クロスサイトスクリプティング(XSS)および
Javaソースコードへの影響について学ぶ

クロスサイトスクリプティング(XSS)とは？

Javaの場合

概要

クロスサイトスクリプティング(XSS)は、アプリケーションがHTTPリクエスト(URL、URLパラメータ、フォームフィールド、ヘッダ、Cookie、ボディ)から信頼性のないデータを受け取り、HTMLコンテキスト(ボディ、属性、スクリプト、スタイルなど)のコマンドとしてではなくテキストとして解釈されるように、特殊文字を適切にエスケープせずにWebページに書き込むことで発生します。

信頼性のないソースからHTTPレスポンスまで、信頼できないデータがたどるフローは、アプリケーションフレームワーク、ビジネスロジック、データ層、ライブラリ、入り組んだコードパスなどによって、XSSが見えにくく非常に複雑になることがあります。

XSSは、以下の3つに分類されます:

反射型**XSS**

反射型XSSは、悪意のあるスクリプトが(通常はURLで)被害者となるユーザに送信されると、被害者のブラウザでそのスクリプトが脆弱なアプリケーションに転送され、アプリケーションによって被害者のブラウザに攻撃が送られて実行されることで発生します。

格納型**XSS**

「持続型XSS」としても知られるこのタイプは、多くの場合、他のタイプよりも危険です。格納型XSSは、攻撃者が脆弱なアプリケーションに悪意のあるスクリプトを送信し、そのアプリケーションでそのデータが(おそらくデータベースに)保存されることで発生します。そ

の後、この悪意のあるスクリプトが被害者のブラウザに送り返され、そこで実行されてしまうのです。

DOMベースXSS

DOMベースのXSSは、完全にユーザのブラウザで発生し、サーバ側のアプリケーションは関与しません。DOMベースのXSS攻撃は、URLからの情報などの信頼できない情報がWebアプリケーションに取り込まれ、それがドキュメントオブジェクトモデル(DOM)に書き込まれて実行される場合に発生する可能性があります。

攻撃

XSSの攻撃は非常に一般的です。Webサイトをクロールしてxss攻撃を送り、最終的にそれがHTMLになるかどうかを確認する自動化ツールがたくさんあります。

影響

クロスサイトスクリプティング(XSS)の脆弱性によって、攻撃者は通常、被害者になりすまして、ユーザが実行できる任意の操作を実行し、ユーザのデータにアクセスすることが可能になります。ユーザのログイン資格情報の取得、Webサイトの仮想的な改ざんの実行、メッセージやルック&フィールの変更、トロイの木馬機能をWebサイトに仕込む、悪意のあるユーザがユーザのシステムにアクセスできるようにバックドアを作成する、などが可能になります。

HTMLやJavaScriptの脆弱性を悪用する攻撃者は、ユーザができることはなんでも実行でき、ユーザが参照できるものは何でも(パスワード、支払い、個人の金融情報など)見ることができるようになるため、XSSの脆弱性は特に危険です。

XSSが特に危険なのは、被害者(ユーザと脆弱なアプリケーションの両方)が悪用されたことに気づかないことが多いからです。

深刻な影響：攻撃者が、銀行取引、電子メール、医療記録などの機密データを保持するアプリケーションにアクセスする可能性があります。

重大な影響：セキュリティ侵害を受けたユーザがアプリケーション内の管理者特権を持っていた場合には、攻撃者が脆弱なアプリケーションを完全に制御し、全てのユーザとそのデータが危険にさらされる可能性があります。

JavaでのXSS

パラメータの入力または出力を削除できる場合は、削除してください。それ以外の場合は、パラメータがページのどこでレンダリングされるかに基づいて、適切なエンコード方式を使用してパラメータをエンコードしてください：

コンテキスト	例	危険な文字	エンコーディング	備考
HTML エンティティ	`<div>{untrusted}</div>`	`&<>"'/`	`&#xHH;`	
HTML 属性	`<input value="{untrusted}">`	英数字以外	`&#xHH;`	これは、`href`、`src`、`style`などの複雑な属性や`onclick`.のようなイベントハンドラに対しては安全ではありません。`javascript:`や`data:`などの安全でないURLやCSS表記を避けるために、許可リストによる十分な検証を行う必要があります。
URL パラメータ	`<a href="/?name={untrusted}">`	英数字以外	`%HH`	

CSS	p { color : {untrusted} };	英数字以外	\HH	これは、url、behavior、-moz-bindingなどの複雑なプロパティには安全ではありません。JavaScriptのURLやCSS表記を避けるために、許可リストによる十分な検証を行う必要があります。
JavaScript	var name = '{untrusted}';	英数字以外	\xHH;	JavaScriptの関数には、許可リストによる検証を行わなければ、信頼できないデータを入力として安全に使用できないものがあります。

JSPを使用する場合

```Java
<c:out value=\"${userControlledValue}\"/>... or
...${fn:escapeXml(userControlledValue)}
```

Springタグに関する推奨事項

Springのタグライブラリを使用してテキストを安全に出力する方法を以下に示します:

```Java
<div>
<spring:escapeBody htmlEscape=\"true\">$
{userControlledValue}</spring:escapeBody>
// for data in HTML context
</div>
<script>
<!--
var str = \"<spring:escapeBody javaScriptEscape=\"true\">$
{userControlledValue}</spring:escapeBody>\";
// for data in JavaScript context
-->
</script>
```

おつかれさまでした！

Javaのクロスサイトスクリプティング(XSS)とは何であるか、そしてシステムをXSSから守るにはどうすればよいのかが、お分かり頂けたと思います。ここで学んだ新しい知識をうまく応用しながら、コーディングをしてください。これをあなたのネットワークで自由に共有してください。また、一般的な脆弱性に関する弊社のその他のレッスンも是非ご覧ください。

この学習モジュールに関して修正などがある場合は、こちらをクリックして、プルリクエストを作成してください！

関連記事：

ブログ：DOM XSS in wix.com

ブログ：Find JavaScript cyber-vulnerabilities for free with CodeSec