第四章：サーバサイドリクエストフォージェリ**(SSRF)**

サーバサイドリクエストフォージェリ(SSRF)および
コードを安全にする方法を学ぶ

サーバサイドリクエストフォージェリとは？

JavaScriptの場合

概要

サーバサイドリクエストフォージェリ(SSRF)の脆弱性によって、攻撃者が標的のアプリケーションまたはアプリケーションプログラミングインターフェイス(API)を悪用して、細工されたリクエストが想定外の宛先に送信される可能性があります。そして、脆弱なアプリケーションが一種の攻撃中継へと変わり、攻撃者が内部システムへアクセスすることが可能になります。

SSRFには2つの条件があります：

1. アプリケーションがサーバまたはローカルホストへのリクエストを実行
2. 攻撃者が外部アプリ/ユーザ入力を介してサーバまたはローカルホストを制御

影響

攻撃者はURLを制御するため、アプリケーションをだまして、攻撃対象のネットワーク内部のURLを呼び出すことができます。これにより、内部リソースの列挙、内部専用APIの悪用、`file://`プロトコルの使用によるローカルシステムのリソース流出などにつながる事が考えられます。

攻撃者は、サーバから送信されるHTTPリクエストのURLの部分を制御できます。この部分がホスト名の一部である場合、攻撃者はHTTPリクエストの送信先も制御できてしまいます。

ネットワークの設定により、このHTTPリクエストは外部、内部、ローカルホスト、あるいは(攻撃者がURLのその部分を制御できる場合)**file://**メソッドを使用したローカルファイルに対するものになります。
また、攻撃者はこの脆弱性を悪用して、通常は攻撃者が到達できない内部サーバやローカルホストを列挙して、それらを操作する可能性があります。
さらに、攻撃者はこの脆弱性を悪用して、攻撃者が制御するサーバに対して公開サーバからアクセスできるようにし、HTTPリクエストに含まれるデータやシークレットなど、ユーザ以外に表示されるべきでない情報を漏洩させる可能性があります。

対策方法

可能であれば、ユーザからの入力で、サーバからリクエストされるURLを完全に制御することはやめてください。アプリケーションで、自由形式のテキストフィールドではなく、ユーザが選択できるオプションのリストを表示できる場合があります。
ユーザからの入力でURLを制御する必要がある場合は、リクエストされたURLが許容できるものであることを確認してください。例えば、許可リストを使用すれば、リクエストできるドメイン、IP、メソッド、パスを制限できます。さらに、拒否リストを使用すれば、ローカルホスト、プライベートネットワーク範囲などを除外することもできます。

おつかれさまでした！

SSRFとは何であるか、そしてお使いのシステムでそれにどう対策すべきかが、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:SSRF Detection With IAST

ブログ：CodeSec: Find this vulnerability straight from your CLI