平成 28 年度卒業論文

# アタックツリー分析を用いたウェブサーバの設計及び実装

# Design and Implementation of Web Server Using Attack Tree Analysis

指導　松野　裕　准教授

3025　笠井　要輔

日本大学理工学部応用情報工学科

# 目次

# 第1章 序論

## 1.1 背景

近年、インターネットにおけるセキュリティ問題は非常に重要になっている。しかしながら大学研究室やあるいは一般企業のウエブシステムはコストなどの面から必ずしも適切なセキュリティ分析が行われているとは言えない(参考文献 1 参照)。課題としては、脅威をできるだけ網羅すること及びその脅威分析に基づいた実装を行うことである。

## 1.2 目的

本研究では、大学研究室のウエブサーバという身近な例にとり、一連のセキュリティ分析を行い、リスク対策を行う。具体的な方法としては、構成分析手法として注目されている、アタックツリーを用いて標準的な脅威分析を行い、それに基づいた実装を行う。まずは、ローカルでサーバーを構築し、サービス妨害についてアタックツリーの分析を行い、対策を実装して攻撃してみる。サーバーをグローバルに公開した際には、アタックツリー分析を行い構築したサーバーと、セキュリティの有識者が自分の知識を頼りに構築したサーバーとでセキュリティ性を比較する。

## 1.3 構成

本論文は、本章も含めて 5 章から構成されている。
第 1 章では、序論として、本研究の背景・目的を述べる。
第 2 章では、本論文で使用するセキュリティ分析の概要について述べる。
第 3 章では、構築するサーバーの構成について説明する。
第 4 章では、実際に構築するサーバーについてセキュリティ分析を行う。
第 5 章では、本研究で得られた成果をまとめ、今後の課題について述べる。

# 第2章 脅威分析の概要

## 2.1 脅威分析

脅威分析とは、システムのセキュリティ性を可視化することである。脅威分析することで、システムの脆弱性や問題点、システムが安全であることを説明することができる。脅威分析は概ね 4 つの項目から構成されている（図 1 参照）。また、それぞれの項目の分析の仕方には様々な手法があり、驚異分析したい対象によって最適な分析手法を選択する必要がある。

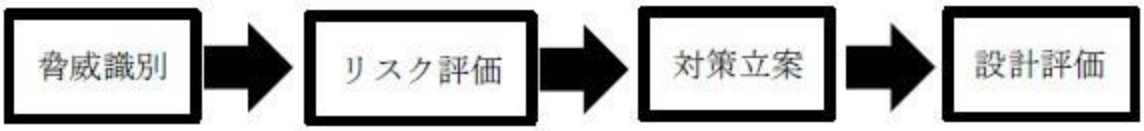



図 1：脅威分析の流れ

下記に脅威分析のそれぞれの項目について説明する。

- 脅威識別…脅威を洗い出す。
- リスク評価…それぞれの脅威についてリスクの度合いを決定する
- 対策立案…各脅威について対策を立案する。
- 設計評価…対策設計が要件を達成しているかを検証する。

## 2.2 様々な分析手法の紹介

2.1 にて説明したそれぞれの分析の項目について、具体的な分析手法を紹介する。

脅威識別

- STRIDE…システム構築・更新において攻撃者がどのような攻撃を仕掛けてくるかを考慮する。
- 5W…本質にアプローチして具体的施策を考慮する。
- RWX…評価対象を悪用系、妨害系などの脅威事象に評価対象をスロットに当てはめることで、脅威をパターンで導出。
- Misuse case…UML を用いて悪意のあるふるまいを洗い出す。

リスク評価

- BugBar…Microsoft が定めた深刻度を用いる。
- DREAD…潜在的損害の大きさ、再現性、悪用性、影響を受けるユーザ、検出可能性のそれぞれの観点から脅威を評価する。
- CVSS…情報システムの脆弱性に対する汎用的な評価手法。
- CRSS…CVSS の応用。影響に関する区分を部分的から軽微、全面的から甚大とする。

- RSMA…リスク値を影響度、発生可能性のリスクレベル判定表によって決定する。

対策立案

- 予防と検知…具体的な対策を考慮する。

設計評価

- Threat Tree…脅威木
- Attack Tree…攻撃木
- Attack / Defense Tree…アタックツリーを行うと共に対策法も行える手法。

## 2.3 今回用いる分析手法について

今回の脅威分析では、分析に脅威識別～リスク評価までを IT 関係で広く利用されている Microsoft の SDL(SRTIDE,BugBar)、設計評価を構成分析手法として注目されている、アタックツリーにて行い、それに基づいた実装を行う。以降から、今回用いる分析手法について詳しく説明する。

## 2.4 STRIDE（脅威識別）

STRIDE とは、STRIDE 手法は Microsoft によって策定された分析手法で、洗い出した脅威は下記表 1 の 6 つに分類することができる（参考文献[5]参照）。また、Threat Modeling Tool という専用フリーソフトウェアにて DFD（データフロー図）を描くと、脅威一覧が自動生成されると共に、それぞれの脅威を自動的に STRIDE の 6 つに分類してくれる。

表 1：STRIDE の分類

| | |
|---|---|
| Spoofing | なりすまし |
| Tampering | 改ざん |
| Repudiation | 否認 |
| Information Disclosure | 情報漏えい |
| Denial-of-Service | サービス妨害 |
| Elevation of Privilege | 権限の昇格 |

## 2.5 BugBar（リスク評価）

BugBar とは、Microsoft が定めた深刻度であり、4 段階の深刻度に分けられる（参考文献[5]参照）。

表 2：深刻度

| | |
|---|---|
| 緊急 | ユーザーの操作なしでコード実行の悪用が行われる |
| 重要 | ユーザーデータの機密性、完全性または可用性が侵害される可能性がある・・ |
| 警告 | 認証要件、または、非デフォルト設定に対してのみ適用性があるなど・・ |
| 注意 | 影響は・・・包括的に緩和される・・ |

## 2.6 アタックツリー（設計評価）

ATA(Attack Tree Analysis)とは、安全分析手法である FTA(Fault Tree Analysis)をセキュリティに応用したセキュリティ分析手法であり、B.Schneier によって発案された（参考文献[4]参照）。攻撃の手段を抽象度の高い順から分解していくことで、そのリスクの度合いを分析する。脅威分析の項目としては、設計評価に当たる分析手法である。また、ADT(Attack Defense Trees)、ACT(Attack Countermeasure Trees)といったセキュリティ要求と組み合わせた分析手法も存在する。

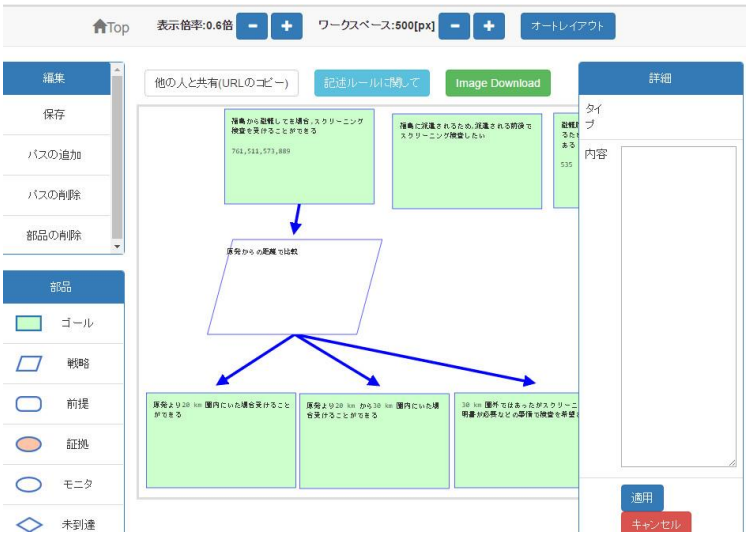
# 第3章 対象サーバーの使用および設計

## 3.1 要件

- 研究室にてサーバーを構築し、グローバルに公開する。
- 通信は全て HTTPS 通信。
- 公開するアプリケーションは、松野研究室の CMS サイト（WordPress)と松野先生が発案している D-Case をブラウザ上にて描ける D-Case Tool。
- 一般ユーザは WordPress の閲覧と、 D-Case Tool の利用（要ログイン）ができる 。
- 管理者はサーバーへのリモートアクセス と、 WordPress の管理者ログイン ができる。

実際に構築したサーバー


松野研究室の CMS サイト




D-Case Tool

## 3.2 物理設計

サーバーの物理構成を表 1 に示す。

表 1：サーバーの物理構成

| パーツ名 | スペック、詳細 |
|---|---|
| OS | Cent OS 7 |
| CPU | I7-6700(3.4GHz) |
| GPU | GTX970 |
| RAM | 16GB |
| ROM | SSD500GB |
| 電源 | 750W |
| LAN | 1000BASE-T/100BASE-TX/10BASE-T×1 |
| LAN ケーブル | ツイストペアケーブル CAT5E,UTP |

## 3.3 論理構成

今回は 1 台の物理コンピューターに必要なアプリケーションをインストールするので、単一サーバーとなる。

## 3.4 実際の構築

現在の構築の状況としては、ローカルサーバーの構築が完成したところである。下記に構築する際に使用したコマンドについて示す。

```
Nano エディタのインストール
# yum -y install nano

SSH のポート番号変更
# nano /etc/ssh/sshd_config

policycoreutils-python のインストール
# yum -y install policycoreutils-python

変更したポート番号の通信許可
# semanage port -a -t ssh_port_t -p tcp 10022

変更したポート番号の通信許可(ファイアウォール)
# nano /etc/firewalld/services/ssh.xml

ファイアウォールの設定変更を反映
# firewall-cmd --reload

SSH サーバー再起動
# systemctl restart sshd.service

Httpd パッケージのインストール
# sudo yum -y install httpd

Apache の設定変更
# sudo nano /etc/httpd/conf/httpd.conf

Apache の起動
# sudo systemctl start httpd.service

Apache を自動的に起動
# sudo systemctl enable httpd.service
```

ファイアウォールの設定変更

```
# sudo firewall-cmd --permanent --add-service=http
# sudo firewall-cmd --reload
```

MariaDB のインストール

```
# sudo yum -y install mariadb-server mariadb
```

MariaDB の設定変更

```
# sudo nano /etc/my.cnf
```

MariaDB の起動

```
# sudo systemctl start mariadb
# sudo systemctl enable mariadb
```

php と関連パッケージのインストール

```
# sudo yum -y install php php-mbstring php-gd php-mysql
```

Apache の再起動

```
# sudo systemctl restart httpd.service
```

WordPress 用のデータベース作成

```
 CREATE DATABASE データベース名;
 GRANT ALL PRIVILEGES ON データベース名.＊TO “ユーザー名”@”localhost” IDENTIFIED BY “パスワード”;
 FLUSH PRIVILEGES
```

wordpress のインストール

```
# curl -LO http://ja.wordpress.org/latest-ja.tar.gz
# tar zxf latest-ja.tar.gz
# sudo ls wordpress
# sudo mv wordpress /var/www/html
# sudo chown -R apache:apache /var/www/html
# cd /var/www/html/wordpress
# sudo mv wp-config-sample.php wp-config.php
# sudo nano wp-config.php
```

# 第4章 サーバーのリスク分析および防御手段の設計

## 4.1 脅威識別

まず、Microsoft の Threat Modeling Tool を用いて DFD を作成した。簡単に下記の DFD の流れを説明する。

まず、「Human User」（一般ユーザー）はブラウザを用いて HTTPS 通信にてウェブアプリケーション(WordPress,D-Case Tool)へアクセスする。この HTTPS 通信を行っている所がインターネット上なので、「Internet Boundary」をいう境界線でインターネットの境界線を表している。また、「User mode or Kernel mode Boundary」という境界線を越えるには、ユーザーログインしないと通信できないことを示している。「Machine Trust Boundary」は、その境界線の右側が全てサーバー側であることを示している。そしてウェブアプリケーションは、ブラウザクライアントの要求に応じて SQL Database、CSS、HTML、PHP を読み込み、結果をブラウザクライアントへ表示する、といった流れになる。

次に「Human User(Administrator)」は、ブラウザクライアントの通信は HTTPS 通信で「Web Application(WordPress)」にユーザーログインをして記事の投稿やページの編集を行う。また、「Managed Application(Win SCP)」からの通信では FTP 通信を行い、サーバーのファイルの読み書きを行う。

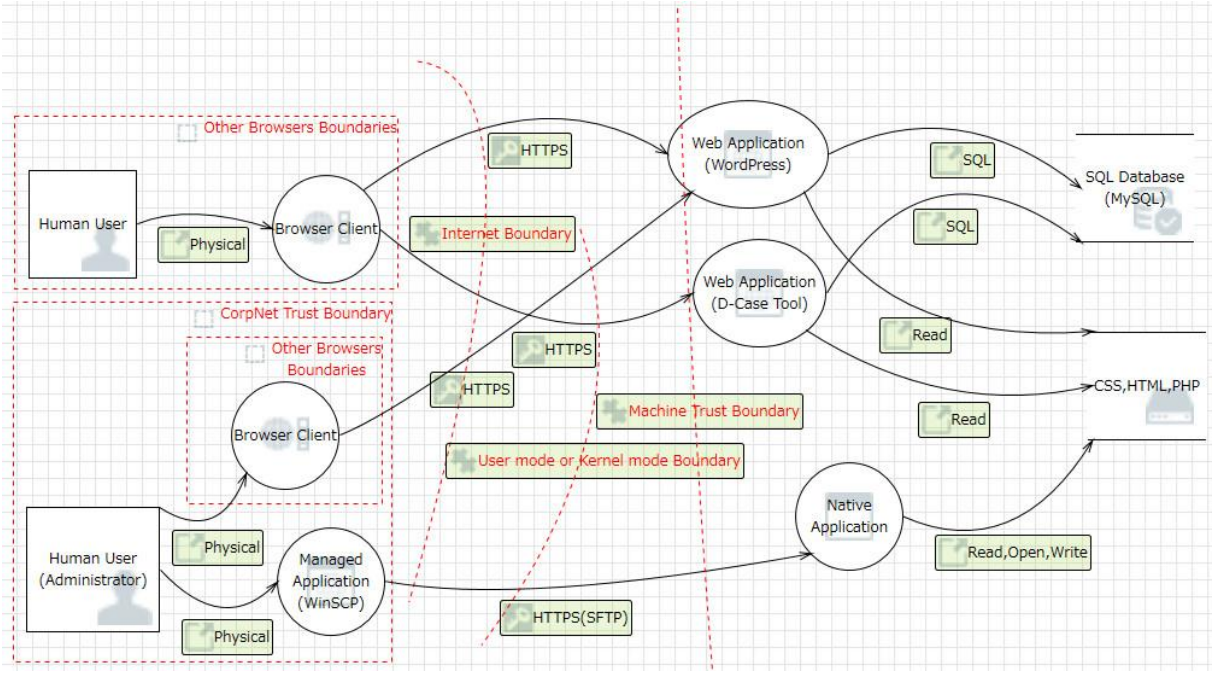



図　今回構築するシステムの DFD

上記の DFD から自動生成された脅威から、システムに対する脅威のみを洗い出した。

表：脅威一覧

| 区間 | 脅威分類 | 脅威 |
|---|---|---|
| Web Application ⇒SQL Database | サービス妨害 | リソース消費(DDos/Dos 攻撃) |
| | 改ざん | SQL インジェクション |
| Browser Client ⇒Web Application | なりすまし | ブラウザクライアントの偽装によるウェブアプリケーションへの不正アクセス |
| | 改ざん | バッファオーバーフロー |
| | 改ざん | クロスサイトスクリプティング |
| | サービス妨害 | ウェブアプリケーションの停止、クラッシュ |
| | サービス妨害 | 外部エージェントが信頼境界を越えるデータを中断する |
| | 権限の昇格 | ブラウザクライアントのコンテキストの偽装 |
| Human User ⇒Browser Client | なりすまし | ユーザーの偽装による不正アクセス |
| | 権限の昇格 | ヒューマンユーザーのコンテキストの偽装 |
| Web Application ⇒Device | なりすまし | デバイスの偽装によってデータを攻撃者のターゲットへ書き込む |
| | サービス妨害 | リソース消費(DDos/Dos 攻撃) |
| Human User⇒ Managed Application | なりすまし | ユーザの偽装による不正アクセス |
| | 権限の昇格 | ヒューマンユーザーのコンテキストの偽装 |
| Managed Application ⇒Native Application | なりすまし | 管理アプリケーションの偽装によるネイティブアプリケーションへの不正アクセス |
| | 改ざん | バッファオーバーフロー |
| | サービス妨害 | アプリケーションの停止、クラッシュ |
| | サービス妨害 | 外部エージェントが信頼境界を越えるデータを中断する |

## 4.2 リスク評価

深刻度が緊急、重要の脅威を対策することとした。

表?：リスク評価

| 脅威分類 | 脅威 | 深刻度 | 対策可否 |
|---|---|---|---|
| サービス妨害 | リソース消費(DDos/Dos 攻撃) | 重要 | 必要 |
| 改ざん | SQL インジェクション | 緊急 | 必要 |
| なりすまし | ブラウザクライアントの偽装によるウェブアプリケーションへの不正アクセス | 警告 | 不要 |
| 改ざん | バッファオーバーフロー | 緊急 | 必要 |
| 改ざん | クロスサイトスクリプティング | 緊急 | 必要 |
| サービス妨害 | ウェブアプリケーションの停止、クラッシュ | 緊急 | 必要 |
| サービス妨害 | 外部からの通信妨害 | 重要 | 必要 |
| 権限の昇格 | ブラウザクライアントのコンテキストの偽装 | 警告 | 不要 |
| なりすまし | ユーザーの偽装による不正アクセス | 重要 | 必要 |
| 権限の昇格 | ヒューマンユーザーのコンテキストの偽装 | 警告 | 不要 |
| なりすまし | デバイスの偽装によってデータを攻撃者のターゲットへ書き込む | 重要 | 不要 |
| なりすまし | ユーザーの偽装による不正アクセス | 重要 | 不要 |
| なりすまし | 管理アプリケーションの偽装によるネイティブアプリケーションへの不正アクセス | 重要 | 必要 |

## 4.3 対策立案

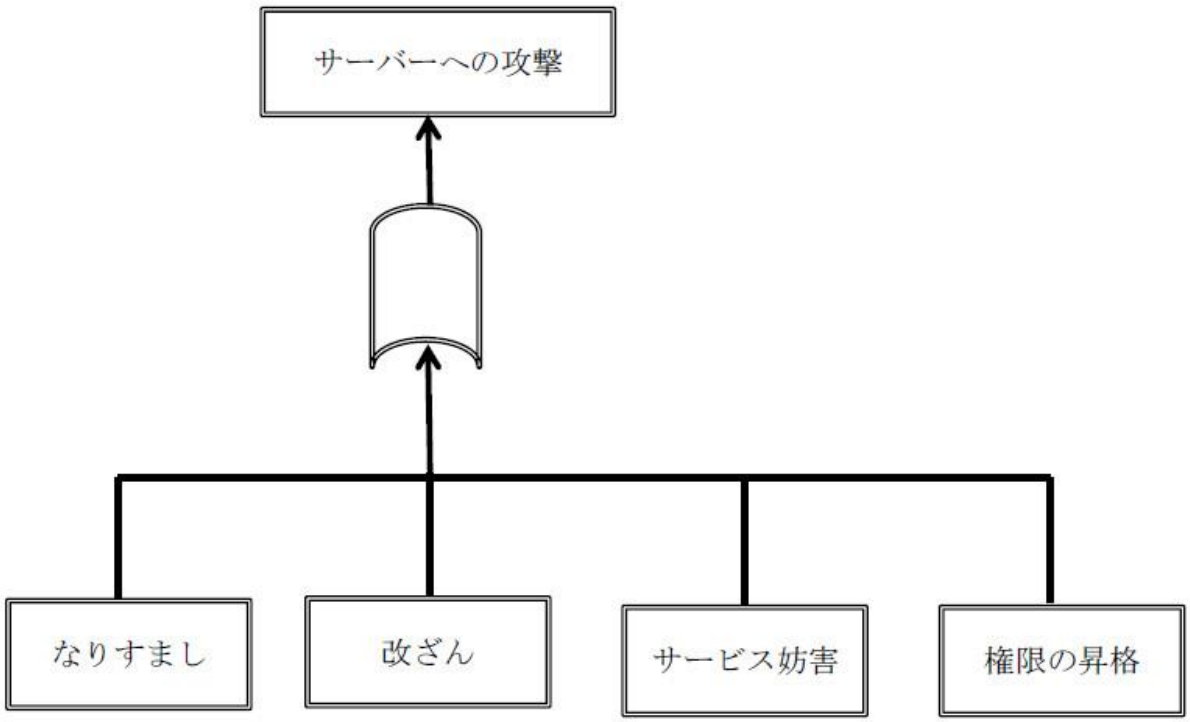
？マークのものは対策方法が分からなかった。

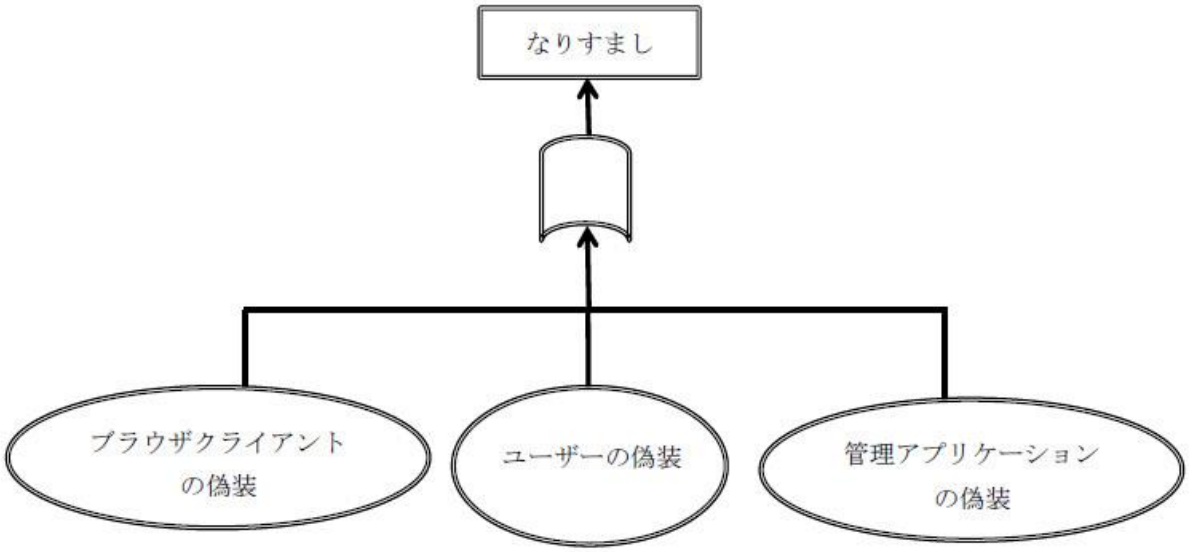
表?：対策立案

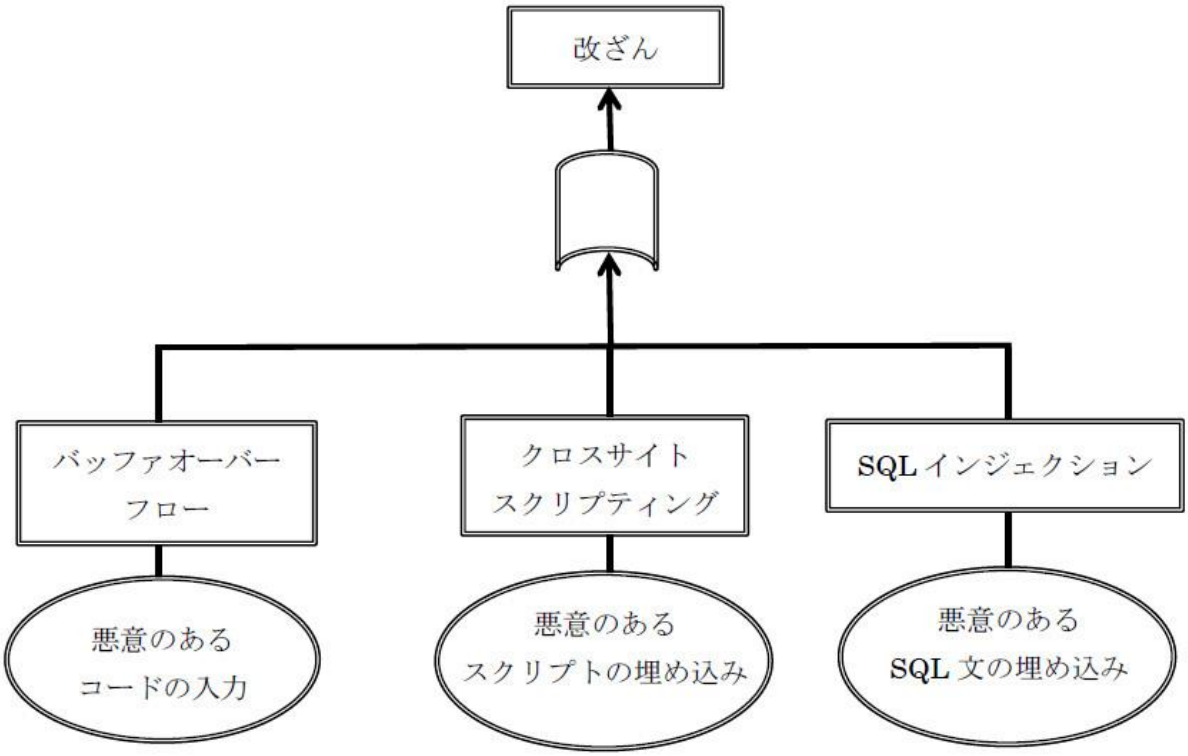
| 脅威 | 対策方法 |
|---|---|
| リソース消費<br>(DDos/Dos 攻撃) | ● 特定 IP のアクセス制限<br>● （ユーザが国内のみなら）海外からのアクセスを制限 |
| SQL インジェクション | ● SQL を埋め込むところで特殊文字を適切にエスケープ<br>● シフト JIS の場合には 1 バイト文字を整理<br>● SQL の記述をなくすために O/R（Object/Relational）マッピングを活用<br>● 攻撃者に役立つ情報を与えないために、不要なエラーメッセージ（データベースが出力するエラーなど）の表示を抑止<br>● バインドメカニズムの利用<br>● WAF（Web Application Firewall）による不正な文字の検出・防御（もしくは、ホワイトリストによる通過の許可）<br>● IDS （Intrusion Detection System）、IPS （Intrusion Prevention System）などによる不正な文字の検出・防御<br>● アカウント分離と権限の最小化による適切なアクセス制御<br>● 管理者アカウントなどの越権から守るために必要個所を適切に暗号化<br>● 万が一の侵入の際に検証可能なようにログを適切に取得 |
| バッファオーバーフロー | ● ポインタではなくデータを渡す |
| クロスサイトスクリプティング | ● 入力された値をただの文字列として認識させる |
| ウェブアプリケーションの停止、クラッシュ | ● サーバーの多重化 |
| 外部からの通信妨害 | ● 通信網を増やす<br>● 通信を有線で行う |
| ユーザの偽装による不正アクセス | ● 外部エンティティを識別するために標準的な認証メカニズムを使用する |
| デバイスの偽装によってデータを攻撃者のターゲットへ書き込む | ● 宛先データストアを識別するために標準の認証メカニズムを使用する |
| 管理アプリケーションの偽装によるネイティブアプリケーションへの不正アクセス | ● 標準の認証メカニズムを使用する |

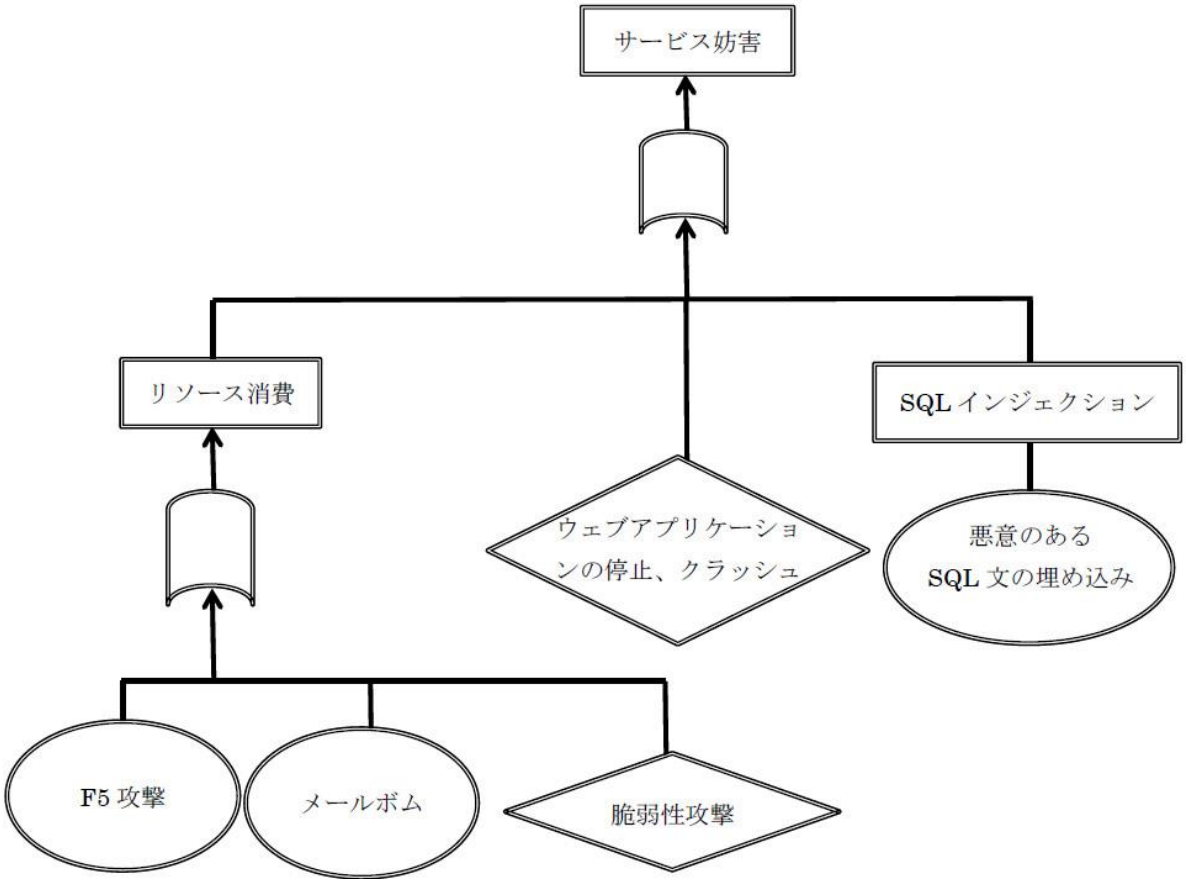
## 4.4 アタックツリーによる脅威の詳細分析

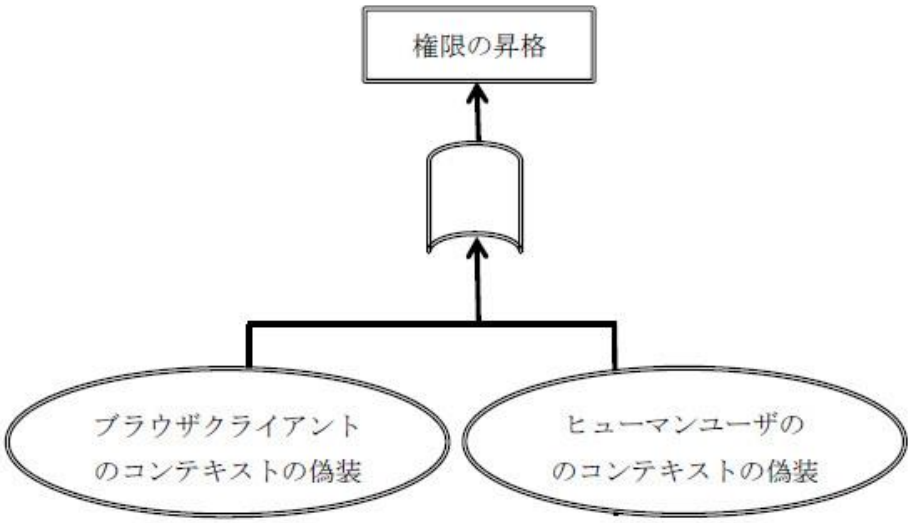
攻撃の組み合わせに何があるかを分析するために、アタックツリーを用いた。ノードが多いのでいくつかに分けて下記に示す。




なりすまし
プラウザクライアント
の偽装
ユーザーの偽装
管理アプリケーション
の偽装


改ざん
バッファオーバー
フロー
クロスサイト
スクリプティング
SQL インジェクション
悪意のある
コードの入力
悪意のある
スクリプトの埋め込み
悪意のある
SQL 文の埋め込み


サービス妨害
リソース消費
SQL インジェクション
ウェブアプリケーションの停止、クラッシュ
悪意のある SQL 文の埋め込み
F5 攻撃
メールボム
脆弱性攻撃


権限の昇格
ブラウザクライアントのコンテキストの偽装
ヒューマンユーザのコンテキストの偽装

# 第5章 結論

## 5.1 専門家のセキュリティによるレビュー

本研究では、セキュリティ分析したものをセキュリティに関する要求工学、システム検証の専門家である産総研の田口研治先生にレビューして頂いた。下記に私が行った脅威分析についてのレビューを示す。尚、本論文に記載されている分析は、このレビューに基づいて修正されたものである。

1） DFD について

1-1）何が目的で、どのように記述したかをより明確にする必要がある。

1-2） 何を意味するのか分からないノードがある。

1-3）Internet Explorer Boundaries から FTP 通信でデバイスにアクセスするのは、整合性が取れないのではないか。

1-4） HTTPS(FTP) 通信ならば相互通信を意味する２本線を書くべきではないか。

1-5）Sandbox Trust Boundary Border の中にサーバーがあるのはおかしい。

2） 2.2 STRIDE

2-1）どの箇所において STRIDE から分析された脅威があるかをより明確にする必要がある。

3） DFD から生成された脅威一覧

3-1)ユーザに対する脅威と、システムに対する脅威が混ざっている。

3-2)日本語がおかしい。

4） 設計評価（アタックツリー）

4-1)ここでは設計はされていないので、「設計評価」ではなく「アタックツリーによる脅威の詳細分析」にするべき。

4-2)DFD から分析されていない脅威がある。

## 5.2 まとめ

今回初めてセキュリティ分析を行い、脅威分析について理解を深めることができた。また、今回行った分析では脅威識別の際に描いた DFD によってその後の分析が大きく変わるので、DFD に間違いがあると、脅威識別以降の分析も全てやり直すことになるので非常に時間がかかった。今現在自動化されているのは脅威識別までだが、対策立案まで自動化できると非常にスムーズに脅威分析が行えると共に、適切なセキュリティを行うことができると考えられる。

また、STRIDE 分析に用いるツールが英語なので今ひとつ理解できない脅威がある。これを理解する為にネットワークセキュリティの理解を深める必要がある。今後は理解できていない脅威の理解、サーバーのグローバルへの公開が目標である。

# 文献


[1] ITmedia セキュリティ対策最大の懸念は「コスト」 （閲覧日 2017 年 1 月 30 日）
http://www.itmedia.co.jp/enterprise/articles/1607/05/news048.html

[2] 信頼できるコンピューティングのセキュリティ開発ライフサイクル(閲覧日 2017 年 1 月 30 日）
https://msdn.microsoft.com/ja-jp/library/ms995349.aspx

[3] 脅威分析研究会 脅威分析超入門 （閲覧日 2017 年 1 月 30 日）
https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxzaWdzdGF3ZWJ8Z3g6MTc4MmE5Mzk1YWUxZDY4OA

[4] 第一回 脅威分析研究会 発表資料　アタックツリーによる脅威分析(閲覧日 2017 年 1 月 30 日）
https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxzaWdzdGF3ZWJ8Z3g6M2VhY2M0NzIwYTA1MTE1MQ

[5] 脅威分析（仕様と設計のセキュリティ分析）（閲覧日 2017 年 1 月 30 日）
https://www.asteriskresearch.com/wp-content/uploads/2016/01/ThreatModeling_requirements_and_design20160204.pdf

[6] セキュリティ開発ライフサイクル（SDL）における QA の役割（閲覧日 2017 年 1 月 30 日）
http://www.juse.or.jp/upload/files/7p_4_0916.pdf

[7] Microsoft 脅威モデルを作成する（閲覧日 2017 年 1 月 30 日）
https://msdn.microsoft.com/ja-jp/library/ff648644.aspx

[8] IPA　共通脆弱性評価システム CVSS 概説（閲覧日 2017 年 1 月 30 日）
https://www.ipa.go.jp/security/vuln/CVSS.html

[9] CCDS 製品分野別セキュリティガイドライン（閲覧日 2017 年 1 月 30 日）
https://www.ccds.or.jp/public/document/other/guidelines/CCDS%E8%A3%BD%E5%93%81%E5%88%86%E9%87%8E%E5%88%A5%E3%82%BB%E3%82%AD%E3%83%A5%E3%83%AA%E3%83%86%E3%82%A3%E3%82%AC%E3%82%A4%E3%83%89%E3%83%A9%E3%82%A4%E3%83%B3_%E8%BB%8A%E8%BC%89%E5%99%A8%E7%B7%A8_Ver.1.0.pdf

[10] Top SE ミスユースケースによるセキュリティ要求分析（閲覧日 2017 年 1 月 30 日）
http://www.fuka.info.waseda.ac.jp/rewg-sub/workshop/201305/IPSJ-REWS-MASG-ex.pdf

[11] 情報セキュリティ大学院大学 脅威分析法（閲覧日 2017 年 1 月 30 日）
https://www.ipa.go.jp/files/000046476.pdf

# 謝辞

本研究にあたり，ご指導，ご助言を頂いた日本大学理工学部応用情報工学科松野裕准教授ならびに木原雅巳教授に深く感謝致します。また、研究においてご協力，ご助言を頂いたシステム検証の専門家である田口研治先生に深く謝致します。