データベース運用サイトへの SQL インジェクション対策

総合技術センター
運営・管理支援分野　藤田　智弘（Tomohiro Fujita）

１．はじめに

就職活動の長期化による学業不振等を防ぐ目的のため、学科固有の就職活動情報を蓄積するデータベースを作成した。現在、このデータベースは Web 上にアップロードされておらず、学生が利用することは出来ない状態となっている。将来的には蓄積した情報を有効活用出来るよう現在作成しているデータベース運用サイトを学生向けに公開したいと考えている。しかし、Web への公開には Web アプリケーションに適切な対策をしていなければ、Web ページの改竄やデータベースからの情報流出などのクラッキングを受ける可能性がある。また、これらの脅威に対して網羅的に対策できる手段はなく、個々の脅威に対して個々の対策を講じる必要がある。

今回は、データベースに対し、大きな脅威となる SQL インジェクションの対策としてプレースホルダをデータベース運用サイトに導入した。

２．SQL インジェクションとは

データベースに対し、非常に驚異的な攻撃である SQL インジェクションについて簡単に説明する。

データベースへのアクセスには SQL と呼ばれる命令文を用いる。Web サイトからデータの検索、追加、更新などの操作は、Web サイトから発せられる SQL によりデータベースへアクセスすることで行われる。Web サイトからデータベースに受け渡す SQL に本来の意図とは異なる命令文を挿入することで、不正な操作を引き起こす事を SQL インジェクションと言う。

３．SQL インジェクションによる影響

データベースに SQL インジェクション攻撃を受けた場合に想定される被害は以下の通りとなる。

- データベース内の全ての情報が流出、改竄、消去される
- データベースを用いたログイン認証を ID、パスワードを用いずログインされる
- データベースサーバー上のファイルの読み出し、書き込み、プログラムの実行などをされる

これらのことから、SQL インジェクションがもたらす影響力は、絶大であるため、Web アプリケーション作成には、SQL インジェクション脆弱性を排除しなければならない。

３．現 Web ページの脆弱性確認

SQL インジェクションの対策前に、現 Web ページがどの程度のセキュリティ強度を持つのか確認を行った。

現在運用しているサイトは、データベースを用いて認証を行っているため、不正な命令を混入させて認証を回避できるかどうかを図１の通り試した。

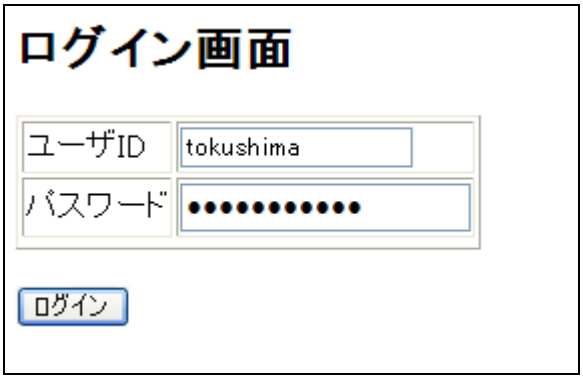

図１　ログイン画面への SQL インジェクション攻撃

パスワードの部分は隠されて見えなくなっているがここでは、「’ OR ‘a’=’a」と入力している。このように入力した場合、SQL インジェクション対策を行っていないと下記の通りに SQL 文が構築される。

select * from x where id=’tokushima’ and pass=’’ OR ’a’=’a’;

上の文の太字網掛け部分がログイン画面で入力されたところであり、本来パスワードが入力

される部分が空欄となる代わりに追加の条件として「OR 'a'='a'」が挿入されている。
これにより、条件が常に真となりパスワードの認証を回避される。また、これにより認証の回避が出来た場合同じ手法を用いて様々な不正な命令を挿入することが出来る。

このように Web ページに不正な命令を混入させ、ログインを試みたが、認証を回避することは出来なかった。確認のため、内部でどのように処理されたのかを確認したところ、以下の通りとなっていた。

select * from x where id='tokushima' and pass='¥' OR ¥'a¥'=¥'a';

パスワード欄に入力された'（以後シングルクォートとする）の前に¥が付随していた。
これは、入力されたシングルクォートが正規の命令文を不正に改竄されないよう文字としてのシングルクォートとして扱うようつけられて印である。（以後この処理をエスケープ処理とする）このことから、極初歩的な SQL インジェクションに対しては対策されていたことがわかった。

４．magic_quotes_gpc

Web アプリケーション側では、シングルクォートのエスケープ処理をする記述はしていなかったため、自動的にエスケープ処理された原因を調査したところ、一部の文字に対して自動でエスケープ処理をしてくれる設定が PHP に初期状態で備わっていることがわかった。

図２に示す magic_quotes_gpc の部分が自動的にエスケープ処理を行う設定部分であり、現状では On になっていることが確認出来た。

```
; Magic quotes for incoming GET/POST/Cookie data.
magic_quotes_gpc = On
```

図２　php.ini の maic_quotes_gpc 設定

５．maig_quotes_gpc 設定の問題点

現時点にて特定を自動的にエスケープ処理してくれるため、特別 SQL インジェクションの対策を追加する必要がないと感じるかもしれないが、magic_quotes_gpc の設定を On のままにしておくと以下のような不具合が発生する場合がある。

- エスケープ処理する必要の無い文字もエスケープされる（文字列に余分な￥が付随する可能性がある）
- エスケープ処理対象外の変数が存在（エスケープ処理が行われない変数読み込み処理が存在する）

このことから、magic_quotes_gpc を On のまま使用するのは完全なるセキュリティ構築の上では好ましくない。

ただし、このままでは文字のエスケープ処理が全く行われないため、Web アプリケーション側で対策を講じる必要がある。現にこの機能を Off にしたところ、先ほどのログインページにて同様の値を入力し SQL インジェクションを行った所認証を回避することが出来た。そのため、PHP にプレースホルダと呼ばれる SQL インジェクション対策の導入を行った。

※補足
magic_quotes_gpc は、PHP6 以降より廃止されている。そのため、導入初期段階では、SQL インジェクションに対する対策がない。バージョンの更新や新規導入の際には注意する必要がある。

６．プレースホルダの導入

SQL インジェクションは、プレースホルダを導入する事で容易に対策することが出来る。これは、プレースホルダが入力される変数を完全なる文字列もしくは数字として SQL 文に組み込んでくれるため、命令文が不正に改竄を行われなくなるためである。

プレースホルダには、動的プレースホルダと静的プレースホルダの２種類がある。どちらも SQL インジェクションに対して堅牢であるが、動的プレースホルダは過去に処理系のバグが存在したため SQL インジェクションを許したことがある。そのため、どちらでも実装可能である場合は静的プレースホルダを用いるのが一般的である。

プレースホルダのパッケージは、PEAR ライブラリに複数存在するが、今回は現在幅広く使われており、メンテナンスが継続している静的プレースホルダの MDB2 パッケージを導入した。MDB2 パッケージの導入は以下の図 3 の通りコマンドを入力することで行うことが出来る。

```
$ pear install mdb2
```

図３　MDB2 のインストール

MDB2 のインストールのみでは、対応するデータベースとの同期をとることが出来ないため、使用しているデータベースに応じて適切なドライバもインストールする必要がある。ドライバのインストールは以下の図４の通りコマンドを入力することで行うことが出来る。また、ここでは、MYSQL のドライバをインストールしているため、コマンドの末尾を mysql としている。

```
$ pear install mdb2_driver_mysql
```

図４　対応 DB ドライバのインストール

７．プレースホルダの検証

Web サイトにユーザ ID「tokushima」とパスワード「 ’OR ‘a’=’a」を入力し認証を回避することが出来るかを再度試した。

結果として、認証を通ることが出来ず、プレースホルダが無事に導入出来ていることが確認出来た。

８．さいごに

プレースホルダを導入する事で、magic_quotes_gpc の設定を Off にした状態でも SQL インジェクションを防ぐことが出来た。ただし、Web アプリケーションには SQL インジェクションの他にもクロスサイトスクリプティング、セッション ID やクッキーに対する脆弱性など数々の脅威が存在する。今後も、このような脅威に対して対策を講じ安心して使用することが出来るよう当 Web ページの改善を進めていきたいと考えている。

参考文献

［1］徳丸浩，『体系的に学ぶ安全な Web アプリケーションの作り方 脆弱性が生まれる原理と対策の実践』，ソフトバンククリエイティブ(株)，2011 年 3 月 5 日

［2］T.Terada の日記，http://d.hatena.ne.jp/teracc/20070125/1169722643，teracc，2011 年 12 月 22 日

［3］PHP のお勉強，http://www.cocoaliz.com/php/index/40/，2011 年 12 月 22 日

［4］徳丸浩の日記，http://www.tokumaru.org/d/20100701.html 徳丸浩，2011 年 12 月 22 日