攻撃とその影響を特定可能とするWebサーバログ相関手法

鐘　揚† 朝倉　浩志‡ 谷川　真樹† 大嶋　嘉人†

†NTT セキュアプラットフォーム研究所
180-8585 東京都武蔵野市緑町 3-9-11
{zhong.yang, tanikawa.masaki, oshima.yoshihito}@lab.ntt.co.jp

‡NTT コミュニケーションズ　アプリケーション&コンテンツ部
108-8118 東京都港区芝浦 3-4-1 グランパークタワー
h.asakura@ntt.com

あらまし Web 関連の脆弱性は毎年 1,500 件前後発生しており，それらを悪用する攻撃は大いなる脅威である．Web サービス事業者にとっては攻撃が発生した際にその攻撃と Web サーバに与えられた影響を特定し迅速に対策を行うことで被害を最小限に留めることが重要である．しかし，既存の対策である WAF やファイル改ざん検知ソフトなどでは攻撃の特定とその影響の特定とを個別に実施する必要があり，対策を行うまでに時間を要していた．この問題を解決するため，本稿では Web サーバへの脆弱性を悪用した攻撃とそれによる影響を特定可能とするためのログ相関手法を提案する．本手法はプロセス ID や送信元ポートによって，HTTP リクエストとその他のサーバ内部イベントとを括り付けることにより，攻撃とその影響の関連付けを行うことを特徴とする．提案手法が攻撃とその影響を特定可能であること及び，Web サーバの処理能力に与える影響が比較的小さいことを実験の結果により示す．

A Log Correlation Method to Identify the Target and the Effect of Web Attacks

Yang ZHONG† Hiroshi ASAKURA‡ Masaki TANIKAWA† Yoshihito OSHIMA†

†NTT Secure Platform Labotatories
3-9-11 Midori-cho, Musashino-shi, Tokyo, 180-8585, JAPAN
{zhong.yang, tanikawa.masaki, oshima.yoshihito}@lab.ntt.co.jp

‡NTT Communications Corporation Applications & Content
Granparktower Bldg. 3-4-11 Shibaura, Minato-ku, Tokyo 108-8118, JAPAN
h.asakura@ntt.com

Abstract Web-related vulnerabilities have been discovered 1,500 cases every year and attacks that exploit these vulnerabilities are serious threats. It is important to determine the targeted web application and the effects when attacked. However, existing countermeasures such as WAF and integrity check software cannot determine the target and the effects efficiently and results in time-consuming for prevention. To overcome this problem, we propose an efficient log correlation method that determines the targeted web application and the effects simultaneously. Our approach is characterized in correlation with HTTP requests and other events using process ID and source port of network connections. The experimental results show that our approach has the capability of determination of web attacks effects and it is relatively small impact on the performance of web server.

1 はじめに

表 1 は CVE(Common Vulnerabilities and Exposures)[1] が公開している 2010 年から 2014 年に発見された毎年の全脆弱性の件数と Web 関連の脆弱性の件数を示している．Web 関連の

表 1: 新たに発見された脆弱性の件数

年	全脆弱性	Web 関連	割合
2010	4,639	1,160	25.0%
2011	4,150	1,104	26.6%
2012	5,289	1,582	29.9%
2013	5,186	1,505	29.0%
2014	7,903	1,773	22.0%

脆弱性は毎年 1,500 件前後発生しており，それらを悪用する攻撃は大いなる脅威である．Web サービスを行う事業者にとっては，攻撃そのものであるアクセスのみならず，その攻撃によって Web サーバに与えられた影響を効率的に特定し，迅速に対策を行うことで被害を最小限に留めることが重要である．例えば，OS コマンドインジェクション攻撃が行われた場合，攻撃を行った HTTP リクエストをネットワークアクセスログや Web アクセスログから発見することも重要だが，その攻撃によって挿入された OS コマンドが Web サーバ上で実際に実行されたかどうかも特定する必要がある．

しかし，既存の Web セキュリティ対策である WAF(Web Application Firewall) やファイル改ざん検知ソフトなどでは攻撃の特定とその影響の特定を個別に実施する必要があり，対策を行うまでに時間を要していた．例えば WAF では攻撃 HTTP リクエストを検知することは可能だが，攻撃による Web サーバへの影響を特定することは困難である．既知の限定的な攻撃であれば遮断することで Web サーバに影響を与えないことも可能だが，未知攻撃や誤検知が発生しやすい環境には適用できない．一方，ファイル改ざん検知ソフトは攻撃により生じたファイルの改ざんを検知することは可能だが，攻撃で狙われた Web アプリケーションや攻撃元 IP アドレスなどの情報がなく，攻撃自体の究明が困難である．

[1] http://cve.mitre.org

この問題を解決するため，本稿では Web サーバの脆弱性を悪用した攻撃とその影響を同時に特定可能なログ相関手法を提案する．本手法は，HTTP リクエストを処理したプロセスを識別するプロセス ID(PID) やそのプロセスが行ったネットワークアクセスの送信元ポートを用いて，HTTP リクエストとファイルアクセスやネットワークアクセス，DB アクセスなどのサーバ内部のイベントとの関連付けを行うことで，攻撃とその影響を同時に特定することを特徴とする．提案手法が攻撃と影響を特定可能であること及び，Web サーバの処理能力に与える影響が比較的小さいことを実験の結果により示す．

2 章では既存の攻撃検知方式を，監視・分析する対象の違いから 2 つ取り上げる．3 章では Web サーバにおける攻撃検知の要件と既存手法の課題を述べる．4 章では提案手法及びその実装について解説を行う．5 章では実験の結果及び評価について述べる．6 章では今後の課題について述べ，本稿をまとめる．

2 Web サーバに対する既存の攻撃検知方式

ここでは攻撃検知方式が監視・分析する対象の違いから 2 つのアプローチを取り上げ，それぞれの概略を説明する．

2.1 ネットワーク型

ネットワーク型攻撃検知方式は通信電文の内容に基づいて攻撃を検知する．利用する情報は HTTP メッセージに現れる攻撃の特徴となる．Web サーバに対する攻撃の検知に特化したものに WAF がある．商用の WAF には Imperva SecureSphere[2]，F5 BIG-IP ASM[3] など，オープンソースなものとしては ModSecurity [1] などが存在する．学術的な論文としてはリクエスト URL に含まれるパラメタに着目した手法 [2]

[2] http://www.imperva.jp/Products/WebApplicationSecurity

[3] https://f5.com/jp/products/modules/application-security-manager

や HTTP リクエストに含まれるトークンの順序性に着目した手法 [3] が提案されている．

2.2 ホスト型

ホスト型攻撃検知方式は OS やアプリケーションが出力する情報に基づいて攻撃検知を行う．OS やアプリケーションが出力する情報とはファイルアクセス，ネットワークアクセスなどのイベントを指している．ホスト型の攻撃検知ソフトとして利用できるものに Tripwire [4] や OSSEC[4] などが存在する．Tiripwire はファイルの変更に着目した攻撃検知を行うアプローチであり，OSSEC はホストが出力するログの異常なパターンを予めルールとして定義し，攻撃検知を行うアプローチをとっている．学術的な論文としてはプログラムのシステムコールの挙動に基づいて攻撃検知を行う手法 [5] が提案されている．

3 Web サーバに対する攻撃検知方式の要件と既存手法の課題

攻撃による被害を最小限に留める観点から，Web サーバの攻撃検知方式の要件として以下の 2 点を同時に満たすことが特に重要である．

攻撃の特定

攻撃の特定とは攻撃の対象となった Web アプリケーションを特定することであり，より厳密には`/index.php` などのリクエスト URI を特定することを指す．攻撃を特定する目的は再度攻撃を受けないために脆弱性の修正などの対策を行うことにある．攻撃が発生したことがわかっても，その対象となった箇所を特定し，対処できなければ再度同じ攻撃を受け，被害を繰り返してしまう．

攻撃による影響の特定

攻撃による影響とは攻撃によって引き起こされたファイルアクセスやコマンド実行などのサーバ内部のイベント (以降単にイベントと呼ぶ) を指している．これらのイベントを特定することは攻撃による被害の範囲・状況等を把握する上で必要である．Web サービス事業者にとってはこの点が把握できれば，ユーザ通告は必要かまた，ユーザ通告する場合はどういった対策をしてもらうべきかを判断することが可能となる．

以上の要件について，OS コマンドインジェクション攻撃を例に挙げて具体的に示す．指定した IP アドレスを ip パラメタに格納し，その IP アドレスに対して ping コマンドを実行した結果を返す ping.php という Web アプリケーションを想定する．Web アプリケーションが ip パラメタへの入力値のチェックを怠っていた場合，以下に例示する HTTP リクエストを送ることで，任意の OS コマンドを実行できる．

```
ip=1.1.1.1;cat /etc/passwd
```

ここで，攻撃の特定とは即ち，

```
GET /ping.php?ip=1.1.1.1;cat /etc/passwd
```

という HTTP リクエストで攻撃の対象となった Web アプリケーションのパス`/ping.php` を特定することである．

攻撃による影響の把握とは攻撃者が挿入した OS コマンド `cat /etc/passwd` を特定することである．実行された OS コマンドから Web サーバのユーザ情報が攻撃者によって取得された可能性が高いため，ID・パスワード情報を変更するようユーザへ的確に指示することが可能となる．

表 2: 既存方式における要件の充足性

攻撃検知方式	攻撃の特定	影響の特定
ネットワーク型	✓	
ホスト型		✓

しかし，2 章で述べた各手法では表 2 に示すように上記の 2 つの要件を同時に満たすことはできない．ネットワーク型攻撃検知方式では攻撃の HTTP リクエストを発見できるが，その攻撃が成功してシステムに影響を及ぼしたか，あるいは脆弱性は存在せずに攻撃が失敗したかまで把握できない．逆に，ホスト型攻撃検知方式では攻撃による影響であるかもしれないサー

[4] `http://www.ossec.net/`

バ内での不審な事象を把握できるものの，それが Web アプリケーションへの攻撃で生じたものなのか，またどの Web アプリケーションへの HTTP リクエストによって起きたのかを知ることができない．

4 提案手法と実装

図 1 に示すように提案手法のキーアイデアは Web サーバへの HTTP リクエストとその HTTP リクエストをアプリケーションが処理する時にサーバ内部で発生する様々なイベントと関連付ける部分にある．そのため HTTP リクエストが攻撃として検知された場合，それによって Web サーバで発生したイベントを同時に把握することが可能となる．攻撃検知を行う部分は例えば著者らが提案した HTTP リクエストに対するアノマリ検知手法 [6] を採用することができる．両者を組み合わせることにより，攻撃による影響の有無や範囲を把握することが実現可能である．次節以降にて対象とするイベントと HTTP リクエストとイベントの関連付け手法について詳しく説明する．

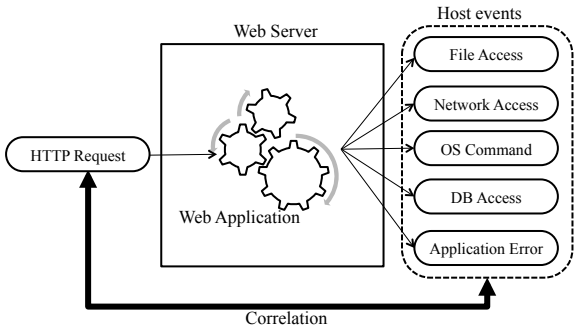

図 1: 提案手法のキーアイデア

4.1 監視対象イベント

CVE などで報告される攻撃を調査したところ，悪用される脆弱性は様々であるが，攻撃成功後に行う行動は限定的である．例えば DB 内に格納されているデータを奪取したり，ファイルを操作して Web ページを改ざんしたり，プログラムコードあるいはコマンドを実行する等の手口に集約される．このため，表 3 に示されるイベントを監視することで攻撃による影響を十分に把握できると考える．

表 3: 影響を把握するために必要なイベント

イベント種類	把握可能な影響
ファイルアクセス	攻撃によるファイルの読出し，作成及び編集
ネットワークアクセス	攻撃によるファイルダウンロード，C&C サーバとの通信及び，他のホストのスキャン
OS コマンド実行	攻撃による OS やアプリケーションへの設定変更
アプリケーションエラー	攻撃によるアプリケーションの異常終了
DB アクセス	攻撃による DB の読出し，書込み

4.2 イベント相関方法

4.2.1 PID，PPID に基づく相関

通常，Web サーバは HTTP リクエストを受け取るとあるプロセスにその HTTP リクエストの処理を割り当てる．例えば，Apache はデフォルトで Prefork というモードで動作しており，各 HTTP リクエストは待機状態にある Web サーバのプロセスの 1 つに割り当てられ処理される．Prefork モードでは 1 つのプロセスは同時に複数の HTTP リクエストを処理しないという特徴がある．この Prefork モードを採用している Web サーバ (Apache[5]，Nginx[6] など) では HTTP リクエストを処理したプロセスの PID を出力可能である．

ファイルアクセス，ネットワークアクセス，OS コマンド実行などのイベントは OS のシステムコールを監視することで取得できる．システムコール監視の場合，システムコールの種類，引数，戻り値の他，システムコールを発行したプロセスの PID も取得可能である．

そのため，PID を用いることで，HTTP リクエストとファイルアクセス，ネットワークアクセス及び OS コマンド実行を関連づけることが可能となる．しかし，PID は有限であり，OS で

[5] http://httpd.apache.org/
[6] http://nginx.org/en/

再利用される可能性及び，Prefork モードでも再度同じプロセスが別の HTTP リクエストに割り当てられる可能性があるため，どのタイミングのどのプロセスから発生したイベントなのかを区別する必要がある．そこで，時間的制約 Δ_a を導入し，ある HTTP リクエストを処理したプロセスの PID と同じ値の PID または PPID を持つイベントであり，かつ HTTP リクエストの発生時刻とそのイベントの発生時刻が Δ_a 以内である場合，そのイベントをその HTTP リクエストに関連するものとする．図 2 に PID，PPID に基づく関連付けの例を示す．HTTP リクエストの PID が 1 であるため，PID が 1 であるファイルアクセス，PPID が 1 であるネットワークアクセス及び OS コマンド実行イベントが関連づけられる．

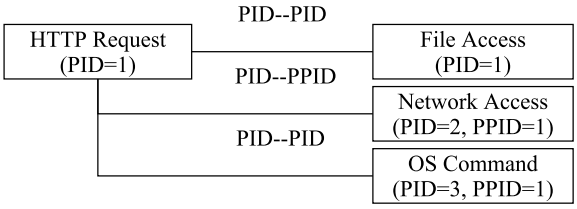

図 2: PID，PPID に基づく相関の例

4.2.2 PID，送信元ポートに基づく相関

Web サーバと DB サーバの分離が一般的である現在，Web アプリケーションと DB 間は TCP 通信を利用している場合が多い．通常，DB アクセスの発端となった HTTP リクエストの PID 情報はこの TCP 通信によって DB サーバ側へは伝搬されないため，PID，PPID に基づく相関を利用できない．しかし，TCP 通信の特性として送信元ポートによって各 TCP セッションを識別するため，DB アクセスの内容は TCP 通信の送信元ポートを参照することで一意に識別可能である．また，前節で述べた通りネットワークアクセスと HTTP リクエストは PID を用いて関連づけられるため，PID と送信元ポートの双方を用いることで，HTTP リクエストと DB アクセスを関連づけることが可能となる．ただし，送信元ポートも有限であり，OS で再利用される可能性があるため，時間制約 Δ_b を導入して関連付けの条件にする．つまり，ある HTTP リクエストを処理するプロセスから発生したネットワークアクセスの送信元ポートとある DB アクセスの送信元ポートが同じ，かつ HTTP リクエスト，ネットワークアクセス，DB アクセスのイベント発生時刻が Δ_b 以内である場合，それぞれ関連するものとする．図 3 に PID，送信元ポートに基づく関連付けの例を示す．同じ PID であることから HTTP リクエストとネットワークアクセスが関連付けられ，同じ送信元ポートであることから，DB アクセスとネットワークアクセスが関連づけられ，よって HTTP リクエストと DB アクセスが関連づけられる．

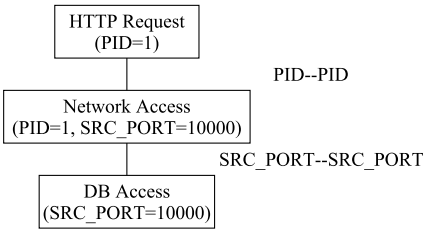

図 3: PID，送信元ポートに基づく相関の例

4.2.3 時刻に基づく相関

アプリケーションエラーは個々のアプリケーションが出力するイベントであり，出力される情報はアプリケーションへの依存性が高く，PID や送信元ポートといった確実に HTTP リクエストと関連付けできる情報が存在しない．しかし，現在の Web サーバの処理スピードの速さを鑑みれば，HTTP リクエストの処理とその処理途中で出力されるエラーの発生の時刻差は大きく違わない．また，エラーの発生数は通常，大量でないことが想定されるため，イベントの発生時刻が近ければ，同一の処理から発生したものとして扱って問題ないと考えられる．よって，HTTP リクエストの発生時刻とアプリケーションエラーの発生時刻の差が Δ_c 以内の場合，それらを関連するものとする．

4.3 実装

提案手法を実装したシステムのアーキテクチャ概要を図 4 に示す．Web アプリケーション及び DB は別のサーバで動作している環境を想定した．Web サーバプログラムには Apache 及び

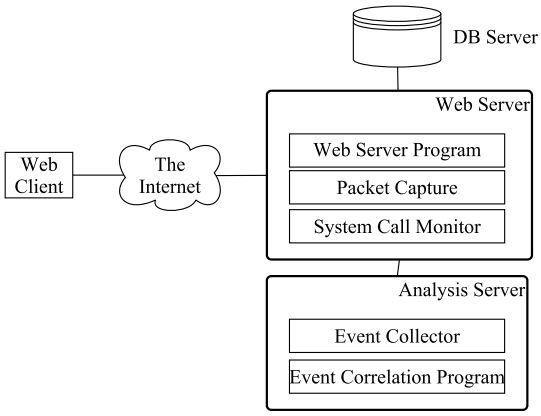

図 4: 提案システムのアーキテクチャ概要

PHP，OS には Ubuntu，DB サーバプログラムには MySQL を用いた．Web サーバ内のファイルアクセス，ネットワークアクセス，OS コマンド実行に関する情報を取得するため，システムコール監視機構である Linux Audit[7] を利用した．Linux Audit は予め設定したルールに基づきカーネル内をフックしてプログラムのシステムコール実行結果を出力する．監視するシステムコールは open，close や access などのファイルアクセスに関するもの，socket や connect などのネットワークアクセスに関するもの，OS コマンド実行を表す execve とした．また，DB アクセスに関する送信元ポートなどの情報を取得するため，パケットキャプチャである tshark を利用した．tshark は MySQL プロトコルを解釈する機能を有しており，それを利用して SQL クエリ，接続ユーザ名等の情報を抽出した．HTTP リクエスト及びアプリケーションエラーに関する情報はそれぞれ Apache から出力されるアクセスログとエラーログから取得した．取得したイベントは分析エンジンに送られ，整形され，イベント相関手法が適用される．イベントの整形には Fluentd[8] を利用し，提案手法に基づきイベント相関を実行する部分は Python スクリプトを用いて実装した．

5 実験および評価

5.1 攻撃と影響の関連付けの可否

Web サーバに脆弱なアプリケーションをインストールし，攻撃を行い，その際に HTTP リクエストとその影響であるイベントが関連づけられるかを調べた．図 5 は phpLDAPAdmin への OS コマンドインジェクション攻撃 (CVE-2011-4075) 時の関連付け後のログを示している．各行はイベントの種類，PID，PPID，イベントを要約した情報を表している．例えば，1 行目は Web アプリケーション `cmd.php` への HTTP リクエストが発生し，それを処理したプロセスの PID が 36205 であることを示している．2 から 10 行目はその HTTP リクエストによってライブラリファイルなどへのアクセスが発生していることを表している．12，13 及び 15 行目は攻撃者によって挿入された OS コマンドが実行され，`/etc/passwd` へのファイルアクセスが行われたことを表している．これらのイベントの PID はそれぞれ異なるが，PID，PPID に基づく相関によって関連付けられたものである．

図 6 に DVWA(Damn Vulnerable Web Application)[9] への SQL インジェクション攻撃時の関連付け後のログを示す．9 行目は DB アクセス専用のネットワークアクセスイベントを表しており，送信元ポートは 48097 である．10 行目は SQL クエリが発行されたことを示しており，同じ送信元ポート 48097 を利用しているため関連づけられたものである．これら一連のログから SQL インジェクション攻撃を含む HTTP リクエストが処理され，実際に DB でその SQL クエリが実行されたことが分かる．

以上のように提案手法は攻撃とその影響を関連づけることが可能であり，両者を同時に特定できる効果があることが確認できた．

[7] `https://people.redhat.com/sgrubb/audit/`

[8] `http://www.fluentd.org/`

[9] http://www.dvwa.co.uk/

```
1  http_req, 36205, 1570, POST /phpldapadmin/htdocs/cmd.php HTTP/1.0
2  file, 36205, 1570, read /var/www/phpldapadmin/lib/schema_functions.php
3  file, 36205, 1570, read /var/www/phpldapadmin/lib/xmlTemplates.php
4  file, 36205, 1570, read /var/www/phpldapadmin/lib/createlm.php
5  file, 36205, 1570, read /var/www/phpldapadmin/lib/page.php
6  file, 36205, 1570, write /var/lib/php5/sess_q30nf1s15k8u1qm7ttd2s1hqb1
7  ...
8  file, 36205, 1570, read /etc/resolv.conf
9  file, 36205, 1570, read /etc/ldap/ldap.conf
10 file, 55649, 36205, read /lib/x86_64-linux-gnu/libc.so.6
11 ...
12 command, 55651, 36205, sh -c cat /etc/passwd
13 command, 55652, 55651, cat /etc/passwd
14 file, 55651, 36205, read /etc/ld.so.cache
15 file, 55652, 55651, read /etc/passwd
16 file, 36205, 1570, read ldaprc
17 file, 36205, 1570, read /dev/urandom
```

図 5: OS コマンドインジェクション攻撃 (CVE-2011-4075) 時の関連付け後のログ (抜粋)

```
1  http_req, 36722, 1570, GET /dvwa/vulnerabilities/sqli/?id=1%2527+union+all+select+1%252
   Cversion%2528%2529&Submit=Submit HTTP/1.1
2  file, 36722, 1570, read /var/www/dvwa/vulnerabilities/sqli/index.php
3  file, 36722, 1570, read /var/www/dvwa/dvwa/includes/dvwaPhpIds.inc.php
4  file, 36722, 1570, read /var/www/dvwa/vulnerabilities/sqli/source/low.php
5  file, 36722, 1570, read /var/www/dvwa/vulnerabilities/sqli/index.php
6  file, 36722, 1570, write /var/lib/php5/sess_2pv0sf4scg8hml4qir8i72cl45
7  ...
8  network, 36722, 1570, connect 127.0.0.1:3306
9  db_network, 36722, 1570, src:127.0.0.1:48097 dst:127.0.0.1:3306
10 db, -1, -1, src:127.0.0.1:48097 dst:127.0.0.1:3306 SELECT first_name, last_name FROM
   users WHERE user_id = '1' union all select 1,version() #'
```

図 6: SQL インジェクション攻撃時の関連付け後のログ (抜粋)

5.2 パフォーマンスへの影響

次に，代表的な Web アプリケーションである osCommerce, phpMyAdmin, Wordpress, MediaWiki に対して提案手法を適用した場合の Web サーバへのパフォーマンス影響を測定した．イベント相関部分やパケットキャプチャは別ハードウェアで Web サーバからオフロードできるため，システムコール監視部分による影響のみを計測した．

5.2.1 HTTP リクエストのスループット

スループットは Apache Bench[10] を使って計測した．多重接続がある環境を想定して，各 Web アプリケーションのトップページに対して並列に 10 アクセス行いながら合計 1000 アクセスを 10 セット繰り返し，その平均値を求めた．その結果，表 4 に示すようにシステムコール監視によって Web サーバに 15%から 40%のスループットの低下がみられた．調査の結果，その主たる原因はライブラリファイルへのアクセスが大量に発生することと分かった．低下度が大きい phpMyAdmin や osCommerce は他の 2 つに比べて，もともとスループットが高いことから，高いスループットにある Web アプリケーションほど低下度が大きい傾向があると言える．

表 4: スループット (リクエスト数／秒)

AP の名称	w/o Audit	Audit	低下度
osCommerce	44.97	27.99	37.8%
phpMyAdmin	44.11	26.19	40.7%
Wordpress	11.78	10.01	15.1%
MediaWiki	9.32	7.78	16.6%

[10] http://httpd.apache.org/docs/2.4/programs/ab.html

5.2.2 ログの容量

各 Web アプリケーションに対して 1 度だけ HTTP リクエストを行ったときに出力されるログの容量を計測した．表 5 にログの行数及びサイズ (KB) を示す．各 Web アプリケーションとも，1 アクセスにつき 100KB 以上のログを出力しており，大規模な Web サイトではディスク容量の圧迫が課題となる．前節で述べたように，

表 5: 1HTTP リクエストあたりのシステムコールログ量

AP の名称	行数	サイズ (KB)
osCommerce	601	145
phpMyAdmin	547	126
Wordpress	553	130
MediaWiki	828	202

ファイルアクセスに関するシステムコールが非常に多いのが原因であるため，それらを除外することで，大幅に削減できると考える．

5.2.3 メモリ使用量

実メモリの使用量は，1000 アクセスを行うときに 1 秒毎に Linux Audit プロセスの実メモリ使用量を測定し，その最大値を求めた．表 6 に示す通り，それぞれ数 MB 程度であった．現在の Web サーバのメモリ搭載量は GB 単位であることを鑑みると，提案手法の実メモリ使用量が Web サーバに与える影響は小さいと考える．

表 6: 実メモリ使用量

AP の名称	サイズ (MB)
osCommerce	4.65
phpMyAdmin	6.82
Wordpress	6.18
MediaWiki	4.56

6 まとめ

本稿では Web サーバに対する攻撃が起きた際，攻撃とその影響を同時に特定可能とするログ相関手法を提案した．本手法では PID や送信元ポートといった情報を用いてイベントの関連付けを行う．実験により提案手法は，攻撃の原因と影響を同時に特定できることを示した．しかし，本手法にはまだ次の 2 つの課題が存在する．まず一つ目はイベント関連付けの精度である．処理時間の短い HTTP リクエストが時間制約 Δ_a 以内に連続して発生し，かつ，同じプロセス (同一 PID) で処理された場合，それらにより発生した個々のイベントがどちらの HTTP リクエストによるものなのか特定することができない．もう一つの課題はシステムコール監視による Web サーバのパフォーマンスへの影響である．先に述べたようにライブラリファイルへのアクセスを監視，記録の対象から除外することなどにより，スループットの向上，ログ容量の削減を図る必要がある．今後は以上の課題を解決し，より正確にイベントを相関させ，パフォーマンス劣化も抑えた監視方法を実現したい．

参考文献

[1] Ryan Barnett. Waf virtual patching challenge: Securing webgoat with modsecurity. In *Breach Security*, 2009.

[2] Kruegel Christopher and Giovanni Vigna. Anomaly detection of web-based attacks. In *Proceedings of the 10th ACM conference on Computer and communications security*, 2003.

[3] Kenneth L Ingham, Anil Somayaji, John Burge, and Stephanie Forrest. Learning dfa representations of http for protecting web applications. In *Computer Networks*, 2007.

[4] Gene H Kim and Eugene H Spafford. The design and implementation of tripwire: A file system integrity checker. In *Proceedings of the 2nd ACM Conference on Computer and Communications Security*, 1994.

[5] Steven A Hofmeyr, Stephanie Forrest, and Anil Somayaji. Intrusion detection using sequences of system calls. In *Journal of computer security*, 1998.

[6] 鐘揚, 朝倉浩志, 高倉弘喜, 大嶋嘉人. Web アプリケーションのパラメタを悪用する攻撃のアノマリ検知手法. コンピュータセキュリティシンポジウム 2014 論文集, 2014.