2F-01

SQL インジェクションの透過的な防御手法*

向佐裕貴† 中村章人†

会津大学†

1 はじめに

SQL インジェクション（SQL Injection: SQLI）は，データベースを利用するシステムにおける最も危険な脆弱性の一種である[1]．SQLI 攻撃が成功すると，一般的に機密性または完全性が侵害され，情報の漏洩，改竄，または破壊を引き起こす．また，ユーザ認証を回避されたり，任意のコマンドを実行されたりする可能性もある．

本論文では，SQLI 攻撃の防御方法を示す．提案手法は，外部由来の入力値を言語処理系レベルで確実に追跡し，危険な SQL 文の実行方法を動的に変更して攻撃を防止する．アプリケーションコードが SQLI 脆弱かどうかに係わらず攻撃を防止できるという意味で透過的である．

2 SQL インジェクション（SQLI）

2.1 SQLI 脆弱性と攻撃

SQL 文を文字列の連結で作成する方法を動的文字列生成または動的 SQL と呼ぶ．SQLI が起きる原因は，動的文字列生成において，適切な処理がされていない外部由来の要素を SQL 文に注入してしまうことに起因する[1]．その結果，意図しない SQL 文を作成・実行してしまう．

2.2 プリペアードステートメント

SQLI 攻撃に対する防御策の一つがプリペアードステートメント（Prepared Statement: PS）の利用である．PS は SQL 文の構造を定義し，外部入力値との組み合わせ後もその構造を変化させない働きを持つ．すなわち，想定する SQL 文の構造を変化させるような外部入力値の注入を防止できる[1]．

3 提案手法：SQLI 防御策

3.1 外部入力の追跡

プログラミング言語における文字列型オブジェクトを拡張した抽象的な言語要素を仮定する．これを拡張文字列システム（Extended String System: ESS）と呼ぶ．ESSは以下の二つの機能を持つ．

- 外部入力マーキング: 文字列中の外部入力値にマーク（目印）を付ける．マークは，文字列中の外部入力部分の位置を表す二つの整数の組（スライス索引）である．
- マーク伝搬: 文字列どうしの連結を行って新しい文字列が作られるときに，マークを伝搬させる．

ESS オブジェクトは，文字列とマークリストの二つで構成する．図 1 に ESS の例を示す．Web アプリケーションにおける認証処理において，ユーザ名とパスワードの組合せをデータベースから検索する場合を想定する．この SQL 文は，二つの外部入力値を連結して作成される．それぞれの値が `"a' OR 1=1; --"` と `"dummy"` であったとすると，まず一つ目の外部入力値のマークは [1,14] となる（a）．これが SQL 断片文字列と連結されて新しい文字列に伝搬すると，[36,49] に更新される（b）．二つ目の外部入力値のマークは当初 [1,6] であるが，これが連結されて SQL 文が完成すると [65,70] に更新される（c）．

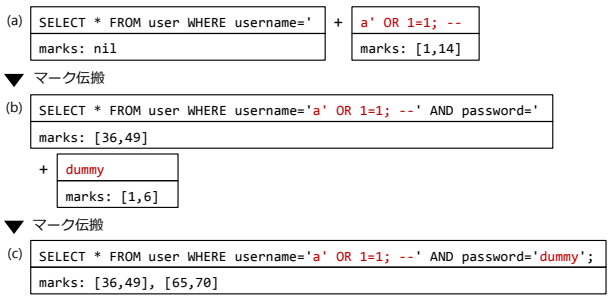

図 1: 外部入力マーキングとマーク伝搬の例

3.2 SQLI 防御手順

ESS を用いて SQLI を防止する手順を示す（図 2）．Web インタフェース（Web IF）はクライアントからの HTTP リクエストを受信し，それをアプリケーションコードに引き渡す．SQL 文を生成するコードをここでは SQL 生成ロジックと呼ぶ．作成し

* “Transparent SQL Injection Defense Method”, Yuki MUKASA, Akihito NAKAMURA
† University of Aizu

た SQL 文はデータベース（DB）ドライバを通じて DB サーバに送られて実行される．

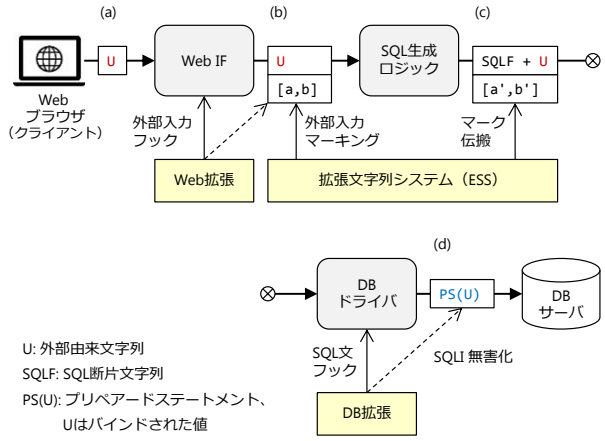

図 2: SQLI 防御手順

システム本来の処理に割り込むために，二つの拡張を行う．Web 拡張は，Web IF を拡張して外部入力をフックする．DB 拡張は，DB ドライバを拡張して SQL 文の送信をフックする．

以下に SQLI 防御手順を示す（図 2 参照）．

(a) 外部入力フック: Web IF は，外部入力を受信したら，値をパースして文字列オブジェクトを生成する．Web 拡張は，ESS の外部入力マーキング機能を呼び出す．

(b) 外部入力マーキング: ESS は各外部入力値に対してマーキングを行い，ESS オブジェクトを生成する．

(c) マーク伝搬: 文字列の連結操作により新しい文字列が生成されるとき，ESS はマークを統合して更新する．

(d) SQLI 無害化（detoxification）: DB 拡張は，SQL 文が DB ドライバに渡されるのをフックして，文字列がマークされているならば PS を生成してパラメータ値（外部入力値）をバインドし，SQL 文を実行する．SQL 文の文字列がマークされていないならば，何もせず元のアプリケーションコードに戻る．

以下に SQLI 無害化の手順を示す（図 3）．

(d1) パラメータ化: 外部入力スライスをプレースホルダ（"?"）に置換した PS を作成する．

(d2) パラメータバインド: DB 拡張は，DB ドライバが提供する関数を呼び出して PS を DB サーバに送る．同様に各外部入力スライスを PS にバインドする．

(d3) 実行: DB 拡張は，DB ドライバが提供する SQL 実行関数を呼び出して，パラメータをバインドした PS を実行する．

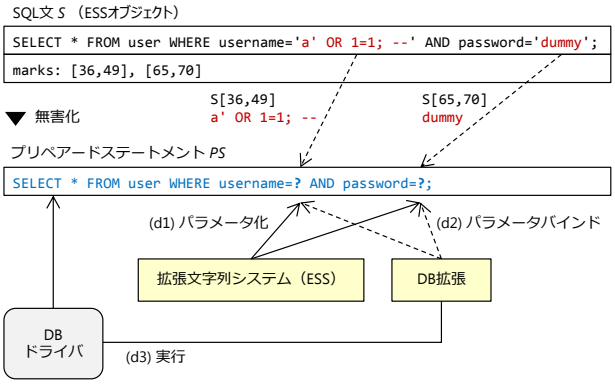

図 3: SQLI 無害化手順

4 実装と評価

Web アプリケーションの開発に広く利用されている Ruby，JavaScript，PHP で提案手法を実装し評価した．Ruby ではメタプログラミング，JavaScript ではソースコード変換を用いた[2]．PHP では拡張モジュールを用いた[3]．

SQLI 用ペネトレーションツール sqlmap を用いて，各実装が SQLI 攻撃を防止できることを確認した．また，本手法の導入による性能負荷を計測したところ，応答時間は十数%（絶対値で数 ms），メモリ使用量は数%（約 1KB）増加する．それぞれ対話的なアプリケーションでは問題にならない．

5 おわりに

本論文では，情報漏洩等の深刻な損害をもたらすSQLI攻撃に対する防御策を示した．提案手法は，言語処理系を拡張して外部入力を確実に追跡し，必要に応じてSQL文の実行方法を変更することで，アプリケーションコードがSQLI脆弱な場合であってもDBアクセスを安全に実行できる．

参考文献

[1] Clarke-Salt, J.: *SQL Injection Attacks and Defense* (*2nd ed.*), Syngress, 2009.

[2] Mukasa, Nakamura: Transparent SQL Injection Defense Method using Programming Language Constructs, IPSJ SIG Technical Report, Vol.2019-DBS-170 No.11, 2019.

[3] 向佐，中村: WordPress におけるプラグイン透過な SQL インジェクション防御策の提案，情報処理学会研究報告，Vol.2019-DPS-181 No.5, 2019.