

# Web Application Firewall (WAF)

読本

改訂第2版

Web Application Firewall を理解するための手引き



**IPA**

独立行政法人 情報処理推進機構  
セキュリティセンター

2011年12月

本書は、以下の URL からダウンロードできます。

「Web Application Firewall(WAF)読本」

<http://www.ipa.go.jp/security/vuln/waf.html>

# 目次

目次.....	1
はじめに .....	2
本書の対象読者.....	2
本書の構成.....	2
本書の注意事項.....	3
第2版の主な改訂内容.....	3
1. WAFによるウェブアプリケーションの脆弱性対策 .....	4
1.1. 攻撃による影響を低減する対策：WAF .....	4
1.2. IPAの取り組みからみるウェブアプリケーションへの攻撃と脆弱性対策の実情.....	5
1.3. WAFに関する取り組み.....	7
2. WAFの概要 .....	9
2.1. WAFとは.....	9
2.2. WAFとFW、IPSの違い.....	10
2.3. WAFの種類.....	12
2.4. WAFが有効な状況.....	14
3. WAFの詳細 .....	16
3.1. WAFの設置.....	16
3.2. WAFの機能.....	18
3.3. WAF機能における留意点.....	28
4. WAF導入におけるポイント.....	30
4.1. 導入判断.....	31
4.2. 導入.....	35
4.3. 運用.....	46
5. IPAにおけるWAF導入・運用事例.....	49
5.1. まえがき.....	49
5.2. 導入判断.....	51
5.3. 導入.....	55
5.4. 運用.....	69
5.5. WAF導入・運用結果総括.....	72
付録A. オープンソースソフトウェアの紹介.....	73
ModSecurity.....	73
WebKnight.....	81
付録B. 商用製品の紹介.....	90
用語集.....	93

# はじめに

---

ウェブアプリケーションの脆弱性を悪用した攻撃からウェブアプリケーションを保護するセキュリティ対策の一つとして、Web Application Firewall (WAF) の導入が挙げられます。独立行政法人情報処理推進機構 (IPA) では、WAF の理解を手助けする情報として、WAF の概要、機能の詳細、導入におけるポイントをまとめた本書を作成しました。

本書が WAF の導入を検討する一助となれば幸いです。

## 本書の対象読者

---

対象読者は、「WAF の導入を検討したいウェブサイト運営者」としています。

本書ではウェブサイト運営者を「ウェブサイトを企画し、運営する立場の組織または個人」と定義します。例えば、ウェブサイト <http://www.ipa.go.jp/> のウェブサイト運営者は IPA となります。

## 本書の構成

---

本書は 5 つの章と 2 つの付録で構成しています。

「1. WAF によるウェブアプリケーションの脆弱性対策」では、IPA の取り組みからみたウェブアプリケーションへの攻撃および脆弱性対策の実情、各機関における WAF に関する取り組みを紹介しています。

「2. WAF の概要」では、WAF に関する概要をまとめています。この章では、WAF とはどのようなものかを理解できることを目的としています。

「3. WAF の詳細」では、WAF の機能をまとめています。この章では、WAF にはどのような機能があり、その機能にどのような留意点があるかを理解できることを目的としています。

「4. WAF 導入におけるポイント」では、WAF を導入する際の「導入判断」・「導入」・「運用」の各フェーズにおける検討すべきポイントをまとめています。

「5. IPA における WAF 導入・運用事例」では、IPA におけるオープンソースソフトウェアの WAF「ModSecurity」<sup>モッドセキュリティ</sup>の導入および運用事例をまとめています。WAF 導入における「導入判断」・「導入」・「運用」の各フェーズにおいて、実際に IPA がどのように検討し取り組んだかを紹介します。「4. WAF 導入におけるポイント」とあわせて読むことで、WAF 導入におけるポイントを理解できることを目的としています。

「付録 A. オープンソースソフトウェアの紹介」では、オープンソースソフトウェア「ModSecurity」<sup>ウェブナイト</sup>、「WebKnight」の導入例を紹介しています。

「付録 B. 商用製品の紹介」では、本書の作成にご協力いただいた企業の WAF 製品・サービスを紹介しています。

## 本書の注意事項

---

本書では、WAF の基本的な機能や動作、課題等を解説しています。WAF 製品によっては、本書の解説にそぐわない場合があります。

## 第 2 版の主な改訂内容

---

第 2 版では、「4. WAF 導入におけるポイント」を拡充し、IPA がオープンソース WAF 「ModSecurity」を導入、運用した事例を「5. IPA における WAF 導入・運用事例」にまとめました。4 章、5 章をあわせて読むことで、WAF の導入から運用までの流れを具体的に理解できるようになりました。

その他、「2.4 WAF が有効な状況」や、付録 A における「ModSecurity」の導入例を改訂しました。

# 1. WAF によるウェブアプリケーションの脆弱性対策

---

本章では、IPA の取り組みからみたウェブアプリケーションへの攻撃と脆弱性対策の実情、各機関における WAF に関する取り組みを紹介します。

## 1.1. 攻撃による影響を低減する対策:WAF

---

WAF は、ウェブアプリケーションの脆弱性を悪用した攻撃からウェブアプリケーションを保護するセキュリティ対策の一つです。WAF はウェブアプリケーションの実装面での根本的な対策ではなく、攻撃による影響を低減する運用面での対策です。

IPA ではウェブアプリケーションの脆弱性を作り込まないために、「安全なウェブサイトの作り方」<sup>1</sup>や「セキュア・プログラミング講座」<sup>2</sup>といった開発者向けの資料を公開しています。しかしながら、ウェブアプリケーションの脆弱性を悪用した攻撃が後を絶たず、脆弱性関連情報流通基本枠組み「情報セキュリティ早期警戒パートナーシップ」<sup>3</sup>においても、ウェブサイト運営者の諸事情からすぐに脆弱性を修正できないのが実情です。このような実情において、ウェブアプリケーションが攻撃の被害にあわないためのセキュリティ対策の一つとして、WAF の使用が有効だと考えます。

---

<sup>1</sup> <http://www.ipa.go.jp/security/vuln/websecurity.html>

<sup>2</sup> <http://www.ipa.go.jp/security/awareness/vendor/programmingv2/index.html>

<sup>3</sup> [http://www.ipa.go.jp/security/ciadr/partnership\\_guide.html](http://www.ipa.go.jp/security/ciadr/partnership_guide.html)

## 1.2. IPA の取り組みからみるウェブアプリケーションへの攻撃と脆弱性対策の実情

本節では、IPA が取り組んでいる施策を通じて得られた、ウェブアプリケーションへの攻撃とウェブサイトにおける脆弱性対策の実情を紹介します。

### 1.2.1. 「JVN iPedia」への攻撃の実情

IPA と一般社団法人 JPCERT コーディネーションセンター (JPCERT/CC) が運営する「JVN iPedia<sup>4</sup>」のアクセスログを、ウェブサイトの攻撃検出ツール「iLogScanner<sup>5</sup>」で解析したところ、2009年1月から2010年12月にかけて、12,194件の攻撃と思われる通信を確認しました(図 1-1)。

#### ウェブサイトを狙った攻撃があったと思われる件数

解析対象のウェブサイト：JVN iPedia (脆弱性対策情報データベース)  
解析したウェブサーバのアクセスログの期間：2009年1月～2010年12月  
攻撃があったと思われる件数：12,194件、攻撃が成功した可能性の高い件数：0件

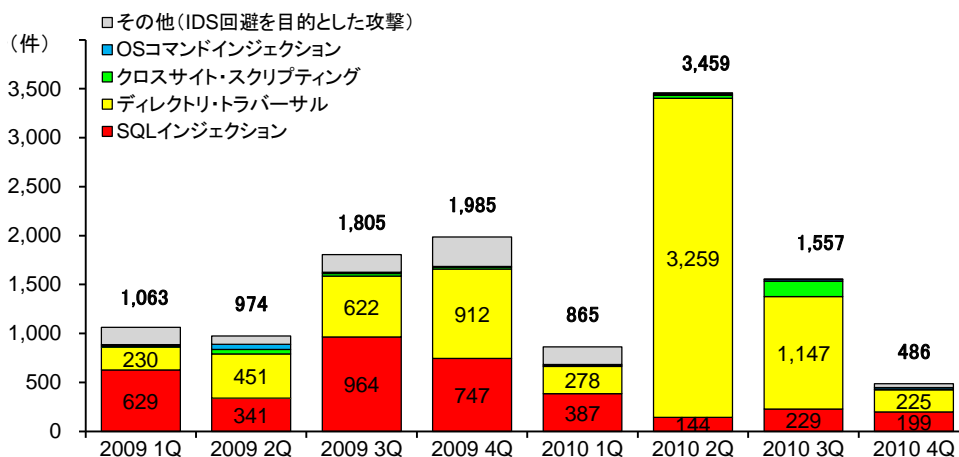


図 1-1 2009年1月から2010年12月の「JVN iPedia」への攻撃件数

企業や組織が運営するウェブサイトはインターネットに公開するため、ファイアウォールなどで、インターネットからウェブサイトへの通信を遮断するわけにはいきません。近年、ウェブアプリケーションの脆弱性を悪用した攻撃により、大手企業のウェブサイトにおける個人情報の漏えい事件がたびたび報道されます。しかし、ウェブサイトを狙った攻撃は、大手企業のウェブサイトだけを対象にしているわけではありません。図 1-1 から分かるように、インターネットに公開しているウェブサイトは絶えず攻撃を受けることとなります。個人、中小企業、大企業といった組織規模を問わず、すべてのウェブサイトが攻撃にさらされる危険があります。

<sup>4</sup> 国内で利用されるソフトウェア等の製品 (OS、アプリケーション、ライブラリ、組込み製品など) の脆弱性対策情報を中心に収集・蓄積する脆弱性対策情報データベース。

<http://jvndb.jvn.jp/>

<sup>5</sup> <http://www.ipa.go.jp/security/vuln/iLogScanner/index.html>

## 1.2.2. 「情報セキュリティ早期警戒パートナーシップ」における脆弱性対策の実情

脆弱性関連情報流通基本枠組み「情報セキュリティ早期警戒パートナーシップ」において、2010年第4四半期（10月から12月）<sup>6</sup>時点で、ウェブサイト（ウェブアプリケーション）の脆弱性関連情報の届出件数は累計で5,338件にのぼります（図1-2）。

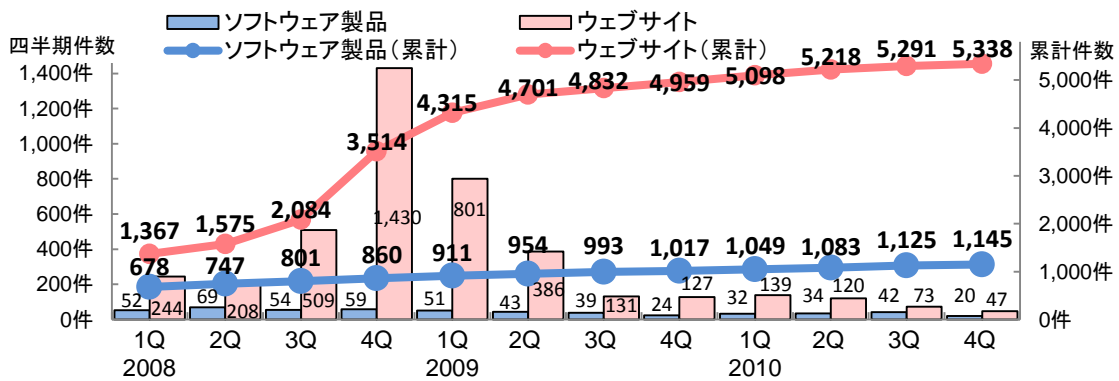


図 1-2 脆弱性関連情報の届出件数の四半期別推移（2010年第4四半期時点）

「情報セキュリティ早期警戒パートナーシップ」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ます。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ありません。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%にのぼります（図1-3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す。

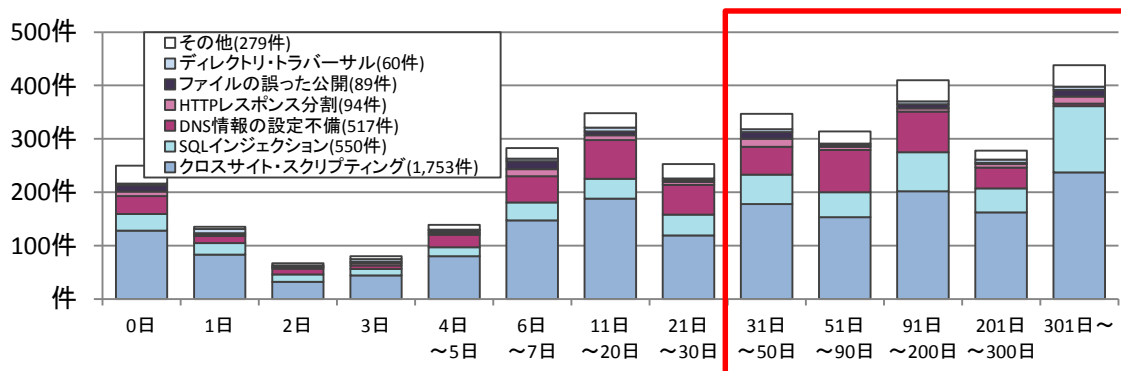


図 1-3 ウェブサイトの修正に要した日数（2010年第4四半期時点）

<sup>6</sup> <http://www.ipa.go.jp/security/vuln/report/vuln2010q4.html>



## 1.3. WAF に関する取り組み

---

本節では、各機関における WAF に関する取り組みを紹介します。

### 1.3.1. KISA の取り組み

---

KISA (Korea Internet & Security Agency) <sup>7</sup>は、ウェブサイトにてオープンソースソフトウェアの WAF を紹介しています。KISA が紹介している WAF<sup>8</sup>は以下の 2 つです。

- Trustwave 社が提供する「ModSecurity」<sup>9</sup>
- AQTRONIX 社が提供する「WebKnight」<sup>10</sup>

KISA は、これらの WAF を提供元企業のウェブサイトだけではなく、自組織のウェブサイトからダウンロードできるようにすることで、WAF の普及を推進しています。さらに、これらの WAF の導入手順書、設定手順書の公開、Q&A の提供、また WAF に関するセミナー紹介などの活動も行っています。

KISA ではオープンソースソフトウェアの WAF を紹介するとともに、ウェブアプリケーション強化ツール「CASTLE」<sup>11</sup>、WebShell<sup>12</sup>検出ツール「WHISTL」<sup>13</sup>も公開しています<sup>14</sup>。

### 1.3.2. OWASP の取り組み

---

OWASP (Open Web Application Security Project) <sup>15</sup>の WAF に関連する活動には、「OWASP Best Practices: Use of Web Application Firewalls」<sup>16</sup>や「OWASP ModSecurity Core Rule Set Project」<sup>17</sup>があります。

「OWASP Best Practices: Use of Web Application Firewalls」では、各種攻撃手法への WAF での防御可否、WAF 導入による効果とリスク、WAF 導入を決定する基準等をまとめ、文書として公開しています。改訂第 2 版公開時点における公開されている文書の最新バージョンは、2008 年 3 月より公開されている Version 1.0.5 です。

「OWASP ModSecurity Core Rule Set Project」では、オープンソースソフトウェアの WAF 「ModSecurity」に設定する検出パターン「Core Rule Set」を開発し、公開しています。このプロジェクトの概要によると、「Core Rule Set」では攻撃に含まれる文字列に焦点を当てた汎用的な作りとなっています。改訂第 2 版公開時点における「Core Rule Set」の最新バージョンは、Version 2.1.2 です。

---

<sup>7</sup> <http://www.kisa.or.kr/>

<sup>8</sup> 本書でも「付録 A. オープンソースソフトウェアの紹介」で ModSecurity, WebKnight の導入例を紹介しています。

<sup>9</sup> <http://www.modsecurity.org/>

<sup>10</sup> <http://www.aqtronix.com/?PageID=99>

<sup>11</sup> [http://toolbox.krcert.or.kr/MMVF/MMVFView\\_V.aspx?MENU\\_CODE=7&PAGE\\_NUMBER=16](http://toolbox.krcert.or.kr/MMVF/MMVFView_V.aspx?MENU_CODE=7&PAGE_NUMBER=16)

<sup>12</sup> WebShell とは、「ウェブサーバに不正にアップロードされるバックドアプログラム」を指します。

<sup>13</sup> [http://toolbox.krcert.or.kr/MMVF/MMVFView\\_V.aspx?MENU\\_CODE=6&PAGE\\_NUMBER=15](http://toolbox.krcert.or.kr/MMVF/MMVFView_V.aspx?MENU_CODE=6&PAGE_NUMBER=15)

<sup>14</sup> 「WHISTL」を利用するためには、KISA への利用申請が必要です。

<sup>15</sup> <http://www.owasp.org/>

<sup>16</sup> [http://www.owasp.org/index.php/Category:OWASP\\_Best\\_Practices:\\_Use\\_of\\_Web\\_Application\\_Firewalls](http://www.owasp.org/index.php/Category:OWASP_Best_Practices:_Use_of_Web_Application_Firewalls)

<sup>17</sup> [http://www.owasp.org/index.php/Category:OWASP\\_ModSecurity\\_Core\\_Rule\\_Set\\_Proiect](http://www.owasp.org/index.php/Category:OWASP_ModSecurity_Core_Rule_Set_Proiect)

### 1.3.3. WASC の取り組み

---

WASC (Web Application Security Consortium) <sup>18</sup>の活動の一つとして、「WAFEC (Web Application Firewall Evaluation Criteria)」<sup>19</sup>の作成があります。WASC では、汎用性のある WAF の評価基準の策定を目標として、WAFEC を作成し公開しています。WASC は、この理由を専門家であっても WAF の評価を行うための基準書の作成が難しく、開発元が異なる WAF を比較することが難しいと説明しています。第 2 版公開時点における WAFEC の最新バージョンは、2006 年 1 月 16 日に公開された Version 1.0 です。

### 1.3.4. PCI SSC の取り組み

---

PCI SSC (Payment Card Industry Security Standards Council) <sup>20</sup>は、カード加盟店を中心としたカード会員データを取り扱う事業者が遵守する必要があるクレジット業界における国際的なセキュリティ基準 PCI-DSS (Payment Card Industry Data Security Standard) <sup>21</sup>を策定しています。

PCI-DSS は、情報セキュリティに対する具体的な実装を要求するセキュリティ基準です。この PCI-DSS の要件 6.6 「Web アプリケーションに対する既知の攻撃に対する防御」において、下記のどちらかを採用することが定められています。

- 定期的なアプリケーションコードの見直し
- WAF の導入

本要件は、2008 年 6 月 30 日までのバージョン 1.1 では「推奨」にとどまっていた。しかし、2008 年 7 月に改定されたバージョン 1.2 において、本要件が「必須」となりました。この PCI-DSS は、2004 年 12 月に策定されて以降、数回バージョンアップがなされており、第 2 版公開時点の最新バージョンは、2.0 です。

---

<sup>18</sup> <http://www.webappsec.org/>

<sup>19</sup> <http://projects.webappsec.org/Web-Application-Firewall-Evaluation-Criteria>

<sup>20</sup> <https://www.pcisecuritystandards.org/>

<sup>21</sup> [https://www.pcisecuritystandards.org/security\\_standards/pci\\_dss.shtml](https://www.pcisecuritystandards.org/security_standards/pci_dss.shtml)

## 2. WAF の概要

本章では、WAF の動作概要、WAF と Firewall (FW)、Intrusion Prevention System (IPS) の違い、WAF の種類、WAF が有効な状況を解説します。

### 2.1. WAF とは

WAF は、ウェブアプリケーションの脆弱性を悪用した攻撃などからウェブアプリケーションを保護するソフトウェア、またはハードウェアです。WAF は脆弱性を修正するといったウェブアプリケーションの実装面での根本的な対策ではなく、攻撃による影響を低減する対策です。

WAF は、WAF を導入したウェブサイト運営者が設定する検出パターンに基づいて、ウェブサイトと利用者間の通信の中身を機械的に検査します (図 2-1)。WAF を使用することで以下の効果を期待できます。

- 脆弱性を悪用した攻撃からウェブアプリケーションを防御する
- 脆弱性を悪用した攻撃を検出する
- 複数のウェブアプリケーションへの攻撃をまとめて防御する

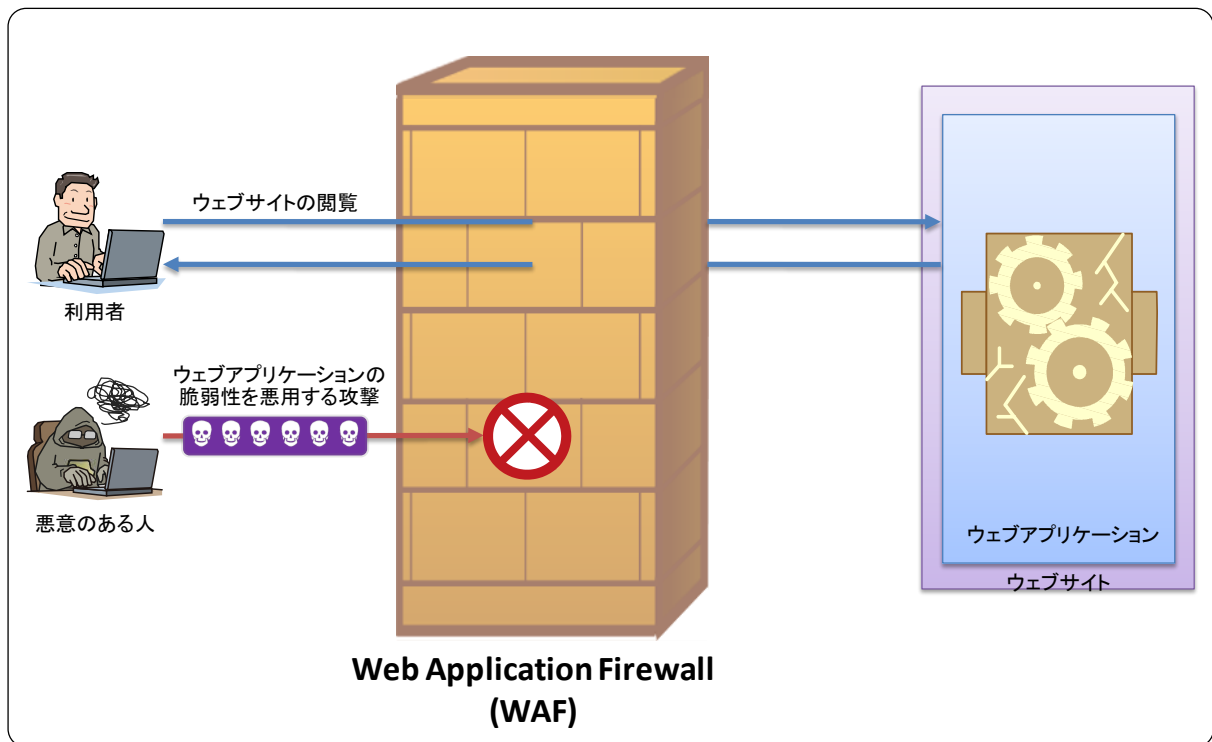


図 2-1 WAF の動作概要

その他に、検出パターンに特徴のある個人情報 (クレジットカード番号等) のパターンを設定しておくことで、特徴のある個人情報が悪意のある人に送信されてしまうことを防ぐといった使い方もできます。

また WAF は、検出パターンに基づき通信の中身を機械的に検査するため、実際に人の目で見ると異なる判定が生じる場合があります。この判定結果により、ウェブアプリケーションの

脆弱性を悪用した攻撃などの悪意ある通信を遮断できない場合や、利用者がウェブサイト閲覧する正常な通信を遮断してしまう場合があります（詳細については、「3.3 WAF 機能における留意点」を参照）。WAF の導入を検討する場合、このことを考慮する必要があります。

## 2.2. WAF と FW、IPS の違い

本節では、WAF と FW、WAF と IPS の違いを解説します。

### 2.2.1. WAF と FW

WAF は名称に「ファイアウォール」を含みますが、FW とは機能が異なります。

FW は、通信における送信元情報と送信先情報（IP アドレスやポート番号等）を基にアクセスを制限するソフトウェア、またはハードウェアを指します。FW を使用すると、サーバで動作しているサービスとの通信を制限できます。例えば、組織内部のファイル共有サービスへのアクセスを組織内部からに限定し、インターネットからそのサービスへのアクセスを禁止するといった対策をとることができます。一般に公開する必要がないサービスへのアクセスを制限することで、それらのサービスに対する不正なアクセスを防止できます。

しかし、企業や組織が運営するウェブサイトはインターネットに公開するため、FW でアクセスを制限するわけにはいきません。このため、FW でウェブアプリケーションの脆弱性を悪用する攻撃を防ぐことはできません（図 2-2）。

一方 WAF は、FW で制限できないウェブアプリケーションへの通信内容を検査することができます。例えば、ウェブアプリケーションの通信内容に、外部からデータベースを不正に操作する「SQL インジェクション攻撃」の特徴的なパターンが含まれていた場合、その通信を遮断するといった対策をとることができます。

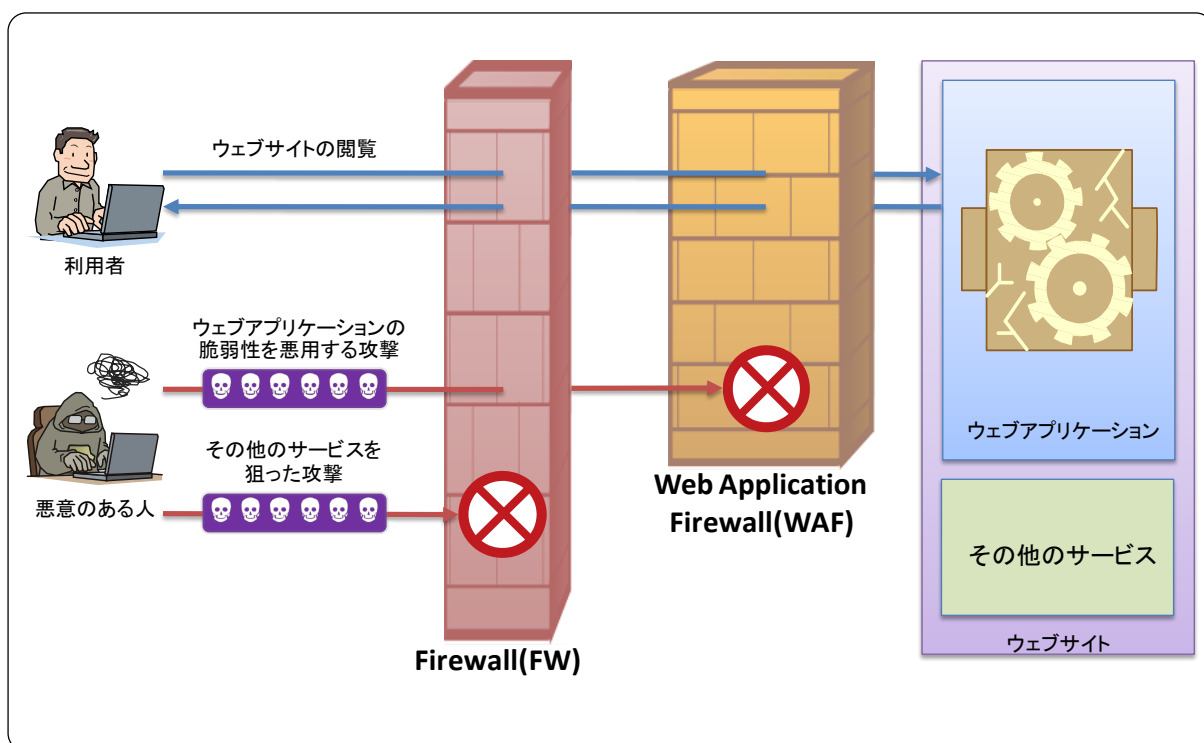


図 2-2 FW と WAF の違い

## 2.2.2. WAF と IPS

WAF と IPS は、どちらも検出パターンに基づいて、通信の中身を検査します。

IPS は、ウェブサイト運営者が設定する検出パターンに基づいて、様々な種類の機器への通信を検査するソフトウェア、またはハードウェアです。一般的に IPS は、様々な種類の攻撃（オペレーティングシステムの脆弱性を悪用する攻撃、ファイル共有サービスへの攻撃等）を防御できます（図 2-3）。IPS は、攻撃の詳細を定義した検出パターンである「ブラックリスト」<sup>22</sup>による検査により、攻撃を防御します。

一方 WAF は、ウェブサイト運営者が設定する検出パターンに基づいて、ウェブアプリケーションへの通信を検査するソフトウェア、またはハードウェアです。IPS が様々な種類の攻撃を防御できることに対して、WAF はウェブアプリケーションへの攻撃を防御できます（図 2-4）。特に、WAF では保護する対象がウェブアプリケーションに特化しており、「ブラックリスト」による検査に加えて、正常な通信を定義した検出パターンである「ホワイトリスト」による検査を行うこともできることが特徴です。

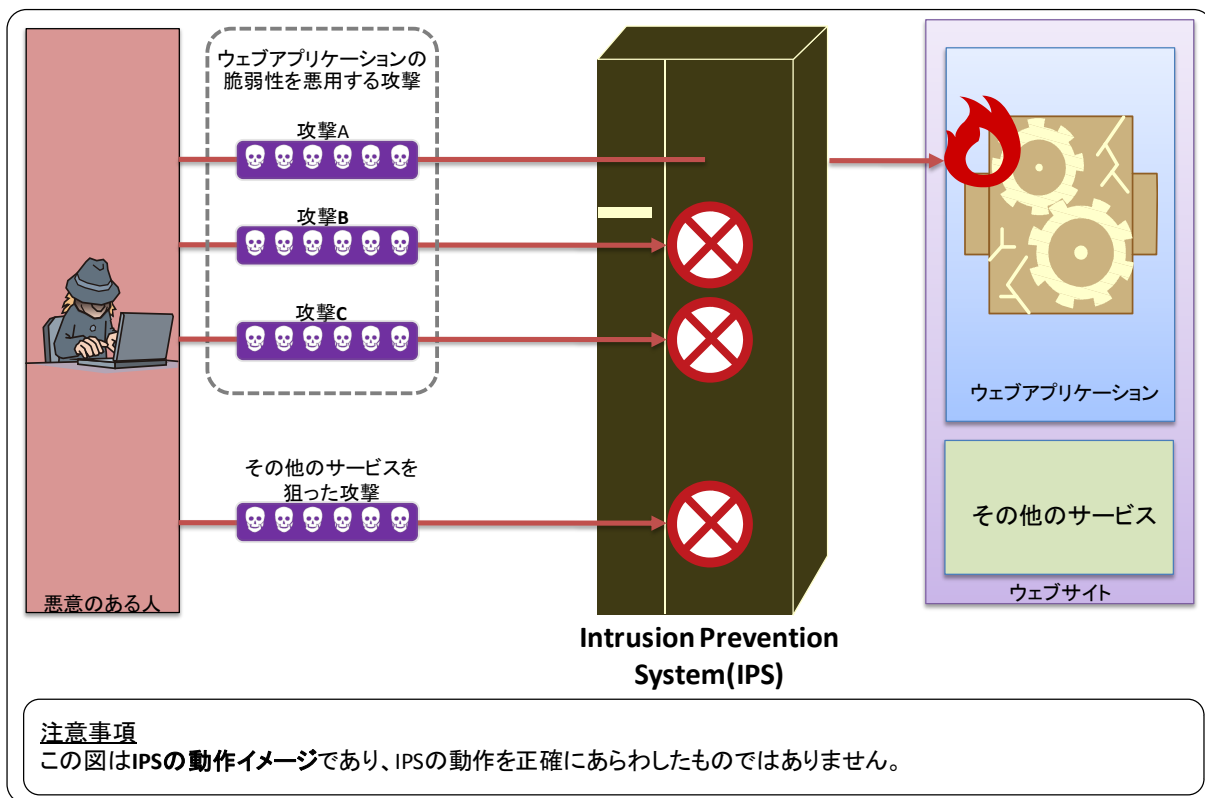


図 2-3 IPS の動作イメージ

<sup>22</sup> 「ブラックリスト」、「ホワイトリスト」については、「3.2 WAF の機能」を参照してください。

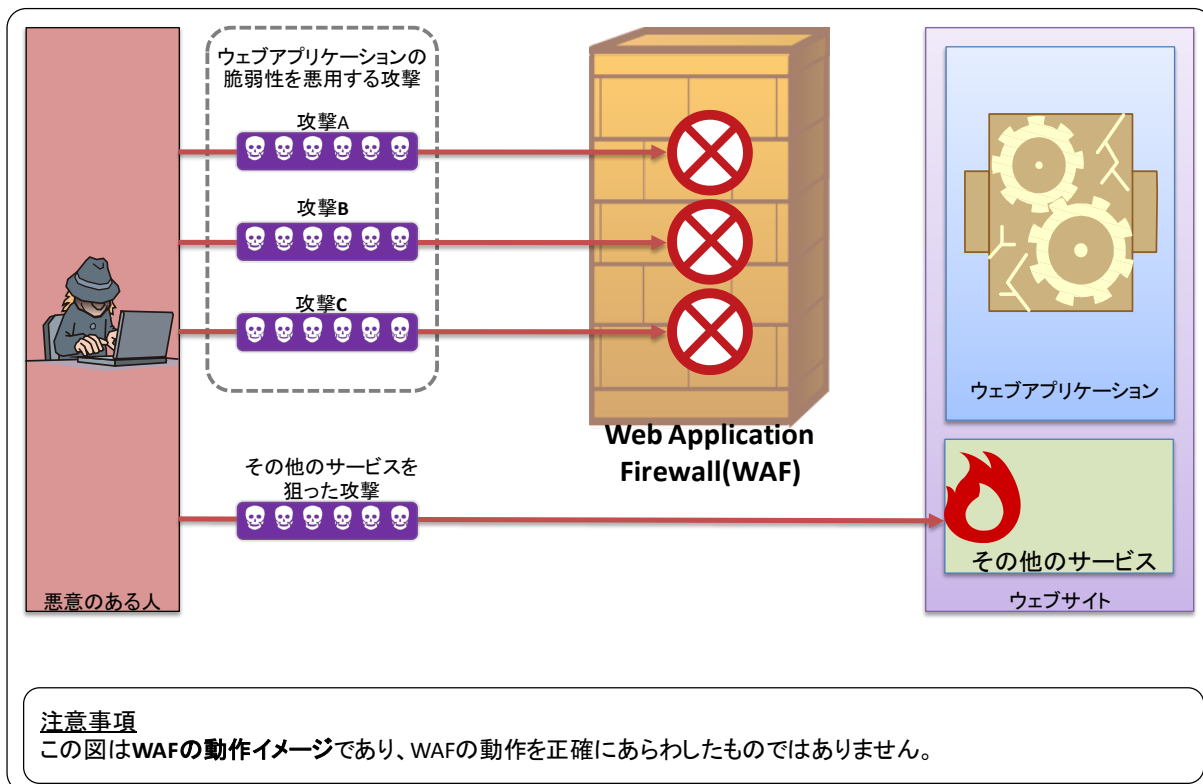


図 2-4 WAF の動作イメージ

## 2.3. WAF の種類

本節では、ライセンスおよび提供形態の 2 つの観点から考える WAF の種類について、解説します。

ライセンスの観点で WAF を考えると、「商用製品」の WAF、「オープンソースソフトウェア」の WAF の 2 種類があります。WAF を導入する場合、導入費用だけでなく、運用費用が生じます。「商用製品」の WAF、「オープンソースソフトウェア」の WAF はそれぞれ導入費用、運用費用が異なります。

また、提供形態の観点で WAF を考えると、「専用機器」として提供されている WAF、「ソフトウェア」として提供されている WAF、「サービス」として提供されている WAF の 3 種類があります。

それぞれの WAF の種類を理解したうえで、WAF を選定することが重要です。

### 2.3.1. ライセンスの観点で考える WAF の種類

#### ■ 「商用製品」の WAF

「商用製品」の WAF（以降、商用 WAF）とは、企業より販売、提供される WAF です。商用 WAF には、共通して以下の特徴があります。

- 販売または提供企業に費用を支払うことで、使用できる
- 販売または提供企業からサポートを受けて、運用できる<sup>23</sup>
- マニュアルが充実しているため、ウェブサイト運営者に WAF に関する情報が提供される

#### ■ 「オープンソースソフトウェア」の WAF

「オープンソースソフトウェア」の WAF（以降、オープンソース WAF）は、ライセンスに従えば、無償で使用できる WAF です。オープンソース WAF には、共通して以下の特徴があります。

- ライセンスに従えば、無償で使用できる
- 提供企業や組織がサポートサービスを提供していないことがあり、ウェブサイト運営者自ら WAF を運用する必要がある。ウェブサイト運営者が WAF に関する知識を十分に有していないと、運用費用が高くなる場合がある
- マニュアルが不足していることがあり、ウェブサイト運営者には WAF に関する知識が要求される

### 2.3.2. 提供形態の観点で考える WAF の種類

商用 WAF、オープンソース WAF それぞれにおいて、WAF の提供形態が異なります（図 2-5）。商用 WAF<sup>24</sup>は、ソフトウェアに限らず、専用機器やサービスとして提供している企業があります。一方で、オープンソース WAF はソフトウェアとしてインターネットで公開されています<sup>25</sup>。

WAF の提供形態によって、WAF の設置場所が変わってきます。WAF の設置場所については、「3.1 WAF の設置」を参照してください。

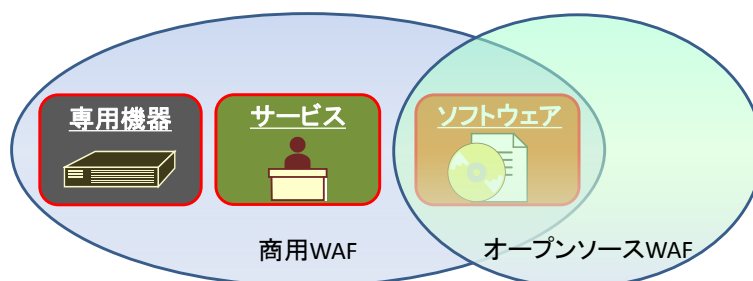


図 2-5 商用 WAF、オープンソース WAF ごとの提供形態

<sup>23</sup> 商用 WAF の運用を外部に委託することもできます。

<sup>24</sup> 「商用製品」の WAF の一部を「付録 B. 商用製品の紹介」にて紹介しています。

<sup>25</sup> 「オープンソースソフトウェア」の WAF の一部を「付録 A. オープンソースソフトウェアの紹介」にて紹介しています。

## 2.4. WAF が有効な状況

本節では、ウェブアプリケーションの脆弱性を悪用した攻撃による被害を防ぐうえで、WAF の導入が有効な状況を解説します。

ウェブサイトを運営するうえで、ウェブアプリケーションの脆弱性を悪用した攻撃を防ぐためには、「脆弱性を作り込まないこと」、「脆弱性が見つかったら修正すること」が重要です。しかし、ウェブサイト運営者の事情により、これらの根本的な脆弱性対策を実施することが困難な状況があります。また、ウェブサイト運営者が直接管理できないウェブアプリケーションへの攻撃を防ぎたい状況もあります。こういった状況においては、WAF の導入が有効な場合があります。

WAF の導入が有効な状況を「事前対策」と「事後対応」の観点から整理すると、図 2-6 のようになります。「事前対策」はウェブアプリケーションの脆弱性を悪用する攻撃によるセキュリティ事件の発生を低減する施策となります。また「事後対応」は事故が起きた場合、被害を最小限に抑え、早期復旧を実現する施策となります。

これより、図 2-6 の(a)から(c)それぞれについて解説していきます。

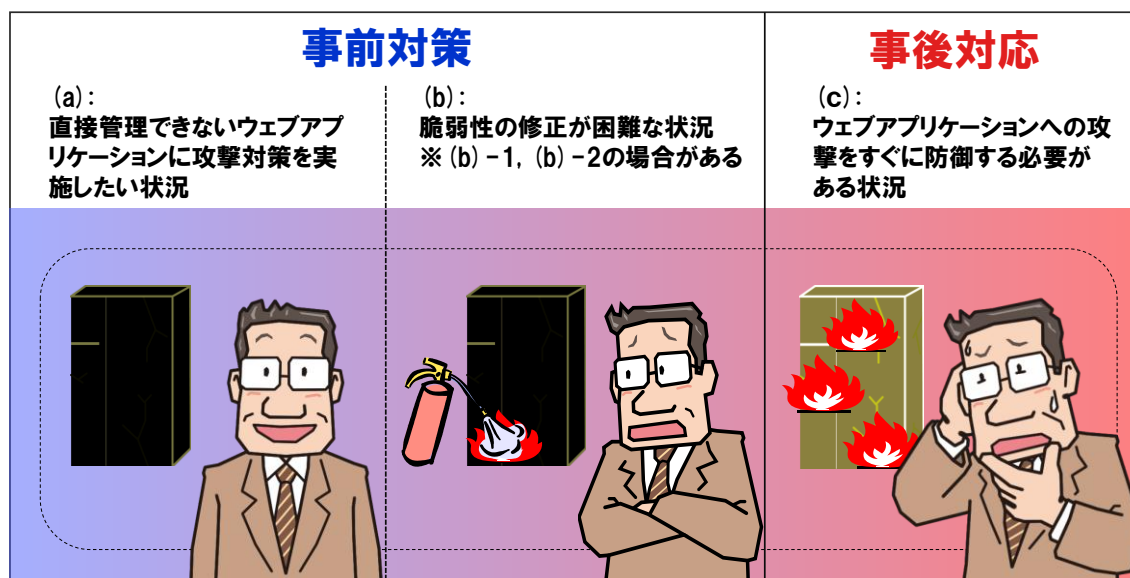


図 2-6 WAF の導入が有効な状況

### (a) 直接管理できないウェブアプリケーションに攻撃対策を実施したい状況

開発者や運営者が異なるウェブアプリケーションにおいて、脆弱性を悪用する攻撃に対して同じ対策を実施したい場合があります。例えば、開発者が異なるウェブアプリケーションをまとめて管理できる立場にあるウェブサイト運営者の場合です。こういったウェブサイト運営者には、異なる地域にグループ企業をもつ大手企業のウェブサイト運営者や、レンタルサーバ等を提供する企業のウェブサイト運営者などが該当します。



## **(b)-1 開発者にウェブアプリケーションの改修を依頼できない状況**

---

ウェブアプリケーションに脆弱性が発見された場合、開発者に直接脆弱性の修正を依頼できないことがあります。

企業や組織がウェブアプリケーションを開発する際、他社に開発を依頼することがあります。仮にこのウェブアプリケーションに脆弱性が発見された場合に、開発企業に脆弱性の修正を依頼できない事態(例：開発事業から撤退している)が生じ得ます。

開発企業にウェブアプリケーションの改修を依頼せずとも、他の企業に改修を依頼することもできます。しかし、改修費用が高くなり予算内で改修できない事態に陥る可能性があります。

## **(b)-2 改修できないウェブアプリケーションに脆弱性が発見された状況**

---

商用製品やオープンソースソフトウェアを使用してウェブサイトを構築した場合、該当ソフトウェアの改修に直接関与できず、脆弱性を修正できないことがあります。

近年、ブログや Wiki に代表されるウェブアプリケーションが商用製品やオープンソースソフトウェアとして提供されています。これらのソフトウェアを利用することで、ウェブアプリケーションを独自開発することなく、ウェブアプリケーションを利用できます。

商用製品に脆弱性が発見された場合、該当ソフトウェアの開発元が脆弱性を修正したバージョン、または修正パッチを提供しない限り、脆弱性を修正できない場合があります。該当ソフトウェアのサポート期間が終了していた場合、脆弱性が修正されない可能性もあります。

オープンソースソフトウェアの場合、利用者自身が脆弱性を確認し修正することもできます。ただし、自組織に脆弱性を修正できる技術者がいない場合、脆弱性を修正できない事態に陥る可能性があります。

## **(c) ウェブアプリケーションへの攻撃をすぐに防御する必要がある状況**

---

ウェブアプリケーションに脆弱性があり、ウェブアプリケーションの脆弱性を悪用する攻撃を受けた場合、その攻撃によりウェブサイトが被害にあってしまうことがあります。

攻撃による被害に気づいた場合、それ以上の被害が生じないように対策を講じることが重要です。そのため、被害原因の調査や原因を解消するために、必要に応じてウェブサイトを停止することがあります。しかし、インターネット中心に事業を展開している企業にとって、ウェブサイトを長期間停止することは機会損失が大きく、事業継続に多大な影響を与えかねません。この事業継続の観点から、ウェブアプリケーションの脆弱性を修正する時間を許容できない場合があります。

## 3. WAF の詳細

本章では、WAF の設置場所、機能、機能における留意点を解説します。

### 3.1. WAF の設置

WAF を設置する場合、設置場所には「ネットワーク」と「ウェブサーバ」の 2 つがあります。WAF を導入する場合、ウェブサイトの構成や可用性、設置場所ごとの WAF の特徴を考慮することが重要です。

なお、商用 WAF やオープンソース WAF の具体的な設置方法は WAF によって異なるため、本書では説明しません。WAF の具体的な設置方法は、WAF の開発元にお問い合わせください。

#### 3.1.1. 設置場所: ネットワーク

ネットワークに設置する WAF (以降、ネットワーク設置型 WAF<sup>26</sup>) は、利用者とウェブサイト間の HTTP 通信 (HTTPS 通信を含む) を、通信経路上に介在することで検査します (図 3-1)。ネットワーク設置型 WAF には、専用機器で提供されるもの、ソフトウェアをインストールしたサーバ機器を利用するものがあります。

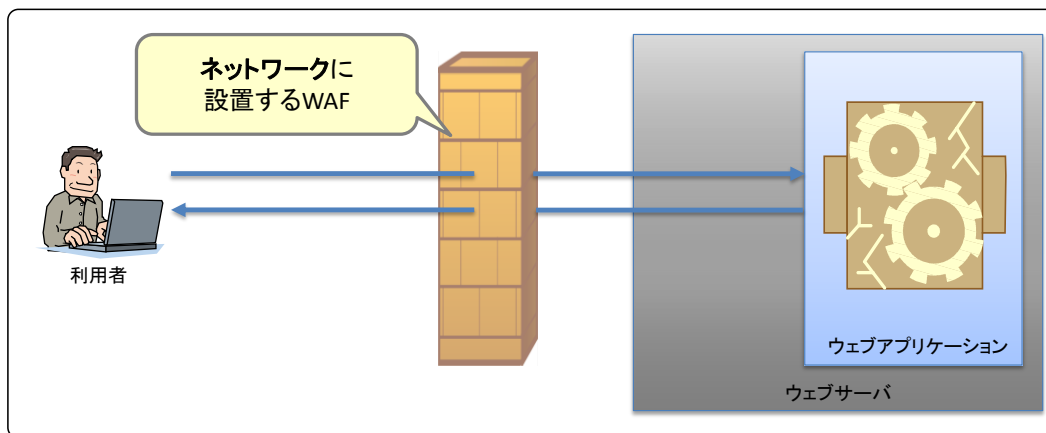


図 3-1 ネットワーク設置型 WAF のイメージ図



ネットワーク設置型 WAF には、ウェブサイトに設置する場合だけではなく、企業や組織における別の拠点などウェブサイトの外側に設置する場合があります。また、企業や組織が設置している WAF をサービスとして提供しているものもあります。

<sup>26</sup> 「ネットワーク設置型 WAF」は、本書での説明のために用意した用語であり、WAF について定着して用いられる用語ではありません。

ネットワーク設置型 WAF の主な特徴は以下の通りです。

- ウェブサーバの動作環境に依存しない
- ウェブサイトを構成するウェブサーバの台数に依存しない
- 既存のウェブサイトに導入する場合、ネットワーク構成を見直す必要がある
- WAF が HTTPS 通信に対応していれば、HTTPS 通信も検査できる
- WAF を導入することで、ウェブサイトの可用性が低下する可能性がある

また、企業や組織が提供する WAF のサービスには、上記 5 つの特徴にくわえて、以下の特徴があります。

- ウェブサイト運営者が自らのネットワークに WAF を設置する必要がない
- WAF を設置する場合に比べて、ウェブサーバやネットワークの構成への影響が小さい

### 3.1.2. 設置場所:ウェブサーバ

ウェブサーバに設置する WAF（以降、サーバインストール型 WAF<sup>27</sup>）は、利用者とウェブサイト間の HTTP 通信をウェブサーバが送受信する際に検査します（図 3-2）。サーバインストール型 WAF には、ソフトウェアで提供され、ウェブサーバの一部として動作するものが多いです。

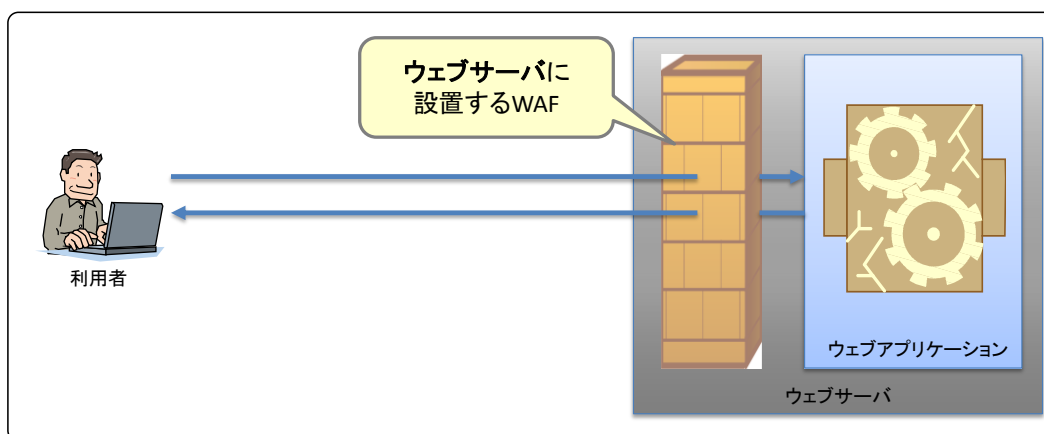


図 3-2 ウェブサーバに設置する WAF のイメージ

サーバインストール型 WAF の主な特徴は以下の通りです。

- ウェブサーバの動作環境に依存する
- ウェブサイトを構成するウェブサーバすべてに導入する必要がある
- 既存のウェブサイトに導入する場合でも、ネットワーク構成を見直す必要はない
- ウェブサーバで HTTPS 通信が処理される（復号と暗号化が行われる）ため、WAF が HTTPS 通信に対応していなくとも、HTTPS 通信を検査できる
- WAF を導入することで、ウェブサーバの性能が低下する可能性がある

<sup>27</sup> 「サーバインストール型 WAF」は、本書での説明のために用意した用語であり、WAF について定着して用いられる用語ではありません。なお、ホスト型 WAF と呼ばれることもあります。

## 3.2. WAF の機能

WAF は、様々な機能を連携させることで、ウェブアプリケーションの脆弱性を悪用する攻撃からウェブアプリケーションを保護します (図 3-3)。WAF の機能には、WAF であれば必ず実装している「基本機能」と、基本機能を補うために WAF が独自に実装している「拡張機能」があります<sup>28</sup>。

なお、WAF の機能を解説する上で、HTTP に関する用語が多く登場します。HTTP の用語については、RFC 2616<sup>29</sup>における用語を使用しています。

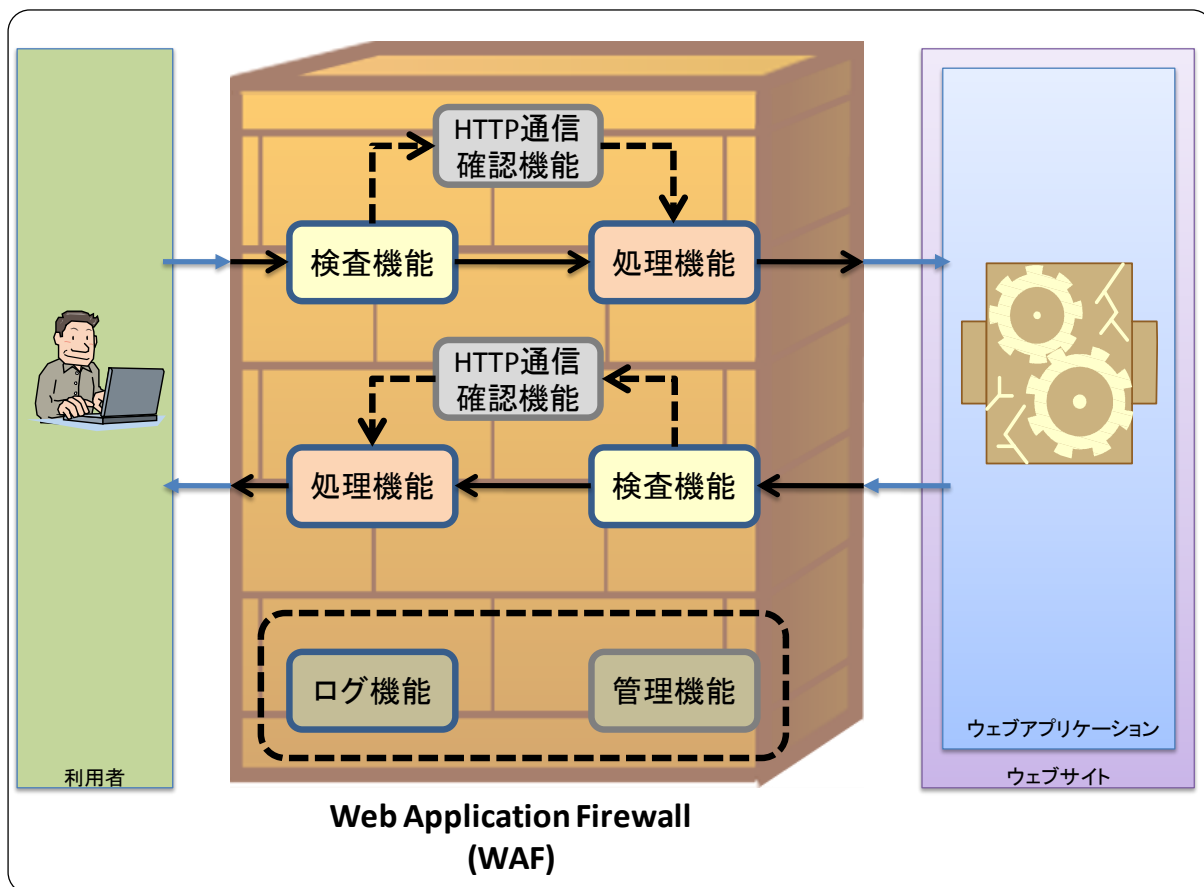


図 3-3 WAF の機能概要

<sup>28</sup> 「基本機能」、「拡張機能」は、本書での説明のために用意した用語であり、WAF について定着して用いられている用語ではありません。

<sup>29</sup> Hypertext Transfer Protocol -- HTTP/1.1  
<http://www.ietf.org/rfc/rfc2616.txt>

### 3.2.1. 基本機能

---

WAF は、利用者とウェブサイト間の HTTP 通信の内容を機械的に検査します。検査の結果、「悪いもの」と判定した HTTP 通信に対して、定義した処理を実行します。この一連の機能が WAF の「基本機能」です。WAF の「基本機能」には以下の機能が含まれます。

- 「検査機能」：HTTP 通信を検出パターンに基づいて検査する
- 「処理機能」：HTTP 通信に対する処理を実行する
- 「ログ機能」：WAF の動作を記録する

#### ■ 検査機能

検査機能とは、定義した検出パターンに基づいて、HTTP 通信内の HTTP リクエストおよび HTTP レスポンス<sup>30</sup>を検査する機能です。HTTP リクエスト、HTTP レスポンスの検査項目には、以下が含まれます<sup>31</sup>。

##### ◆ HTTP リクエストの検査項目

- リクエストライン（メソッド、URI、クエリストリング）
- ヘッダ（ジェネラルヘッダ、リクエストヘッダ、エンティティヘッダの各フィールド）
- メッセージボディ（POST データ）

##### ◆ HTTP レスポンスの検査項目

- ステータスライン（ステータスコード）
- ヘッダ（レスポンスヘッダフィールド）
- メッセージボディ

また、検査機能で使用する検出パターンは、定義方法により「ブラックリスト」と「ホワイトリスト」の 2 つがあります。

---

<sup>30</sup> HTTP レスポンスの検査は、攻撃によりウェブサーバから悪意のある人に重要な情報や不必要な情報を応答しないことを目的として使用されます。

<sup>31</sup> 検査項目は WAF によって異なります。

## ◆ 「ブラックリスト」

「ブラックリスト」は、HTTP 通信における「不正な値、またはパターン」を定義した検出パターンを指します。「ブラックリスト」を使用して HTTP 通信を検査した場合、WAF は HTTP 通信の内容が「不正な値、またはパターン」に合致したときに、その HTTP 通信を不正な通信として検出します<sup>32</sup>。

一般的に「ブラックリスト」には、ウェブアプリケーションの脆弱性を悪用した攻撃における特徴的な値またはパターンが定義されます。「ブラックリスト」に基づいて検査する場合、WAF は既知の攻撃からウェブアプリケーションを防御できます。「ブラックリスト」では、すでに特徴が分かっている攻撃しか検出できません。「ブラックリスト」で最新の攻撃を検出するには、新しい攻撃手法が判明次第、随時「ブラックリスト」を更新する必要があります。

## ◆ 「ホワイトリスト」

「ホワイトリスト」は、HTTP 通信における「正しい値、またはパターン」を定義した検出パターンを指します。「ホワイトリスト」を使用して HTTP 通信を検査した場合、WAF は HTTP 通信の内容が「正しい値、またはパターン」に合致しないときに、その HTTP 通信を不正な通信として検出します<sup>33</sup>。

一般的に「ホワイトリスト」には、ウェブアプリケーションの設計に基づいてウェブアプリケーションのパラメータ毎に正しい値またはパターンが定義されます。「ホワイトリスト」に基づいて検査する場合、開発者が想定していない値または型をウェブアプリケーションが受け取ることがなくなります。「ホワイトリスト」はウェブアプリケーションの設計に依存するため、ウェブアプリケーション毎に「ホワイトリスト」を作成する作業が必要となります。

なお、WAF は防御対象をウェブアプリケーションに限定できるため、「ホワイトリスト」を使用できます。

「ブラックリスト」、「ホワイトリスト」の長所・短所を表 3-1 にまとめました。その他の WAF 機能における留意点については、「3.3 WAF 機能における留意点」を参照してください。

表 3-1 「ブラックリスト」・「ホワイトリスト」の長所・短所

長所・短所	「ブラックリスト」	「ホワイトリスト」
長所	どのウェブアプリケーションでも同じ「ブラックリスト」を使用できる	正しい値、またはパターン以外を不正な通信として検出できるため、未知の攻撃にも対応できる
短所	新しい攻撃手法が判明したら、随時「ブラックリスト」を更新する必要がある	ウェブアプリケーション毎に「ホワイトリスト」を定義する必要がある

<sup>32</sup> 文献等において、ネガティブセキュリティモデルと呼ばれることがあります。

<sup>33</sup> 文献等において、ポジティブセキュリティモデルと呼ばれることがあります。

## ■ 処理機能

処理機能とは、「検査機能」または「HTTP 通信確認機能」（後述）で検出された不正な HTTP 通信に対して、定義した処理を実行する機能です。この機能で定義できる処理方法は 3 つあります。

### ◆ 通過処理

通過処理とは、検出された不正な HTTP 通信をそのまま利用者またはウェブサイトへ送信する処理方法です。この処理方法は、一般的に WAF の導入時に HTTP 通信を検証したり、検出した不正な HTTP 通信を記録する場合に設定されます。

### ◆ エラー処理

エラー処理とは、検出された不正な HTTP 通信の代わりに、WAF がエラー応答を生成して、利用者またはウェブサイトへ送信する処理方法です（図 3-4）。一般的にこのエラー処理で送信するエラー応答の内容を、WAF で任意のものに編集できます。

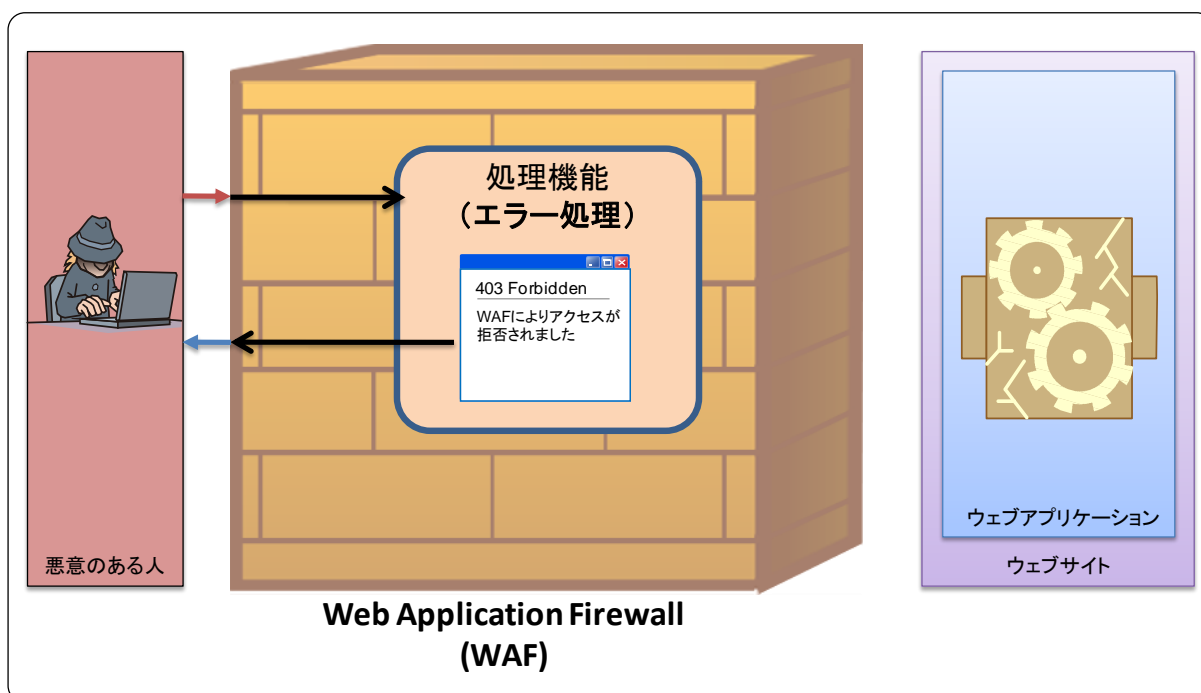


図 3-4 WAF の処理機能（エラー処理）

## ◆ 遮断処理

遮断処理とは、検出された不正な HTTP 通信を意図的に破棄する処理方法です (図 3-5)。WAF が HTTP 通信を破棄する際には、「利用者またはウェブサイトに HTTP 通信切断応答を送信する」、「HTTP 通信に何も応答しない」といった方法が使用されます。

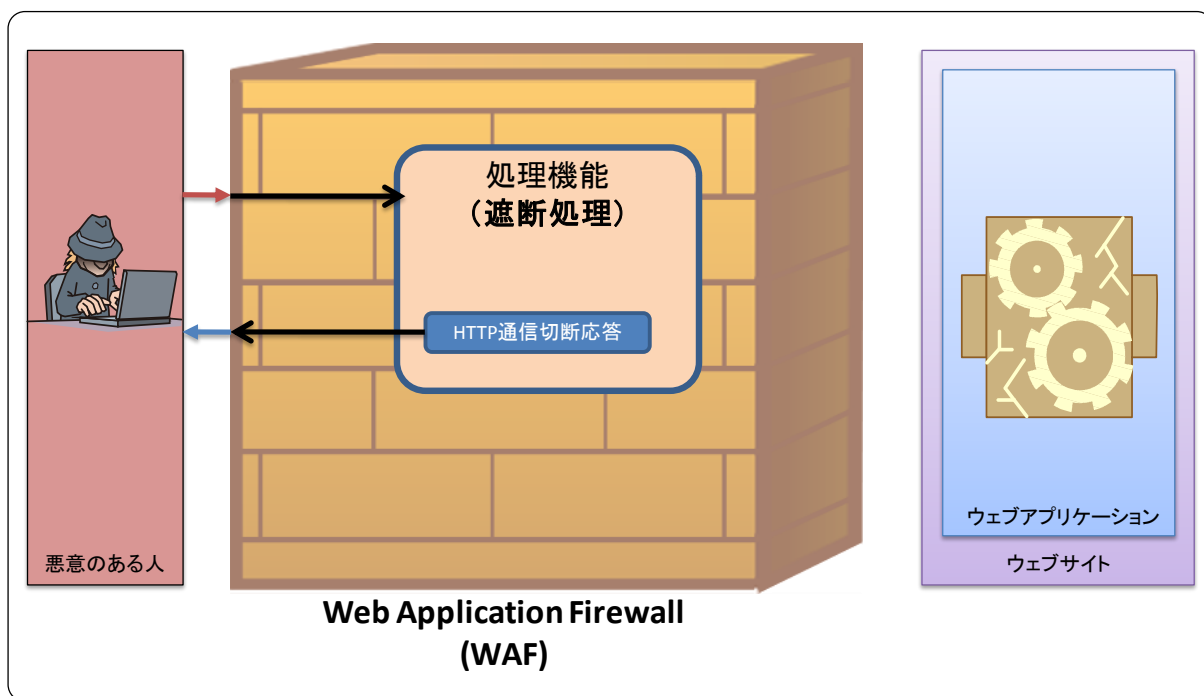


図 3-5 WAF の処理機能 (遮断処理)



また WAF によっては、以下の処理方法も選択できるものがあります。

#### ◆ **書き換え処理**

書き換え処理とは、検出された不正な HTTP 通信の一部を WAF が書き換えて、利用者またはウェブサイトに送信する処理方法です (図 3-6)。この処理方法は、特徴的な文字列を含む HTTP 通信を検出した場合でも、HTTP 通信を継続したい場合に設定されます。該当する HTTP 通信として、HTTP リクエストの場合はクロスサイト・スクリプティング攻撃や SQL インジェクション攻撃などの攻撃通信が挙げられ、HTTP レスポンスの場合は重要な情報や不必要な情報が挙げられます。

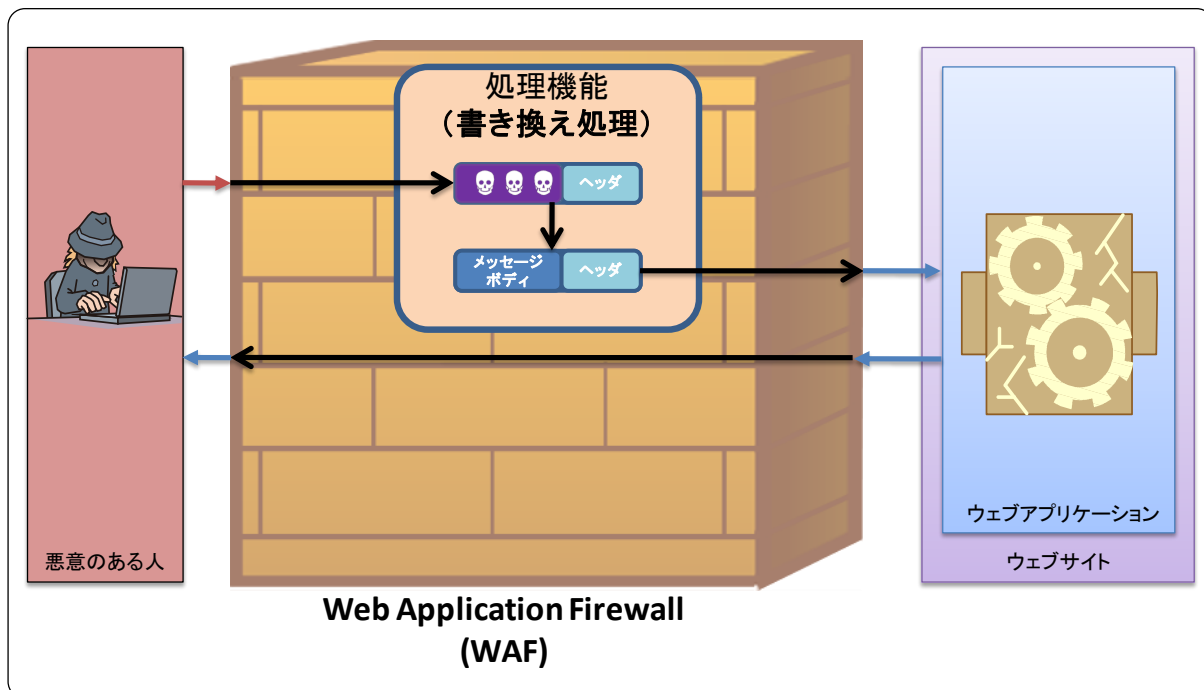


図 3-6 WAF の処理機能 (書き換え処理)

## ■ ログ機能

ログ機能とは、「検査機能」により検出された不正な HTTP 通信や WAF の動作（以降、ログ）を記録する機能です。一般的に WAF のログは、ファイルまたはデータベースに記録されます。ログには、記録される内容から 2 種類があります。

### ◆ 監査ログ

監査ログには、検査機能により検出された不正な HTTP 通信とその HTTP 通信に対する処理方法が記録されます。監査ログに記録される主な内容としては、検出日時、選択した処理方法、接続元 IP アドレス、接続先 URL、HTTP 通信において不正だと判断した箇所（HTTP リクエストヘッダ、HTTP メッセージボディ等）、検出パターンなどがあります。

監査ログはウェブサイト運営者が検出した不正な HTTP 通信を確認する他、WAF の管理機能のレポート（後述）にも使用されます。

### ◆ 動作ログ

動作ログには、WAF 自身の動作情報やエラー情報が記録されます。動作ログに記録される主な内容としては、情報発生日時、WAF の起動・停止・設定の変更等の動作情報、エラー情報などがあります。

動作ログは、ウェブサイト運営者が WAF の動作が正常であるか確認することに使用されます。

### 3.2.2. 拡張機能

---

「基本機能」とは別に、WAF が独自に実装している「拡張機能」<sup>34</sup>があります。WAF の「拡張機能」として、例えば以下の機能が挙げられます。

- 「HTTP 通信確認機能」<sup>35</sup>：HTTP 通信におけるセッション<sup>36</sup>の正当性を確認する
- 「管理機能」：WAF を管理する上で利便性を高める

#### ■ HTTP 通信確認機能

HTTP 通信確認機能とは、HTTP 通信のセッションにおけるパラメータや HTTP リクエスト等の正当性を確認する機能です。「検査機能」では、主に個々の HTTP 通信の内容を検査するため、セッションに関する脆弱性を悪用した攻撃等を防御することは困難です。この機能では、セッションに関する脆弱性を悪用した攻撃等を防御するための確認方法を提供します。この機能が提供する確認方法は、3 つあります。

#### ◆ セッションにおけるパラメータ確認<sup>37</sup>

セッションにおけるパラメータの確認とは、セッションにおいてパラメータが改ざんされていないか確認する方法を指します。

ブラウザがウェブサイトへ送信する HTTP リクエストは、利用者が自由に送信できるものです。ウェブアプリケーションがセッション固有の情報を特定のパラメータ<sup>38</sup>に格納して利用者と送受信する場合、悪意のある人がそのパラメータを改ざんすることで、セッション固有の情報が任意のものに書き換えられてしまいます。

それに対し、セッションにおけるパラメータ確認では、セッションにおける HTTP レスポンスの特定のパラメータを WAF が一時的に保存し、次の HTTP リクエストでそのパラメータ値と保存した値が一致するか確認することで、パラメータ改ざんを防御します。

特定のパラメータの例として、HTTP ヘッダの Cookie、hidden パラメータ等が挙げられます。また、WAF の中には、パラメータを利用者に送信する前に暗号化するものもあります。

---

<sup>34</sup> すべての WAF が実装しているわけではありません。

<sup>35</sup> 「HTTP 通信確認機能」は、本書での説明のために用意した用語であり、WAF について定着して用いられている用語ではありません。

<sup>36</sup> セッションとは、利用者とウェブサイト間で交わされる HTTP 通信の一連の流れを指します。

<sup>37</sup> 「セッションにおけるパラメータ確認」は、本書での説明のために用意した用語であり、WAF について定着して用いられている用語ではありません。

<sup>38</sup> HTTP リクエストの Cookie ヘッダ、HTTP リクエストの POST データ等があります。

## ◆ HTTP リクエストの正当性確認<sup>39</sup>

HTTP リクエストの正当性確認とは、セッションにおいて利用者から送信される HTTP リクエストが正当な HTTP リクエストであるか確認する方法を指します。

ウェブアプリケーションに CSRF (Cross-Site Request Forgeries/クロスサイト・リクエスト・フォージェリ) の脆弱性<sup>40</sup>がある場合、悪意のある人が利用者を用意した罠に誘導することで、予期しない処理が実行されてしまいます。

それに対し、HTTP リクエストの正当性確認は、セッションにおける HTTP レスポンスに WAF が第三者に予測されない秘密情報を付与して、次の HTTP リクエストでその秘密情報が送付されているか確認することで、CSRF 攻撃を防御します。

## ◆ ウェブサイトの画面遷移確認<sup>41</sup>

ウェブサイトの画面遷移確認とは、セッションにおいて利用者から送信される HTTP リクエストヘッダの画面遷移が適切なものであるか確認する方法を指します。

ウェブアプリケーションの脆弱性を悪用する攻撃のうち、利用者を悪意のある人が作成したウェブページに誘導するような攻撃<sup>42</sup>では、利用者は悪意のある人が用意した罠に誘導されます。このような攻撃では、正常な画面遷移とは異なり、悪意のある人が用意した罠から脆弱なウェブアプリケーションに画面遷移します。

それに対し、ウェブサイトの画面遷移確認は、ウェブアプリケーションにおける画面遷移を定義して、HTTP リクエストの Referer ヘッダがその定義に一致するか確認することで、誘導型の攻撃を防御します。確認した結果、定義した画面遷移に一致しない場合、不正な HTTP 通信として検出します。

この確認方法を使用すると、以下の問題点が生じる可能性があります。

- ウェブサイトにおける可用性が低下する
- 検索エンジン対策において不利となる

---

<sup>39</sup> 「HTTP リクエストの正当性確認」は、本書での説明のために用意した用語であり、WAF について定着して用いられている用語ではありません。

<sup>40</sup> CSRF の脆弱性の詳細については、以下をご参照ください。

「安全なウェブサイトの作り方」：<http://www.ipa.go.jp/security/vuln/websecurity.html>

「知っていますか？脆弱性 (ぜいじゃくせい)」：[http://www.ipa.go.jp/security/vuln/vuln\\_contents/index.html](http://www.ipa.go.jp/security/vuln/vuln_contents/index.html)

<sup>41</sup> 「ウェブサイトの画面遷移確認」は、本書での説明のために用意した用語であり、WAF について定着して用いられる用語ではありません。

<sup>42</sup> CSRF 攻撃が該当します。

## ■ 管理機能

管理機能とは、WAF を運用する上で WAF の利便性を高める機能です。この機能があることで、WAF を導入したウェブサイト運営者が「ログ機能」で記録された監査ログを解析して、不正な HTTP 通信の検出および処理件数を確認したり、「検査機能」の検出パターンを随時更新する手間を省いたりすることが可能となります。「管理機能」には、以下のようなものが挙げられます。

### ◆ レポートの生成

レポートの生成とは、「ログ機能」で記録されたログを解析して統計情報をレポートとして出力する機能です。出力されるレポートは、一般的に一定期間毎（1 週間毎、1 ヶ月毎、1 年間毎等）に生成されます。出力されるレポートでは、不正な HTTP 通信と判断された接続元アドレス毎の検知回数、接続先 URL とそのアクセス回数、検知理由等がまとめられます。

### ◆ 管理者への通知

管理者への通知とは、「検知機能」や「HTTP 通信確認機能」により不正な HTTP 通信が検出された場合、あらかじめ指定した管理者にメールなどの手段で通知する機能です。この機能では一般的に不正な HTTP 通信を検出したら、管理者に通知されます。

### ◆ 「ホワイトリスト」の自動生成

「ホワイトリスト」の自動生成とは、WAF を経由する HTTP 通信から自動的に「ホワイトリスト」を生成する機能です。「ホワイトリスト」の短所には、ウェブアプリケーション毎に「ホワイトリスト」を生成しなければいけないため、「ホワイトリスト」の生成に手間がかかるという点があります。この機能は、「ホワイトリスト」の短所を補うためのものです。

### ◆ 「ブラックリスト」の自動更新

「ブラックリスト」の自動更新とは、「ブラックリスト」を自動的に更新する機能です。「ブラックリスト」の短所には、攻撃手法に依存しているため、新しい攻撃手法が発見されたら、その攻撃の検出パターンを定義して「ブラックリスト」を更新しなければいけないという点があります。この機能は、「ブラックリスト」の短所を補うためのものです。

### 3.3. WAF 機能における留意点

WAF を導入したとしても、すべてのウェブアプリケーションの脆弱性を悪用した攻撃を防御できるわけではありません。また、WAF を導入することで、利用者の正常な HTTP 通信を WAF が防御してしまい、ウェブサイトの可用性を低下させる可能性もあります。WAF を導入する場合、以下の留意点を理解した上で使用することが重要です。

#### 3.3.1. 「検査機能」における偽陽性 (false positive) ・偽陰性 (false negative)

WAF が利用者とウェブサイト間で交わされる HTTP 通信を検査する場合、HTTP 通信を機械的に検査するが故に、偽陽性 (false positive)、偽陰性 (false negative) が生じる可能性があります。

偽陽性、偽陰性とは、「不正な HTTP 通信」(陽性) または「正常な HTTP 通信」(陰性) と判定されるべき HTTP 通信がそれぞれ逆の判定をされてしまう判定エラーを指します。

偽陽性とは、本来「正常な HTTP 通信」であるにもかかわらず、「不正な HTTP 通信」と判定される判定エラーです。偽陽性が生じると、利用者の正当な HTTP 通信が WAF で防御される場合があります。

偽陰性とは、本来「不正な HTTP 通信」であるにもかかわらず、「正常な HTTP 通信」と判定される判定エラーです。偽陰性が生じると、ウェブアプリケーションの脆弱性を悪用する攻撃を WAF が防御しない場合があります。

WAF の「検査機能」で「ブラックリスト」、「ホワイトリスト」どちらの検出パターンを使用するかによって、偽陽性、偽陰性が生じる要因が異なります。WAF を導入する場合、「ブラックリスト」、「ホワイトリスト」それぞれの要因を理解して使用するとよいでしょう。

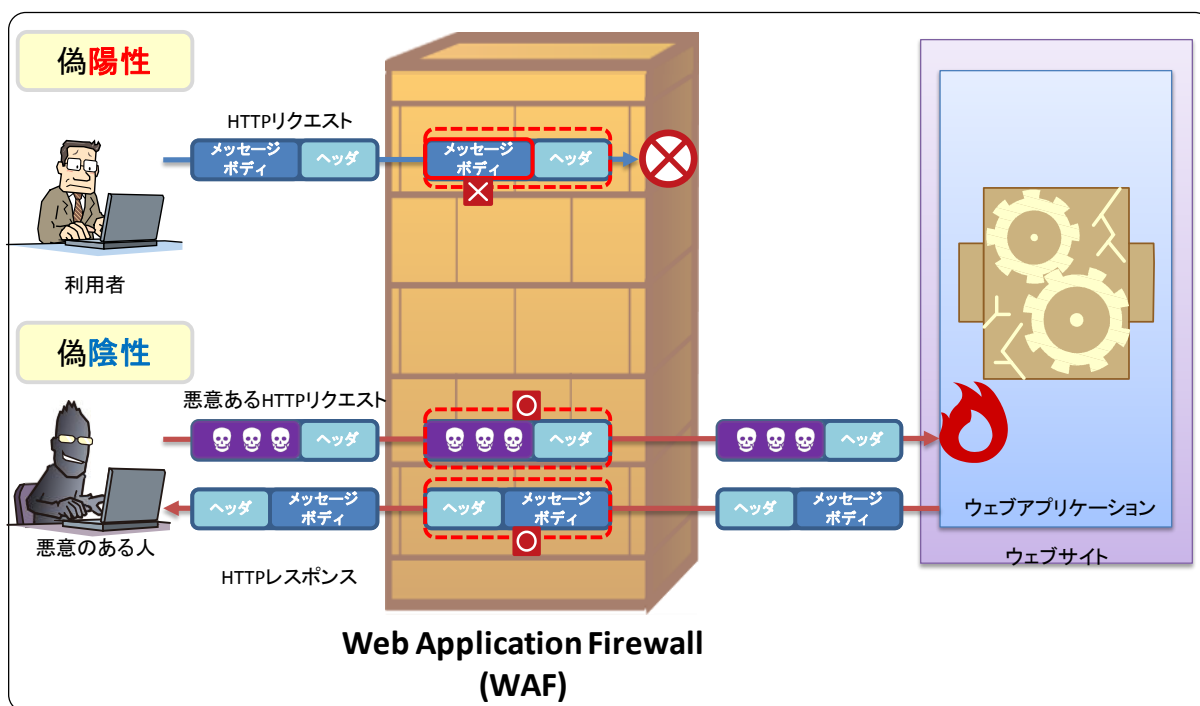


図 3-7 「検査機能」における偽陽性・偽陰性

## ■ 「ブラックリスト」における留意点

どのようなウェブアプリケーションであっても、同じ「ブラックリスト」を導入できます。一般的に既知の攻撃を防御できるように、WAF の開発元が「ブラックリスト」を提供しています。しかし、この「ブラックリスト」のすべての検出パターンを有効にすると、偽陽性が発生する可能性があります。もし正常な HTTP 通信が遮断された場合、その「ブラックリスト」の検出パターンを無効にせざるを得ません。

## ■ 「ホワイトリスト」における留意点

ウェブアプリケーションにおいて、自由な入力 that 許されている入力フォームがある場合、正常とみなす入力値を定義することは困難です。この場合、「ホワイトリスト」において、該当する入力フォームにおける検出パターンを作成することができず、結果として攻撃を検知できない場合があります。

また、ウェブアプリケーションの仕様を変更した場合、「ホワイトリスト」も併せて変更する必要があります。ウェブアプリケーションの仕様変更を検出パターンに適切に反映できないと、偽陽性、偽陰性が生じる可能性があります。

### 3.3.2. WAF で防御できない不正な HTTP 通信

---

ウェブアプリケーションに存在する脆弱性の種類によっては、WAF を導入しても、その脆弱性を悪用する攻撃を防御できない場合があります。

例えば、ウェブアプリケーションにおける認可制御に問題があり、特定の利用者だけに許可する機能がそれ以外の利用者にも使用できてしまうという脆弱性がある場合、HTTP 通信自体は WAF にとって正常な通信と判定されてしまいます。

## 4. WAF 導入におけるポイント

本章では、WAF 導入におけるポイントを理解していただくために、WAF の導入における流れを「導入判断」、「導入」、「運用」という 3 つの工程に分けて、それぞれの工程において検討すべき内容や注意すべき事項について説明します。

WAF 導入においては、まず「導入判断」において、攻撃による影響を低減する運用面での対策として WAF を導入するか否かを判断します。この工程における要点は、セキュア・プログラミング等の根本的な対策を検討した上で、自組織のウェブサイトにおける WAF の費用対効果を検討することです。詳しくは、「4.1 導入判断」を参照してください。

WAF 導入を決定したら、続いて「導入」において、WAF の導入および運用の計画を策定し、その計画に則って WAF を導入します。この工程における要点は、WAF の導入および運用に関わる関係者との作業調整、WAF を導入するための検証作業です。詳しくは「4.2 導入」を参照してください。

WAF を導入したら、それで終わりではありません。「運用」において、WAF を運用計画に則って運用します。この工程における要点は、必要に応じて運用手順を見直すことです。詳しくは「4.3 運用」を参照してください。



図 4-1 WAF の導入における 3 つの工程



## 4.1. 導入判断

---

「導入判断」においては、攻撃による影響を低減する運用面での対策として WAF を導入するか否かを判断します。WAF を導入するか否かを判断を行った時点で、この工程は完了です。

本節では、「導入判断」におけるポイントについて、「導入検討」と「WAF 選定」、「導入判断」という 3 つの項目に分けて説明します。なお、本節では「導入検討」、「WAF 選定」、「導入判断」の順に対応することを想定しています。

### 4.1.1. 導入検討

---

WAF 導入においてまず運営しているウェブサイトにおいて、WAF が脆弱性対策として有効であるか否かについて検討します。

ウェブアプリケーションに脆弱性が存在すると、脆弱性を悪用した攻撃の被害を受けてしまいます。そのため、脆弱性を修正することが最も望ましい対応です。しかしながら、「2.4 WAF が有効な状況」で説明したように、WAF の導入が脆弱性を悪用した攻撃に対して有効な場合があります。

このような場合、次の点に注意して WAF の導入を検討します。

#### ■ 防御したい攻撃への対応を確認

脆弱性を狙った攻撃の種類によっては、WAF で防御できない可能性があります（「3.3 WAF 機能における留意点」を参照）。WAF の導入検討においては、事前に防御したい脆弱性を悪用する攻撃を、WAF で防御できるか調査します。

ウェブサイト運営者だけでは、防御したい脆弱性を悪用する攻撃を WAF で防御できるか判断が難しい場合、事前に WAF ベンダに確認するとよいでしょう。



事前に WAF ベンダに確認するなど、防御したい攻撃に WAF が対応しているか確認しましょう

### 4.1.2. WAF 選定

---

防御したい攻撃に WAF が対応していることが確認できたら、運営しているウェブサイトに適した WAF を選定します。WAF を選定する観点には、例えば「ウェブサイトの構成への影響」や「ウェブサイトの性能への影響」、「ウェブサイトの運用への考慮」などがあります。

## ■ ウェブサイトの構成への影響

WAF の設置場所に関してはこちら 3.1 章

WAF を導入することで、ウェブサイトの構成（ネットワーク構成やウェブサーバにインストールしているソフトウェア等）が変わります。WAF の選定にあたっては、ウェブサイトの構成への影響を考慮し、WAF を選定しましょう。

例えば、ウェブサイトのネットワーク構成を変更できない場合、サーバインストール型 WAF や企業や組織がサービスとして提供している WAF を選定できます。一方で、ウェブサーバの構成を変更できない場合、ネットワーク設置型 WAF を選定できます。



事前にウェブサイトの構成を確認し、WAF を選定しましょう。

## ■ ウェブサイトの性能への影響

WAF の設置場所に関してはこちら 3.1 章

どの WAF を選定した場合であっても、少なからずウェブサイトの性能に影響を与えます。WAF の選定にあたっては、WAF がウェブサイトの性能に与える影響を考慮し、WAF を導入した後もウェブサイト性能要件を満たす WAF を選定しましょう。

例えば、ネットワーク設置型 WAF を導入した場合、ウェブサイトの可用性に影響を与えます。一方で、サーバインストール型 WAF を導入した場合、ウェブサイトの性能に影響を与えます。



ウェブサイトの性能への影響を考慮し、WAF を選定しましょう。

## ■ ウェブサイトの運用への考慮

WAF の機能に関してはこちら 3.2 章

ウェブサイトの運用方針に適した WAF を選定するために、WAF の機能を比較して検討します。比較する WAF の主な機能は以下の通りです。

- WAF の導入や設定変更、運用の容易さ（ユーザインターフェイスを含む WAF のログ機能と管理機能）
- WAF 自身の故障によるサービスへの影響（WAF の耐障害性）



WAF の運用まで意識して、ウェブサイトの運用方針に適した WAF を選定しましょう。

### 4.1.3. 導入判断

---

防御したい脆弱性を悪用する攻撃に WAF が対応していることを確認し、ウェブサイトの運用方針に適した WAF を選定できたら、ウェブサイトに WAF を導入するか否かを決定します。

#### (1) WAF 導入・運用の概算費用の見積もり

選定結果をもとに、WAF の導入・運用費用を見積もりましょう。

WAF の導入費用には、WAF 製品の価格、WAF を設置するための費用（WAF の設定、WAF の動作検証等にかかる人件費など）が発生します。WAF の運用費用には、WAF 製品の保守費用、WAF を運用するための費用（ログの確認、WAF 自身の更新、検出パターンの更新等にかかる人件費など）が発生します。

WAF の導入においては、導入費用だけではなく運用費用がかかります。長期的に WAF を運用することを意識して、運用費用を見積もりましょう。また、運用費用については、導入する WAF によって異なります。商用 WAF の導入を検討する場合、WAF ベンダに確認するとよいでしょう。

#### (2) 費用対効果から WAF の導入判断

一般的には WAF の費用（「導入費用」と「運用費用」の合計）と WAF の効果を根本的な脆弱性対策などと比較し、WAF が根本的な脆弱性対策などよりも費用対効果が大きい場合、WAF を導入します。

「2.4 WAF が有効な状況」の(b)-1 の状況を考えてみましょう。「ウェブアプリケーションの改修」、「WAF の導入」等で同等の効果が得られる場合、それぞれの施策にかかる費用を比較します（図 4-2）。比較した結果、もっとも費用がかからない施策が「WAF の導入」だった場合、「WAF の導入」が実施する候補となります。あとは、実施する施策の短所<sup>43</sup>を加味して、WAF の導入を決定します。

一方で、「2.4 WAF が有効な状況」の(c)の状況を考えてみましょう。この状況では「WAF の導入」における費用は他の施策より高いかもしれませんが。しかしウェブアプリケーションの脆弱性を悪用する攻撃を防ぐことが急務となります。攻撃により受けている被害を考慮し、WAF の導入を決定する場合があります。

---

<sup>43</sup> この例では、脆弱性があったとしても、その脆弱性が修正されるわけではありません。

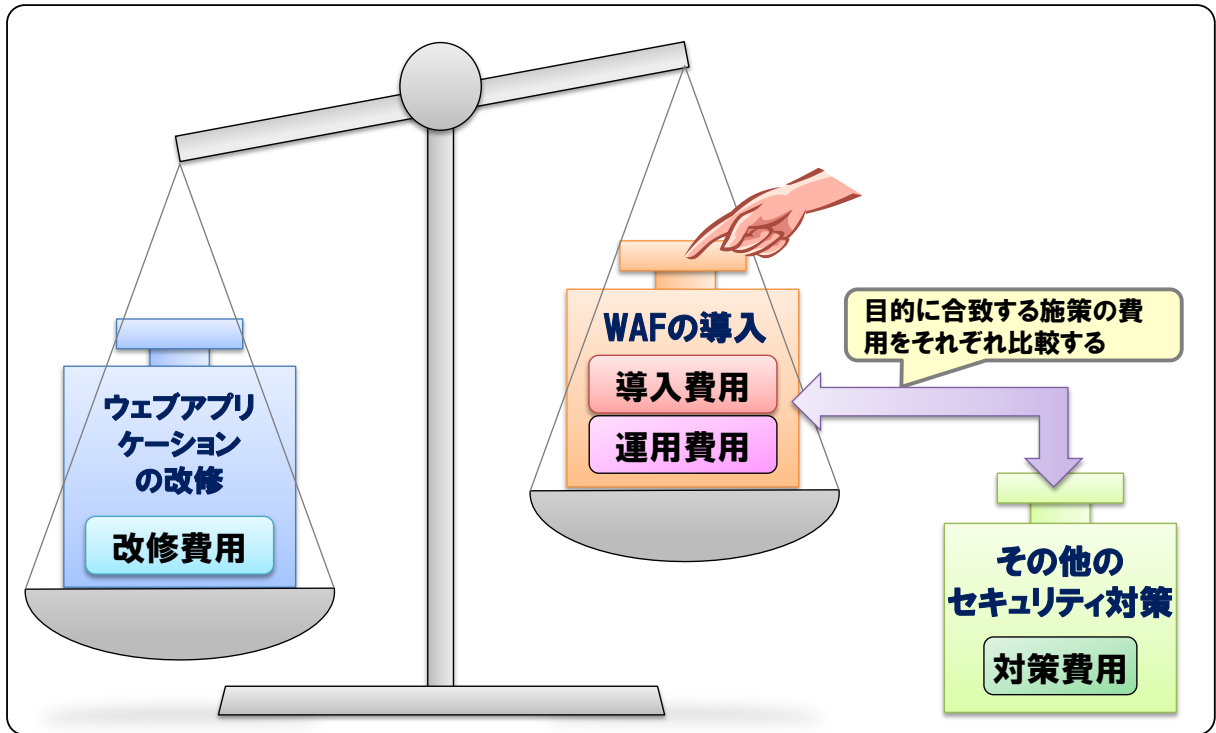


図 4-2 「WAF の導入」や他の施策との費用比較

### (3) 予算確保

費用対効果を試算した結果、WAF の導入が望ましいと判断した場合、実際に WAF を導入するための予算確保を行きましょう。この際には、導入費用だけではなく、必ず運用費用についても確保しておきましょう。



導入費用だけではなく運用費用も試算して、予算を確保しましょう。

## 4.2. 導入

---

「導入」の工程においては、WAF の導入や運用を計画し、その計画に則って WAF を導入します。WAF を導入し、本番運用を開始した時点でこの工程は完了です。

本節では、「導入」におけるポイントについて、「関係者間調整」、「導入計画」、「運用計画」、「検証」の4つの項目に分けて説明します。この工程においては、「検証」を最後に実施し、それ以外の「関係者間調整」、「導入計画」、「運用計画」は場合によっては同時に実施することを想定しています。

### 4.2.1. 関係者間調整

---

WAF の導入にあたって、影響を受ける可能性がある関係者に WAF 導入で生じる作業等を説明しましょう。本節ではまずウェブサイト運営者が WAF を設置する場合の例を2つ挙げ、どのような関係者がいるのかを説明します。その後、関係者と調整すべき項目をまとめています。



関係者にきちんと説明すると、WAF の導入や運用時にトラブルが発生した場合、きちんと連携ができスムーズなトラブル対応ができます。

### ウェブサイト運営者が WAF を設置する場合の例

---

#### ■ ウェブサイト運営者がネットワークに商用 WAF を設置する場合

ウェブサイト運営者がネットワークに商用 WAF を設置する場合、事前に調整が必要な関係者としてウェブサイトの「ネットワーク管理者」、「ウェブアプリケーション開発者<sup>44</sup>」、「WAF ベンダ」を想定できます。

---

<sup>44</sup> この例では、ウェブアプリケーション開発者が開発、保守を担当していますが、実際には開発、保守の担当者が異なる場合があります。

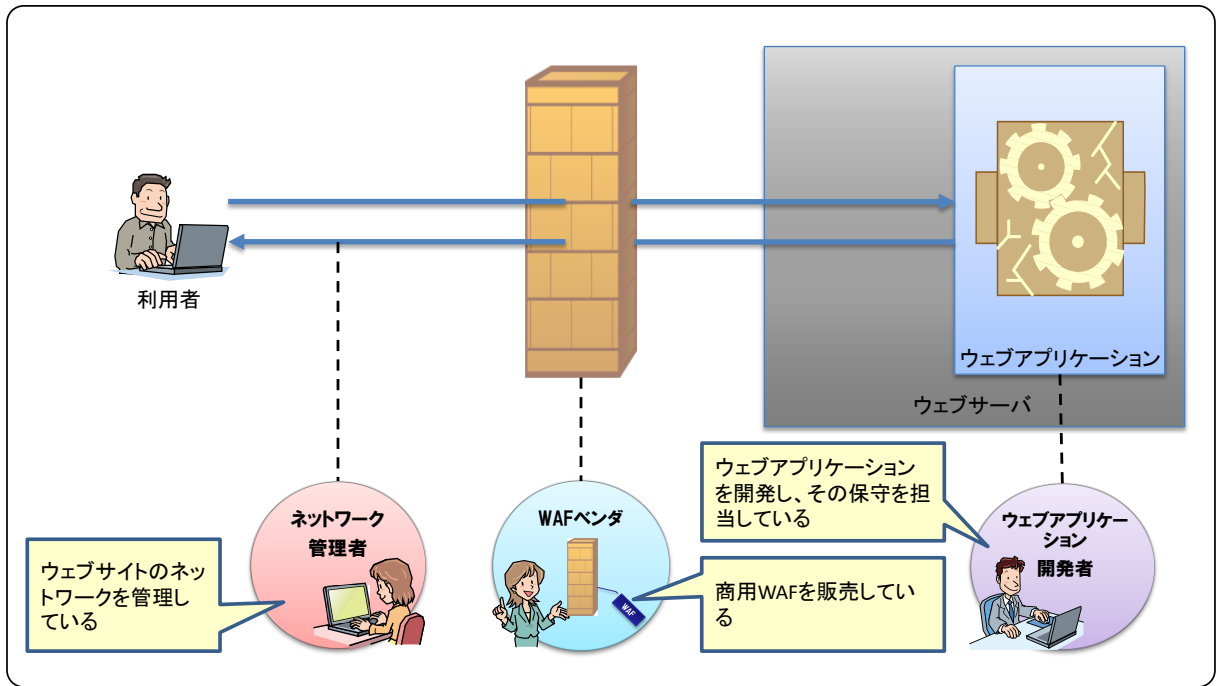


図 4-3 ネットワークに商用 WAF を設置する場合の関係者

### ■ ウェブサイト運営者がウェブサーバにオープンソース WAF を設置する場合

ウェブサイト運営者がウェブサーバにオープンソース WAF を設置する場合、事前に調整が必要な関係者として、ウェブサーバの「サーバ管理者」、「ウェブアプリケーション開発者」を想定できます。

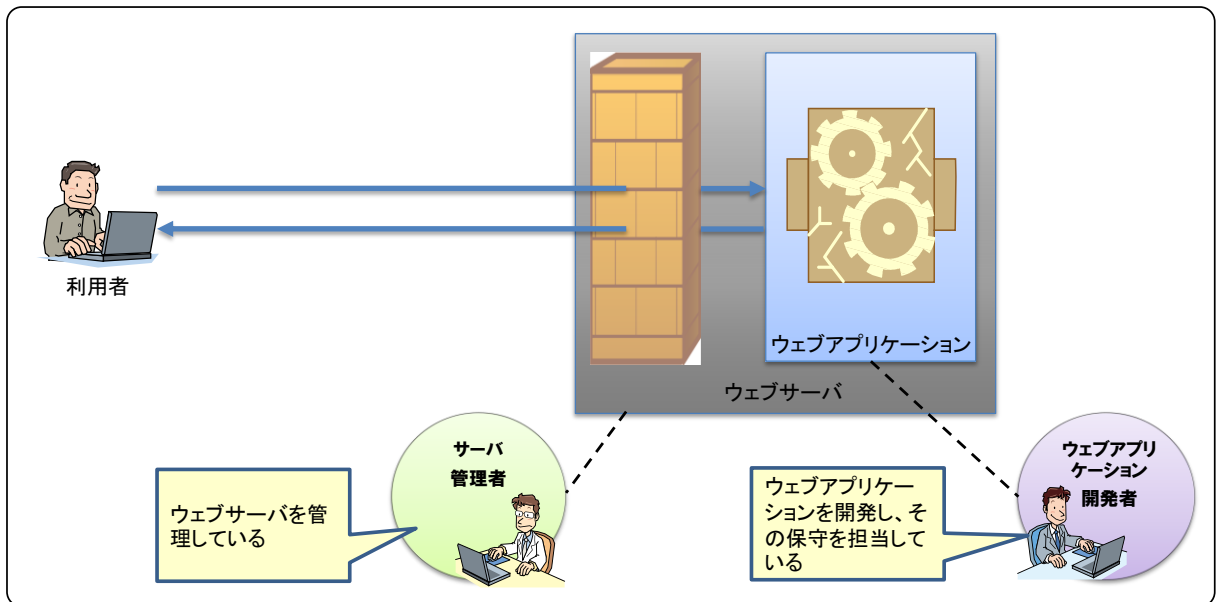



図 4-4 ウェブサーバにオープンソース WAF を設置する場合の関係者

## 関係者と調整すべき項目

### ■ ネットワーク管理者

**ネットワーク管理者**




- ・ WAF設置に伴うネットワーク構成への影響
- ・ WAF設置に伴うネットワークのスループットへの影響  
(単位時間に処理できるパケット数等)
- ・ WAF自身の障害発生時のネットワークへの影響

ネットワーク設置型 WAF を導入する場合、少なからずウェブサイトのネットワーク構成に影響を与えます。そのため、WAF 導入における影響を事前に説明しておくといでしょう。

また、ネットワーク設置型 WAF、サーバインストール型 WAF を問わず、WAF のアップデートや、ブラックリストの更新方法によっては外部のサーバとの通信が発生します。そのため、ネットワークの構成によっては、FW の設定変更など、ネットワーク管理者に作業を依頼する必要があります。

### ■ サーバ管理者

**サーバ管理者**




- ・ WAFのインストールに伴うサーバ資源への影響  
(CPU使用率、MEM使用量、ハードディスク使用量等)
- ・ WAFインストール作業に伴う、サーバへの影響
- ・ WAF自身の障害発生時のサーバへの影響

サーバインストール型 WAF を導入する場合、WAF をインストールするウェブサーバのサーバ資源へ少なからず影響を与えます。そのため、WAF 導入における影響を事前に説明しておくといでしょう。

また、WAF の導入に伴って、WAF 以外に必要なソフトウェアのインストールやウェブサーバソフトウェアの設定変更など、サーバの構成に影響を与える可能性があります。そのため、WAF がウェブサービスの障害の原因となりえます。障害が発生した場合の体制や作業範囲の切り分けなどについて、事前調整をしておくといでしょう。

### ■ ウェブアプリケーション開発者

**ウェブアプリケーション開発者**



- ・ WAFを導入することによるウェブアプリケーションのサポート契約等への影響の有無
- ・ ホホワイトリスト作成時の協力体制
- ・ 問題発生時の調査協力体制  
(偽陽性の発生等)

WAF の導入がウェブアプリケーションのサポート契約に影響を与え、ウェブアプリケーション開発者によるサポートを受けられなくなる可能性があります。事前に WAF の導入によるウェブアプリケーションのサポート契約等への影響を確認しましょう。

WAF のホワイトリストは、ウェブアプリケーションの設計に基づいて定義されるため、ホワイトリストの作成については、ウェブアプリケーション開発者の協力が重要です。また、ブラック

リストにおいても適用してはいけない検出パターンがないかなどを事前に確認しておくことで WAF の導入がスムーズに進む場合があります。

また、WAF 導入後にウェブサイトの閲覧における問題が発生した場合、原因（WAF かウェブアプリケーションか）の切り分けが必要な場合があります。このような場合、ウェブアプリケーション開発者の協力を得られると、切り分けがスムーズに進む可能性があります。

## ■ WAF ベンダ



- ・ 問題発生時の窓口、またはエスカレーション先
- ・ WAF ベンダのサポート範囲の明確化

WAF による障害など、問題が発生したときに、早い段階で WAF ベンダのサポート窓口にお問い合わせできるように、サポート窓口を確認しておきましょう。また、WAF ベンダが問題に対してどこまでサポートしてくれるのかを事前に確認しておくことで、問題への対応がスムーズに進む場合があります。



## 4.2.2. 導入計画

WAF の導入を円滑に行うためには、事前に計画を立てましょう。導入計画において、検討する項目を図 4-5 にまとめました。これらの項目を検討し、導入計画をきちんと立てておくことで、スムーズに WAF の運用を開始できます。



導入後に発生する WAF の設定変更やトラブル発生時に備えて、導入計画で検討した内容を導入手順書としてまとめておくことを推奨します。

### WAF導入環境の事前確認

- WAFを導入する環境を調査し、ネットワーク構成の変更や他のソフトウェアのインストールが必要ないか確認します。

### WAFの初期設定

- WAFを導入するにあたり、WAFの初期設定を決めます。

### WAFの動作検証

- WAFの導入後から運用開始までに検証しておくべき検証内容と検証期間を決めます。

### WAF導入時の導入体制

- WAFを導入する際の導入体制を決めます。

図 4-5 WAF の導入計画において検討する項目

図 4-5 の各項目におけるポイントを以下にまとめます。

### ■ WAF導入環境の事前確認

WAF の導入計画を立てるにあたり、まず WAF を導入するウェブサイトやウェブサーバの環境を調査します。例えば、ネットワーク設置型 WAF を設置する場合、少なからずネットワーク構成に影響を与えるため、WAF に関する設定以外で変更が必要ないかなどを確認します。また、サーバインストール型 WAF を設置する場合、WAF の動作に必要なソフトウェアを追加でインストールする必要がないかなどを確認します。

### ■ WAFの初期設定

WAF の導入時の設定について、事前に検討しておくべき主な項目は以下の通りです。

## ◆ 検査機能における対応範囲

ウェブサイトの構成や WAF を設置する場所によっては、複数のウェブアプリケーションを保護することができます。このとき特定のウェブアプリケーションのみを保護したい場合があります。また、WAF は誤検知を起こす可能性があるため、あらゆる攻撃を WAF で防御するのではなく、特定の攻撃のみ防御する場合があります。

WAF で防御したい攻撃や保護したいウェブアプリケーションを洗い出し、WAF の検査機能や拡張機能などの設定を検討します。

## ◆ ログの出力

3.2.2 章で説明したように、WAF は複数のログを出力します。これらのログは、運用においてすべて必要ではないため、取得するログを検討するとよいでしょう。合わせて、WAF のログの出力先やログを保存する期間などを検討しておくといよいでしょう。

## ■ WAFの動作検証

3.3 章で説明したように、WAF を導入したとしても、すべてのウェブアプリケーションの脆弱性を悪用した攻撃を防御できるわけではありません。また、WAF を導入することで、利用者の正常な HTTP 通信を WAF が防御してしまい、ウェブサイトの可用性を低下させる可能性もあります。そのため、WAF 導入した後に、すぐに運用を開始するのではなく、動作検証をするとよいでしょう。

この段階では、どの様な検証を行うのか？どのような結果ができればよいのか？いつまで検証をおこなうのか？といった点について、事前に検討しておくといよいでしょう。検討すべき詳細な内容については、「4.2.4 検証」を参照してください。

## ■ WAF導入時の導入体制

WAF の導入においては、WAF 導入担当者が関係者と相互に連携しながら有効な支援を受けることができるように体制を整備しておくといよいでしょう。また、問題発生時の報告先を明確にしておくといよいでしょう。

### 4.2.3. 運用計画

---

WAF を導入しても、導入時の状態ですべての攻撃を防御できるわけではありません。そのため、定期的に WAF の検出パターンを更新したり、場合によっては WAF をアップデートする必要があります。WAF の運用開始後に、運用をスムーズに行うために、事前に運用計画を立てておくといよいでしょう。



運用におけるトラブルを回避するためにも、運用計画の段階で検討した内容を運用手順書としてまとめておくことを推奨します。

## ■ 運用体制

WAF の運用においては、WAF 運用担当者が関係者と相互に連携しながら有効な支援を受けることができるように体制を明確にしておきましょう。WAF の設定変更を伴う検出パターンの更新や WAF のアップデートについては、承認ルートを定め変更管理のできる体制を整備しておくといでしょう。

また、問題発生時の報告先を明確にしておくといでしょう。

## ■ 運用ポリシー

WAF の運用では様々な作業が生じます (図 4-6)。WAF の導入および運用には WAF 担当者以外にも多くの関係者が関わるため、これらの作業主体や責任範囲が曖昧になりがちです。そのため、運用計画ではこれらの作業を想定し、「だれが?」、「いつ?」、「どうするか?」といったことを定めた運用ポリシーを検討します。運用ポリシーでは、特に「誰が主体で対応するのか」をきちんと決めておきましょう。

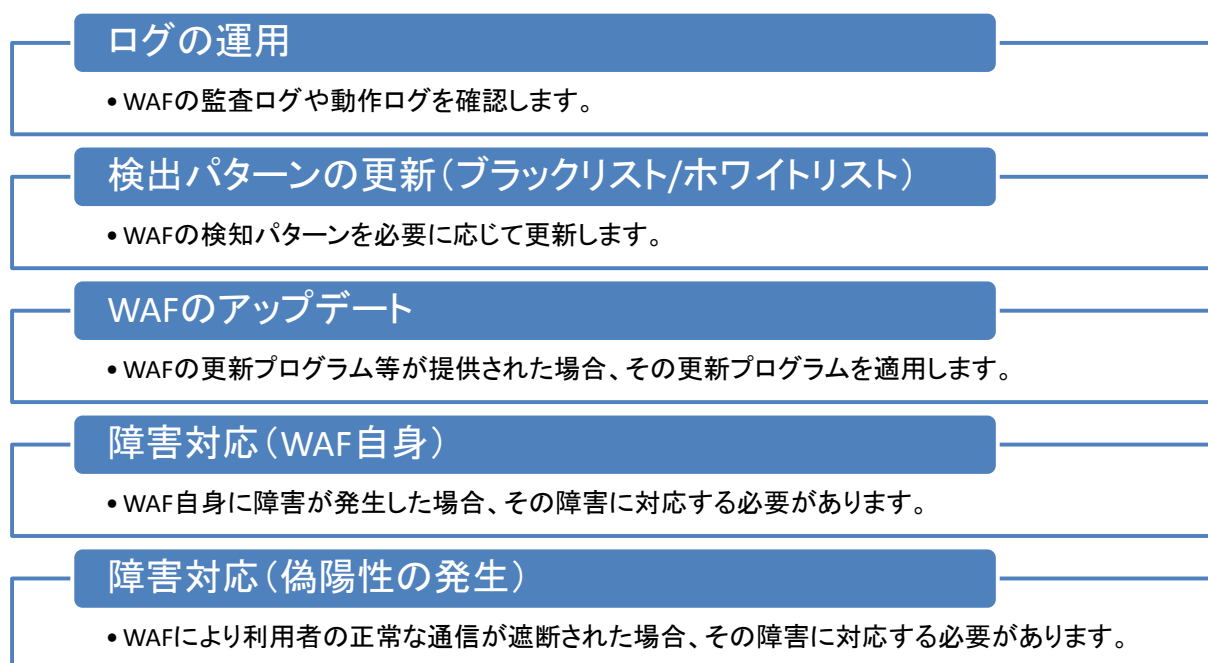


図 4-6 WAF の運用において生じる作業

WAF の運用において生じる作業におけるポイントを以下にまとめます。

### ◆ ログの運用

どのような頻度でどういった目的でログを確認するのか事前に検討しておきます。偽陽性の発生有無や攻撃傾向の把握等を行いたい場合では、短い頻度 (例えば、数時間おきに 1 回、1 日 1 回等) で WAF の監査ログを確認することを検討します。一方で、WAF が正常に動作しているか確認したい場合では、WAF の動作ログを確認することを検討します。

#### ◆ 検出パターンの更新(ブラックリスト/ホワイトリスト)

検出パターンを更新する頻度や方法と、検出パターンの更新によるウェブサイトへの影響を検討しておきます。

検出パターンを更新する頻度や方法は、ブラックリスト、ホワイトリストで異なります。ブラックリストを採用している WAF の場合、新しい攻撃に対応する度に検出パターンを更新します。一方で、ホワイトリストを採用している WAF の場合、ウェブアプリケーションを変更する際にホワイトリストを見直します。

検出パターンを更新することで、利用者の HTTP 通信が一時的に遮断される場合があります。また、新しい検出パターンを適用することで、偽陽性が生じる恐れもあります。そのため、新しい検出パターンの検証手順や更新手順を事前に検討しておくといでしょう。

#### ◆ WAF のアップデート

WAF の更新プログラムのリリースを確認する頻度や更新プログラムを適用する検証手順、ウェブサイトへの影響を検討しておきます。WAF をアップデートすることで、利用者がウェブサービスを利用できなくなる場合があります。そのため、WAF のアップデート手順や、更新手順、アップデートの可否を判断する承認ルートを事前に検討しておくといでしょう。

#### ◆ 障害対応(WAF 自身)

障害の対応体制や手順を検討しておきます。WAF 自身に障害が発生し WAF 自身が機能しなくなった場合、利用者がウェブサービスを利用できなくなる可能性があります。特に専用機器で提供される WAF をウェブサイトのゲートウェイに設置している場合、その WAF にハードウェア障害などが発生すると、同じウェブサイトで動作する他のサービスへの影響が懸念されます。WAF 障害時の影響範囲を考慮し、障害が発生した場合、どのような体制で対応するかを検討しておきます。

#### ◆ 障害対応(偽陽性の発生)

偽陽性の発生した場合の回避策や体制、手順を検討しておきます。偽陽性が発生した場合は、利用者がウェブサービスを利用できなくなる可能性があります。偽陽性が発生した場合、どのような体制で対応するのかについて検討しておきます。

## 4.2.4. 検証

WAFの運用を開始する前に、検証期間を設定して十分にWAFの動作や機能を検証しましょう。十分に検証することで、運用開始後のトラブルを回避できます。本節では、WAFの検証を図4-7の流れに沿って説明します。

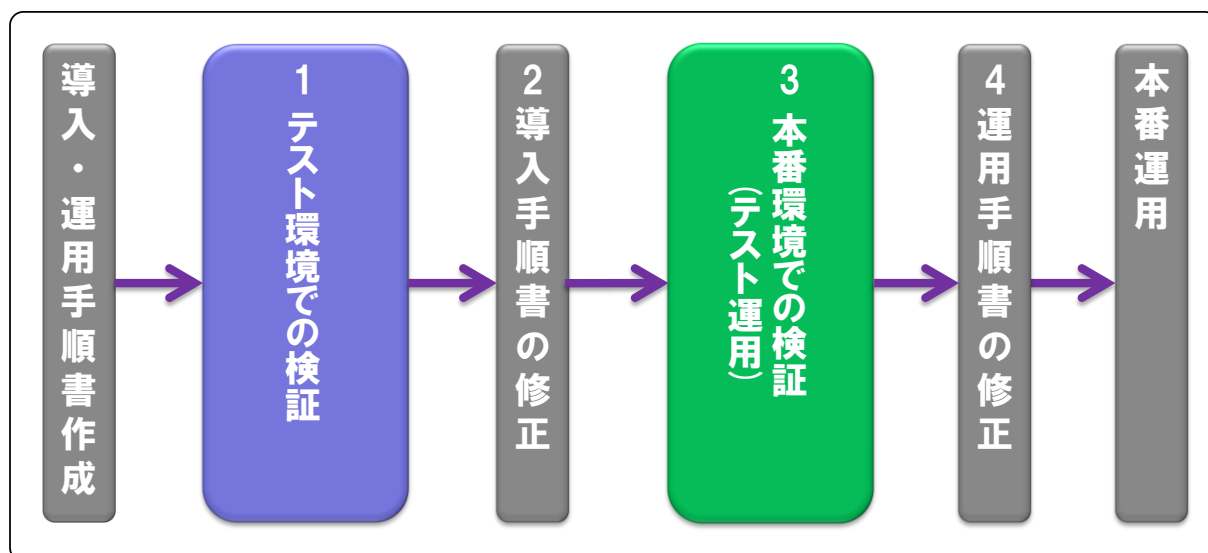


図 4-7 検証の流れ

WAFの検証にあたっては、ウェブサイトのサービスに影響を与えないように、まずテスト環境を使用します。この際、本番環境と同じ構成になっているテスト環境を使用することで、本番運用を想定した検証が容易になります。どうしてもテスト環境を準備することができない場合は、「3 本番環境での検証 (テスト運用)」から実施することになります。この場合、導入手順書で想定していない事象や導入手順書の誤りなどが原因で、利用者がウェブサービスを利用できなくなる等のリスクがあります。

テスト環境での検証が完了したら、本番環境にWAFを導入してテスト運用を開始します。このとき、WAFの通過処理（「3.2.1 基本機能」を参照）を利用して、HTTP通信に影響を与えない状態でログ等を確認していきます。

全ての検証を終えて問題がなかったら、WAFの本番運用を開始します。

検証における各項目のポイントを以下にまとめます。

### (1) テスト環境での検証

テスト環境での検証において、実施しておくべき検証内容は以下の通りです。

#### ◆ 導入手順の検証

導入手順書に沿ってテスト環境へWAF導入することで、導入手順書の妥当性を確認します。WAFの設置場所によらず、WAFは利用者とウェブサーバの間に介在します。そのため、WAFの導入手順に不備があると、サービスが停止してしまう可能性があります。テスト環境を利用し、WAFの導入手順に問題がないか確認しましょう。

## ◆ 偽陽性の検証

偽陽性が発生しないことを確認します。ウェブサイトが利用者に対して正常なサービスを提供できなくなることを防ぐため、偽陽性が発生しないことを十分に検証しましょう。検証方法の一つとしては、ウェブブラウザを用いて手動でウェブサイトアクセスし、すべてのリンクをクリックし、遮断しないことを確認するといった方法があります。

ウェブサイト運営者が自ら WAF を導入する場合、偽陽性の検証を自ら実施する必要があります。一方で WAF ベンダ等に WAF の導入を依頼する場合、偽陽性の検証を WAF ベンダに相談するとよいでしょう。

## ◆ 偽陰性の検証

WAF の検出パターンの設定間違いによる偽陰性が発生しないことを確認します。例えば、検出パターンにブラックリストを採用している場合、攻撃の種類（SQL インジェクション等）毎に WAF が検出する代表的な攻撃パターンをウェブサイトに対して送信します。その結果、防御対象の脆弱性を悪用した攻撃を WAF が検出するか確認します。一方、検出パターンにホワイトリストを採用している場合、保護対象となるページとパラメータに対して、通常入力が想定されない特定のパターンを入力します。その結果、許可されていない通信として WAF が検出することを確認します。

ウェブサイト運営者が自ら WAF を導入する場合、偽陰性の検証も自ら実施する必要があります。一方で WAF ベンダ等に WAF の導入を依頼する場合、偽陰性の検証を WAF ベンダに相談するとよいでしょう。

## ◆ 性能への影響の検証

必要に応じて WAF 導入による性能への影響を次の観点から測定します。

- レスポンスタイム、ターンアラウンドタイム  
ウェブサイト利用者の視点から、レスポンスタイムまたはターンアラウンドタイムを測定し、WAF の導入により、実際のウェブサイト利用に支障がないことを確認します。
- スループット  
ネットワーク設置型 WAF では、単位時間に処理できる HTTP リクエスト数を測定し、WAF を導入した状態であっても、ウェブサイトの可用性の目標に達することを確認します。
- リソース消費  
サーバインストール型 WAF では、インストールしたサーバの CPU 使用率、メモリ使用量、ハードディスク使用量を測定し、WAF を導入した状態であってもサーバの性能目標値を満たすことを確認します。

## (2) 導入手順書の修正

テスト環境での検証において、WAF を導入する際の初期設定を変更したり、導入手順が原因で問題が発生した場合には導入手順書を見直します。また、偽陽性や偽陰性の検証において、検証手順を変更した場合も、導入手順書を見直します。

### **(3) 本番環境での検証(テスト運用)**

本番環境に WAF を導入し、テスト運用を開始します。これまでに行ってきたテスト環境での検証では、利用者からのあらゆる通信を想定した検証ができたとは言い切れません。そのため、すぐに本番運用を開始した場合、偽陽性による通信の遮断が発生するなどサービスへ影響を与えるトラブルが発生する可能性があります。脆弱性を悪用した攻撃により被害が生じているなど、すぐに対処が必要な場合を除いて、必ずテスト運用期間を設けましょう。

テスト運用では、WAF の通過処理を利用し、一定期間ウェブサイトの可用性や、偽陽性と偽陰性の発生を確認します。ウェブサイトの運用に支障がでた場合には、このテスト運用で問題を解消し、「2 導入手順書の修正」に戻り、検証を進めましょう。

また運用手順書に沿って「ログの運用」、「検出パターンの更新」、「WAF のアップデート」など実際の運用と同様の運用を行い、運用手順書に不備がないか確認します。

### **(4) 運用手順書の修正**

本番環境での検証において、運用手順書に沿って運用した結果、問題が発生した場合には運用手順書を見直します。

## 4.3. 運用

「運用」では、運用手順書に則って WAF を運用します。ウェブサイトから WAF を取り除くか、ウェブサービスを停止するまで、この工程を続けることが重要です。

本節では、「運用」でのポイントについて、「通常運用」と「緊急対応」、「保守」の 3 つの項目に分けて説明します。なお、それぞれの項目は場合によっては同時に実施することを想定しています。

### 4.3.1. 通常運用

WAF を効果的に活用するためには以下の運用を行なうことが重要です。

#### ■ WAF のアップデートや検出パターンの更新

WAF を運用していると、WAF の開発元から更新プログラムがリリースされることがあります。その更新プログラムを WAF に適用することで、新機能の追加や機能改善、場合によっては WAF 自身の脆弱性が修正されます。

また、ブラックリストを利用した WAF を利用している場合、WAF の開発元から新たな攻撃に対応した検出パターンが提供されることがあります<sup>45</sup>。一方、ホワイトリストを利用した WAF を利用している場合、ウェブアプリケーションに変更したときに、検出パターンを見直す必要が生じます<sup>46</sup>。検出パターンを更新することで、脆弱性を狙った攻撃からウェブアプリケーションを保護し続けることができます。

WAF の更新プログラムや検出パターンについては、運用手順書に基づき WAF へ適用を検討します。WAF のアップデートや検出パターンを更新する場合、「導入」における検証と同様に、まずテスト環境で検証することが大切です。テスト環境での検証に問題がなかった場合、本番環境でアップデート作業を実施します。テスト環境での検証に問題があった場合、運用手順書の見直しも検討します。



アップデート作業では、その都度アップデート作業を検証しましょう。必要があれば運用手順書も見直すことが重要です (図 4-8 を参照)。

<sup>45</sup> これは検出パターンとしてブラックリストを使用している場合です。

<sup>46</sup> これは検出パターンとしてホワイトリストを使用している場合です。



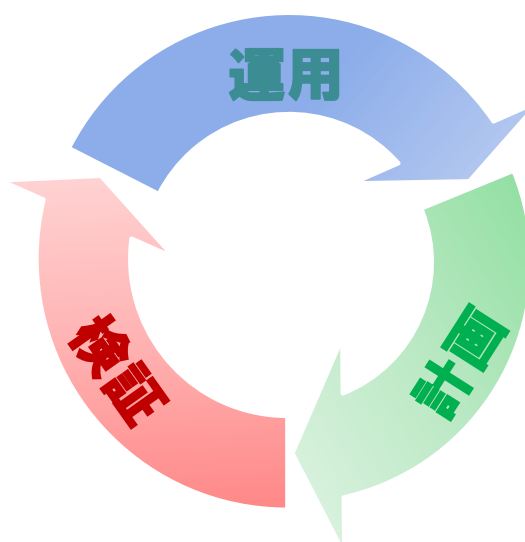


図 4-8 アップデート作業における運用サイクル

### ■ 定期的なログの確認

運用手順書に基づき、WAF のログを定期的に確認します。

監査ログの確認作業により、偽陽性の発生有無を確認できます。偽陽性が発生していた場合、「4.3.2 緊急対応」を実施します。加えて、監査ログを確認することで、現在の攻撃手法の傾向を把握することが可能となり、ウェブサイトのセキュリティ対策を立てる上での指針の一つになります。

また、動作ログの確認作業では、WAF の障害発生の予兆を確認します。動作ログに障害の疑いがある場合、事前に対応可能な予防策（保守部品の準備など）があれば対応します。

### 4.3.2. 緊急対応

---

緊急対応は発生頻度が少ないわりに、発生したときの影響は大きいです。以下のような事象が発生することを前提として、定期的に訓練することを推奨します。

#### ■ WAF 自身の障害

WAF 自身に障害が発生した場合は、運用手順書に基づき速やかに対応します。

#### ■ 偽陽性の発生

偽陽性の発生はサービスへの影響が大きいため、運用手順書に基づき速やかに対応する必要があります。

### 4.3.3. 保守

---

商用 WAF を導入した場合は、必ず保守契約を結んでおく必要があります。

#### ■ ハードウェア保守

専用機器で提供される商用 WAF を導入した場合、ハードウェアに関する保守契約を結んでおく必要があります。また、サーバインストール型 WAF を導入する場合でも、インストールしたサーバ自体の保守契約を結んでおく必要があります。

ハードウェアの保守契約を結んでいないと、ハードウェア部品の故障により、WAF を動作させることができなくなり、ウェブアプリケーションの脆弱性を悪用した攻撃に対して、ウェブサイトが無防備な状態となる可能性があります。

#### ■ ソフトウェア保守

WAF 自身の更新プログラムやブラックリストの更新を受けるために、ソフトウェアに関する保守契約を結んでおく必要があります。

ソフトウェアの保守契約を結んでいないと、ブラックリストが更新できないため最新の攻撃手法に対応できない状態や、WAF 自身に脆弱性がある状態が発生する可能性があります。

## 5. IPA における WAF 導入・運用事例

4章では、WAF 導入におけるポイントについて紹介しました。本章では、オープンソース WAF 「ModSecurity」を、実際に IPA が導入・運用した事例を紹介します。また、IPA での WAF 運用事例を通して得られた運用におけるポイントについても紹介します。

### 5.1. まえがき

IPA では 2010 年にウェブサイト運営者向けセキュリティ対策セミナーを開催しました。そのセミナーにおいて、「WAF を日本語で紹介している文献が少ない」、「WAF の導入事例がないため、導入に向けて何を検討していいのかわからない」などのウェブサイト運営者の悩みが分かりました。

このため、IPA は自ら WAF を導入し、WAF ができること、できないことなど WAF に関する知識を蓄積し、蓄積した情報を導入事例として公開します。

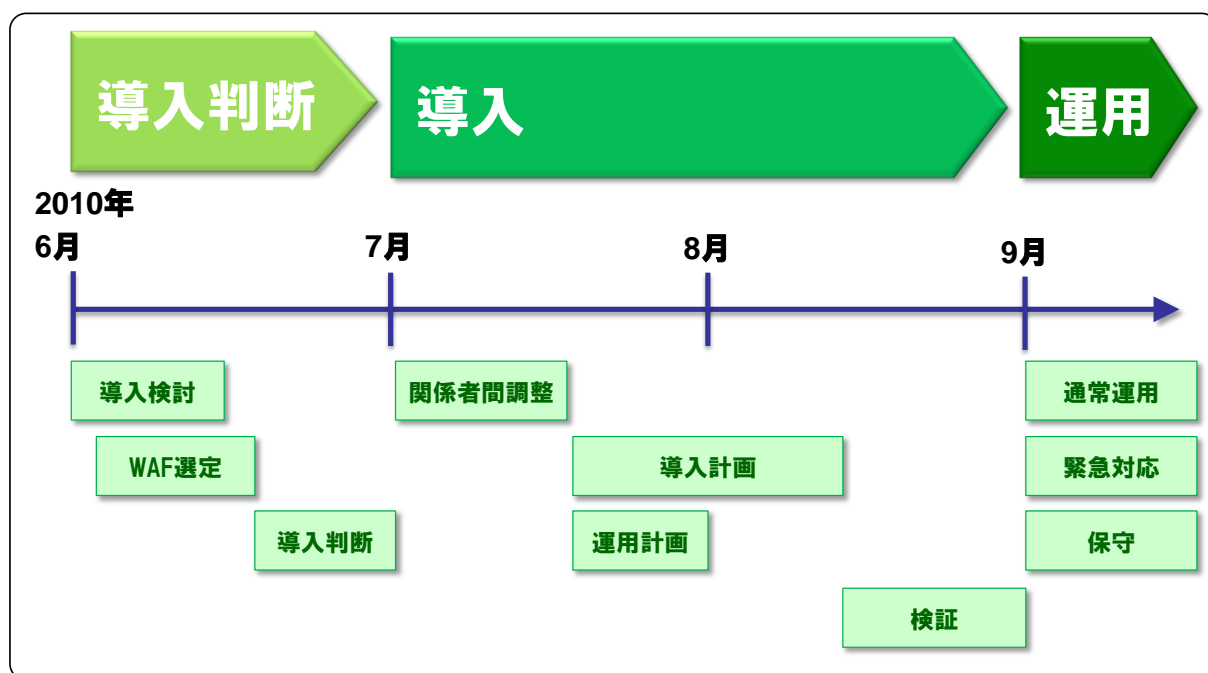


図 5-1 IPA における WAF 導入スケジュールの実情

実際には 4 章で定義した「WAF を導入して運用を開始するまでの 3 つの工程」である、「導入判断」「導入」「運用」の流れに沿って、図 5-1 に示すスケジュールで、WAF を導入しました。この導入に要した期間は、約 2 ヶ月です。また、本章を執筆している時点で、約 4 ヶ月間 WAF を運用しています。

次節以降では、「導入判断」「導入」「運用」それぞれの工程において、実際に「IPA が何を検討したのか?」「どのような作業を行ったのか?」といった項目を紹介します。

今回、国内で利用されるソフトウェア等の製品(OS、アプリケーション、ライブラリ、組み込み製品など)の脆弱性対策情報を中心に収集・蓄積する脆弱性対策情報データベース「JVN iPedia」<sup>47</sup>に WAF を導入しました。

なお、本章は図 5-2 に示す「WAF 推進業務担当者」からの視点で説明します。

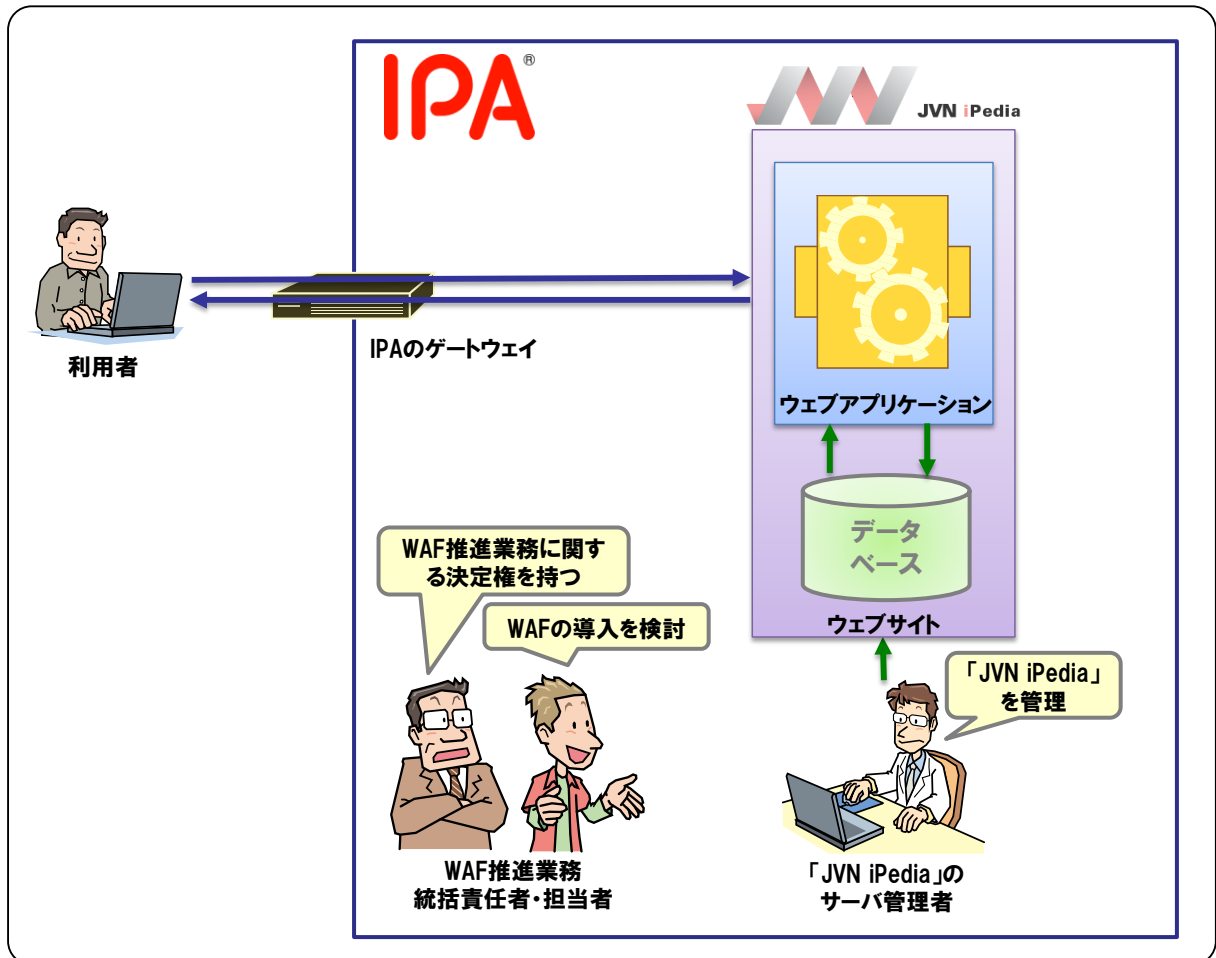


図 5-2 導入検討開始時に判明していた「JVN iPedia」の運用の関係図

<sup>47</sup> <http://jvdb.jvn.jp/>

## 5.2. 導入判断

---

本節では、WAF を導入して運用を開始するまでの 3 つの工程のうち、最初に行う工程である「導入判断」について紹介します。IPA が考える「導入判断」におけるポイントは、4 章に記載してありますので、合わせて読むとより理解が深まります。

### 5.2.1. 導入検討 [事例]

---

「導入検討」におけるポイントはこちら 4.1.1 章

5.1 章にて紹介した理由から、今回はテストケースとして WAF を導入しました。WAF を導入した「JVN iPedia」には脆弱性の存在は確認されていないため、WAF の導入は必要ではありませんでした。

なお、本来は、2.4 章「WAF が有効な状況」や 4.1.1 章「導入検討」を参考にして、WAF の導入を検討します。

### 5.2.2. WAF 選定 [事例]

---

「WAF 選定」におけるポイントはこちら 4.1.2 章

WAF を選定するにあたり、WAF の導入に関する決定権をもつ「WAF 推進統括責任者<sup>48</sup>」に相談しました。また、実際に WAF を導入する「JVN iPedia」のウェブサーバの構成等を、「JVN iPedia」のサーバ管理者にヒアリングしました。

#### ■ WAF 推進統括責任者への相談

まず WAF 推進統括責任者に WAF を導入することについて、相談しました。この時の相談では、「WAF を導入する目的」、「WAF を導入することによるメリット・デメリット」などを説明しました。その結果、「JVN iPedia」へ WAF を導入することの承諾を得ました。

続いて、導入する WAF の種類について相談しました。その結果、今回の WAF 導入では、商用 WAF ではなく、オープンソース WAF を導入するとの結論に至りました。

オープンソース WAF を導入するとの結論に至った理由は、今回の WAF 導入はテストケースであり、WAF ベンダによるサポートや WAF の機能が充実している商用 WAF を導入するのではなく、オープンソース WAF を導入することで、導入で苦勞する点などを実体験から蓄積できると考えたためです。

---

<sup>48</sup> 実際にはウェブサーバ運用の予算について決定権のある人が望ましいです。

## ■ 「JVN iPedia」の管理者へのヒアリング

「JVN iPedia」のウェブサーバの構成について、「JVN iPedia」のサーバ管理者にヒアリングしました。「JVN iPedia」のサーバ管理者へのヒアリングから得られた「JVN iPedia」のウェブサーバの構成は以下の通りです。

- Linux で構築されている
- 「JVN iPedia」のウェブサーバソフトウェアとして、「Apache」を使用している

また、サーバ管理者へのヒアリングから、「JVN iPedia」のウェブサーバは、「MyJVN」<sup>49</sup>へのリバースプロキシとしても機能していることがわかりました。この「MyJVN」には API の提供など動的コンテンツが多く存在しています。「MyJVN」への通信を WAF で検査すると、偽陽性が発生する可能性が高く、「MyJVN」サービスへの影響が懸念されます。このため、「JVN iPedia」へ WAF を導入する際、「MyJVN」への通信について WAF による検査の対象外にする<sup>50</sup> 必要があることがわかりました(図 5-3)。

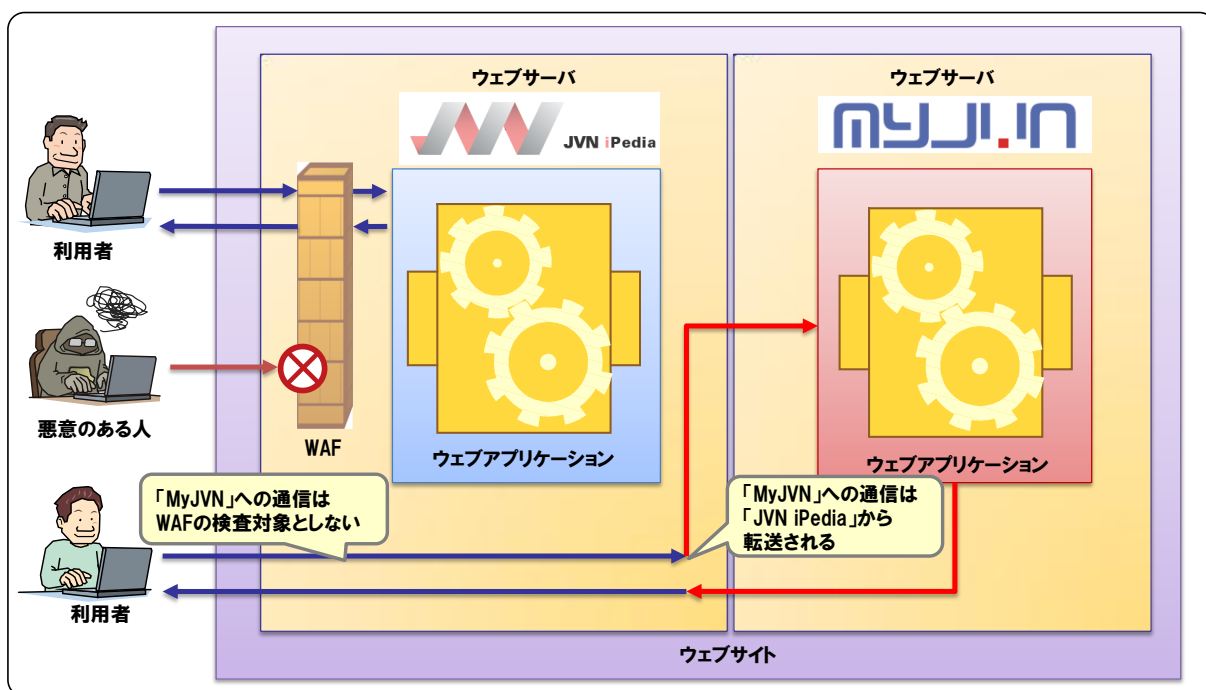


図 5-3 「JVN iPedia」と「MyJVN」の関係

一方、「JVN iPedia」のサーバ管理者に運用に関わる関係者について確認し、加えてその関係者の役割についてもヒアリングしました。このヒアリング結果、新たにわかった関係者は以下の方々です。これらの関係者の関係を図 5-4 にて紹介します。

- IPA のネットワーク管理者(「JVN iPedia」のサーバ管理者とは別部門が管理している)
- JVN iPedia の開発業者

<sup>49</sup> <http://jvndb.jvn.jp/apis/myjvn/>

<sup>50</sup> 検査対象外等の設定については、「5.3.2 導入計画 [事例]」で検討しています。

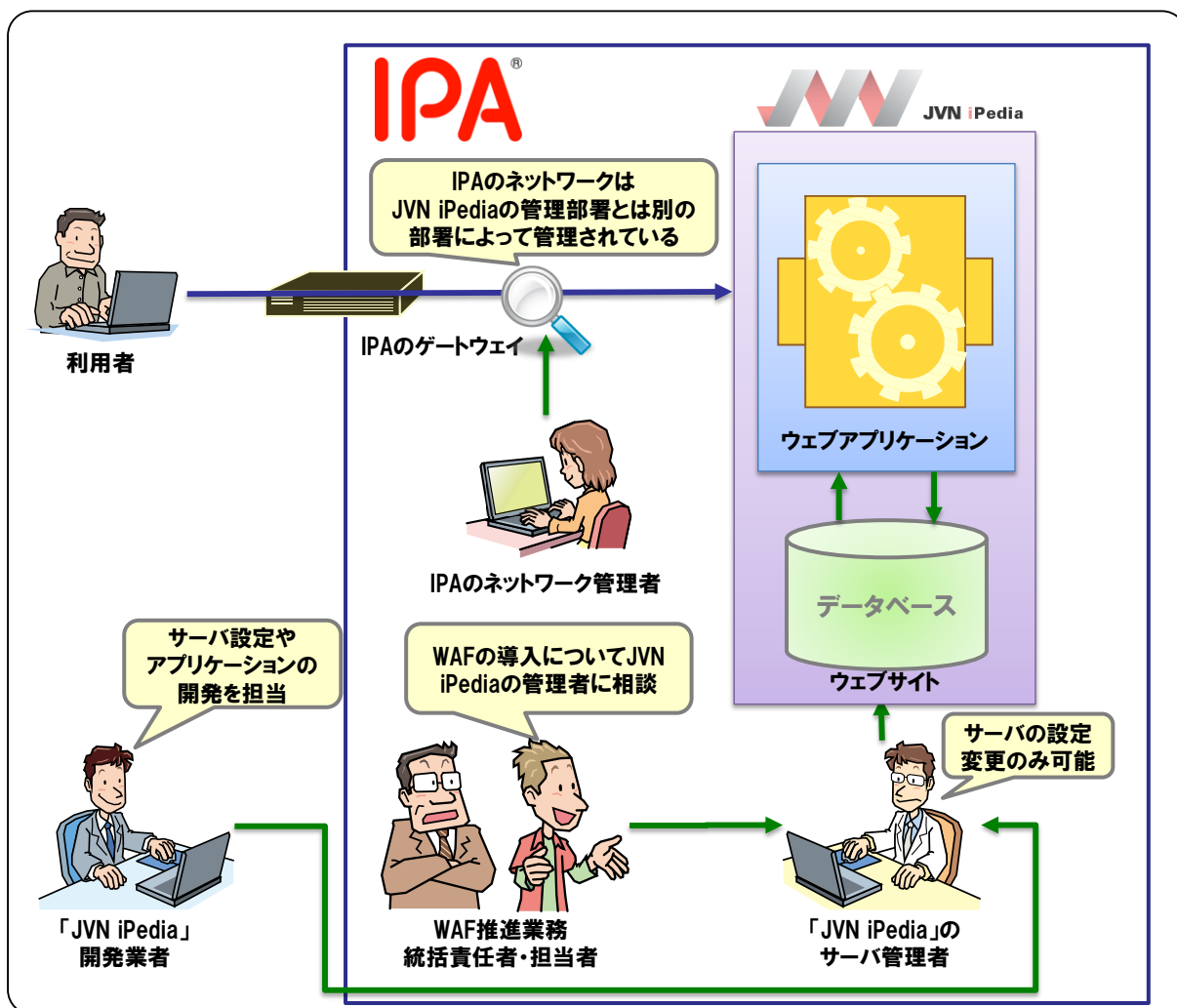


図 5-4 実際のJVN iPedia の運用の関係図

**POINT**

事前ヒアリングにより、WAF の選定に必要な情報を得ることができました。

**POINT**

事前ヒアリングにより、「MyJVN」に関する設定など導入時に検討が必要な情報を得ることができました。

**POINT**

事前ヒアリングにより、導入時に調整が必要な関係者を知ることができました。

WAF推進統括責任者への相談、「JVN iPedia」のサーバ管理者へのヒアリング結果を踏まえて、今回導入する WAF を選定しました。

ヒアリング結果から抽出した、WAF を選定するための要件は以下の通りです。

- 本来は、トータルコストを算出し要件定義をすることが必要だ。しかし、今回はテストケースであり、WAF を導入することで苦勞した点などを蓄積するために WAF を導入するため、オープンソース WAF を利用したい。
- 今回のテストケースのために、IPA のネットワーク構成を変更するなど別部門へ負担をかけることは難しい。そのため、できる限りネットワーク構成の変更は避けたい。
- 「JVN iPedia」は、Linux で構築されており、ウェブサーバソフトウェアとして「Apache」を使用している。できる限り「JVN iPedia」の環境を変えることなく WAF を動作させたい。

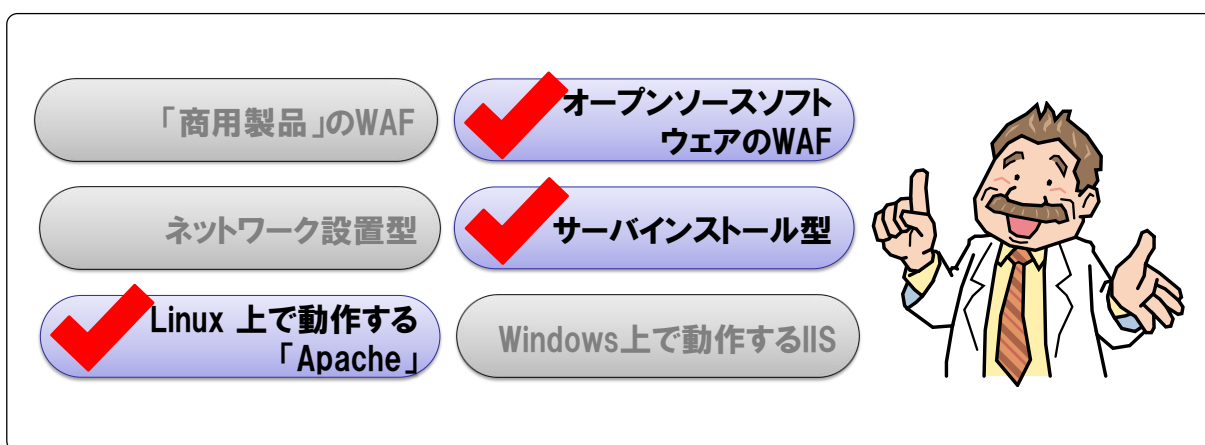


図 5-5 IPA が導入する WAF の要件

これまでの検討結果(図 5-5)より、次の WAF を導入することとしました。

選定結果: オープンソース WAF

**ModSecurity<sup>51</sup>**

### 5.2.3. 導入判断[事例]

導入する WAF 選定を終えたのち、WAF 推進統括責任者に WAF の選定過程、選定結果を説明しました。また、選定した WAF の導入にかかる導入費用、WAF の運用にかかるであろう運用費用の概算を説明しました。その結果、WAF を導入することについて最終的な許諾を得ました。

<sup>51</sup> 本書では、「ModSecurity」の詳細な説明を割愛します。なお、インストール手順については、「付録 A. オープンソースソフトウェアの紹介」を参照してください。



## 5.3. 導入

本節では、WAF を導入して運用を開始するまでの 3 つの工程のうち「導入」について紹介します。IPA が考える「導入」におけるポイントは、4.2 章に記載してあります。合わせて読むとより理解が深まります。

### 5.3.1. 関係者間調整 [事例]

「関係者間調整」におけるポイントはこちら 4.2.1 章

まず「JVN iPedia」のサーバ管理者へのヒアリングから得られた関係者に WAF の導入について説明しました。調整対象の関係者は図 5-6 の通りです。

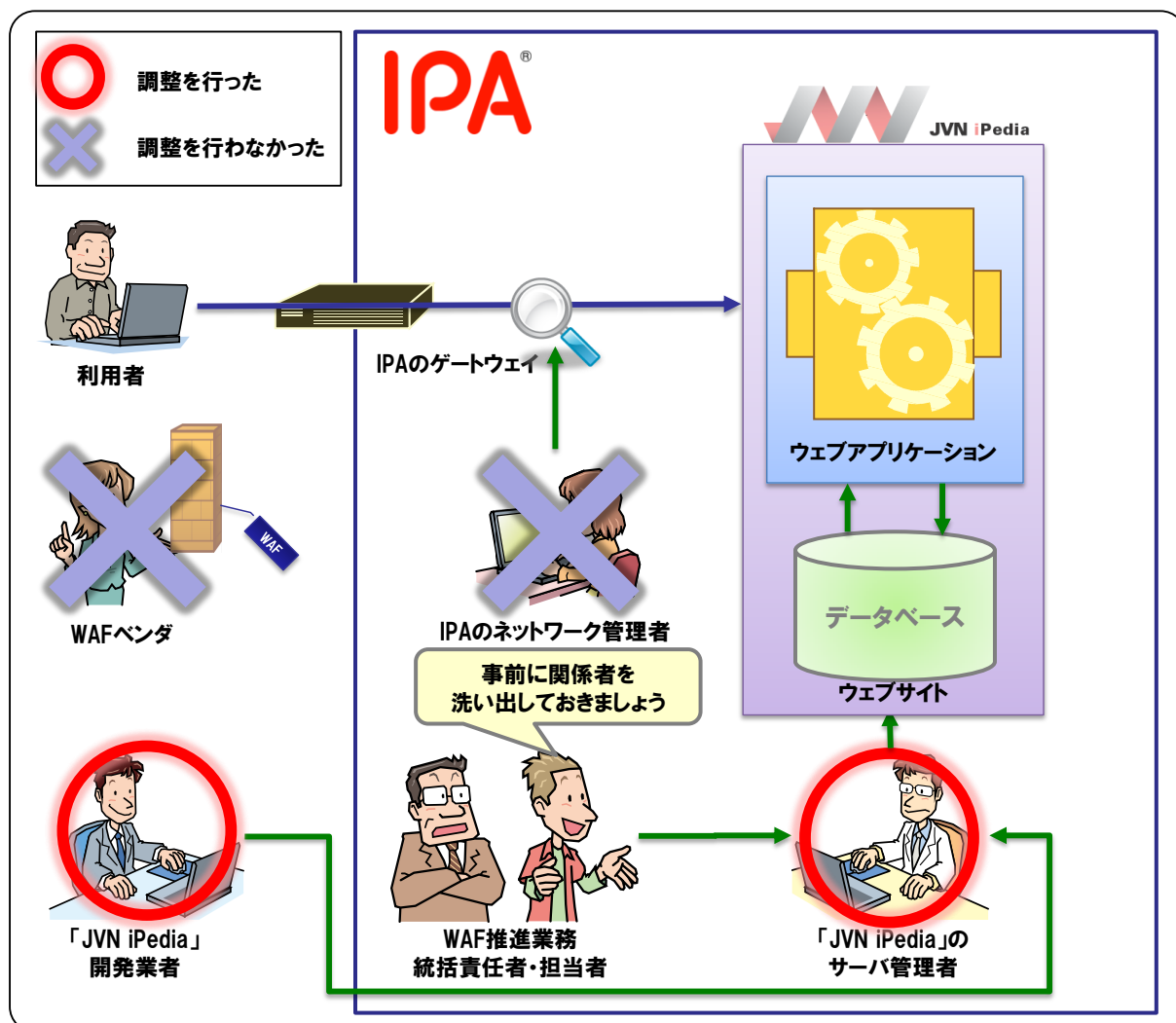


図 5-6 WAF 推進業務担当者が調整を行った関係者

以降本項では、実際の調整内容および調整結果の詳細について紹介します。

## ■ IPA のネットワーク管理者との調整

今回導入した「ModSecurity」は、サーバインストール型 WAF<sup>52</sup>です。ネットワーク構成の変更は必要ありません。このため、IPA のネットワーク管理者に調整する必要はありませんでした。



## ■ 「JVN iPedia」のサーバ管理者との調整

「ModSecurity」は、「Apache」のモジュールとして動作する WAF です。インストールやアップデートを行う際には、「Apache」の再起動が必要となり、利用者が「JVN iPedia」を一時的に利用できなくなるなど、導入におけるリスクを「JVN iPedia」のサーバ管理者に説明しました。



また、3.3 章で説明したように WAF を導入した場合、偽陽性などの誤検知が発生し利用者の正常な通信を遮断してしまう可能性があります。このような運用におけるリスクを「JVN iPedia」のサーバ管理者に説明しました。

一方でリスクを最大限低減するための施策として、導入を行う前に導入テスト・偽陽性の発生に関する確認などの検証を十分に検討している事を説明しました。

この結果、WAF を導入することについて、サーバ管理者の了承を得ました。

なお、サーバ管理者には「導入判断」の段階で一度ヒアリングしています。当然、その時点でも WAF を導入する目的、効果、リスクを簡単に説明していました。再度これらを説明した理由としては、「導入判断」の時点と異なり、導入する WAF が決まったことがあげられます。導入する WAF の特徴が明確になったため、その WAF の特徴、リスクなどを中心に再検討し、これらを詳細に説明しました。

## ■ 「JVN iPedia」開発業者との調整

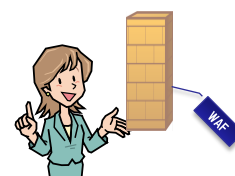
「JVN iPedia」は、公募（企画競争）により選定された開発業者が作成したウェブアプリケーションです。そのため、「ModSecurity」の導入が「JVN iPedia」のウェブアプリケーションに影響を与える可能性はないか、開発業者に確認しました。



その結果、WAF 導入が「JVN iPedia」のウェブアプリケーションへ影響を与えないことを確認できました。

## ■ WAF ベンダとの調整

「ModSecurity」は、オープンソース WAF でありサポートサービスを前提にはしません。そのため、WAF ベンダに調整する必要はありませんでした。



<sup>52</sup> サーバインストール型 WAF の特徴は、3.1.2 章を参照してください。

## 5.3.2. 導入計画 [事例]

「導入計画」におけるポイントはこちら 4.2.2 章

WAF の関係者との調整が終わった後、WAF の導入計画を立てました。IPA では図 5-7 の順序で WAF を導入することとしました。



図 5-7 IPA の導入計画

以降本項では、導入計画において実際に検討した内容および検討結果を、図 5-7 の項目ごとに紹介します。

### (1) 導入前の事前確認

導入計画を立てるにあたり、「ModSecurity」を導入する「JVN iPedia」のウェブサーバの詳細な環境について、事前に次の内容を確認しました。

#### ◆ ハードウェア構成について確認

WAF を導入するウェブサーバのハードウェア構成として、主にハードディスクの残量を確認しました。これは、「ModSecurity」を導入することで、ハードディスクに出力されるログの全体量が増加するためです。

「JVN iPedia」のウェブサーバにおけるハードディスクの残量を確認した結果、ハードディスクの容量は十分な余裕がないことがわかりました。この結果、導入計画の段階でログの出力量について、検討し調整することとしました(詳細は (2) 初期設定の決定で紹介)。

#### ◆ 必須ソフトウェアのインストール状況について確認

「ModSecurity」をインストールするためには、事前にいくつかのソフトウェアをインストールしておく必要があります<sup>53</sup>。これらのソフトウェアが現在「JVN iPedia」が動作しているウェブサーバにインストールされているかどうかを確認しました。

<sup>53</sup>詳細については「付録 A. オープンソースソフトウェアの紹介」を参照してください。

この結果、「ModSecurity」をインストールする際に、いくつかのソフトウェアをインストールする必要があることがわかりました。

## (2) 初期設定の検討

### ◆ 「ModSecurity」で遮断するための検出パターンの決定

今回の導入はテストケースであり、「ModSecurity」に同梱されている「Core Rule Set」<sup>54</sup>がどの程度有効であるか確認をすることも、目的の一つとしました。

しかしながら、「Core Rule Set」に含まれるすべてのルールを適用した場合、偽陽性の発生など誤検知が発生する可能性が高く、運用への影響が懸念されます。そのため、まずは攻撃が成功してしまった場合、影響が深刻な脆弱性である「SQL インジェクション」に関する検出パターンのみ有効としました。

有効にした「Core Rule Set」における検出パターンは次の通りです。

- modsecurity\_crs\_41\_sql\_injection\_attacks.conf

### ◆ 「ModSecurity」及び「Core Rule Set」のバージョンを決定

「ModSecurity」、「Core Rule Set」とともに、導入時点の最新バージョンを導入しました。導入を検討した際の最新バージョンは次の通りです。

- ModSecurity v2.5.12
- Core Rule Set v2.0.7

### ◆ ログの出力の有無および保存期間の決定

「ModSecurity」が出力するログには、「検知ログ<sup>55</sup>」「監査ログ」「動作ログ」の三種類があります。これらのログの出力の要否について検討しました。

今回の WAF 導入の目的は、「WAF のできる事、できない事を確認すること」ですので、全てのログを出力し、WAF の動作を確認することとしました。しかし、(1) 導入前の事前確認で紹介したように「JVN iPedia」のハードディスクは、十分な余裕がない状況でした。そのため、出力された「監査ログ」「動作ログ」をウェブサーバの「アクセスログ」に比べて短い期間だけ保存しておく設定にしました。

「監査ログ」「動作ログ」の保存期間

- 第 5 世代まで(5 週間分)

### ◆ その他設定の決定

5.2.2 章で紹介したように、「JVN iPedia」は「MyJVN」へのリバースプロキシとしても機能しています。「MyJVN」は API の提供など動的コンテンツが多く、WAF の導入により偽陽性が発生する可能性が高いと考えました。偽陽性の発生は、サービスへの影響が大きいことから、「MyJVN」への通信は「ModSecurity」による検査の対象外としました。

<sup>54</sup> 「Core Rule Set」は、OWASP が開発している検出パターンです。OWASP、「Core Rule Set」については、「1.3.2 OWASP の取り組み」を参照してください。

<sup>55</sup> 「ModSecurity」では、「監査ログ」とは別に検知結果だけを「Apache」のエラーログに出力可能です。

ここでは、「MyJVN」への通信を検査対象外とした設定例を紹介します。以下の設定を「Apache」の設定ファイルに記述します。この設定では URI が「/en/apis/myjvn」または「/apis/myjvn」の HTTP 通信を、「ModSecurity」による検査の対象外にします

(参考) [MyJVN] への通信を検査対象外とする設定

```
<LocationMatch "^(/en)?(/apis)?/myjvn">  
  SecRuleInheritance Off  
</LocationMatch>
```

### (3) 導入手順の検討

「ModSecurity」の導入が原因で「JVN iPedia」のサービスが停止してしまうと、利用者が「JVN iPedia」を利用できなくなります。そのため、本番環境に導入する前にテスト環境（仮想環境）で動作確認や偽陽性の発生等を検証することとしました。

IPA が実際に検討した導入手順は次の通りです（図 5-8）

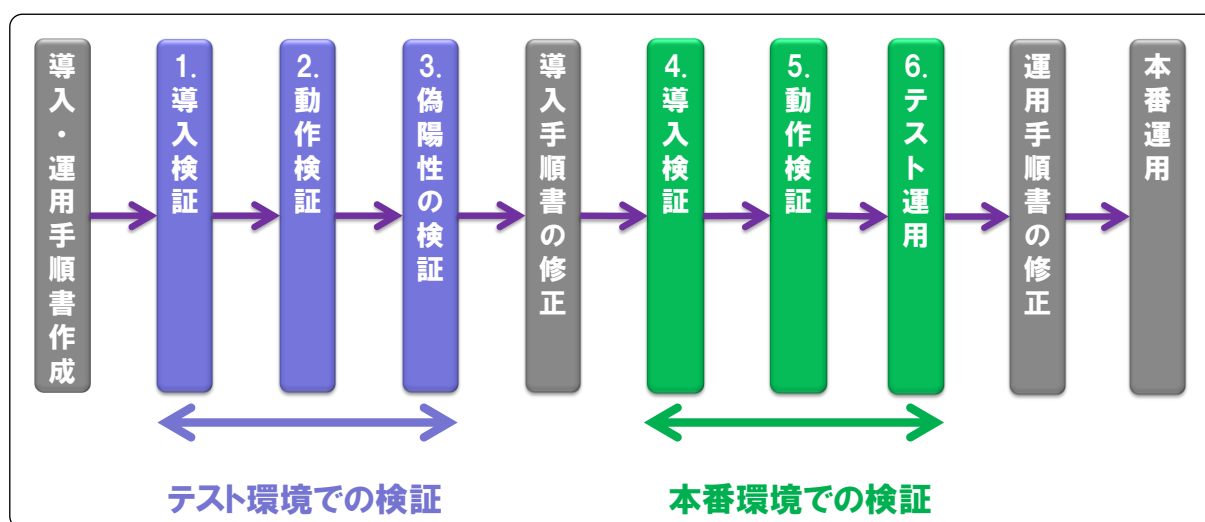


図 5-8 IPA の導入手順

なお、仮想環境上に構築するテスト環境の構成は、ソフトウェアのバージョンを含め、本番環境とほぼ同じ構成となっています。また、検証内容についてもこの時点で検討しました。この検証内容の詳細については、5.3.4 章で紹介します。

### (4) 導入に関する対応の検討

次に実際に「JVN iPedia」に「ModSecurity」を導入する際の対応を検討しました。

#### ◆ 導入日時及び利用者への通知日時について検討・調整

「ModSecurity」を導入する際は、サービスが一時的に停止する可能性があります。一方で、「JVN iPedia」のメンテナンスを行う際は、ウェブページ上で事前に告知するとの運用ポリシーを設けています。そのため、事前に「ModSecurity」の導入日時と、ウェブページ上で利用者へ通知する日時を「JVN iPedia」の管理者に調整しました。

## ◆ 導入作業スケジュール及び連絡体制の明確化・調整

導入作業スケジュールを明確にしました。具体的には、WAFの導入担当者に加えて、緊急時にWAF導入中止等を判断する責任者の予定を確保しました。

また、WAFの導入が原因によるサービスの停止など、緊急時に必要となる連絡体制を明確にしました。

### (5) 導入手順書の作成

「導入計画」におけるこれまでの(1)～(4)の検討結果をもとに、以下の2つの目的で、導入手順書を作成しました。

- 導入作業時のタイプミスなどを極力減らすため
- 担当者が変わった場合でも、誰でも導入できるようにするため

なお、今回作成した導入手順書には次のような内容を記載しています (図 5-9)。

## 導入手順書

1. 「ModSecurity」導入手順
  - 1.1. 動作に必要なソフトウェアのインストール手順
  - 1.2. インストール手順
  - 1.3. 検証手順
2. 「ModSecurity」設定変更手順
  - 2.1. ログファイルの管理設定変更手順
  - 2.2. ルール(シグネチャ)の変更手順
  - 2.3. 「ModSecurity」のアップデート手順
  - ⋮

**TIPS**

- A. 「ModSecurity」の動作モード切替手順
- B. 「ModSecurity」の取り外し手順
- ⋮

導入手順書にはこんな内容が記載されているんですね




図 5-9 作成した導入手順書の内容

### 5.3.3. 運用計画 [事例]

「運用計画」におけるポイントはこちら 4.2.3 章

WAF の導入計画を立てると同時に、WAF の運用計画も立てました。本項では、運用計画における実際の検討内容および検討結果を紹介します。

#### (1) 運用ポリシーの作成

「誰が?」「いつ?」「いつまでに?」「どうする?」といった事を定める運用ポリシーを作成しました。ここでは、実際に IPA が作成した運用ポリシーを紹介します。なお、ここで登場する「誰が?」の担当者は、図 5-4 で紹介した関係者が該当します。

##### ◆ 「ModSecurity」のアップデートポリシー

「ModSecurity」のアップデートにおいては、「Apache」の再起動が必要となり、「JVN iPedia」のサービスが一時的に停止する可能性があります。そのため、新しい「ModSecurity」がリリースされた場合、必ずアップデートするのではなく、必要な場合のみアップデートすることとしました。

IPA の「『ModSecurity』のアップデートポリシー」では次のように規定しています (図 5-10)。

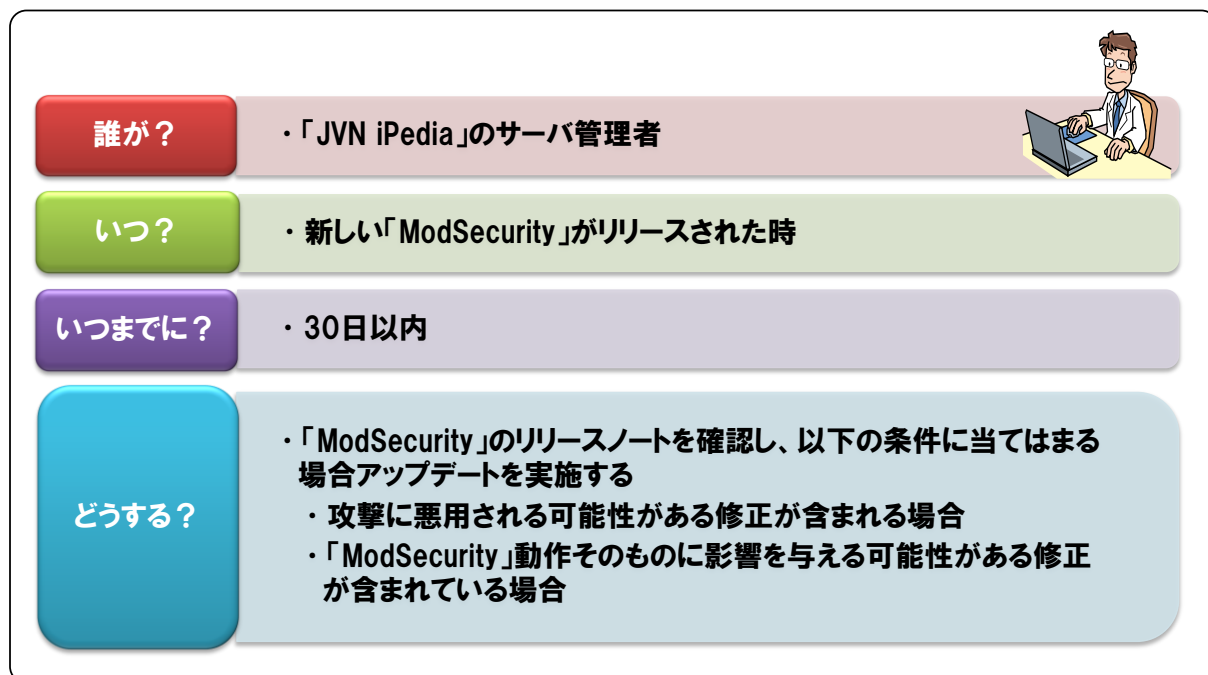


図 5-10 「ModSecurity」のアップデートポリシー

##### ◆ 「Core Rule Set」のアップデートポリシー

「Core Rule Set」のアップデートにおいても「ModSecurity」のアップデートと同様に、「Apache」の再起動が必要です。このため「Core Rule Set」がリリースされた場合、必ずアップデートするのではなく、必要な場合のみアップデートすることとしました。

IPA の「『Core Rule Set』のアップデートポリシー」では次のように規定しています (図 5-11)。

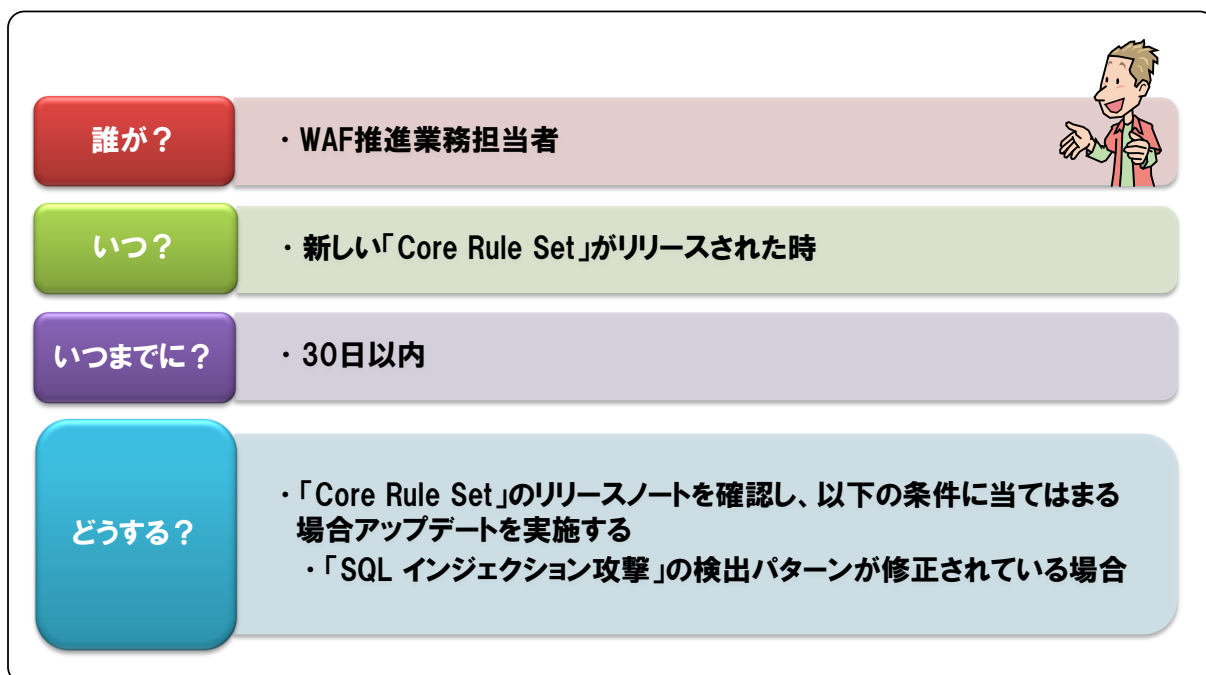


図 5-11 「Core Rule Set」のアップデートポリシー

◆ 「ModSecurity」のログ確認ポリシー

偽陽性によるサービスへの影響を考慮し、「ModSecurity」のログを日々確認することで、早い段階で偽陽性を発見できる仕組みを作ることとしました。

「ModSecurity」導入当初、IPAの「ログ確認ポリシー」では次のように規定していました（図 5-12）。

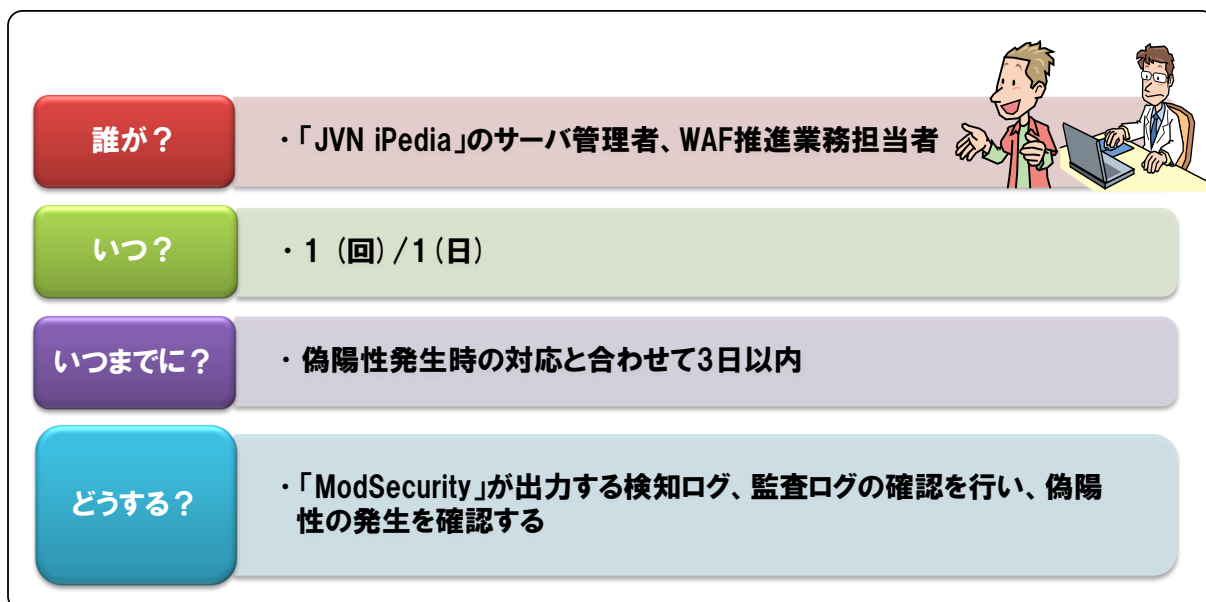


図 5-12 ModSecurity のログ確認ポリシー

なお、本番運用開始から 2011 年 1 月初旬まで、「JVN iPedia」のサービスに影響を与える偽陽性は発生していません。そのため、本書を執筆している時点では、ログの確認を 1（回）/1（週）に変更しています。



#### ◆ 偽陽性発生時の対応ポリシー

偽陽性発生時の対応はログ確認と同様に可能な限り早い対応が必要であると考えました。IPAの「偽陽性発生時の対応ポリシー」では次のように規定しています(図 5-13)。


誰が?	・ WAF推進業務担当者	
いつ?	・ ログ確認時に偽陽性の発生が確認できた時	
いつまでに?	・ ログ確認の対応と合わせて3日以内	
どうする?	・ 「ModSecurity」をウェブサーバから切り離す。 ・ 偽陽性発生時の「Core Rule Set」の把握と「ModSecurity」ユーザーメーリングリストなどから原因を調査。 ・ 「Core Rule Set」の見直しを検討する。	

図 5-13 偽陽性発生時の対応ポリシー

#### ◆ 「ModSecurity」障害発生時の対応ポリシー

「ModSecurity」に障害が発生した場合、「Apache」が停止してしまうなど「JVN iPedia」の運用に影響を与える可能性があります。このように「ModSecurity」の障害発生時は、サービスへの影響が大きいと想定できることから、即日対応することとしました。

IPAの「『ModSecurity』障害発生時の対応ポリシー」では次のように規定しています(図 5-14)。


誰が?	・ 「JVN iPedia」のサーバ管理者	
いつ?	・ 「ModSecurity」の障害によりウェブサーバが停止した時	
いつまでに?	・ 即日対応	
どうする?	・ 「ModSecurity」をウェブサーバから切り離す。 ・ 障害の原因を「ModSecurity」ユーザーメーリングリストなどから調査。 ・ 調査結果をもとに対処策検討	

図 5-14 「ModSecurity」障害発生時の対応ポリシー

## (2) 運用手順書の作成

「運用計画」で決めた運用ポリシーをもとに、導入手順書と同様の目的から運用手順書を作成しました。

今回作成した運用手順書には次のような内容を記載しています(図 5-15)。

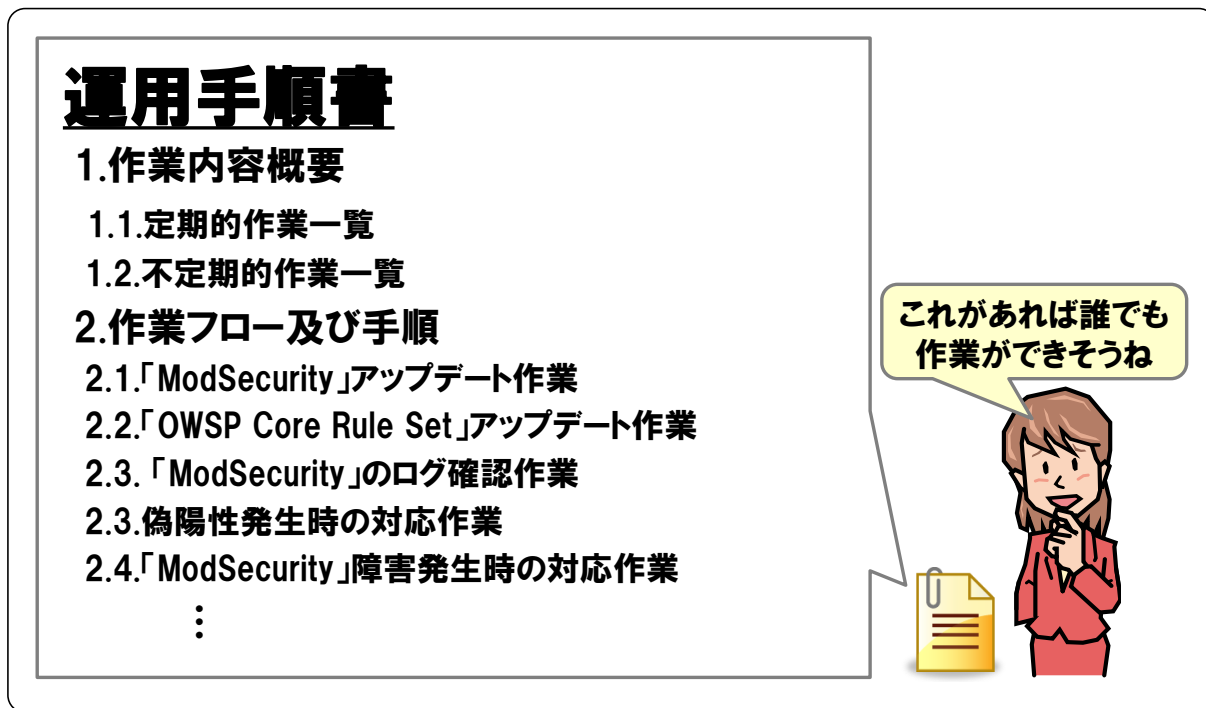


図 5-15 IPA が作成した運用手順書の内容

### 5.3.4. 検証[事例]

「検証」におけるポイントはこちら 4.2.4 章

WAF の導入計画、運用計画を立てた後、テスト環境において検証しながら WAF の導入を進めることとしました。実際に行った検証内容及び検証結果は次の通りです（図 5-16）

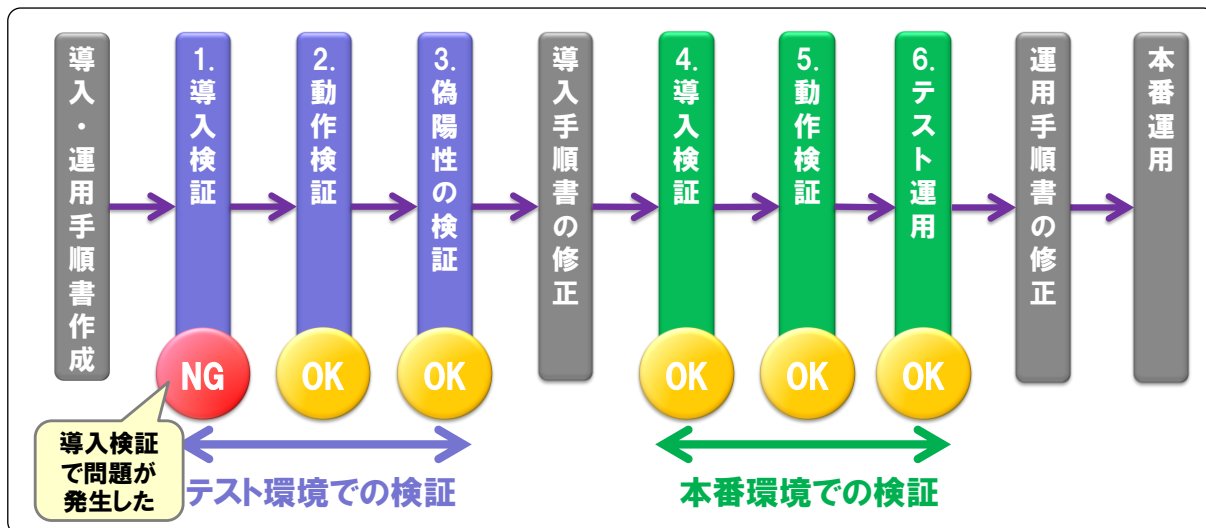


図 5-16 IPA が行った検証内容と検証結果

まずテスト環境で 1.から 3.の 3つの項目を検証しました。そして、検証項目 1.において発生した問題について原因を確認し、本番環境での検証の前に導入手順書を見直しました。続いて、本番環境で 4.から 6.の 3つの項目を検証しました。検証項目 6.において、問題がないことを確認し、最終的に本番運用を開始しました。

以降本項では、図 5-16 の各項目の検証内容、検証結果を紹介します。また、問題が発生した検証項目 1.については、問題への対応も紹介します。

## (1) 導入検証[テスト環境]

### ◆ 検証内容

テスト環境(仮想環境)上に構築した、「JVN iPedia」に導入手順書に基づき「ModSecurity」、をインストールします。インストール後、「Apache」を再起動し、「Apache」が起動することを確認します。

### ◆ 検証結果

「ModSecurity」をインストール後、「Apache」を再起動したところ、「Apache」が起動しないという問題が発生しました。

### ◆ 対応

「ModSecurity」のメーリングリストのアーカイブ<sup>56</sup>を調査したところ、同様の現象が発生している事が確認できました。記載されていた対処方法に基づき対処<sup>57</sup>したところ、「Apache」が起動することを確認できました。



## (2) 動作検証[テスト環境]

### ◆ 検証内容

「ModSecurity」に設定した「Core Rule Set」で検出可能な文字列(たとえば「@@version」)を「JVN iPedia」の検索フォームに入力して検索します。その結果、検索結果は表示されず遮断されていることを確認します<sup>58</sup>。また、「ModSecurity」の検知ログおよび監査ログが出力されることを確認します。

### ◆ 検証結果

「JVN iPedia」の検索フォームに「ModSecurity」が検出する文字列を入力して検索した結果、検索結果は表示されず遮断されていることを確認できました。また、検知ログおよび監査ログが出力されていることを確認できました。



## (3) 偽陽性の検証[テスト環境]

### ◆ 検証内容

「JVN iPedia」利用者の HTTP 通信を再現し、「JVN iPedia」に送信します。その結果、検知ログを確認し、ModSecurity が通信を遮断しないことを確認します。なお、遮断した場合は監査ログより通信内容を確認します。

<sup>56</sup> [http://sourceforge.net/mailarchive/forum.php?forum\\_name=mod-security-users](http://sourceforge.net/mailarchive/forum.php?forum_name=mod-security-users)

<sup>57</sup> ModSecurity のコンパイルオプションを変更しました。現在は ModSecurity のマニュアルに記載してあります。

<sup>58</sup> 検索結果が表示される場合、ウェブサーバは HTTP レスポンスコード 200 で応答しますが、「ModSecurity」が HTTP 通信を遮断した場合、今回の IPA の設定では HTTP レスポンスコード 403 が応答されます。この検証項目では、HTTP レスポンスコード 403 が応答されることを確認しました。

HTTP 通信の再現にあたり、今回は、「iLogScanner」と実際の「JVN iPedia」のアクセスログを使用しました。具体的には、「JVN iPedia」のアクセスログから「iLogScanner」を用いて攻撃通信と判定されたログを特定しました。次に、アクセスログから攻撃通信と判定されたログを省き、利用者からの通信と思われるログのみ抽出しました。この抽出したログをもとに、利用者の HTTP リクエストを再現し、テスト環境上の「JVN iPedia」へ送信しました。

なお、「JVN iPedia」は GET メソッドによるアクセスのみで構成されているため、HTTP 通信の再現にアクセスログを用いることができました。しかし、POST メソッドによるアクセスを含むウェブアプリケーションにおいては、通常のアクセスログからでは再現されない POST メソッドによるアクセスも再現して偽陽性の検証をする必要があります。

#### ◆ **検証結果**

「JVN iPedia」の一ヶ月分のアクセスログから利用者の HTTP 通信を再現して、偽陽性の発生を検証しました。この結果、検知ログには何も出力されないことを確認しました。そのため、検証時点で設定していた「Core Rule Set」では、利用者が「JVN iPedia」を使っているだけでは偽陽性が 1 件も発生しないと判断しました。



#### ■ **【補足】: 偽陰性の検証**

(3) 「偽陽性の検証[テスト環境]」において、偽陰性の発生有無も合わせて検証しました。

その結果、4 件の偽陰性の発生を確認しました。これは「iLogScanner」と「Core Rule Set」の差異によるものと考えます。検証時点で設定していた「Core Rule Set」では防御できない攻撃もあると判断しました。

この防御できない攻撃に関しては「Core Rule Set」をアップデートしたことで解決しました(5.4.1.(2)を参照)。

なお、「偽陰性の検証」でも、「偽陽性の検証」と同様に「iLogScanner」と実際の「JVN iPedia」のアクセスログを活用しました。具体的には、「iLogScanner」を用いて、「JVN iPedia」のアクセスログから攻撃通信と判定されたログを特定しました。このログをもとに、脆弱性を悪用した攻撃の HTTP リクエストを再現し、「JVN iPedia」へ送信しました。

#### (4) **導入検証[本番環境]**

##### ◆ **検証内容**

テスト環境(仮想環境)での検証と同様に、導入手順書に基づき「JVN iPedia」の本番環境に「ModSecurity」をインストールします。ただし、テスト環境(仮想環境)での検証とは異なり、偽陽性の発生によるサービスへの影響の懸念から「ModSecurity」の設定<sup>59</sup>を変更し、攻撃を検出しても遮断しないようにします。「ModSecurity」のインストール後、「Apache」を再起動し、起動することを確認します。

<sup>59</sup> 「ModSecurity」における SecRuleEngine の設定を DetectionOnly に変更しました。この設定に関しては、「付録 A. オープンソースソフトウェアの紹介」を参照してください。

#### ◆ 検証結果

「ModSecurity」のインストール後、「Apache」を再起動したところ、起動することを確認できました。



### (5) 動作検証[本番環境]

#### ◆ 検証内容

「ModSecurity」に設定した「Core Rule Set」で検出可能な文字列（たとえば「@@version」）を「JVN iPedia」の検索フォームに入力して検索します。その結果、「ModSecurity」の検知ログ、監査ログが出力されることを確認します。ただし、テスト環境(仮想環境)での検証とは異なり、検索結果が表示されることを確認します<sup>60</sup>。

#### ◆ 検証結果

「JVN iPedia」の検索フォームに「ModSecurity」が検出する文字列を入力して検索した結果、検知ログおよび監査ログが出力されていることを確認できました。また、検索結果が表示されることを確認できました。



### (6) テスト運用[本番環境]

#### ◆ 検証内容

攻撃を遮断しない設定を有効にした状態で、2週間ほど「ModSecurity」を運用します。このテスト運用期間中は、運用手順書に基づき、日々ログを確認して、偽陽性が発生していないことや攻撃通信を検出していることを確認します。

#### ◆ 検証結果

2週間ほど「ModSecurity」を運用した結果、偽陽性が発生していないことを確認できました。一方で、152件のSQLインジェクション攻撃を検知したことも確認できました。

このテスト運用の結果から、攻撃を遮断しない設定を解除し、「ModSecurity」の本番運用を開始しました。



事前にテスト環境で検証したことで、本番環境へのWAF導入では検証項目1.のインストールにおける問題が生じませんでした。そのため、「JVN iPedia」のサービスが停止することはありませんでした。

<sup>60</sup> この検証時には、「ModSecurity」を遮断しない設定としているため、テスト環境での検証と異なり、攻撃を検知してもHTTPレスポンスコード200が返されます。

## 5.4. 運用

本章では、WAF を導入して運用を開始するまでの 3 つの工程のうち「運用」について導入・運用経験をもとに紹介します。

### 5.4.1. 通常運用 [事例]

「通常運用」におけるポイントはこちら 4.3.1 章

本項では、本書執筆時点までに IPA の通常運用を紹介します。

#### (1) 定期的なログ確認

「運用計画」において作成した運用手順書にもとづき、定期的に「ModSecurity」が出力する検知ログを確認しています。ここでは、検知ログの確認方法および本書執筆時点までの運用実績を紹介します。

##### ◆ 確認方法

IPA は、「iLogScanner」を用いて「ModSecurity」が出力した検知ログから、「JVN iPedia」への攻撃の状況を確認しています。2010 年 8 月に公開したウェブサイト攻撃の検出ツール「iLogScanner V3」では、「Apache」のエラーログファイルを元に「ModSecurity」が検出・遮断した結果を解析する機能を搭載しました。ログの確認には、「iLogScanner」のこの機能を使用すると便利です。

また、あわせて「iLogScanner」の結果と監査ログを照合しています。これは、「ModSecurity」が誤って遮断した HTTP リクエスト（偽陽性）が存在しないことを確認するためです。ログ解析などによる攻撃状況の把握は、実施するセキュリティ対策を立てる上での指針の一つになります。偽陽性の発生をいち早く確認し対処するためや、ウェブサーバへの攻撃を確認するために常日頃からログを確認する習慣をつけることを推奨します。

##### ◆ 運用実績

本書執筆時点までの SQL インジェクション攻撃の検出数および偽陽性の発生に関する実績を紹介합니다(表 5-1)。

表 5-1 SQL インジェクション攻撃の発生及び偽陽性の発生に関する実績

月	SQL インジェクション攻撃の検出数	偽陽性の発生件数
2010 年 9 月	149 件	0 件
10 月	14 件	0 件
11 月	27 件	0 件
12 月	125 件	1 件 <sup>61</sup>

<sup>61</sup> 運用に影響を与える偽陽性ではありませんでした。

また、「ModSecurity」を運用したことで、「iLogScanner」では検出できない POST メソッドで送信されるデータに含まれる攻撃も、「ModSecurity」では検出できることがわかりました。



POST メソッドで送信されるデータを検査対象にするなど、WAF ならではの機能が有効であることを確認できました。

## (2) 「ModSecurity」および「Core Rule Set」のアップデート

運用計画の中で作成した運用手順書にもとづき、定期的に「ModSecurity」、「Core Rule Set」のリリース状況を確認しています。仮に新しいバージョンが公開された場合、アップデート内容を確認するなどして、アップデートの実施を計画します。また、アップデートを実施する際には必ずテスト環境で検証<sup>62</sup>します。

以降本項では、改訂第 2 版の執筆時点までに IPA が取り組んだ「ModSecurity」や「Core Rule Set」のアップデート作業を紹介します。

### ◆ 「Core Rule Set v2.0.8」へのアップデート

「ModSecurity」をテスト運用していた 2010 年 8 月 27 日に、「Core Rule Set v2.0.8」がリリースされました。更新内容を確認したところ、「SQL インジェクション」に関わる検出パターンが変更されていました。そのため、運用手順書に記載されている運用ポリシーにもとづき「Core Rule Set v2.0.8」へのアップデートを検討しました。

アップデートを行う際は、まず導入手順書に基づきテスト環境で「Core Rule Set v2.0.8」を検証しました。導入手順書に基づくテスト環境での検証作業において、問題がないことを確認しましたので、本番環境の「Core Rule Set」を「Core Rule Set v2.0.8」にアップデートしました。

この結果、アップデート前までは検出できなかった一部の SQL インジェクション攻撃を検出・遮断することが可能になりました。



検出パターンをアップデートすることで、WAF の検知性能があがりました。

### ◆ 「Core Rule Set v2.0.10」へのアップデート検討

2010 年 11 月 18 日に「Core Rule Set v2.0.9」、2010 年 11 月 29 日に「Core Rule Set v2.0.10」と新しいバージョンが立て続けにリリースされました。両バージョンとも「SQL インジェクション攻撃」の検出パターンが更新されていました。そのため、運用手順書に記載されている運用ポリシーにもとづき最新バージョンの適用を検討しました。

導入手順書に基づきテスト環境で検証したところ、「Core Rule Set v2.0.10」をそのまま適用した場合、偽陽性が発生する可能性があることがわかりました。具体的には、「●●● and ▲▲▲」といった名前の製品が存在した場合、「JVN iPedia」でこういった製品名を検索すると、その検索通信を「ModSecurity」が遮断してしまいます。このため、本書執筆時点では本番環境の「Core Rule Set」を最新バージョンにアップデートしていません。

<sup>62</sup> すでに本番運用を開始しているため、「5.3.4 検証[事例]」で紹介したテスト運用を行っていません。



## ◆ 「ModSecurity v2.5.13」へのアップデート検討

2010年11月29日に「ModSecurity v2.5.13」がリリースされました。「ModSecurity v2.5.13」の更新内容に、「ModSecurity」の動作そのものに影響を与えると思われる修正が含まれていることを確認しました。そのため、運用手順書に記載されている運用ポリシーにもとづき「ModSecurity v2.5.13」へのアップデートを検討しました。

導入手順書に基づきテスト環境で「ModSecurity v2.5.13」のインストール手順を検証したところ、導入手順書にまとめたインストール手順ではエラーが発生してしまい、正常にインストールできませんでした。本書執筆時点では、「ModSecurity」のメーリングリスト等で回避方法の記載を確認できていません。このため、「ModSecurity v2.5.13」へのアップデートを見合わせています。



「ModSecurity」のアップデートにおいて問題が発生する可能性があります。また、「Core Rule Set」のアップデートにより偽陽性が発生する可能性があります。

### 5.4.2. 緊急対応 [事例]

「緊急対応」におけるポイントはこちら 4.3.2 章

緊急対応として次の2つ対応を想定しています。

- 「ModSecurity」障害発生時の対応
- 偽陽性発生時の対応

本書執筆時点までには、緊急対応は発生しておりません。しかし、これらの問題発生時の対応方法については、運用手順書に記載（5.3.3章で紹介）してあり、これに従って対処を行うこととなっています。

### 5.4.3. 保守 [事例]

「保守」におけるポイントはこちら 4.3.3 章

「ModSecurity」は、オープンソースソフトウェアのWAFであるため、通常、WAFベンダと保守契約を結ばません。このため、「ModSecurity」に障害が発生した場合でも、問い合わせ先はユーザーメーリングリストなど限られています。実際にIPAでも、「ModSecurity」の検証において「Apache」が起動しないといった問題（「5.3.4 検証[事例]」を参照）が発生した際、メーリングリストを参照して解決しました。

IPAの事例はたまたま同様の事例が他のユーザの環境で発生していたため、メーリングリストを調べることで解決できましたが、解決できない問題が発生する場合があります。オープンソースソフトウェアのWAFを導入する際は、この点に十分注意する必要があります。



保守契約がなく、メーリングリスト等から独自に調査する必要がありました。

## 5.5. WAF 導入・運用結果総括

最後にオープンソース WAF「ModSecurity」を IPA が導入・運用をおこなった実績をもとに、IPA が考える「WAF の有用性」について紹介します。また、IPA が考える「WAF 導入時の課題」について、「WAF 導入時の一般的な課題」と、「オープンソース WAF 導入時における課題」と 2 つの観点から紹介します。

### ■ WAF 導入経験から得られた WAF の有用性

今回オープンソース WAF「ModSecurity」を導入し、「SQL インジェクション攻撃」の検出パターンのみを有効にした状態で運用しました。その結果、月々数十件程度の攻撃を検出・遮断しました。脆弱性を狙った攻撃からウェブアプリケーションを防御する手段として WAF が有効であることを確認できました。

また、POST メソッドで送信されてくる攻撃通信を解析し検査する機能など WAF ならではの機能が有効であることも確認できました (5.4.1 章(1))。

### ■ WAF 導入経験から得られた導入時の課題<sup>63</sup>

#### ◆ WAF 導入時の一般的な課題<sup>64</sup>

インストールの失敗や検出パターンをそのまま利用した場合に偽陽性が発生するなど、WAF の課題を確認しました。(5.4.1 章(2))。またこれらの問題は、事前に「ModSecurity」のインストールや偽陽性の発生確認等を検証することで、トラブルを未然に防ぐことができました。

このように、事前に検証することが重要であることがわかりました。

#### ◆ オープンソース WAF 導入時における課題

基本的にサポート契約がないオープンソース WAF「ModSecurity」を利用したため、インストールの失敗や偽陽性が発生した際などには、メーリングリスト等を利用し、独自に調査しました (5.4.3 章)。

このように、導入・運用において、人的コストがかかることを再確認できました。



WAF は、脆弱性対策の一つとして有効である。



一方で、WAF は導入に関するコストだけでなく、運用にも人的コストがかかる。

<sup>63</sup> この課題は検出パターンにブラックリストを採用している「ModSecurity」導入経験に基づき、IPA が考えた課題です。検出パターンにホワイトリストを採用している WAF を利用する場合、この課題がそのまま当てはまらない可能性に留意してください。

<sup>64</sup> これらの課題は、商用 WAF では発生しない可能性があります。

## 付録 A. オープンソースソフトウェアの紹介

この付録では、オープンソースソフトウェアとして提供されている、ModSecurity、WebKnight の概要、導入例を紹介します。



この付録における導入例は、評価環境で実施した一例であり、動作環境により導入手順や設定手順が異なることにご注意ください。

### ModSecurity

#### 概要

「ModSecurity」は、米 Trustware 社<sup>65</sup>が GPLv2 ライセンスのもと提供しているオープンソースソフトウェアです。「ModSecurity」は、ウェブサーバソフトウェア「Apache」のモジュールとして動作します。この付録を執筆した時点の最新版は、「ModSecurity 2.5.13」です。「ModSecurity」の動作環境については、表 A-1 を参照してください。

表 A-1 ModSecurity の動作環境

項目名	値
対象 OS	各種 Unix、Windows
対象ウェブサーバ	Apache 2.x <sup>66</sup>

<sup>65</sup> <http://www.trustware.com/>

<sup>66</sup> ModSecurity 2.x は、Apache 1.x を対象としていません。

## 導入の流れ

---

この項では、「ModSecurity 2.5.13」を表 A-2 の環境に導入する流れを紹介します。「ModSecurity」の細かな設定等については、開発元が提供しているドキュメント<sup>67</sup>を参照してください。

表 A-2 ModSecurity のテスト環境

項目名	値
OS	CentOS release 5.5 (Final) Kernel 2.6.18-194.26.1.el5
ウェブサーバ	Apache 2.2.17

### (1) ダウンロード

「ModSecurity」は、以下のウェブサイトからダウンロードできます。

ダウンロードサイト：<http://www.modsecurity.org/download/>

### (2) インストール

ここでは、「ModSecurity」をソースファイルからコンパイルしてインストールします。

「ModSecurity」を動作させるためには、別途必要なソフトウェアがあり、事前にインストールする必要があります。ここでは、前提ソフトウェアのインストール方法については割愛します。

#### ■ 前提ソフトウェア

- mod\_unique\_id
- libapr
- libapr-util
- libpcre
- libxml2
- liblua 5.1.x
- libcurl 7.15.1 又はそれ以上

---

<sup>67</sup> <http://www.modsecurity.org/documentation/index.html>

前提ソフトウェアをインストールした後、「ModSecurity」のインストールを実施します<sup>68</sup>。これらの作業は、便宜上全て root ユーザで行います。

```
# tar xvfz modsecurity-apache_2.5.13.tar.gz
# cd modsecurity-apache_2.5.13/apache2/
# ./configure
# make
# make test
# make install
```

次に、「ModSecurity」を使用するために「Apache」の設定変更を行います。

「Apache」の設定ファイル (httpd.conf) に下記内容を追記します。

```
# vi /usr/local/httpd/conf/httpd.conf
Include conf/extra/httpd-modsecurity.conf
```

### (3) 設定

インストールが完了しましたら、「ModSecurity」の設定を行います。

まず、「ModSecurity」を使用するために、検出パターンファイルを設定します。ここでは、無償で公開されている「Core Rule Set」を使用します。

「Core Rule Set」は、「ModSecurity」に同梱されていますので、適した場所（ここでは、/usr/local/modsecurity2 以下）へコピーします。

```
# mkdir /usr/local/modsecurity2
# cp -r rules /usr/local/modsecurity2
```

次に、「Apache」の「ModSecurity」用の設定ファイル (httpd-modsecurity.conf) として下記内容を記述します。今回は検出パターンのうち、「SQL インジェクション」に関する検出パターンのみ使用します。

```
# vi /usr/local/httpd/conf/extra/httpd-modsecurity.conf
LoadFile /usr/local/libxml2/lib/libxml2.so
LoadFile /usr/local/lua/lib/liblua5.1.so
LoadModule security2_module modules/mod_security2.so

Include /usr/local/modsecurity2/rules/modsecurity_crs_10_config.conf
Include /usr/local/modsecurity2/rules/base_rules/modsecurity_crs_41_sql_injection_attacks.conf
```

また、「ModSecurity」の設定ファイル (modsecurity\_crs\_10\_config.conf) として、下記に内容を記述し、「Apache」を再起動します。

<sup>68</sup> バージョンによっては、Apache の configure 時に、「--with-pcre=」オプションが必要になる場合があります。詳しくは、ModSecurity のインストールマニュアルを参照してください。

```
# vi /usr/local/modsecurity2/rules/modsecurity_crs_10_config.conf
SecComponentSignature "core ruleset/2.0.10"

SecRuleEngine On
SecDefaultAction "phase:2,deny,log"

SecAuditEngine On
SecAuditLogRelevantStatus "^(?:5|4(?:!04))"
SecAuditLogType Serial
SecAuditLog /var/log/httpd/modsec_audit.log
SecAuditLogParts "ABIFHKZ"

SecDebugLog /var/log/httpd/modsec_debug.log
SecDebugLogLevel 3
```

参考までに「ModSecurity」の設定ファイルの主な設定項目を表 A-3 にまとめます。詳細については、「ModSecurity」のドキュメントを参照してください。

表 A-3 ModSecurity の主な設定項目

設定項目	説明	補足
SecRuleEngine	動作モード	On - 遮断 Off - 無効 DetectionOnly - 検知のみ
SecDefaultAction	デフォルトアクションの設定	検出パターンに合致したときの ModSecurity の動作を定義する。なお、検出パターン (SecRule) による個別設定が優先されます。
SecAuditEngine	監査ログの動作モード	On - 全て記録 Off - 無効 RelevantOnly - SecAuditLogRelevantStatus に合致したステータスコードのみを記録
SecAuditLogRelevantStatus	監査ログの記録対象とするサーバ応答コード	SecAuditEngine が RelevantOnly の場合
SecAuditLog	監査ログの出力場所 (ファイル名含む)	-
SecAuditLogType	監査ログの出力タイプ	Serial - 1 ファイルに記載 Concurrent - セッション毎に個別ファイルを生成
SecAuditLogStorageDir	個別ファイルの出力先ディレクトリ	SecAuditLogType が Concurrent の場合に設定 Serial 設定の場合はコメントアウト
SecAuditLogParts	監査ログの出力項目 (AZ は必須)	A - AuditLog ヘッダー B - リクエストヘッダー C - リクエストボディ D - Reserved E - レスポンスボディ F - レスポンスヘッダー G - Reserved H - 追加情報。パターンにマッチしたアクセ

		スだとここにタグが付与される。 I - ファイルを除外した、コンパクトなリクエストボディ J - Reserved K - トランザクションにマッチした全てのルール Z - 最後の境界線
SecDebugLog	デバッグログの出力場所 (ファイル名含む)	-
SecDebugLogLevel	デバッグログに出力するログレベル	0 - no logging. 1 - errors (intercepted requests) only. 2 - warnings. 3 - notices. 4 - details of how transactions are handled. 5 - as above, but including information about each piece of information handled. 9 - log everything, including very detailed debugging information.

「ModSecurity」が出力するログファイルを事前に作成しておきます。

```
# touch /var/log/httpd/modsec_audit.log
# touch /var/log/httpd/modsec_debug.log
```

これらのログファイルについては、ログ管理の設定をしておくとう便利です。「5 IPA における WAF 導入・運用事例」における IPA のログ管理の設定を例として、紹介します。

```
# vi /etc/logrotate.d/httpd
/var/log/httpd/modsec_audit.log /var/log/httpd/modsec_debug.log {
    weekly
    compress
    rotate 5
    create 600 httpd httpd
    missingok
    postrotate
        /bin/kill -usr1 `cat /var/log/httpd/httpd.pid 2> /dev/null` 2> /dev/null || true
    endscript
}

# /etc/rc.d/init.d/crond restart
```

## (4) 検証

「ModSecurity」の運用を開始する前に、検出パターンに合致した HTTP 通信を「ModSecurity」が遮断しないように設定し、十分な検証を行います。「ModSecurity」の設定ファイル (mod\_security\_crs\_10.config.conf) を下記に変更し、「Apache」を再起動します。

```
# vi /usr/local/modsecurity2/rules/modsecurity_crs_10_config.conf  
  
SecRuleEngine On  
  
↓  
  
SecRuleEngine DetectionOnly
```

この設定を行うことで、「ModSecurity」は実際に遮断せず検知のみを行います。「ModSecurity」でパターンファイルが有効になっているか確認するため、ブラウザから「ModSecurity」をインストールしたウェブサーバに対して、以下の URL でアクセスします。このアクセスにおいて、パラメータ「id」に「SQL インジェクション」の脆弱性を悪用した場合に含まれる「and 1=1;--」という文字列を設定しています。この文字列は (3) で設置した検出パターンファイルで検知できるものです。

```
http://ウェブサーバの IP アドレス/example.html?id=and 1=1;--
```

上記のようなアクセスを行うと、「ModSecurity」は「SQL インジェクション」の脆弱性を悪用した攻撃と判定して、ログを出力します。「ModSecurity」の初期設定では、ログは下記のログファイルに記録されます。このログファイルにて、「and 1=1;--」が検知できているか確認します。

```
# tail /var/log/httpd/error_log  
[Thu Dec 09 19:44:36 2010] [error] [client 192.168.0.1] ModSecurity: Warning. Pattern match  
"%b(%) (?=:<>|<=>|<>|=) ?%1%b[%"'"' %c2%b4%%e2%x80%x99%%e2%x80%x98](%d+)[  
%"'"' %c2%b4%%e2%x80%x99%%e2%x80%x98] (?=:<>|<=>|<>|=) ?%"'"' %c2%b4%%e2%x80%x99%%e2%x80%x98](%w+)[%"'"' %c2%b4%%  
e2%x80%x99%%e2%x80%x98] ?=:<>|<=>|<>|=) ?%"'"' %c2%b4%%e2%x80%x99%%e2%x80%x99%%e2%x80%  
0%x98]%"'"' %c2%b4%%e2%x80%x99%%e2%x80%x98)*"?%s+(and|or)%s+(%s%"'"'  
%"'"' ..." at ARGS:id. [file  
"/usr/local/modsecurity2/rules/base_rules/modsecurity_crs_41_sql_injection_attacks.conf"] [line "425"]  
[id "950901"] [rev "2.0.10"] [msg "SQL Injection Attack"] [data " and 1=1"] [severity "CRITICAL"]  
[hostname "192.168.0.139"] [uri "/"] [unique_id "TQCzFH8AAAEAAAeVCf8AAAAA"]
```

前述した確認方法はあくまでも例となります。実際の HTTP 通信のログ出力結果を参考に、「遮断されてはいけない HTTP 通信が遮断されていないか？」(偽陽性)、「遮断しなければいけない HTTP 通信が通過していないか？」(偽陰性)などを確認します。問題があった場合、「ModSecurity」の設定ファイルや検出パターンファイルを見直します。この検証作業を偽陽性などの問題が発生しなくなるまで繰り返し行います。



## (5) 運用

検証で問題が発生しないことを確認したら、実際に「ModSecurity」の運用を開始します。運用を開始するために、「ModSecurity」の設定ファイルを下記に変更し、「Apache」を再起動してください。

```
# vi /usr/local/modsecurity2/rules/modsecurity_crs_10_config.conf
SecRuleEngine DetectionOnly
↓
SecRuleEngine On
```

本設定変更を行うことで、「ModSecurity」は設定した検出パターンファイルに基づき、実際にウェブアプリケーションに対する攻撃を遮断します。

## (6) アンインストール

「ModSecurity」のマニュアルでは、アンインストール手順を記載しておりません。ここでは、「ModSecurity」を無効化する手順を紹介します。

```
# vi /usr/local/modsecurity2/rules/modsecurity_crs_10_config.conf
SecRuleEngine On
↓
SecRuleEngine Off
```

これにより、「ModSecurity」は、遮断・検知ともに行わない設定になり、ウェブサーバへの影響を最小限に抑えることができます。

または、下記のように設定ファイルをコメントアウトすることでも無効化できます。

```
# vi /usr/local/httpd/conf/httpd.conf
Include conf/extra/httpd-modsecurity.conf
↓
#Include conf/extra/httpd-modsecurity.conf
```

## (7) TIPS:「iLogScanner V3.0」との連携

(4)の検証で説明をしたように「ModSecurity」の検知結果はログファイルを目視で確認する必要があります。しかしながら、IPAが公開している「iLogScanner V3.0」は、「Core Rule Set」を少し変更することで、「Apache」のエラーログファイルを元に「ModSecurity」が検知・遮断したデータを解析できます。

「iLogScanner」で「ModSecurity」の検知・遮断したデータを解析するためには、「Core Rule Set」の検出パターンに脆弱性に対応した"tag"名称を付与するだけです。例えば、「SQLインジェクション」に対応した"tag"名称は、以下のようになります。

```
tag:'WEB_ATTACK/SQL_INJECTION'
```

「SQL インジェクション」の"tag"名称を「Core Rule Set」に付与する例を紹介합니다。(4)の検証で使った「SQL インジェクション」の検出パターンに"tag"名称を付与してみましょう。この検出パターンは検出パターンファイルの 424,425 行目にありました。

```
424 行目  SecRule REQUEST_FILENAME|ARGS_NAMES|ARGS|XML:/* "\b(%d+) ?(?:=<>|<=>|<|>|=)
        ?%1%b|[%"'"/\:\;\.\-\_]{1,4}(%d+)[%"'"/\:\;\.\-\_]?(?:=<>|<=>|<|>|=) ?[%"'"/\:\;\.\-\_]{2,4}b|[%"'"/\:\;\.\-\_]{1,4}(%w+)[%"'"/\:\;\.\-\_]?(?:=<>|<=>|<|>|=) ?[%"'"/\:\;\.\-\_]{3,4}b|([%"'"/\:\;\.\-\_]{1,4})?%s+(and|or)%s+([%"'"/\:\;\.\-\_]{1,4})?%s+([%"'"/\:\;\.\-\_]{1,4})?%w+([%"'"/\:\;\.\-\_]{1,4})?"
        "phase:2,rev:'2.0.10',capture,multiMatch,t:none,t:urlDecodeUni,t:htmlEntityDecode,t:replac
425 行目  eComments,t:compressWhiteSpace,t:lowercase,ctl:auditLogParts+=E,block,msg:'SQL Injection
        n Attack',id:'950901',logdata:'%(TX.0)',severity:'2',setvar:tx.msg=%{rule.msg}',setvar:tx.sql_injection_score=+%{tx.critical_anomaly_score},setvar:tx.anomaly_score=+%{tx.critical_anomaly_score},setvar:tx.%{rule.id}-WEB_ATTACK/SQL_INJECTION-%{matched_var_name}=%{tx.0}"
```

この検出パターンを以下のように変更します。赤字が変更箇所となります。

```
424 行目  SecRule REQUEST_FILENAME|ARGS_NAMES|ARGS|XML:/* "\b(%d+) ?(?:=<>|<=>|<|>|=)
        =) ?%1%b|[%"'"/\:\;\.\-\_]{1,4}(%d+)[%"'"/\:\;\.\-\_]?(?:=<>|<=>|<|>|=) ?[%"'"/\:\;\.\-\_]{2,4}b|[%"'"/\:\;\.\-\_]{1,4}(%w+
        +)[%"'"/\:\;\.\-\_]?(?:=<>|<=>|<|>|=) ?[%"'"/\:\;\.\-\_]{3,4}b|([%"'"/\:\;\.\-\_]{1,4})?%s+(and|or)%s+
        ([%"'"/\:\;\.\-\_]{1,4})?%s+([%"'"/\:\;\.\-\_]{1,4})?%w+([%"'"/\:\;\.\-\_]{1,4})?"
        "phase:2,rev:'2.0.10',capture,multiMatch,t:none,t:urlDecodeUni,t:htmlEntityDecode,t:repl
425 行目  aceComments,t:compressWhiteSpace,t:lowercase,ctl:auditLogParts+=E,block,msg:'SQL Inje
        ction Attack', tag:'WEB_ATTACK/SQL_INJECTION',id:'950901',logdata:'%(TX.0)',severity:'2
        ',setvar:tx.msg=%{rule.msg}',setvar:tx.sql_injection_score=+%{tx.critical_anomaly_score},set
        var:tx.anomaly_score=+%{tx.critical_anomaly_score},setvar:tx.%{rule.id}-WEB_ATTACK/SQL
        _INJECTION-%{matched_var_name}=%{tx.0}"
```

変更が終わりましたら、(4)で試したリクエストでアクセスを行うと、「ModSecurity」が「SQL インジェクション」の脆弱性を悪用した攻撃と判定して、ログを出力します。

```
# tail /var/log/httpd/error_log
[Thu Dec 09 20:11:18 2010] [error] [client 192.168.0.1] ModSecurity: Warning. Pattern match "\b(%d+)
+)?(?:=<>|<=>|<|>|=) ?%1%b|[%"'"/\:\;\.\-\_]{1,4}(%d+)[%"'"/\:\;\.\-\_]?(?:=<>|<=>|<|>|=) ?[%"'"/\:\;\.\-\_]{2,4}b|[%"'"/\:\;\.\-\_]{1,4}(%w+
+)[%"'"/\:\;\.\-\_]?(?:=<>|<=>|<|>|=) ?[%"'"/\:\;\.\-\_]{3,4}b|([%"'"/\:\;\.\-\_]{1,4})?%s+(and|or)%s+([%"'"/\:\;\.\-\_]{1,4})?%s+([%"'"/\:\;\.\-\_]{1,4})?%w+([%"'"/\:\;\.\-\_]{1,4})?"
"phase:2,rev:'2.0.10',capture,multiMatch,t:none,t:urlDecodeUni,t:htmlEntityDecode,t:replaceComments,t:compressWhiteSpace,t:lowercase,ctl:auditLogParts+=E,block,msg:'SQL Injection Attack', tag:'WEB_ATTACK/SQL_INJECTION',id:'950901',logdata:'%(TX.0)',severity:'2',setvar:tx.msg=%{rule.msg}',setvar:tx.sql_injection_score=+%{tx.critical_anomaly_score},setvar:tx.anomaly_score=+%{tx.critical_anomaly_score},setvar:tx.%{rule.id}-WEB_ATTACK/SQL_INJECTION-%{matched_var_name}=%{tx.0}"
[line "425"] [id "950901"] [rev "2.0.10"] [msg "SQL Injection Attack"] [data " and 1=1"] [severity "CRITICAL"] [tag "WEB_ATTACK/SQL_INJECTION"] [hostname "192.168.0.139"] [uri "/" ] [unique_id "TQC5Vn8AAAEAAAqOoPcAAAA"]
```

上記の様に、「tag "WEB\_ATTACK/SQL\_INJECTION"」が出力されていれば設定完了です。

# WebKnight

---

## 概要

---

WebKnightは、AQTRONIX社がGPLのもと提供しているオープンソースソフトウェアです。WebKnightは、WindowsのInternet Information Service(IIS)のISAPIフィルタとして動作します。この付録を執筆した時点の最新版は、WebKnight 2.2です。WebKnightの動作環境については、表A-4を参照してください。

表 A-4 WebKnight の動作環境

項目名	値
対象 OS	Windows
対象ウェブサーバ	IIS 5.0, 6.0, 7.0 <sup>69</sup> ISAPI フィルタをサポートするウェブサーバ

---

<sup>69</sup> ISAPI フィルタのインストールが必要です。

## 導入の流れ

---

この項では、WebKnight 2.2 を表 A- 5 の環境に導入する流れを紹介します。WebKnight の細かな設定等については、開発元が提供しているドキュメント<sup>70</sup>を参照してください。

表 A- 5 WebKnight のテスト環境

項目名	値
OS	Windows Server 2003
ウェブサーバ	IIS 6.0
備考	IIS は IIS5.0 プロセス分離モードで実行する

### (1) ダウンロード

WebKnight は、AQTRONIX 社のウェブサイトからダウンロード可能です。

ダウンロードサイト：<http://www.aqtronix.com/?PageID=99#Download>

---

<sup>70</sup> <http://www.aqtronix.com/?PageID=99>

## (2) インストール

WebKnight のインストールには、次の3つの方法があります。

- Windows インストーラ「WebKnight.msi」
- インストール用 VB スクリプト「install.vbs」
- 手動インストール

ここでは、Windows インストーラを利用し、グローバルフィルタとして WebKnight をインストールします。(1) でダウンロードしたファイルの中にある「WebKnight.msi」をダブルクリックし、インストールを行ないます(図 A-1)。正常にインストールを完了した後、IIS を再起動すると、ISAPI フィルタに WebKnight が追加され、正常に動作していることが分かります(図 A-2)。

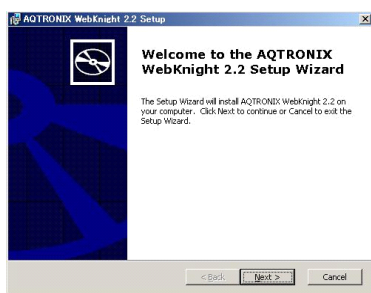


図 A-1 Windows インストーラ

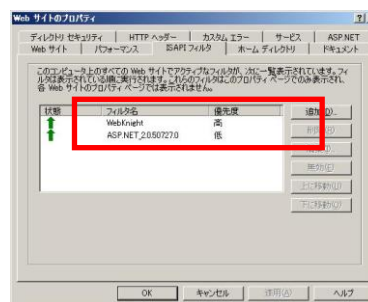


図 A-2 IIS の ISAPI フィルタ

「WebKnight.msi」

## (3) 設定

WebKnight の設定は、「WebKnight Configuration」ツールを利用して行います。

Windows のスタートメニューから「すべてのプログラム」, 「AQTRONIX WebKnight」とメニューを開いていき、「WebKnight Configuration」を実行します。「Open Configuration」ウインドウが起動したら、「WebKnight.xml」を選択し、[OK]ボタンをクリックします(図 A-3)。



図 A-3 「Open Configuration」ウインドウ

「WebKnight Configuration」ツールが起動します(図 A-4)。参考までに「WebKnight Configuration」ツールの設定項目の概要を表 A-6 にまとめています。

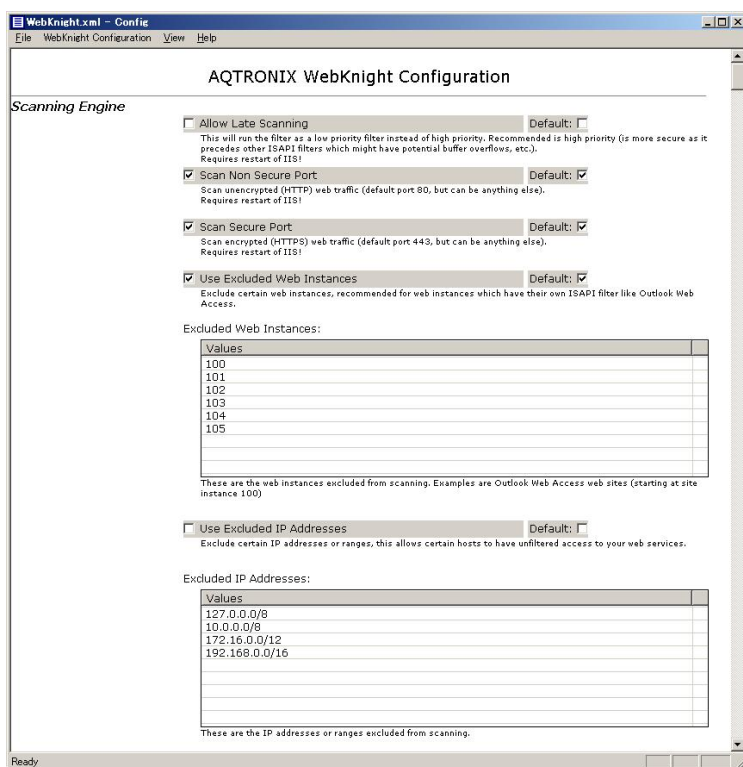


図 A-4 「WebKnight Configuration」 ツール

表 A-6 「WebKnight Configuration」 ツール設定項目の概要

カテゴリ	内容	備考
Scanning Engine	HTTP/HTTPSに対するスキャンの可否、フィルタ対象外 IP アドレスなどを設定	
Incident Response Handling	不正アクセスを検知した場合の動作を設定	
Logging	ログ機能の可否、ログの取得内容などを設定	
Connection	監視する IP アドレスや、アクセスを遮断する IP アドレスなどを設定	
Authentication	アカウントブルートフォースアタックなど認証に関する内容を設定	
Request Limits	コンテンツ、URL、クエリストリングスなど制限をかけるリクエスト長を設定	
URL Scanning	URL で拒否する文字列などの拒否項目の設定	”URL Denied Sequence”の内容で自サイトのアクセスを遮断することがないか要確認
Mapped Path	ディレクトリトラバーサル攻撃などで拒否する文字列、アクセスを許可するパスを設定	”Allowed Paths”については、下記注意参照
Requested File	拒否するファイルの文字列や拡張子などを設定	自サイトで許可しているファイルを遮断する可能性がある所以要確認
Robots	ロボットに対する遮断動作について設定	
Headers	サーバヘッダの変更や遮断するヘッダの内容について設定	

ContentType	リクエストヘッダの Content-Type のチェックの可否などを設定	
Cookie	Cookie に対する拒否項目の設定	
User Agent	ブラウザなどのユーザエージェントに対する拒否項目の設定	
Referrer	Referrer に対するスキャンの可否や特定ドメインからのホットリンクの制限などを設定	
Methods	許容または拒否するメソッドを設定	
QueryString	拒否するクエリストリングの文字列などを設定	
Global Filter Capabilities	グローバルフィルタの適用可否、POST データに対する拒否項目の設定	
SQL Injection	SQL インジェクションに利用されるキーワードの設定	
Web Applications	ウェブアプリケーションの許容について設定	

設定時には、次のことに注意してください。

- “Mapped Path” の “Allowed Paths” にウェブサイトのパスを必ず指定する必要があります。ここにパスの指定がないと、そのサイトへの通信が遮断されます。
- “Requires restart of IIS” の記載がある設定を変更した場合は、IIS の再起動が必要になります。
- Post データの検査は、デフォルトでは未実施であるため、必要に応じて “Global Filter Capabilities” にある Postdata 関連の設定を有効にします。

#### (4) 検証

偽陽性、偽陰性が発生していないか検証します。検証の際の手順として、一例を以下に示します。

##### ① 「Response Log Only」の設定

WebKnight の設定に問題がないか確認するため、まずは、通信を遮断しないように、Response Log Only モードに設定します。これにより、WebKnight で異常と検知した通信についてはログに記録されますが、通信自体はそのまま継続されます。

Response Log Only モードの設定方法は、「WebKnight Configuration」の「Incident Response Handling」カテゴリ内の「Response Log Only」にチェックを入れます。

② ログの確認 (WebKnight)

WebKnight で異常と検出されたログを確認し、正常な通信が検知されていないか確認します。ログは次のどちらの方法でも確認することができます。

- WebKnight 付属ツール「Log Analysis」(図 A-5) を使用する

Windows のスタートメニューから「すべてのプログラム」, 「AQTRONIX WebKnight」とメニューを開いていき、「Log Analysis」を実行することで、「Log Analysis」を起動できます。

- WebKnight のログファイルを直接閲覧する

初期設定であれば、ログファイルは C:\Program Files\AQTRONIX WebKnight\Log Files 配下に日付毎に作成されます。

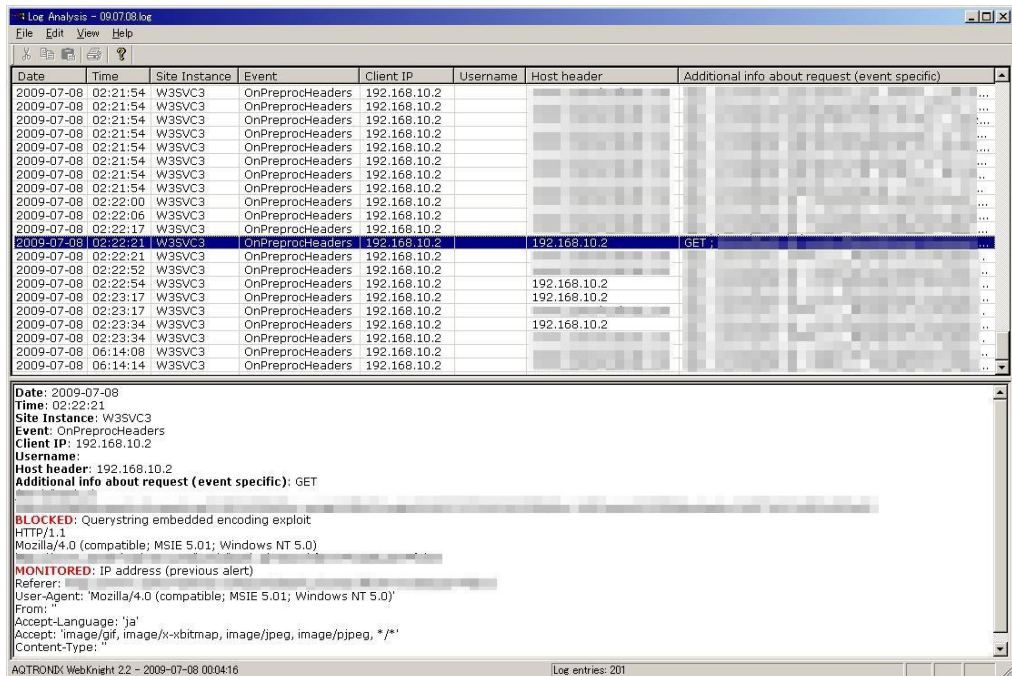


図 A-5 「Log Analysis」

③ ログの確認 (IIS)

IIS のログを確認し、WebKnight で検出されなかった攻撃がないか確認します。

②、③の結果、偽陽性、偽陰性が発生している場合は、再度 (3) に戻り、設定の見直しを実施します。偽陽性、偽陰性が生じなくなるまで、この手順を繰り返します。



## (5) 運用

(4) において問題が発生しないことが確認できたら運用に入ります。その場合、不正な通信の検出後の処理を決定しておきます。ここでは、(4) で設定した「Response Log Only」の設定を解除し、「不正を検出したらエラー画面を返す」処理に変更します。この場合、エラー画面をカスタマイズすることも WebKnight 初期設定のエラー画面を返すこともできます。

### ① 「Response Log Only」の解除

「WebKnight Configuration」の「Incident Response Handling」カテゴリ内の「Response Log Only」のチェックを外します。

### ② エラー画面の変更

WebKnight の初期設定<sup>71</sup>では、不正な通信を検知した場合、WebKnight が直接ブラウザに図 A-6 のエラー画面を送信します。

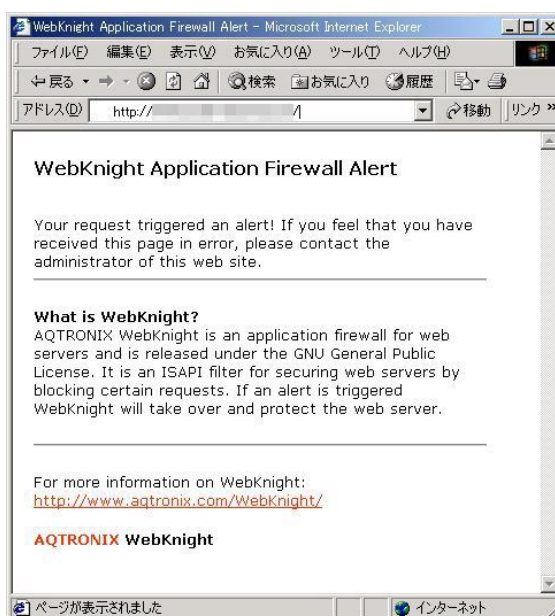


図 A-6 WebKnight 初期設定のエラー画面

エラー画面をカスタマイズする場合は、「WebKnight Configuration」の「Incident Response Handling」カテゴリ内の「Response Directly」を無効にし、「Response Redirect」を有効にします。そして、「Response Redirect URL」に、用意したエラー画面のパスを指定します（図 A-7）。

なお、「Response Redirect」で指定するエラー画面パスは、「WebKnight Configuration」の「Mapped Path」カテゴリ内の「Allowed Paths」で指定する必要があります。

<sup>71</sup> 「WebKnight Configuration」の「Incident Response Handling」カテゴリ内の「Response Directly」が有効になっている場合を指します。

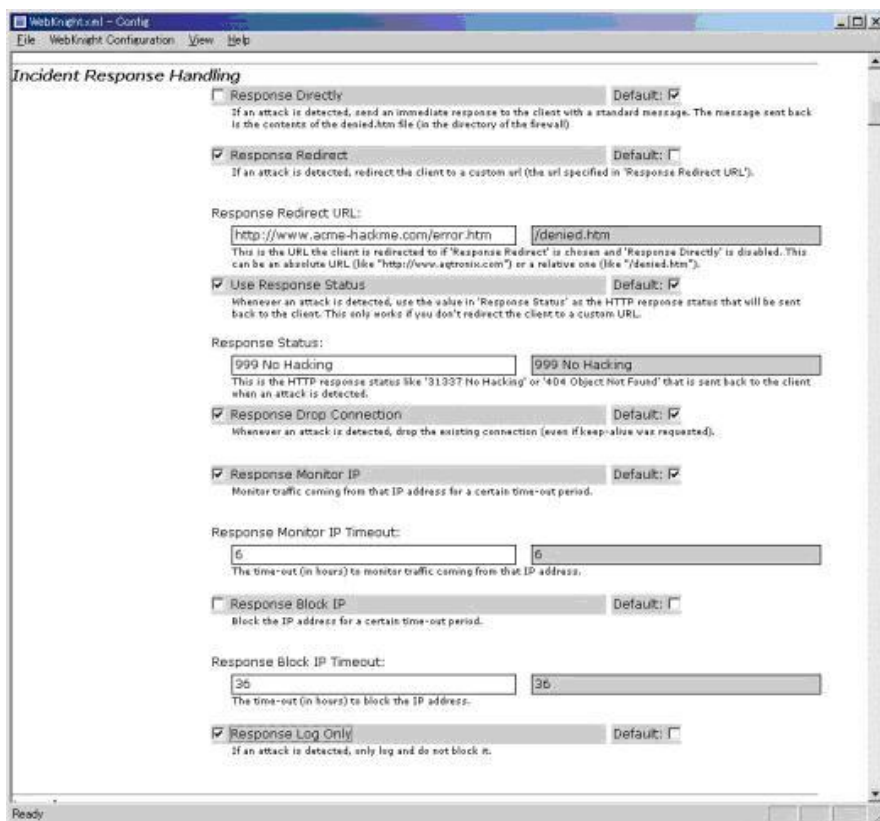


図 A-7 "Response Redirect"の設定

### ③ ログの確認 (IIS)

「Response Log Only」の設定を外した場合、エラーの応答方法により IIS へ出力されるログが異なります。

以下に「SQL インジェクション」の脆弱性を悪用する攻撃で記録されたログを示します。パラメータ「param」に SQL 文が記録されていることが分かります。

#### Response Directly で運用した場合

WebKnight のログ

```
2009-07-10 ; 08:42:30 ; W3SVC3 ; OnPreprocHeaders ; 192.168.10.2 ; ; 192.168.10.2 ; GET ; <URI> ;
param=SELECT+*+FROM+Users+WHERE+&submit=%83T%81%5B%83%60 ; BLOCKED: Possible SQL injection in
querystring ; HTTP/1.1 ; Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0) ; <Referer>
```

IIS のログ

なし

## Response Redirect で運用した場合

### WebKnight のログ

```
2009-07-10 ; 08:52:49 ; W3SVC3 ; OnPreprocHeaders ; 192.168.10.2 ; ; 192.168.10.2 ; GET ; <URI>;  
param=SELECT+*+FROM+Users+WHERE+&submit=%83T%81%5B%83%60 ; BLOCKED: Possible SQL injection in  
querystring ; HTTP/1.1 ; Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0) ; <Referer>;
```

### IIS のログ

```
2009-07-10 08:52:49 192.168.10.2 - W3SVC3 WIN2K-SVR 192.168.10.2 80 - - - 302 0 141 414 10 HTTP/1.1  
192.168.10.2 Mozilla/4.0+(compatible;+MSIE+5.01;+Windows+NT+5.0) - <Referer>
```

## (6) アンインストール

WebKnight のアンインストールには、インストール時と同様に 3 つの方法があります。ここでは、OS の機能を利用したアンインストール方法について説明します。また、アンインストール後は、IIS を再起動します。

### ◆ Windows の「プログラムの追加と削除」からアンインストールする


OS の機能である「プログラムの追加と削除」を利用します。「AQTRONIX WebKnight 2.2」を選択し、「削除」をクリックすることで、アンインストールが実行されます。

なお、WebKnight のアンインストール後、ログの情報（初期設定では、C:¥Program Files¥AQTRONIX WebKnight ¥LogFiles）が残ります。不要であれば、OS の再起動後に該当フォルダを削除してください。

## 付録 B. 商用製品の紹介

本付録では、WAF 製品・サービスの利用推進を目的として、本書の作成にご協力いただいた企業の WAF 製品・サービスを開発元企業名の「あいうえお順」に掲載します。本付録に掲載している WAF 製品・サービスについては、各製品の開発元企業、サービスの提供元企業にお問い合わせください。


### 第 1 版 第 1 刷で掲載した企業

	
開発元企業	株式会社ジェイピー・セキュア
URL	<a href="http://www.jp-secure.com/cont/products/">http://www.jp-secure.com/cont/products/</a>

	
提供元企業	株式会社セキュアスカイ・テクノロジー
URL	<a href="http://www.scutum.jp/index.html">http://www.scutum.jp/index.html</a>

	
開発元企業	株式会社ソリトンシステムズ
URL	<a href="http://www.soliton.co.jp/products/net_security/netattest/waf/index.html">http://www.soliton.co.jp/products/net_security/netattest/waf/index.html</a>

	
開発元企業	日本電気株式会社
URL	<a href="http://www.nec.co.jp/soft/siteshell/">http://www.nec.co.jp/soft/siteshell/</a>

	
開発元企業	Barracuda Networks, Inc.
URL	<a href="http://www.barracudanetworks.com/ns/products/web-site-firewall-overview.php">http://www.barracudanetworks.com/ns/products/web-site-firewall-overview.php</a>



**Webセキュリティ on-Demand**  
**Webアプリケーションファイアウォール**

提供元企業	株式会社日立システムズ
URL	<a href="http://www.hitachi-systems.com/solution/s005/web/index.html">http://www.hitachi-systems.com/solution/s005/web/index.html</a>

**第 1 版 第 2 刷で掲載した企業**



開発元企業	Imperva Inc.
URL	<a href="http://www.imperva.jp/products/web_application_security.asp">http://www.imperva.jp/products/web_application_security.asp</a>



開発元企業	Citrix Systems, Inc
URL	<a href="http://www.citrix.co.jp/products/cns/how-it-works/firewall.html">http://www.citrix.co.jp/products/cns/how-it-works/firewall.html</a>

**第 1 版 第 3 刷で掲載した企業**



開発元企業	株式会社ネットファイア
URL	<a href="http://www.netfire.jp/product.html">http://www.netfire.jp/product.html</a>

## 第2版 第2刷で掲載した企業



IT agility. Your way.

開発元企業	F5 Networks, Inc.
URL	<a href="http://www.f5networks.co.jp/product/bigip/asm/index.html">http://www.f5networks.co.jp/product/bigip/asm/index.html</a>



開発元企業	Penta Security Systems, Inc.
URL	<a href="http://www.pentasecurity.co.jp/jpn/product/webWppleIntro.do">http://www.pentasecurity.co.jp/jpn/product/webWppleIntro.do</a>

# 用語集

## HTTP(Hypertext Transfer Protocol)

ウェブサーバとウェブブラウザがデータを送受信するために使われるプロトコル。

## HTTPS 通信

本書では、SSL(Secure Socket Layer)やTLS(Transport Layer Security)を用いて暗号化した HTTP 通信を指す。

## SSL(Secure Socket Layer)

インターネット上で情報を暗号化して送受信するプロトコル。個人情報やクレジットカード番号などを安全に送受信できる。オンラインバンキングなどで利用されている。

## SQL (Structured Query Language)

リレーショナルデータベース (RDB) において、データベースの操作やデータの定義を行うための問い合わせ言語。SQL 文には、CREATE 文などでデータの定義を行う DDL (Data Definition Language: データ定義言語) や SELECT 文、UPDATE 文、GRANT 文などでデータベース操作やアクセス権限の定義を行う DML (Data Manipulation Language: データ操作言語) などがある。

## ウェブアプリケーション

ウェブサイトで稼動するシステム。一般に、Java, ASP, PHP, Perl などの言語を利用して開発され、サイトを訪れた利用者に対して動的なページの提供を実現している。

## ウェブサイト

特定のドメイン(例: <http://www.ipa.go.jp/>)を構成する要素のまとまりを指す。ウェブサイトを構成する要素には、ウェブページやウェブアプリケーション、それらが動作するウェブサーバやデータベースが動作するデータベース・サーバ等がある。

## ウェブサーバ

ウェブページやウェブアプリケーションが動作するソフトウェア、または物理的なサーバ機器を指す。本書においては、ことわりなくウェブサーバと言った場合、物理的なサーバ機器を指す。

## オープンソース

ソフトウェアのソースコードが公開されており、再頒布の自由が可能であること。

## 脆弱性

ウェブアプリケーション等におけるセキュリティ上の弱点。コンピュータ不正アクセスやコンピュータウイルス等により、この弱点が攻撃されることで、そのウェブアプリケーションの本来の機能や性能を損なう原因となり得るもの。また、個人情報等が適切なアクセス制御の下に管理されていないなど、ウェブサイト運営者の不適切な運用により、ウェブアプリケーションのセキュリティが維持できなくなっている状態も含む。

## プロトコル

通信規約。ネットワーク上でデータを流すための約束事をまとめたもの。

本ページは白紙です



著作・制作 独立行政法人情報処理推進機構（IPA）

編集責任 小林 偉昭

執筆者 勝海 直人 甲斐根 功  
中村 勝敏 NEC システムテクノロジー株式会社

協力者 齊藤 和男 株式会社ジェイピー・セキュア  
高木 浩光 独立行政法人産業技術総合研究所  
山岸 正 株式会社 日立製作所  
大谷 慎吾 株式会社ラック  
徳丸 浩

板橋 博之 大森 雅司 木曾田 優 金野 千里  
相馬 基邦 谷口 隼祐 永安 佑希允 渡辺 貴仁

協力会社 株式会社 Imperva Japan  
NEC システムテクノロジー株式会社  
株式会社ジェイピー・セキュア  
株式会社セキュアスカイ・テクノロジー  
株式会社ソリトンシステムズ  
株式会社ネットファイア  
株式会社日立情報システムズ  
マクニカネットワークス株式会社

※独立行政法人情報処理推進機構の職員については所属組織名を省略しました。

Web Application Firewall (WAF) 読本 改訂第2版

---

[発行] 2010年 2月16日 第1版 第1刷  
2010年 5月12日 第1版 第2刷  
2010年10月 8日 第1版 第3刷  
2011年 2月28日 改訂第2版 第1刷  
2011年 5月12日 改訂第2版 第2刷  
2011年12月27日 改訂第2版 第3刷

[著作・制作] 独立行政法人情報処理推進機構 セキュリティセンター

## 情報セキュリティに関する届出について

IPA セキュリティセンターでは、経済産業省の告示に基づき、コンピュータウイルス・不正アクセス・脆弱性関連情報に関する発見・被害の届出を受け付けています。

ウェブフォームやメールで届出ができます。詳しくは下記のサイトを御覧ください。

URL: <http://www.ipa.go.jp/security/todoke/>

### コンピュータウイルス情報

コンピュータウイルスを発見、またはコンピュータウイルスに感染した場合に届け出てください。

### 不正アクセス情報

ネットワーク(インターネット、LAN、WAN、パソコン通信など)に接続されたコンピュータへの不正アクセスによる被害を受けた場合に届け出てください。

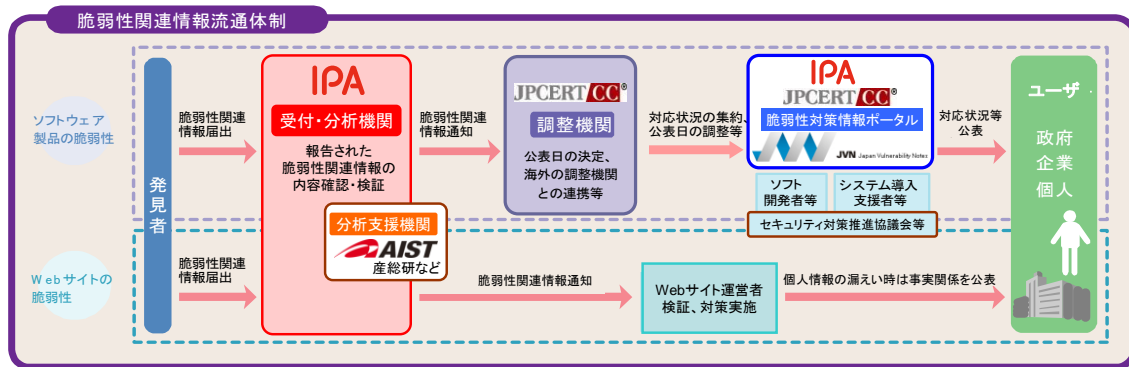
### ソフトウェア製品脆弱性関連情報

OSやブラウザ等のクライアント上のソフトウェア、ウェブサーバ等のサーバ上のソフトウェア、プリンタやICカード等のソフトウェアを組み込んだハードウェア等に対する脆弱性を発見した場合に届け出てください。

### ウェブアプリケーション脆弱性関連情報

インターネットのウェブサイトなどで、公衆に向けて提供するそのサイト固有のサービスを構成するシステムに対する脆弱性を発見した場合に届け出てください。

## 脆弱性関連情報流通の基本枠組み「情報セキュリティ早期警戒パートナーシップ」



※IPA: 独立行政法人 情報処理推進機構、JPCERT/CC: 一般社団法人 JPCERT コーディネーションセンター、産総研: 独立行政法人 産業技術総合研究所

# IPA

独立行政法人 情報処理推進機構

〒113-6591

東京都文京区本駒込二丁目28番8号

文京グリーンコートセンターオフィス16階

<http://www.ipa.go.jp/>

セキュリティセンター

TEL: 03-5978-7527 FAX 03-5978-7518

<http://www.ipa.go.jp/security/>