# 異常検出手法を用いたSQLインジェクション攻撃の検出

角田 直樹　　安井 浩之　　松山 実
武蔵工業大学

## 1. 概要

近年，SQL インジェクション攻撃が急増しており，その理由の一つとして攻撃力の強化があげられる．また侵入検知システムのシグネチャにマッチしない攻撃も可能になり，検出が困難になってきている．この問題を解決するためには異常検出手法が有効であると考えられる．本報告では Web アプリケーションに渡されるパラメータの値のアルファベット数，数字数，記号数を特徴量とし，マハラノビス距離を用いてパラメータの異常を検出することで SQL インジェクション攻撃を検知する手法を提案するとともに，その評価について述べる．

## 2. マハラノビス距離

マハラノビス距離とはあるデータとデータの集合(グループ)との距離尺度であり，データがどのグループに属するかを判別する分析で使われる．ユークリッド距離と違い，グループのデータのバラつきを考慮した距離を求めることができる．楕円形にデータが分布しているグループがあった場合，データが広がっている方向に未知データがあり，その未知データとの距離を測る場合には距離はそれほど大きくなりにくい．逆にデータが広がっていない方向に未知データがあった場合は距離が大きくなりやすい．

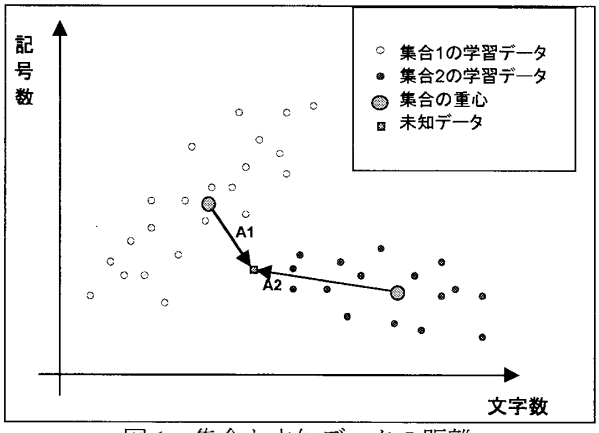


図1：集合と未知データの距離

図1の場合，集合の重心から未知データへのユークリッド距離は集合1のほうが集合2より短いため，未知データは集合1に属することになるが，マハラノビス距離では集合2のほうが距離が短くなり，未知データは集合2に属することになる．

Detection for SQL Injection with Anomaly Detection Method
Naoki Tsunoda, Hiroyuki Yasui, Minoru Matsuyama
Musashi Institute of Technology

## 3. SQLインジェクション

SQL インジェクション攻撃は Web アプリケーションの入力フォーム等に不正な入力をし，送信することによってサーバ側で予期せぬ動作を引き起こす．この攻撃を防止するために，Web アプリケーション自体，あるいは侵入検知システムなどで入力された文字列に危険な単語が入力されていないか，不正と思われる記号がないかをチェックする対策が採られている．しかし，シグネチャベースのマッチングを回避する攻撃が考えだされており，検出が難しくなってきている．

## 4. 提案手法

Web アプリケーションに渡されるパラメータは人間にとって意味のある値であることが多く，使用される文字種の構成に特徴がでるのではないかと考えた．一方，SQL インジェクション攻撃の場合には，正常利用したときとは違った文字の組み合わせになると予測される．そこで入力されたパラメータ値のアルファベット，数字，記号それぞれの数を特徴量とし異常検出を行う．

最初に Web アプリケーションを正常利用したときに得られるパラメータを学習する．正常利用とは Web アプリケーション開発者が予期した通りの利用方法で，不正な入力やミス入力がない場合である．得られた学習データから特徴量の重心を求め，それを正常な集合の重心とする．この正常な集合の重心と未知データを用いてマハラノビス距離を求める．この正常な集合の重心は Web アプリケーションに渡されるパラメータ毎に学習し，求める．

次に検査時には Web アプリケーションへ送られてきたパラメータを未知データとし，学習時に求めた正常な集合の重心とのマハラノビス距離を求める．求めた距離がしきい値以下ならば正常と判断し，それより大きければ異常と判断する．しきい値は学習データの中で最長のマハラノビス距離とした．

## 5. 実験

実験用にショッピングサイトを構築し，ユーザ登録画面の入力フォームのうち名前，メールアドレス，住所，ID，パスワードを SQL インジェクション攻撃の対象とした．

最初に学習データ(250 件)を用いて，正常な値を

学習させ，次に SQL POWER INJECTOR[1]という SQL インジェクション攻撃を行うツールを使い実際に Web アプリケーションに攻撃コード(985 件)を送信した．このツールは Web アプリケーションに SQL インジェクションの脆弱性が無いかチェックするためのツールである．まず最初に SQL インジェクション攻撃のデータから提案手法の検出率を求めた．また比較としてマハラノビス距離ではなくユークリッド距離(しきい値は学習データで最長の距離)でも同様に検出率を求めた．さらに学習時に使った正常データとは別の正常データを送信し，正常なデータを異常と判断してしまう誤検出率についても調べた．

次に Web アプリケーションファイアウォールを導入し，本手法と組み合わせて，同じ実験を行った．Web アプリケーションファイアウォールとは Web アプリケーションに特化したファイアウォールである．今回はシグネチャベースの Green SQL[2]を利用した．一般的に Web アプリケーションファイアウォールは図 2 のように配置され，Web サーバの手前でデータを受け取り，チェックを行い，問題が無ければ Web サーバへ渡すという動作をするが，GreenSQL は，図 3 のように，Web/Data Base サーバにインストールするタイプの Web アプリケーションファイアウォールである．

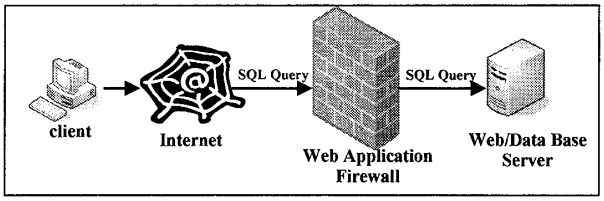

図 2：一般的な Web アプリケーションの配置例

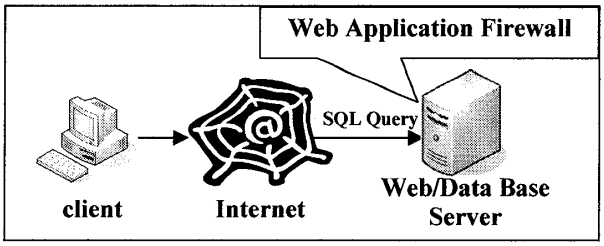

図 3：GreenSQL の設置

## 6. 結果と考察

実験結果を表 1～表 3 に示す．

表 1：マハラノビス距離使用時の結果

| | 名前 | mail | 住所 | ID | password |
|---|---|---|---|---|---|
| 検出率(%) | 83.4 | 75.1 | 72.2 | 76.8 | 81.7 |
| 誤検出(%) | 2.5 | 7.0 | 6.0 | 10.0 | 14.0 |

表 2:ユークリッド距離使用時の結果

| | 名前 | mail | 住所 | ID | password |
|---|---|---|---|---|---|
| 検出率(%) | 71.7 | 66.6 | 56.9 | 70.7 | 71.8 |
| 誤検出(%) | 2.0 | 5.5 | 3.5 | 7.5 | 7.0 |

表 3：GreenSQL と本手法を組み合わせた結果

| | 名前 | mail | 住所 | ID | password |
|---|---|---|---|---|---|
| 検出率(%) (GreenSQL) | 86.4 | 86.4 | 86.4 | 86.4 | 86.4 |
| 誤検出(%) (GreenSQL) | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 |
| 検出率(%) (GreenSQL+本手法) | 93.4 | 91.0 | 91.2 | 94.7 | 92.6 |
| 誤検出(%) (GreenSQL+本手法) | 2.5 | 7.5 | 6.0 | 10.0 | 14.0 |

表 1 と表 2 はマハラノビス距離を使った場合とユークリッド距離を使った場合の結果である．マハラノビス距離を使ったほうが検出率が高いが，誤検出率はユークリッド距離の方が低くなった．ユークリッド距離では正常な集合がしきい値を半径とした球状になるため，マハラノビス距離と比べて，必要以上に正常な集合が大きくなってしまう．そのため，マハラノビス距離よりユークリッド距離のほうが未知データが正常な集合に入りやすい結果となり，マハラノビス距離より検出率は下がり，誤検出率は上がったと考えられる．

次に，表 3 の GreenSQL と本手法の組み合わせではシグネチャベースと組み合わせて使うことによって検出率を向上させることができた．今回は GreenSQL か本手法のどちらかが異常，あるいは不正と判断したものは，もう一方が正常と判断しても異常，あるいは不正としたため，誤検出の数が増えてしまった．しきい値を上げることで誤検出を減らすことができるが，検出率が下がってしまう．この問題を解決するために，本手法で検出した攻撃パターンをシグネチャとして登録していくことで Web アプリケーションファイアウォールの検出率を向上させることができると考えられる．

## 7. まとめ

アルファベット数，数字数，記号数を使ってマハラノビス距離を求めることはユークリッド距離を使うより有効であることを確認できた．しかし本手法のみでは満足できる結果にならなかった．そこで，シグネチャベースの Web アプリケーションファイアウォールと異常検出の本手法，二つを組み合わせて使ったところ，検出率が向上したことを確認できた．しかし誤検出率が上がってしまうので，それが今後の課題である．

## 参考文献

[1]Green SQL - http://www.greensql.net/
[2]SQL POWER INJECTOR
http://www.sqlpowerinjector.com/