Vol.2015-DPS-162 No.54
Vol.2015-CSEC-68 No.54
2015/3/6

# 大容量 WAF ログデータの分析手法の検討

石崎夕香里[†1]　　後藤厚宏[†1]

**概要:** 現在，数多くの Web サイトが世の中に存在し，ビジネスの現場においても個人の利用においても，人々が Web サイトを閲覧し利用するのは当たり前の時代となった．しかし，その反面で，個人情報の漏えい， ウェブサイトの改ざん， ウイルス感染サイトへの誘導など， Web アプリケーションソフトウェアの脆弱性をついた攻撃は増加し，その被害件数も増加する一方である．ここに，Web アプリケーションに対するセキュリティ対策が，追いついていない現実がある．本稿では，Web アプリケーションのセキュリティ対策の一つである，WAF (Web Application Firewall)の導入に着目した．WAF により，Web アプリケーションの脆弱性に関する被害件数は少なくなることが予想される．それにも拘わらず，企業組織において，WAF の導入率は約 20%程度である．WAF の問題点の一つに過剰検知をチューニングする難しさと大変さがある．その解決方法について，ログ分析作業を支援する手法（1.外部データの活用，2.過去データの統計分析，3.ログの可視化）を提案し，その評価について述べた．

# Study on Analysis Method for Huge Log Data of Web Application Firewall

YUKARI ISHIZAKI[†1]　ATSUHIRO GOTO[†1]

**Abstract:** It has become common that there are many web sites in the business and the private scene. On the other hand, the attack aimed at the vulnerable web sites such as private information leakage, web alternation, leading to virus infection site, has been dramatically increasing. The number of victims also has been increasing. The security measures for web application are insufficient. In this paper, we focused on introduction of WAF (Web Application Firewall) as one of the security measures. Introducing WAF would make the damage of web application vulnerability decrease. However, the rate of the introduced WAF in the business organizations is about 20 percent. It is the reason that tuning the results of false positives on WAF is difficult task. Then we propose an analyzing approach of logs (1.the use of external data, 2.statistical analysis used past data, 3.log visualization method), and show an evaluation for this solution.

## 1. はじめに

不正アクセスによる被害件数は，増加をたどる一方だが，中でも，ウェブサイトの改ざんにより情報が漏えいする被害は顕著である．ウェブサイトを狙う攻撃の内容には，クロスサイトスクリプティング， クロスサイトリクエストフォージェリ， OS コマンドインジェクション， SQL インジェクション， セッション管理の不備， ディレクトリトラバーサル， といったものがある．これらの攻撃により，被害を受けた企業は， 多額の損害を受ける．Web サーバの改ざん被害での事故対応にかかる被害額は 4800 万円から 1 億円と言われている[1]．インシデントの顕在化・初動対応，被害状況の調査，復旧作業，対外説明，社内体制の整備，といった事故処置にかかる費用が発生するからである．また， 被害額だけではなく，企業の信用問題にも大きな影響を与え，ビジネスチャンスを失うことにもなりかねない．

また,四半期毎に表示された「ウェブ改ざんの届出件数」は,2012 年までの四半期の最高件数が 16 件だったインシデントに対し，2013 年には 27 件まで件数が伸びている.その届出内容を見ると，改ざんされたサイトの多くは， セキュリティ対策の不十分なパソコンで閲覧するとウイルスに感染するように細工されており，閲覧者のパソコンをウイルスに感染させることがウェブ改ざんの主な目的であったと言える．

不正アクセスの被害内容や届出から，ウェブサイト改ざんに関する内容が多いことが分かった．では，ウェブサイトの改ざんを防ぐためにはどうすべきであるか，主な方法として次の点が挙げられる．

（1） Web アプリケーションの改修
（2） Web サーバの設定・アップデート
（3） Web Application Firewall(WAF)の導入

Web アプリケーションの修正に関して，「情報セキュリティ早期警戒パートナーシップ」[2]では，届出られた脆弱性関連情報をウェブサイト運営者に送付し，ウェブサイト運営者に届出があった脆弱性の修正を依頼している．しかしながら，すべてのウェブサイト運営者がすぐに脆弱性を修正できるわけではない．脆弱性の修正に 31 日以上の日数を要した届出は，全体の 53%にのぼる（図 1 の赤枠部分）．SQL インジェクションのように悪用された場合に危険度の高い脆弱性であっても，ウェブサイト運営者の諸事情により，脆弱性の修正には時間を要しているのが実情である．

†1 情報セキュリティ大学院大学
Institute of Information Security

Vol.2015-DPS-162 No.54
Vol.2015-CSEC-68 No.54
2015/3/6

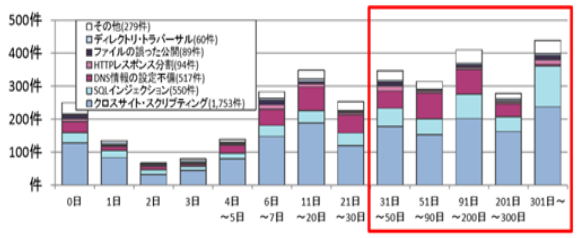


図 1　ウェブサイトの修正に要した日数（2010 年第 4 四半期時点）[出典 IPA]

一方, WAF は, システム稼動前でも稼動後でも導入が可能であり, 周辺システムへの影響も, Web アプリケーションに比べると少なくてすむ.

ウェブサイト改ざんによる被害事例を見ると, WAF によって防御できる攻撃も多く見られる. また, 昨今の WAF は, SSL で暗号化された通信にも, 難読化された JavaScript 攻撃コードにも対応している. 一方 ブラックリスト方式を採用した WAF において, 次の点で問題がある.

(1)　未知の攻撃を検知できない.

ブラックリストを用いた場合, すべての攻撃をブラックリストとして設定できたとしても, 未知の攻撃[7]に対しては, 検知漏れを起こす可能性がある. また, シグニチャ更新までの期間が長い.

(2)　過剰検知（False Positives）の精査に時間を費やす.

WAF 導入時に正常な通信をいきなりドロップできない為, チューニングによって過剰検知を減らしていく. モニタリング後のチューニングを見込んだ緩めのポリシーを初期値として採用している WAF 製品が多い為である. また, 本番運用が始まってからも, 新しいシグネチャの対応やアプリケーションの改修時には, 再チューニングが必要である.

一方, 決めたパターンだけを通過させ, それ以外はすべて遮断するホワイトリスト方式では, 未知の攻撃には対応しても, パターンの定義を作ることが難しい点や, 融通が効かない点がある. 例えば, 半角で入力するというパターンで, ユーザがたまたま全角で入力したら, ブロックされてしまうといったものである. それぞれの長所, 短所はあるが, 本研究では, ブラックリスト方式による WAF に関する研究を取り上げる.

上述した WAF の問題点において, 本稿では 上記(2)に挙げた, 過剰検知を精査するチューニングの問題について考察する. WAF のチューニングを効率化し, ログ精査の作業時間を減らす事により, WAF を用いたセキュリティ対策を積極的に検討する材料となり得る.

## 2.　WAF の仕組み

### 2.1　WAF の概要

WAF は以下のような機能によって, 外部からの攻撃を防ぎ, 個人情報やサーバの情報が攻撃者に渡らないよう, Web サーバや Web アプリケーションを保護する.

（1）　入力値検査

パラメータに対するホワイトリストあるいはブラックリストによる検査を行う. 現在, 多くのログは, このブラックリストによる検査により出力される. 本研究でもこの部分に大きく関わる.

（2）　画面遷移のチェック

外部からの入り口となるページのチェック, Web アプリケーション内のすべての遷移のチェックを行う.

（3）　hidden フィールド操作の防止

hidden フィールドや Cookie, ラジオボタンなどの選択肢の値がクライアント側で改変されていないかチェックし, 改変されていたらエラーとする.

（4）　自動スキャナ, ボットからのアクセスを防止

既知の脆弱性を探るための自動スキャンやボットからのアクセスを検知して, Web スクレイピングを防止, 貴重な IP 情報を保護する.

（5）　L7 DOS 攻撃, ブルートフォース攻撃, パスワードリスト攻撃からの防御

失敗の多いログイン要求を絞り込むことによって, 攻撃を検知, 危険を軽減する.

（6）　個人情報や機密情報をマスクするデータカード

Web アプリケーションから送出される情報にクレジットカード番号などの個人情報が混入していた場合, 表示を自動的にマスクすることで, クライアントの Web ページへ表示されることを防ぐ.

WAF の基本的な機能は, ホワイトリストおよびブラックリストによる入力値の検査である[3]. 通常, 入力値検査は, 正規表現などの手法を用いてパターンマッチングにより行われ, 検知した攻撃を遮断する. ホワイトリストは, 管理者が明示的に設定したトラフィックを通過させる為のリストである. ブラックリストは, 管理者による明示的な設定ないし管理システムから定期的にダウンロードするシグネチャにマッチしたトラフィックを遮断させる為のリストである.

### 2.2　ブラックリスト方式による WAF の基本動作

SQL インジェクションを例にとって説明する. 攻撃者は Web ブラウザの Username 欄に, ’ OR1=1# という文字列を挿入して, SQL 文からデータベースを操作しようとする. WAF に届いたパケットは, シグネチャの内容に合致した為, SQL インジェクション攻撃を発しようとした攻撃者のブラウザはブロックされ, ブロックされた結果が WAF のデータベースにログとして記録される.

### 2.3　シグネチャについて

以下に示すシグネチャセットを, アプリケーションに応

じて最適に設定する．商用の WAF では，ほとんど同じ内容のシグネチャが適用される．ここでは，F5 Networks 社の BIG-IP ASM のシグネチャを例に挙げて説明する．

シグネチャとは WAF が保持している検知ルールとなり，保持数とし大よそ 2,000 個以上となる[4]. 各シグネチャは 20 種類程の攻撃タイプによって分類されており，どれかに所属する[4].

シグネチャは，チェックする機能ごとにシグネチャセットとしてもまとめられている．シグネチャセットをどのように利用するかで，ログの出方も変わる．

例えば，標準シグネチャセットでは，一般的な検知を行い，攻撃タイプは全種類網羅し，シグネチャ数は 1510 である. IPA 準拠のものでは，代表的な 5 種類の攻撃タイプを検知するシグネチャのセットを利用し，シグネチャ数は 1310 である．標準シグネチャセットと IPA 準拠を合わせた組み合わせたセットのシグネチャ数は 1846，全てのシグネチャを利用する All Signatures セットでは，シグネチャ数は 2362 となる．

### 2.4 シグネチャと過剰検知が発生しやすい箇所との関係

WAF に入ってくるリクエストとシグネチャを照合する際，過剰検知を起こしやすい場面や箇所を次に記す．

(1) ユーザのリクエストに対して，WAF に設定している文字コード（Web サーバに設定しているものと同じ）で，デコードし，シグネチャと照合する際，過剰検知が起きやすい．

(2) ファイルのアップロードの際，ユーザのリクエストにバイナリ文字が入っている為，過剰検知が起きやすい．

(3) ユーザ画面の自由記入欄等，HTML タグ等起票すると，過剰検知が起きやすい．

これら過剰検知が発生しやすい箇所についての知識は，ログの分析時に役立つ．

## 3. WAF の導入プロセスと課題

### 3.1 WAF 導入のプロセス

WAF の初期導入は，セキュリティポリシーの策定，セキュリティ要件を定義し，WAF のシグネチャに反映させることから始まる．

次に 2〜3 週間モニタリングを行う．このモニタリングで収集したログを分析するのがポリシーチューニングである．WAF のログから過剰検知精査を実施し，最適なチューニングを行うには，おおよそ 2 週間程かかる．

この後，WAF の本番運用/セキュリティ監視（24 時間 365 日）のフェーズに移る．本番運用時もアタックシグネチャによる最新のセキュリティホールへの対応を含め，サブシステム追加やアプリケーションの改修/追加時などに再度ポリシーチューニングを行う．

費用面においても同様に，WAF 設計，導入，ポリシーチューニング，セキュリティ運用，と大きく分けられるが，ポリシーチューニングのコストが，他のネットワーク機器等の導入時に比べてコストがかかる部分である．

次に，WAF 導入プロセスのサイクルを説明する．日数は規模により変動する．

- 初期設置作業： 計 40 日

作業詳細：

① 最新シグネチャ情報の提供・作業準備 …5 日
設計書作成，最新シグネチャの確認．

② シグネチャアップロード，設定作業 …1 日
WAF 機器に，設計書内容の設定と最新シグネチャを導入する．シグネチャにより検知されたデータは，最初はブロックされず，ログだけが記録される．

③ アプリケーション全打鍵作業 …14 日
顧客（Web アプリケーション担当者）に，一通り想定される入力を実施してもらう．

④ ログ取得作業 …1 日
WAF により，検知されたログの取得．

⑤ ログ精査作業 …14 日
検知されたログの内容が，外部の脅威による攻撃のログなのか，アプリケーションのテスト打鍵で入力した際に検知されたログなのか（つまり，アプリケーションに取って必要な入力値），後者であれば過剰検知として，検知されたログのシグネチャに対しては，ブロック化作業を実施しない．

Web アプリケーションのサイトの作り方，入力値の数によって出力されるログの数は違う．1 日に 2 万行以上のログが出てしまう場合もある．

このログの精査には，明らかに脅威によるものか，過剰検知なのか，大量データの正しい分類と，セキュリティの専門的な判断が必要となる．第 5 章で検討する．

⑥ アプリケーション担当者（通常はユーザ）との調整 …1 日
ログ精査の結果を報告し，アプリケーション担当者に過剰検知のログ内容を確認してもらう．また，脅威により検知されたログの場合，アプリケーションの改修が直ぐにできるものであれば直ぐに修正をしてもらう．

⑦ シグネチャ方針策定 …3 日
過剰検知かどうかの結果を受け，ブロック化を有効にするシグネチャを決定，設定を行う．

⑧ ポリシー有効化作業 …1 日
シグネチャにより検知されたデータは，ログが記録され，ブロックもされる（防御モード）．

- 次年度以降チューニング作業：

初期設置から，半年後あるいは 1 年後，最新のシグネチャへアップデートする．作業内容は初年度設置作業と同じで

あるが，①の日数が少し短くなる．

上記に述べてきたように，WAF 導入には，時間と労力が多くかかり，それが後年も続くのである．

### 3.2 抽出されたログの容量

上記 3.1④ログ取得作業で，ログが抽出される．例えば，あるケースでは，2 週間程のアラートログが 191MB 以上あり，この CSV ファイル[5]をエクセル等で扱うことは難しい．

### 3.3 ログの精査

では，上記 3.1④で取得される何万行もあるアラートログを，上記 3.1.⑤のログ精査結果として，どのようにまとめなくてはいけないのだろうか．

シグネチャ ID，シグネチャ/Violation 名，対象 URL，検知した HTTP リクエスト，検知理由（どの部分や文字列で検知されたか），推奨設定，ソース IP，検知日時，検知件数，等，以上の項目をベースに，統計資料と見解を作成し，方針を提示すべきである．解析作業者または管理者は，アラートとしてあがった数あるログの中から，過剰検知を分類し，真の攻撃に対しては対策を施さなくてはいけない（攻撃対象データが，シグネチャにマッチングするものはブロックする設定をする）．

### 3.4 過剰検知の増加要因

アラートログや，過剰検知が出力される数は，案件毎様々である．それでは，多数の過剰検知が出力される要因は何であろうか．

#### 3.4.1 Web application の作り方

Web Application の画面遷移数，入力値，URL の数等，Web Application の作り方によって，アラートの数は変わる．それらが多ければ多い程，比較的過剰検知のアラートログが生成される可能性も多い．また，入力値の種類にもより，数字だけを入力するようなパラメータでは，ほとんど，アラートとして検知されることはないが，アルファベットや記号文字が混在すると，クロスサイトスクリプティングや SQL インジェクションといった攻撃と見なされてしまうこともある．

#### 3.4.2 シグネチャの選び方

上記 2.3 で示したシグネチャセットの組み合わせ（標準 F5 社推奨，IPA 推奨，標準＋IPA，All）の中から，ALL を選んだ場合には，過剰検知のアラートログが多く生成される．

#### 3.4.3 WAF のバージョンアップとモバイル機器

今回例に挙げている BIG-IP ASM では，セッションハイジャックを防ぐ為，Web　アプリケーションから払い出されたセッション ID(Cookie)のハッシュ値を計算し，それを（ASM Cookie）として付与する．クライアントから戻ってきたそのセッション ID(Cookie)のハッシュ値を計算し，ASM Cookie 値と同じであれば，改ざんされていないと判断する．

最近のアンドロイド OS は，ある一定期間が過ぎても Cookie を消すことはなく，いつまでも保持し続ける仕様のようである．その為，BIG-IP がバージョンアップした際，古い OS のバージョンの ASM Cookie を持ち続けるモバイル機器は，ASM Cookie が照合されず攻撃と見なされ，いつまでもブロックされてしまうことが稀にある．このように，WAF のバージョンアップと周辺環境の機器との関係や有効にする機能により，過剰検知の発生も起こり得るのである．

### 3.5 WAF の大容量ログと過剰検知に対する対応

3.1 では，WAF の導入にあたり，チューニングを含めて，手間と日数が掛かる事を述べた．3.2 では，どのようなログが抽出されるのか，1 ログあたりの例を示した．3.3 では，3.2 が大量になったログを最終的にどのような形に仕上げなくてはならないのか，という点を述べた．3.4 では，過剰検知が増加する，つまりログも伴って増加する要因を記したが，それらの要因を解決してログを減らすことは，根本的な手段とは考えていない．

本稿では，ログが大量であっても，ログの分類，分析の負荷を軽減することを目的とする．それにより，手間とログ精査にかかる日数を短縮し，解析作業者だけではなく，ユーザ管理者でも，ログを精査できるようにしたいと考える．

これらにより，WAF のチューニングの煩わしさが減ることと，管理者でもログを扱えることで，WAF の過剰検知の精査に時間を費やす点が解消され，WAF 導入へのハードルが低くなるのではないだろうかと考える.また，チューニングの効率化はコスト面での削減にもつながる．

## 4. ログ分析手法の提案

### 4.1 従来のログ分析手法

今回取り上げるログの分析は，多く利用されているブラックリストによるシグネチャルールベースに基づいたケースを想定している．

ブラックリスト方式によるログ抽出後，従来は以下の手順で，ログを抽出，分析の前段階の整形，正常検知と過剰検知を判定する評価を実施していた．ログの整形には，プログラミングによる分類（Classification）を用いて，類似攻撃パターンのグループ分けとその件数を算出し，評価は人間（Analyst）によるものである．

Vol.2015-DPS-162 No.54
Vol.2015-CSEC-68 No.54
2015/3/6

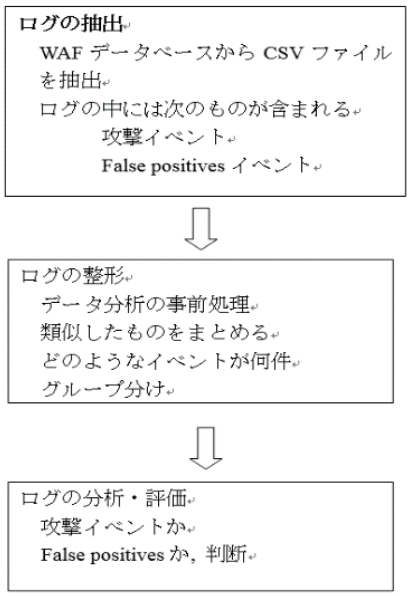


図 2　従来のログ分析手法

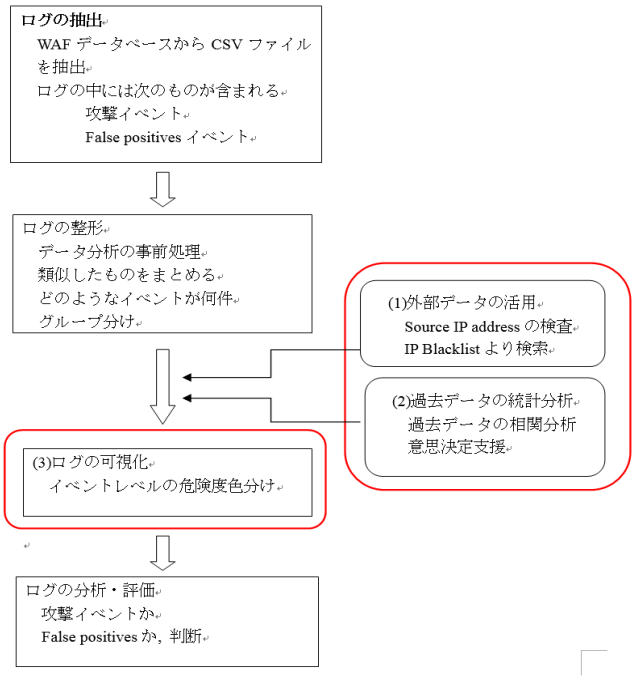


図 3　提案の全体像

### 4.2　提案の全体像

今回の研究では，従来行っているログの精査に，次の提案を加える．

(1)　外部データの活用
(2)　過去データの統計分析
(3)　ログの可視化

(1)では，世界中のブラックリストを登録している DB から Source IP アドレスの検査を試みた．(2)では，過去のデータを統計的に整理し，正常検知と過剰検知の判定基準の規則性を導いた．(3)では，(1), (2)の結果から，イベントの危険度のレベルを可視化し，正常検知の可能性を示した．

赤枠　　現状からの追加部分

### 4.3　ログの抽出

WAF データベースから，シグネチャにより検知したログを抽出する．ファイル形式は，CSV ファイルである．今回は，ブラックリスト形式を採用している為，ログの中身は，攻撃による検知ログ（正常検知），攻撃ではないが検知されたログ（過剰検知）が含まれる．

### 4.4　ログの整形

4.3 で抽出した csv 形式のログを整形する．

何千行，何万行とあるので，まずは，類似したグループ毎まとめ，後に正確なログ分析を行う為に，役立つ形に持っていくことが必要である．プログラミングにより，具体的には同じシグネチャで検知されたものをキーとして，攻撃タイプが同じイベントをグループ化する．どのイベントが何件といった形にする．

例えば，5 日分のログで 1 万行を 10 パターン，20 日分のログで 30 万件を 50 パターンといった具合である．

＜整形後のログ＞

図 4　整形後のログ

＜出力項目＞

No，シグネチャ ID，シグネチャ/Violation 名，対象 URL，件

数, 検知 HTTP リクエスト（一例）, 検知理由, 推奨設定, SourceIP（一例）, 検知日時（一例）

### 4.5 外部データの活用

イベントのソース IP アドレスが, IP UTILITIES.NET のサイト[6]のブラックリストに登録されているかどうか検索する. ソース元データベースは SPAMHOUSE[7]で, 世界中のブラックリストが登録されている.整形したログの表の項目に検査結果を付加する. この作業により, 正常検知かどうかの第一段階の切り分けが出来るのではないかと考えている. SPAMHOUSE 以外にも, BLACKLISTALERT.ORG[7]というサイトも利用が可能である. また, セキュリティベンダー各社は独自の Blacklist を持っている. 正常検知の可能性がある

### 4.6 過去データの活用

過去データの統計分析を行い, 明らかに攻撃であるものの傾向を掴むことにより分析者を支援する. 例として, 正常検知のソース IP アドレス, 送信国, 攻撃タイプ, 検知数, URI, レスポンス等がある.

### 4.7 ログ分析支援情報の可視化

外部データや過去データから得られた分析の支援情報としてログ危険度レベルをスコア化し, 分析支援情報として可視化する. 例として解析者がログの正常検知 と過剰検知を分析に時間をかけることなく実施できるようログに正常検知のより可能性の高いものを色づけする.

## 5. 提案手法の評価

### 5.1 評価に用いたデータ

今回の評価では, 次の 5 種類のログデータを用いた. ログＰは検証環境を用いて取得したログである. 他４つはこれまでに分析を実施した４つのケースの整形済のログである. ログＰとログＡ社は, 過去データの活用（4.6 節）で使用し, ログＢ社, Ｃ社, Ｄ社分は, 分析対象のログとして使用した.

表 1 評価に用いたデータ

| | 対象のサイト | イベント数 |
|---|---|---|
| P（7 か月分） | 検証環境 | 769 |
| A 社（6 か月分） | サービス業 | 320 |
| B 社 | 卸売業 | 6 |
| C 社 | 製造業 | 8673 |
| D 社 | 製造業 | 13468 |

### 5.2 外部データ活用についての評価

最初に, SPAMHOUSE を利用して検証環境 P のログの検証を行った. ブラックリストに登録されていた IP アドレスは正常検知の中の 1 割程であった. これは, 攻撃者による, Tor や踏み台ホストの利用, アドレスを変換する為の経路追加, あるいは NAT をする, といった背景が考えられる.

B, C, D 社のログで, アドレス判定を実施し, スパム登録されていたものの有無を示した.

正常検知の判定の目安になったが, 登録されているアドレスが過剰検知のものも中にあった.


図 5 Blacklist 判定追加

### 5.3 過去データの活用についての評価

本評価においては, ①送信元の国名, ②攻撃タイプ毎の検出の正しさ, ③同一シグナチャーによる検知件数, ④入力値の URI, ⑤レスポンスコード, の 5 種類について, 過去データ（ログＰとログＡ社）から特徴と取り出し, 分析の支援情報とすることを試みた.

#### ① 送信元の国

正常検知の送信元の国, 上位 20 は表 3 のような内容となった. この結果より, 日本以外の国からの攻撃の割合が多いことが分かる.

表 2 送信元の国

| | 国 | 件数 |
|---|---|---|
| 1 | CN | 118 |
| 2 | US | 107 |
| 3 | NL | 45 |
| 4 | FR | 38 |
| 5 | DE | 32 |
| 6 | TH | 31 |
| 7 | IN | 24 |
| 8 | JP | 24 |
| 9 | RO | 22 |
| 10 | RU | 16 |
| 11 | RS | 15 |
| 12 | CA | 11 |
| 13 | BR | 9 |
| 14 | PK | 8 |
| 15 | EC | 7 |
| 16 | PL | 7 |
| 17 | CO | 6 |
| 18 | VN | 6 |
| 19 | ID | 5 |
| 20 | AR | 4 |

② 攻撃タイプ毎の検出の正しさ

攻撃タイプを Command Execution, XSS, HTTP Response Splitting, Path Traversal, SQL Injection の 5 種類に分類し，検出の正しさとの関係を分析した．Command Execution, Path Traversal は正常検知で検出できる場合が多いのに比べ，SQL Injection のアラートは過剰検知が多い傾向があった．

③ 検知件数

検知件数があまりに多い場合（3 桁以上）は，過剰検知の可能性が高いことが分かった．攻撃者の場合，ログインができないことが分かると何度もトライすることは少なく，逆にログインに成功した場合でも目的が果たせれば，相当な数でアクセスすることはないと考えられる．また，DoS 攻撃のような場合は，シグネチャによる検知ではなく，ある閾値を超えたアクセスから判断するふるまい検知の機能（1 分間に 10 回以上のアクセスなど）の方で，攻撃と判断される．

④ URI

攻撃が正しく検知できた（正常検知）場合に URI に入る文字列は，上位 20 は以下のような内容となった．

表 3 URI

| | URI | 件数 |
|---|---|---|
| 1 | / | 165 |
| 2 | /phpmyadmin/scripts/setup.php | 40 |
| 3 | /MyAdmin/scripts/setup.php | 36 |
| 4 | /pma/scripts/setup.php | 23 |
| 5 | /w00tw00t.at.blackhats.romanian.anti-sec:) | 16 |
| 6 | /manager/html | 16 |
| 7 | /cgi-bin/php | 16 |
| 8 | /epegrec/systemSetting.php | 10 |
| 9 | /HNAP1/ | 6 |
| 10 | /web-console/ServerInfo.jsp | 6 |
| 11 | /muieblackcat | 6 |
| 12 | /cgi-bin/php5 | 6 |
| 13 | /robots.txt | 5 |
| 14 | /status | 5 |
| 15 | /jmx-console/HtmlAdaptor | 4 |
| 16 | /dawdwadwaad | 4 |
| 17 | /cgi-bin/php.cgi | 4 |
| 18 | /cgi-bin/php4 | 4 |
| 19 | /cgi-bin/php-cgi | 4 |
| 20 | /sprawdza.php | 4 |

上位は，”/”，”phpmyadmin”，”scripts”といった，不正アクセスを伺わせるものが多い．

⑤ レスポンスコード

レスポンスで「404 Not found」が返ってきていると，アクセスに失敗したことを表し（正常にアクセスできた場合は 200 番のステータスコードが返る），不正アクセス，すなわち，正常検知の可能性が高い傾向にある．

最近は，Web サーバ及び WAF では，ブルートフォース攻撃に対して，403 エラーを返してしまうとそのコンテンツサーバがサーバ上に存在することが分かってしまうので，404 を返すようにしている．

### 5.4 ログの可視化による分析判断

前節で得られた傾向をスコア化し，攻撃による検知か，過剰検知か判断する．

実際に B 社で発生したログを整形，分類し，ログの危険度のレベルにより色付けをした．（赤：危険度高， 黄色：危険度中）．

図 6 ログの危険度レベル

・黄色のログは，404 のレスポンスコードを返しており，ドイツからの通信だった．

・赤色のログは，URI が”/”で，中国からの通信であり，更に送信元がブラックリストに登録されていた．

B,C,D 社の正常検知と過剰検知に分類した正解率である．

表 4 TP と FP の正解率

| | イベント数 | パターン数 | 判定 （正解） | | 正解率 （%） |
|---|---|---|---|---|---|
| | | | True Positives | False Positives | |
| B 社 | 6 | 4 | 4(4) | 0(0) | 100 |
| C 社 | 8673 | 8 | 0(0) | 8(8) | 100 |
| D 社 | 13468 | 16 | 2(3) | 14(13) | 68.8 |

### 5.5 評価のまとめ

ブラックリストへ登録されている攻撃者のソース IP アドレスの登録は 1 割程と少なかった．しかし，URI の統計やレスポンスコードを見ると，攻撃者の動きが読み取り易く，正常検知の判定もし易い．正常検知の可能性をレベル化し，可視化することで，解析者の判断支援につながり，解析作業の短縮化が図れた．

一方，ブラックリストへ登録されていても，顧客側では過剰検知として対処するイベントもあり，その点での判定は難しい面がある．

## 6. まとめと今後の課題

社会問題にもなっている不正アクセスによる Web 改ざん等を減らす為，WAF の導入を検討した．しかし，WAF の導入には，(1)未知の攻撃を検知できない.(2)過剰検知（過剰

検知）の精査に時間を費やす．という主な 2 点のハードルがある． 本研究では，過剰検知の精査を効率化する点に着目し研究内容を述べてきた．この問題点の根本として，大容量のログをどのように分析するか，という課題がある．大容量のログを分類，分析する為に，第 5 章でログの分析技法を検討した．効率良く，ログを精査できることで，解析者の負担と解析日数を減らし，可能であればユーザである管理者もログを扱えるようにすることができれば望ましいと考え取り組んだ．

ログ分析手法の提案として，(1)外部データの活用，(2)過去データの統計分析，(3)ログの可視化を提案し，評価を行った．

評価結果から，外部データの活用として試行したソースアドレスの検索では，ブラックリストに合致する登録の数は 1 割と少なかった．外部データとして利用できるものは，他に多くあるため，それらを活用する効果を継続して検討したい．

一方，過去データからの統計分析では，URI やレスポンスなど，攻撃者の行動を連想させる要素を可視化することは，解析者がより判断しやすい指標を与えやすく，分析の精度を上げる点で効果が見られたと思う．今後は，他にも様々ある統計学的手法を試し，ログ分析に，より最適な手法の探索の検証を実施していく必要がある[8][9].

最後に，可視化について，外部データと統計分析の結果よりスコア化されたものを反映し色付けすることで，解析者の指標に役立つことが評価できた．しかし，可視化を完全に自動化するという課題が残っている．

ログを抽出，整形，分析する一連のログ精査作業において，作業日数を軽減できる見込みができたため，上述した課題を継続して検討しきたい．

## 参考文献

1) “ウェブ感染型マルウェア監視”．株式会社サイバーディフェンス研究所. https://www.cyberdefense.jp/services/malware.html, (参照 2014-03-27).
2) ”情報セキュリティ早期警戒パートナーシップガイドライン”．独立行政法人 情報処理推進機構. http://www.ipa.go.jp/security/ciadr/partnership_guide.html, (参照 2015-01-06).
3) Tammo Kruger, Chrstian Gehl, Konrad Rieck, Pavel laskov. TocDoc: A Self-Healing Web Appliation Firewall, 2010 ACM.
4) “BIG-IP ASM Web アプリケーションファイアウォール セットアップガイド(V11.5.1 対応)”. F5 Networks Japan. http://www.f5networks.co.jp/shared/pdf/BIG-IP_ASM_Easy_Setup.pdf, (参照 2014-08-02).
5) 久保 望. “Excel ファイルも開けるタブ切り替え型の多機能 CSV エディター「SmCsvEdit」”．窓の杜. http://www.forest.impress.co.jp/article/2008/11/27/smcsvedit.html, (参照 2014-08-27).
6) ”IP UTILITIES.NET”. http://www.iputilities.net/blacklist.html, (参照 2014-12-01).
7) ”SPAMHOUSE”. The Spamhouse Project Ltd. http://www.spamhaus.org/, (参照 2014-12-01).
8) Tadeusz Piertraszek. Using Adaptive Alert Classification to Reduce False Positives in Intrusion Detection. In Proceedings of the Symposium on Recent Advances in Intrusion Detection (RAID), pages 102-124. Springer, 2004.
9) Klaus Julisch. Clustering Intrusion Detection Alarms to Support Root Cause Analysis. ACM Transaction on Information and System Security Vol.6, No.4, November 2003, Pages 443-471.