHttpRequest）では、メインコンテンツとスクリプトを取得したドメイン（オリジン）と、オリジンとは異なるドメイン（クロスオリジン）との通信を認めないという考え方。
  - 同一オリジンの判定例（表8.3）
  - XHR2
    - 許可ヘッダ（Access-Control-Allow-Origin）が適切に設定されている場合のみ通信を可能とする。
- （教科書p.240-241の図8.29-8.30,表8.3を参照）

【次回予告】
第14回
演習（2）
（期末試験対策）

また来週！