

# Web Application Firewall (WAF) Guide

2<sup>nd</sup> Edition

A Handbook to Understand Web Application Firewall



IT SECURITY CENTER,  
INFORMATION-TECHNOLOGY PROMOTION AGENCY, JAPAN

December 2011

This guide is available for download at:

Web Application Firewall (WAF) Guide

<http://www.ipa.go.jp/security/vuln/waf.html> (Japanese web page)

Translated by Hiroko Okashita (IPA), Dec 27 2011

# Contents

---

Contents.....	1
Preface .....	2
Intended Readers .....	2
Organization of the Guide .....	2
Caution on the Use of the Guide .....	3
What's New in the 2 <sup>nd</sup> Edition.....	3
1. Web Application Vulnerability Countermeasures with WAF .....	4
1.1. WAF: Mitigate the Impact of Attacks .....	4
1.2. Current Situation of Attacks against Web Applications and Vulnerability Countermeasures .....	5
1.3. Approach to the Use of WAF.....	7
2. WAF Overview .....	9
2.1. What WAF Is.....	9
2.2. Difference between WAF and FW & WAF and IPS .....	10
2.3. Types of WAF .....	13
2.4. Situations Where WAF is Effective.....	15
3. WAF Specifics.....	18
3.1. WAF Deployment.....	18
3.2. WAF Features .....	20
3.3. Points to Remember with WAF Features .....	30
4. WAF Introduction .....	32
4.1. Decision of Introduction.....	33
4.2. Introduction .....	37
4.3. Operation .....	48
5. WAF Introduction Case Study at IPA .....	51
5.1. Introduction .....	51
5.2. Decision of Introduction.....	53
5.3. Introduction .....	57
5.4. Operation .....	71
5.5. Summary of WAF Introduction and Operation.....	75
Appendix A. Open Source Software WAF.....	76
ModSecurity.....	76
WebKnight .....	84
Appendix B. Commercial WAF .....	93
Terminology .....	96

# Preface

---

Web Application Firewall (WAF) is one of the security measures to protect web applications from the attacks that try to exploit the vulnerabilities in web applications. Information-technology Promotion Agency (IPA) has prepared this guide to help website operators understand what the WAF is, what it can do and how to introduce it.

IPA hopes this guide will help website operators protect their web applications with a WAF.

## Intended Readers

---

This guide is intended for the website operators who are considering the possibility of introducing a WAF.

In this guide, a website operator means an entity or individual that has and operates a website. For example, the website operator of <http://www.ipa.go.jp/> is IPA.

## Organization of the Guide

---

This guide is composed of 5 chapters and 2 appendixes.

Chapter 1, "Web Application Vulnerability Countermeasures with WAF," presents the current situation of attacks against Web applications and vulnerability that IPA analyzed from its activities and the approaches by some security organizations.

Chapter 2, "WAF Overview," explains about WAF. The purpose of this chapter is to help the website operators understand what the WAF is.

Chapter 3, "WAF Specifics," gives the details on the WAF. The purpose of this chapter is to help the website operators understand the features of the WAF and the points when using those features.

Chapter 4, "WAF Introduction," shows the points to be considered through each of three phases of the WAF introduction: decision of introduction, introduction and operation.

Chapter 5, "WAF Introduction Case Study at IPA" presents a case study of introducing and operating an open source WAF "ModSecurity" at IPA, and shows what IPA actually considered and did through each phase of "decision of introduction", "introduction" and "operation". By reading Chapter 5 together with Chapter 4 "WAF Introduction", IPA hopes that the readers will understand the important points in introducing a WAF.

Appendix A, "Open Source Software WAF," gives the introduction case study of ModSecurity and WebKnight.

Appendix B, "Commercial WAF," provides the commercial WAF products of the vendors that contributed to this guide.

## **Caution on the Use of the Guide**

---

This guide provides the information on the general features, behaviors and issues of the WAF. Some WAF products may behave differently from what is shown in this guide.

## **What's New in the 2<sup>nd</sup> Edition**

---

In the 2<sup>nd</sup> edition, Chapter 4 “WAF Introduction” has been expanded, and a case study of introducing an open source WAF “ModSecurity” at IPA is added as Chapter 5, “WAF Introduction Case Study at IPA”. Reading Chapter 4 and 5 together will help the readers understand from introduction to operation of WAF in detail.

Other contents such as those in Section 2.4 “Situations Where WAF is Effective” and the ModSecurity case study in Appendix A are also revised.

# 1. Web Application Vulnerability Countermeasures with WAF

---

This chapter introduces the current situation of attacks against web applications and vulnerability countermeasures that IPA analyzed from its activities and the approaches by some security organizations.

## 1.1. WAF: Mitigate the Impact of Attacks

---

WAF is one of the security measures to protect web applications from attacks that try to exploit the vulnerabilities in web applications. WAF is an operational security measure that mitigates the impact of attacks, not a fundamental solution that eliminates the vulnerability in web application implementation.

IPA offers the guidelines for the application developers to help eliminate the vulnerabilities in web applications, such as "How to Secure Your Web Site"<sup>1</sup>, and "Secure Programming Course"<sup>2</sup>. Still, the attacks that exploit the vulnerabilities in web applications show no sign of end, and as seen through the Information Security Early Warning Partnership<sup>3</sup>, it is not that easy for the website operators to quickly fix the vulnerability in their website for various reasons. Under this circumstance, the WAF can be an effective security measure to protect web applications.

---

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

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

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

## 1.2. Current Situation of Attacks against Web Applications and Vulnerability Countermeasures

This section introduces the current situation of attacks against web applications and vulnerability countermeasures for websites that IPA analyzed from its activities.

### 1.2.1. Attacks against JVN iPedia

IPA analyzed access log of JVN iPedia<sup>4</sup>, a database of vulnerability countermeasure information operated by IPA and JPCERT Coordination Center, with a website attack detection tool iLogScanner<sup>5</sup>. Between January 2009 and December 2010, 12,194 transactions that seemed to be an attack were detected (Figure 1-1).

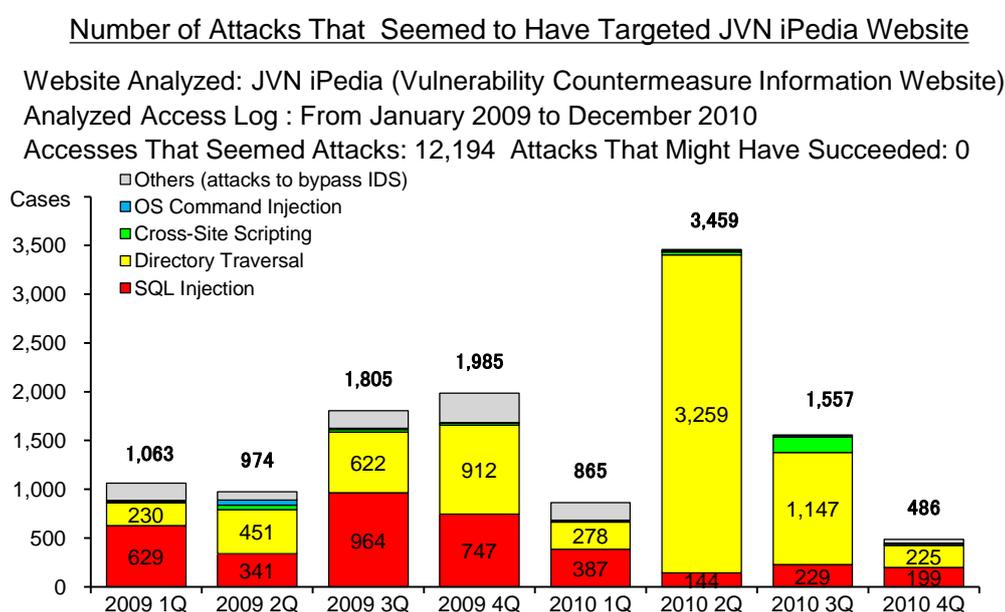


Figure 1-1 Number of Attacks against JVN iPedia between January 2009 and December 2010

Because the websites operated by businesses and organization are open to the public on the Internet, the transactions from the Internet to their website cannot be blocked by firewall. In recent years, information leak of personal information through the website of big companies has been often covered in news media. The target of the attacks against websites, however, is not limited to those of big companies. As shown in Figure 1-1, any website on the Internet can be attacked anytime. Regardless of the size of the company, all websites are potentially exposed to attacks.

<sup>4</sup> A database of vulnerability countermeasure information collected on software products, such as operating systems, applications, libraries and embedded systems, used in Japan.  
<http://jvndb.jvn.jp/en/>

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

## 1.2.2. Vulnerability Countermeasure through Information Security Early Warning Partnership

The total number of vulnerabilities reported through the Information Security Early Warning Partnership, a vulnerability-related information distribution framework, is 5,338 as of the 4<sup>th</sup> quarter of 2010 (October – December)<sup>6</sup> (Figure 1-2).

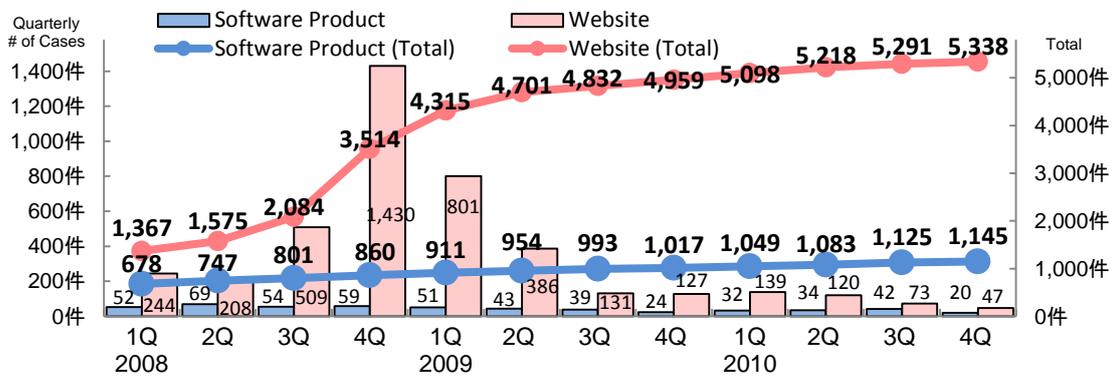


Figure 1-2 Quarterly Shift of Vulnerability-Related Information Reported (as of 4Q of 2010)

Information Security Early Warning Partnership requests the website operators to eliminate the vulnerability reported. However, not all website operators can take actions immediately. For about 53% of reported vulnerabilities, it took more than 31 days to fix the vulnerability (marked with red-box in Figure 1-3). Fact is that it takes a long time to fix vulnerability even though it is a critical one like SQL injection vulnerability for various reasons.

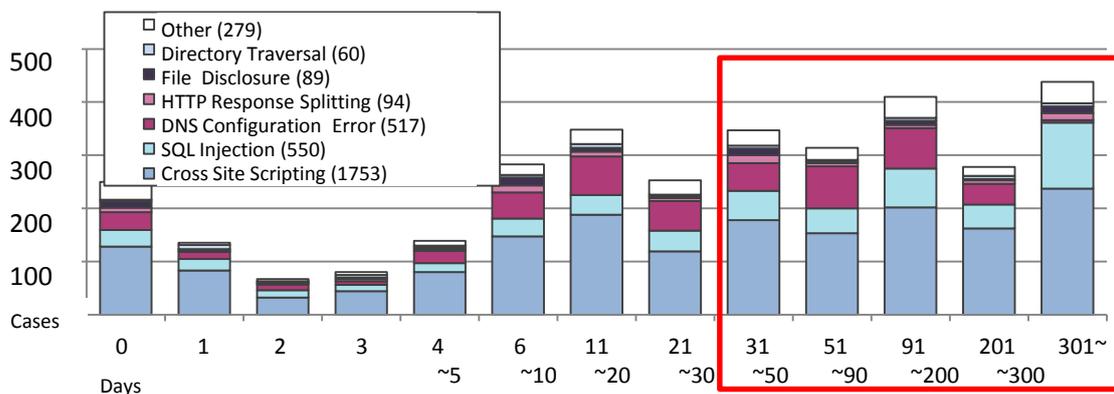


Figure 1-3 Number of Days It Took to Fix Website (as of 4Q of 2010)

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

## 1.3. Approach to the Use of WAF

---

This section introduces the approaches to the use of the WAF by some security organizations.

### 1.3.1. KISA

---

KISA (Korea Internet & Security Agency)<sup>7</sup> introduces open source software WAF on its website. Currently, 2 WAF <sup>8</sup> are listed.

- “ModSecurity”<sup>9</sup> by Trustwave
- “WebKnight”<sup>10</sup> by AQTRONIX

KISA promotes the use of the WAF by making the download of those WAF software available on its website instead of just linking to the original provider’s website. KISA also offers the introduction guides, setup guides and Q&A, as well as the information on seminars.

In addition to introducing open source software WAF, KISA offers a web application security enhance tool "CASTLE"<sup>11</sup> and a WebShell<sup>12</sup> detection tool “WHISTL”<sup>13</sup> as well<sup>14</sup>.

### 1.3.2. OWASP

---

OWASP (Open Web Application Security Project)<sup>15</sup> is working on "OWASP Best Practices: Use of Web Application Firewalls"<sup>16</sup> and “OWASP ModSecurity Core Rule Set Project”<sup>17</sup> as its projects.

The OWASP Best Practices: Use of Web Application Firewalls project documents and publishes the information about the WAF, such as whether a WAF can prevent various attacking techniques, the merit and demerit, and selection criteria when introducing a WAF. The latest version as of the release of this guide (2<sup>nd</sup> Edition) is the Version 1.0.5 published in March 2008.

The OWASP ModSecurity Core Rule Set Project develops and releases the rules for anomaly detection called “Core Rule Set” used for the open source WAF “ModSecurity”. The project explains that the Core Rule Set is general-purpose and focuses on the strings included in attacks. The latest version of the Core Rule Set as of the release of this guide (2<sup>nd</sup> Edition) is the Version 2.1.2.

---

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

<sup>8</sup> This guide also presents the installation case study of ModSecurity and WebKnight in "Appendix A. Open Source Software WAF."

<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) (Korean)

<sup>12</sup> WebShell is a backdoor program that is maliciously uploaded to the Web server.

<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) (Korean)

<sup>14</sup> To use WHISTLE, it is required to apply to KISA.

<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\\_Project](http://www.owasp.org/index.php/Category:OWASP_ModSecurity_Core_Rule_Set_Project)

### **1.3.3. WASC**

---

WASC (Web Application Security Consortium)<sup>18</sup> is working on WAFEC (Web Application Firewall Evaluation Criteria)<sup>19</sup> as one of its projects. WASC develops the WAFEC aiming to establish a versatile evaluation standard of the WAF. WASC explains the reason as to why WASC has tasked itself with development the standard that it is difficult to develop an evaluation standard of the WAF even for the experts, and therefore too much for an individual WAF developer to compare various WAFs to develop a standard. The latest version of WAFEC as of the release of this guide (2<sup>nd</sup> Edition) is the Version 1.0 published in January 16, 2006.

### **1.3.4. PCI SSC**

---

PCI SSC (Payment Card Industry Security Standards Council)<sup>20</sup> has developed PCI-DSS (Payment Card Industry Data Security Standard)<sup>21</sup>, an international security standard for the payment card industry required for the member stores and merchants who process the payment card data.

PCI-DSS is a security standard that requires the implementation of concrete information security measures. The requirement 6.6 in the PCI-DSS says "For public-facing web applications, address new threats and vulnerabilities on an ongoing basis and ensure these applications are protected against known attacks by *either* of the following methods."

- Reviewing public-facing web applications via manual or automated application vulnerability security assessment tools or methods, at least annually and after any changes
- Installing a web-application firewall in front of public-facing web applications

This requirement's compliance level was "recommended" in the Version 1.1 that was valid until June 30, 2006. It became "required" in the Version 1.2 released in July 2008. PCI-DSS has been updated several times since its initial release in December 2004. The latest version as of the release of this guide (2<sup>nd</sup> Edition) is the Version 2.0.

---

<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 Overview

This chapter explains how the WAF works, the difference between the WAF and firewall (FW) or Intrusion Prevention System (IPS), the types of the WAF, and the situations where the WAF is effective.

### 2.1. What WAF Is

The WAF is hardware or software that protects web applications from attacks that exploit the vulnerabilities in web applications. The WAF is a security measure that mitigates the impact of attacks, not a fundamental solution that eliminates the vulnerability in web application implementation.

The WAF mechanically inspects the transactions between a website and its users based on the WAF rules created by the website operator (Figure 2-1). By using a WAF, the following benefits are expected:

- **Protect web applications from attacks that try to exploit vulnerabilities.**
- **Detect attacks that try to exploit vulnerabilities.**
- **Protect multiple web applications from attacks.**

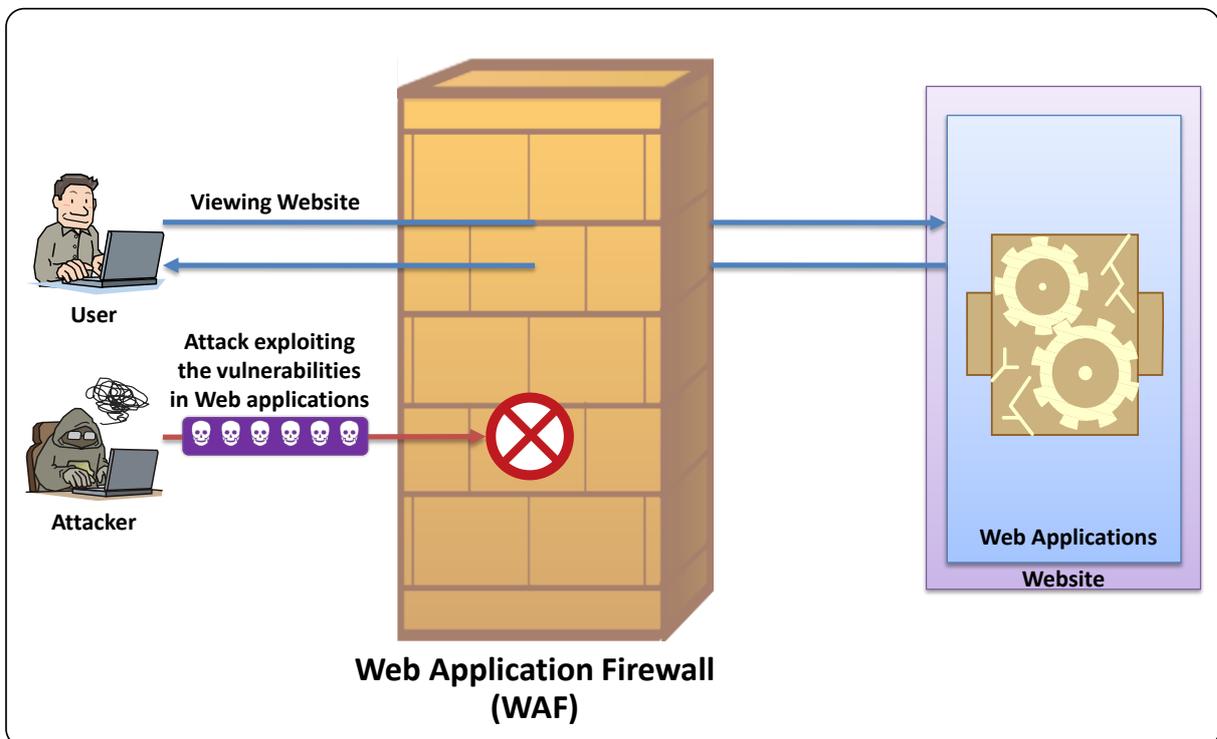


Figure 2-1 WAF Behavior (Image)

In addition, by defining a rule that will detect distinctive personal information (such as a credit card number) to the rules, the WAF can be used to prevent the personal information from being transmitted to an attacker.

Because the WAF does this filtering mechanically based on the rules, sometimes filtering errors may occur, where a resulting judgment is different from the one a person may make. For this, it is possible that a malicious transaction, such as an attack that tries to exploit the vulnerability, could be let through or a legitimate user's access to the website may be blocked (for details, see "3.3 Points to Remember with WAF Features"). When considering introduction the WAF, these points must be paid attention to.

## **2.2. Difference between WAF and FW & WAF and IPS**

---

This section explains the difference between the WAF and FW, and the WAF and IPS.

### **2.2.1. WAF and FW**

---

With a word "firewall" in its name, the WAF may sound like a kind of a firewall, but the WAF is different from firewall.

The FW is software or hardware that enforces access control based on the source and destination information (such as IP address and ports) in packets. By using the FW, it is possible to put restrictions on the transactions with the services running on the server. For example, the website operator can limit the access to the organization's internal file sharing service to from the one that is originated within the organization and prohibit the access over the Internet. By limiting the access to the services unnecessary to be open to the public, it is possible to prevent unauthorized access to those services.

The website of the businesses and organizations meant to be published on the Internet cannot limit the access to the website over the Internet. Thus, the FW may not prevent an attack that exploits the vulnerabilities in web applications (Figure 2-2).

On the other hand, the WAF can inspect the content of the packet to the web applications the FW cannot enforce the control. For example, by having a WAF rule that detects the characteristics of the SQL injection attack which tries to remotely manipulate the database, it is possible to block that offensive packet.

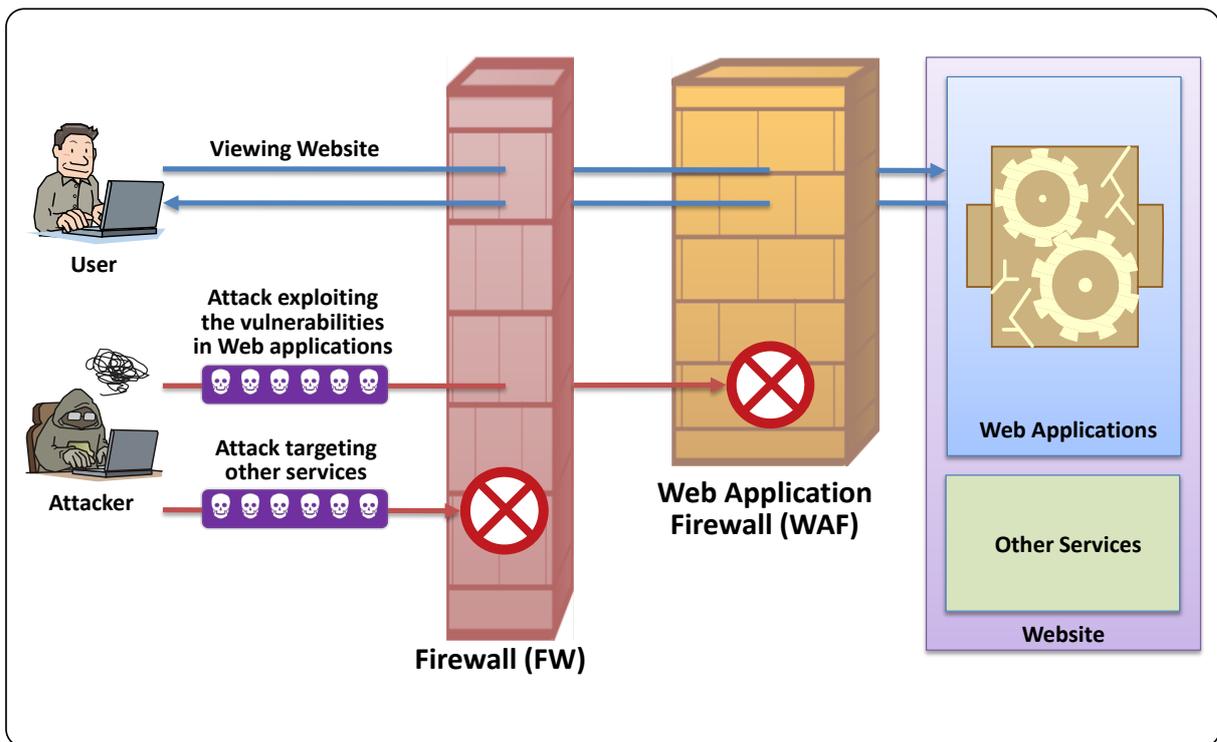


Figure 2-2 Difference between FW and WAF

### 2.2.2. WAF and IPS

The WAF and the IPS both inspect the content of the transactions based on the rules.

The IPS is software or hardware that inspects the transactions to the various devices based on the rules that the operator has defined. In general, the IPS prevents various types of attacks (such as the ones that exploit the vulnerabilities in the OS and attacks against file sharing services)(Figure 2-3). The IPS blocks the attacks by inspecting the transactions using a blacklist<sup>22</sup>, which is a list of the rules that have defined the detail of attack patterns and techniques.

On the other hand, the WAF is software or hardware that inspects the transactions to the web applications based on the rules that the operator has defined. While the IPS can prevent the attacks against various devices, the WAF can prevent only the attacks against web applications (Figure 2-4). The WAF is specialized to protect web applications and it can inspect the transactions using not only a blacklist, but also a whitelist, which is a list of the rules that has defined the characteristics of the legitimate transactions.

<sup>22</sup> For more information on a blacklist and whitelist, see "3.2 WAF Features."

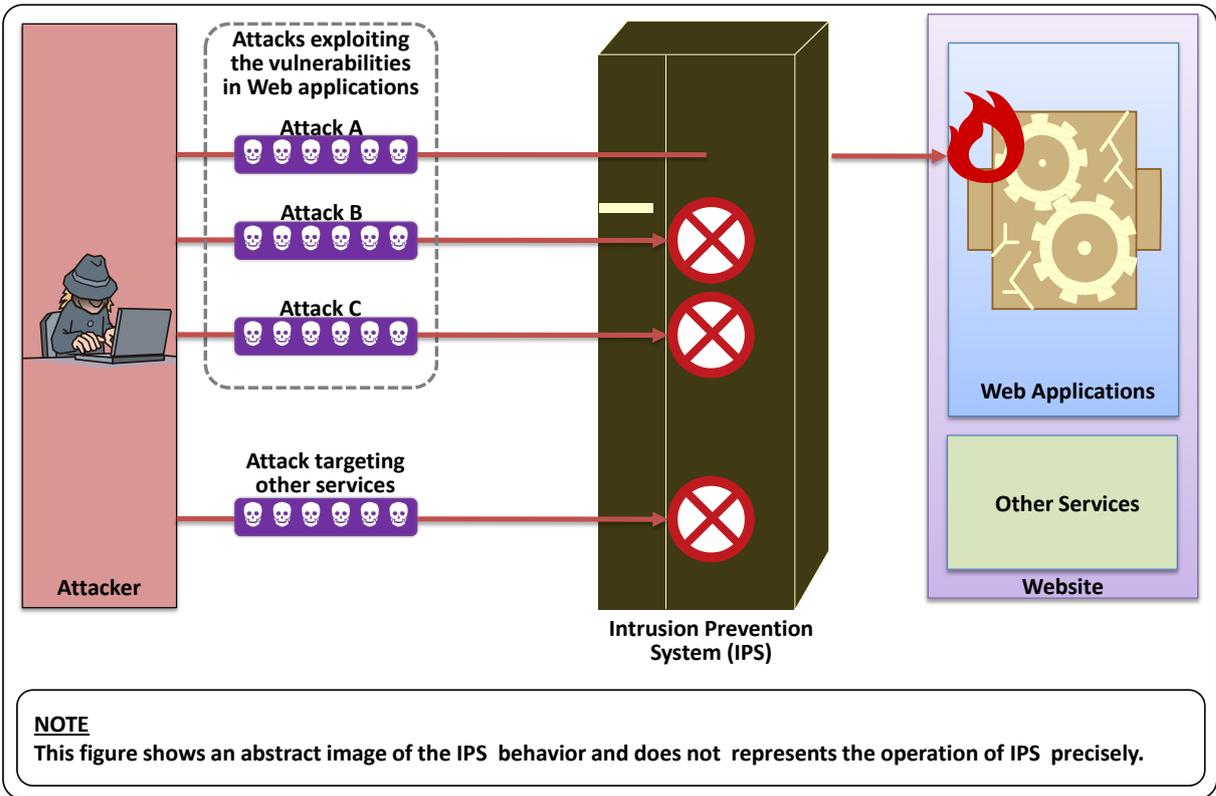


Figure 2-3 IPS Behavior (Image)

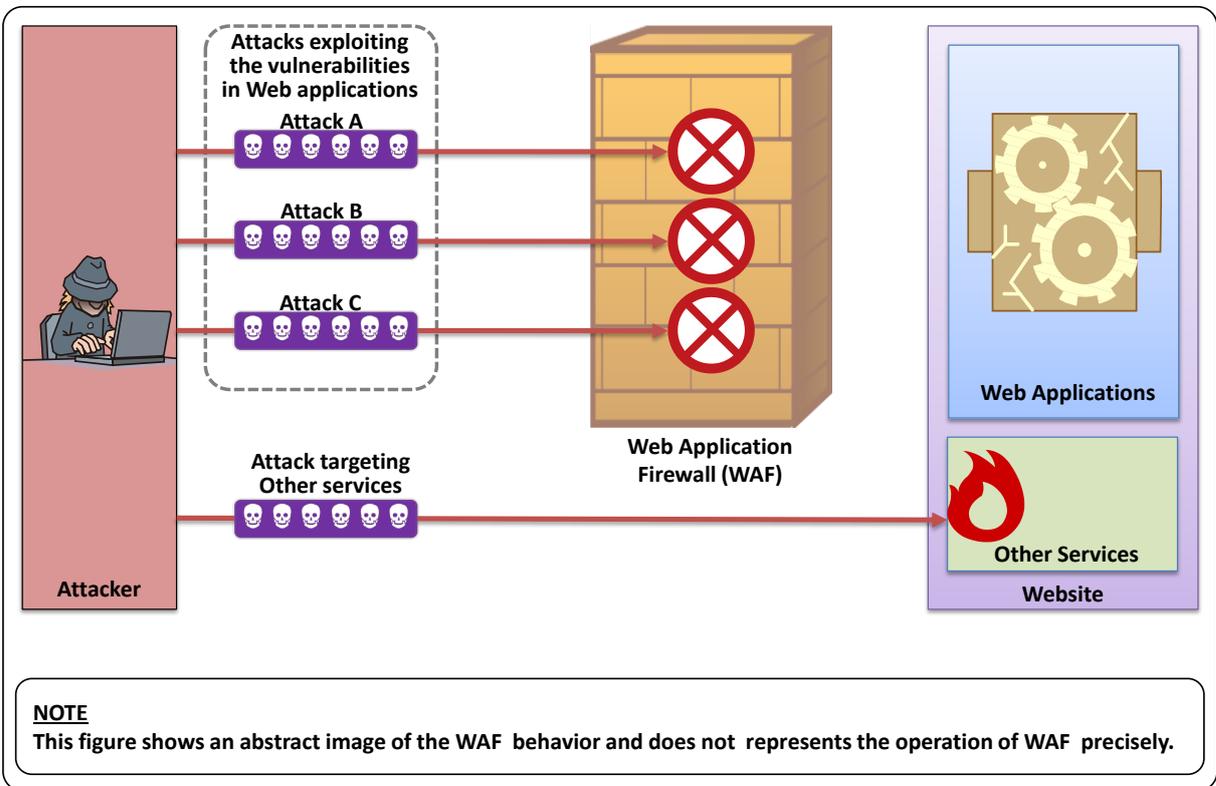


Figure 2-4 WAF Behavior (Image)

## **2.3. Types of WAF**

---

This section explains the type of WAF from the aspect of licensing and the form of the provision.

From the aspect of licensing, there are 2 types of the WAF: commercial WAF products and open source WAF software. When using a WAF, the initial cost and operational cost are required. Those costs are different for the commercial ones and open source software ones.

From the aspect of the form of the provision, there are 3 types of WAF: those that are provided as a specialized equipment, as a software, and as a service.

It is important to understand the advantages and disadvantages of each type of WAF and select an appropriate one.

### **2.3.1. Types of WAF from the aspect of Licensing**

---

#### **■ Commercial WAF Products**

A commercial WAF product (hereafter referred to as “commercial WAF”) is a WAF product the business vendors sell and provide. The commercial WAFs have the following characteristics in common.

- One can use it by paying for it to the seller or provider.
- The website operator can use a support service available from the seller or provider for the operation<sup>23</sup>.
- The manuals are well prepared, thus the operator can obtain the information about the WAF when needed.

#### **■ Open Source WAF Software**

An open source WAF software (hereafter referred to as “open source WAF”) is a WAF that can be used freely as long as following the open source license. The open source WAFs have the following characteristics in common.

- Anyone can use freely as long as following the open source license.
- Since a support service may not be available from the provider, the website operator needs to operate and maintain the WAF. If the operator may not have a good knowledge of the WAF, the operational cost may rise.
- The manuals are often scarce, thus the operator is required to have a good knowledge of the WAF.

### **2.3.2. Types of WAF from the Aspect of the Form of the Provision**

---

The form of the provision is different for the commercial WAF and open source WAF (Figure 2-5). The commercial WAF<sup>24</sup> is available not only as software but also as a specialized equipment and a service. On the other hand, the open source WAF is offered as software on the Internet<sup>25</sup>.

Depending on the form of the provision, where to deploy a WAF changes. For more information, see “3.1 WAF Deployment”.

---

<sup>23</sup> It is possible to outsource the operation of the commercial WAF.

<sup>24</sup> Some commercial WAFs are introduced in the "Appendix B. Commercial WAF”.

<sup>25</sup> Some open source WAFs are introduced in the "Appendix A. Open Source Software WAF”.

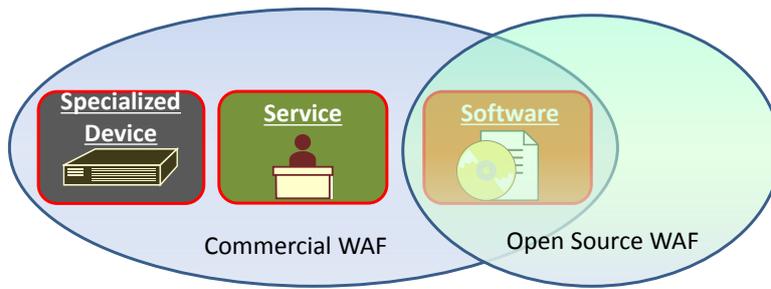


Figure 2-5 Form of the Provision for Commercial WAF and Open Source WAF

## 2.4. Situations Where WAF is Effective

This section explains in what situations the use of the WAF is effective to prevent the damage induced by attacks that exploit vulnerability in web applications.

As a website operator, to prevent attacks that exploit vulnerability in web applications, it is important to make sure that all the necessary countermeasures for the known vulnerabilities are implemented in the first place and to eliminate vulnerability promptly when a new one is found. However, sometimes it may be difficult to take a fundamental countermeasure and eliminate vulnerability for various reasons. In another case, a website operator may want to prevent attacks against web applications which he or she cannot manage directly. In these cases, the WAF may be effective.

The Figure 2-6 shows the situations where introducing the WAF is effective from the viewpoint of proactive measure and incident response measure. A proactive measure will reduce the occurrence of a security incident that exploits vulnerability in web applications. On the other hand, an incident response measure will reduce the damage of a security incident to the minimum, should it happen, and allow a faster recovery.

In what follows, each case of (a) (b) (c) in Figure 2-6 is explained.

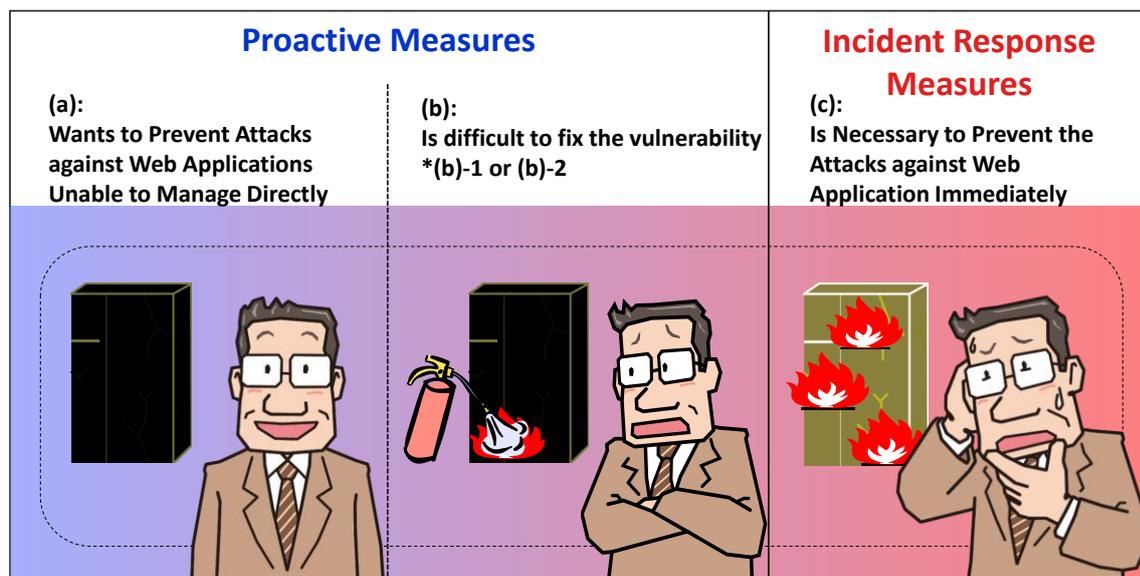


Figure 2-6 Situations Where WAF is Effective

## **(a) When the Website Operator Wants to Prevent Attacks against Web Applications Unable to Manage Directly**

---

There is a case where a website operator wants to implement the same security measures against attacks that exploit vulnerability in the web applications whose developer and operator varies. It applies for a website operator who is in a position to use and manage various web applications developed by different developers, for example, a website operator of a major business that has a number of subsidiaries in different areas or a website operator of a business that offers server hosting services.

### **(b)-1 When It Is Difficult to Have the Developer Fix Vulnerability in the Web Application**

---

When vulnerability is found in a web application, sometimes it may be difficult to have the application developer directly fix vulnerability in the web application.

When a business or organization decides to develop a web application, it may outsource the application development to an outside company. When a vulnerability is found in the web application, there might be a case where having the company that developed the application fix the vulnerability is difficult (e.g. the company is no longer in the software development business).

It is possible to have some other company fix the vulnerability in the application, but the cost could be much higher and over budget, making the modification infeasible.

### **(b)-2 When Vulnerability Is Found in the License-Protected Web Application**

---

When a website is created with a commercial product or open source software, it may be difficult to be actively involved with and make sure of the modification of the product or software.

In recent years, web applications, such as Wiki and Blog applications, are available both as commercial and open source software, enabling anyone to use a web application without developing it oneself.

When vulnerability is found in the commercial products, it is up to the software developers whether and when to fix them and provide a fixed version or security patch. If the support period for a software product is already over, it could be possible that the vulnerability is left as it is.

As for open source software, the user organization can confirm the vulnerability and modify the software if the organization has the capability. If the organization does not have an in-house capability, it may have no choice but leave the vulnerability unfixed.

### **(c) When It Is Necessary to Prevent the Attacks against Web Application Immediately**

---

When a web application has vulnerability and is attacked by exploiting the vulnerability, it could inflict the damage to the website.

When noticing the damage caused by attacks, it is critical to act immediately to stop the damage from spreading. To do so, sometimes it is necessary to stop the web service to investigate the cause and damage or to fix the problem. For the companies that rely on the Internet for their business, however, the longer the website is shut down, the bigger the loss of business opportunity is and it may pose a big impact on their business continuity. From the aspect of business continuity, there is a case where taking time to fix vulnerability is infeasible.

## 3. WAF Specifics

This chapter explains where to deploy a WAF, the WAF features and the points to remember with those features.

### 3.1. WAF Deployment

When deploying a WAF, there are 2 possible locations: on the network and on the web server. When considering where to deploy the WAF, it is important to take into account the configuration and availability of the website, and the WAF characteristics that differ depending on the deployment location.

The detailed deployment method is out of the scope of this guide since it differs depending on each commercial and open source WAF. Contact the WAF vendor as needed.

#### 3.1.1. Deployment Location: On Network

When deploying a WAF on the network (hereinafter referred to as “network-based WAF<sup>26</sup>”), it is located on the path between the users and the website, and inspects the HTTP (as well as HTTPS) transactions (Figure 3-1). A network-based WAF can be a specialized hardware or a server that is installed with WAF software.

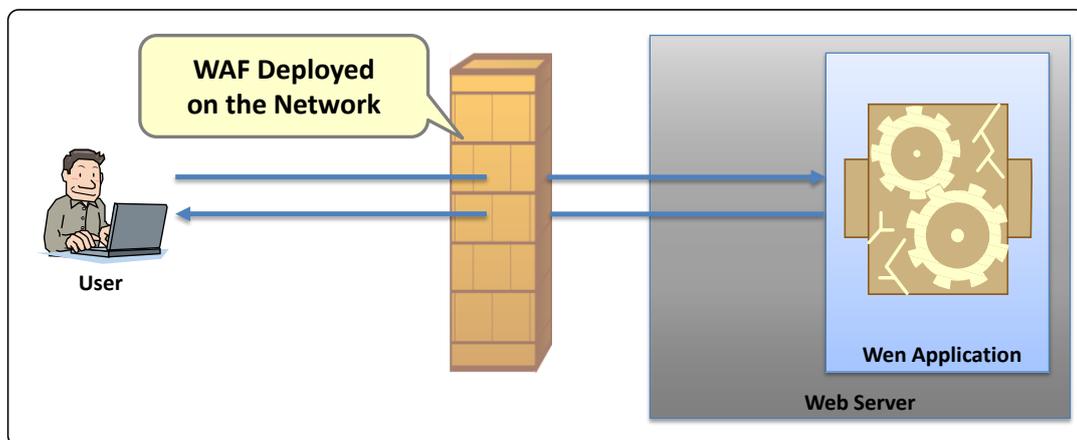


Figure 3-1 Network-Based WAF (Image)



A network-based WAF can be deployed not only on the website but also outside the website such as at another business site. The website operator can also use a commercial WAF service provided by some businesses.

<sup>26</sup> A “network-based WAF” is a coined word used in this guide for convenience and not an established term.

A network-based WAF has the following characteristics.

- Its operation does not depend on the operational environment of the web server.
- Its operation does not depend on the number of the web servers that constitute the website.
- When deploying a WAF to the existing website, the reconfiguration of the network is required.
- If the WAF supports the HTTPS, it is possible to inspect the HTTPS, too.
- By using the WAF, it is possible that the availability of the website may decrease.

In addition to them, the commercial WAF service has the following characteristics.

- The website operator does not have to deploy a WAF on its own network.
- Compared to deploying a WAF on its own network, the impact to its web server or network configuration is small.

### 3.1.2. Deployment Location: On Web Server

---

When deploying WAF on the web server (hereinafter referred to as “server-based WAF<sup>27</sup>”), the WAF inspects the HTTP transactions between the users and the website when the web server receives and sends the packets (Figure 3-2). In general, a server-based WAF is often provided as software and works part of the web server.

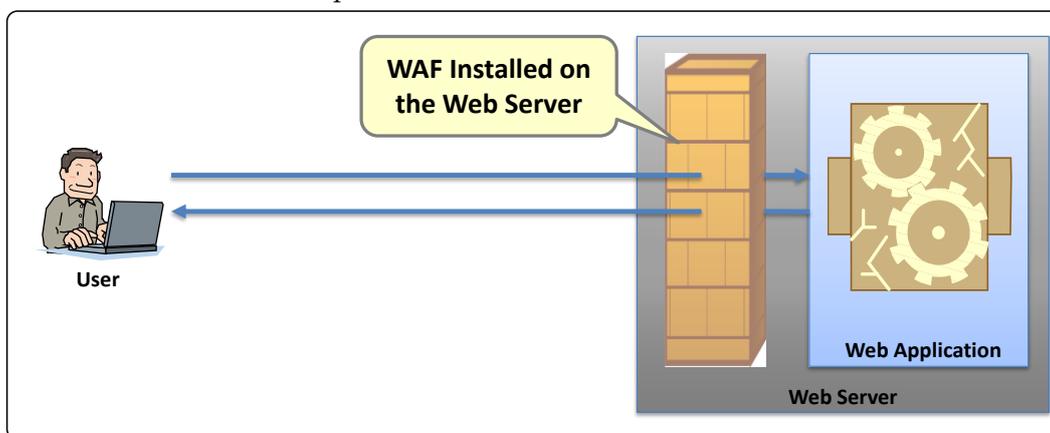


Figure 3-2 Server-Based WAF (Image)

A server-based WAF has the following characteristics.

- Its operation depends on the operation environment of the web server.
- It needs to be deployed to all web servers that constitute the website.
- There is no need to reconfigure the network when deploying a WAF to the existing website.
- Since the HTTPS packets are processed (encrypted and decrypted) on the web server, the HTTPS transactions can be inspected even if the WAF does not support HTTPS.
- By using the WAF, it is possible that the performance of the web server may decrease.

---

<sup>27</sup> A “server-based WAF” is a coined word used locally in this guide for convenience and not an established term. It is also known as a “host-based WAF”.

## 3.2. WAF Features

The WAF protects the web applications from the attacks that try to exploit the vulnerabilities in the web applications using the various features in combination (Figure 3-3). A WAF has 2 levels of features: the “basic features” that all WAFs have and the “advanced features” that are uniquely implemented in each WAF to supplement the “basic features”<sup>28</sup>.

In this section, many HTTP terms are presented in the course of explaining the WAF features. Here, the HTTP terms defined in RFC 2616<sup>29</sup> are used.

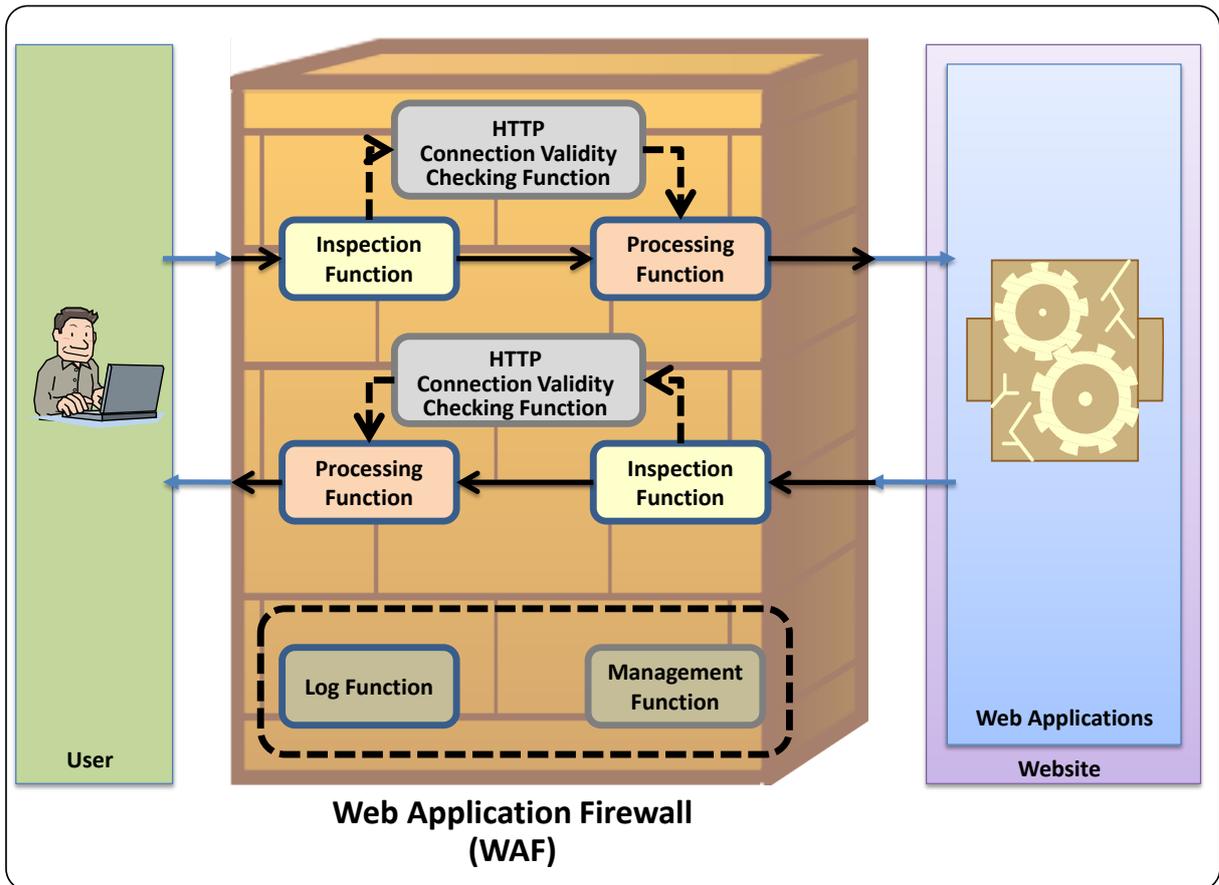


Figure 3-3 Overview of WAF Features

<sup>28</sup> The terms, the "basic features" and the "advanced features," are used locally in this guide for convenience and not the established terms.

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

### 3.2.1. Basic Features

---

The WAF inspects the HTTP transactions between the users and the website mechanically. When an HTTP transaction is deemed "malicious", the WAF executes a preset action. A collection of these proceedings is the "basic features" of the WAF. The "basic features" include the following:

- The Inspection Function: inspect the HTTP connections based on the WAF rules.
- The Processing Function: execute the actions preset to process the HTTP connections detected in the inspection.
- The Log Function: log the WAF behavior.

#### ■ Inspection Function

The "inspection function" is a function that inspects the HTTP request and HTTP response<sup>30</sup> based on the predefined rules. The inspection items for the HTTP request and HTTP response include the following<sup>31</sup>.

##### ◆ Inspection Items for HTTP Request

- The request line (method, URI, query string)
- The header field (general-header field, request-header field, entity-header field)
- The message body (POST data)

##### ◆ Inspection Items for HTTP Response

- The status line (status code)
- The header field (response header field)
- The message body

There are 2 types of rules used by the "inspection feature" depending on the definition: blacklist and whitelist.

---

<sup>30</sup> The inspection of the HTTP response is performed to prevent the web server from responding to an attacker with confidential or unnecessary information.

<sup>31</sup> The inspection items differ depending on the WAF.

## ◆ **Blacklist**

A blacklist is a list of the rules that define "unacceptable values and patterns" for the HTTP transactions. When the WAF inspects an HTTP transaction using the blacklist and if the content of the transaction has a match with the values or patterns on the list, the WAF judges and detects that the HTTP transaction is a malicious one<sup>32</sup>.

In general, a blacklist defines the characteristic values and patterns that are often used in the attacks that try to exploit the vulnerabilities in web applications. By taking the blacklist approach, the WAF can protect the web application from the known attacks. In other words, the blacklist-based inspection can detect only the attacks whose characteristics have already known. To detect the latest attacks using the blacklist, the list needs to be kept updated as soon as a new attacking technique is reported.

## ◆ **Whitelist**

A whitelist is a list of the rules that define "**acceptable, permitted values and patterns**" for the HTTP transactions. When the WAF inspects an HTTP transaction using the whitelist and if the content of the transaction **does not match with the "acceptable, permitted values or patterns" on the list**, the WAF judges and detects that the HTTP transaction is a malicious one<sup>33</sup>.

In general, a whitelist defines the legitimate values and patterns for the parameters based on the design of the web application. By taking the whitelist approach, the WAF can prevent the web application from being fed with a value or the type of the value that the web application developers do not anticipate. Since the whitelist-based inspection depends on the design of the web application, a whitelist needs to be created for each web application.

Since the WAF can limit its target of protection to the web application, it is possible to take the whitelist approach.

Table 3-1 summarizes the advantages and disadvantages of the blacklist and whitelist. For the points to remember with other WAF features, see "3.3 Points to Remember with WAF Features".

Table 3-1 Advantages and Disadvantages of Blacklist and Whitelist

	Blacklist	Whitelist
Advantage	The same blacklist can be used for all web applications.	Since it judges anything that does not match with the legitimate values or patterns as malicious, it can prevent previously unknown attacks as well.
Disadvantage	As soon as a new attacking technique is reported, the list needs to be updated.	A whitelist needs to be created for each web application.

<sup>32</sup> It may be also called "the negative security model" in some documents.

<sup>33</sup> It may be also called "the positive security model" in some documents.

## ■ Processing Function

The “processing function” is a function that executes the preset actions against the malicious HTTP transactions detected by the “inspection function” or the HTTP connection checking feature (as discussed hereinafter). There are 3 actions this function can take.

### ◆ Passing

Passing is an action where the WAF just lets through the HTTP transactions detected in the inspection to the user or website. In general, this action is set to assess the HTTP transactions during the early stage of the WAF introduction or to log the HTTP transactions detected in the inspection.

### ◆ Error Processing

Error processing is an action where the WAF generates and sends back an error response to the user or website, on behalf of the original HTTP connection detected in the inspection (Figure 3-4). In general, the WAF can create and send the error response with arbitrary message.

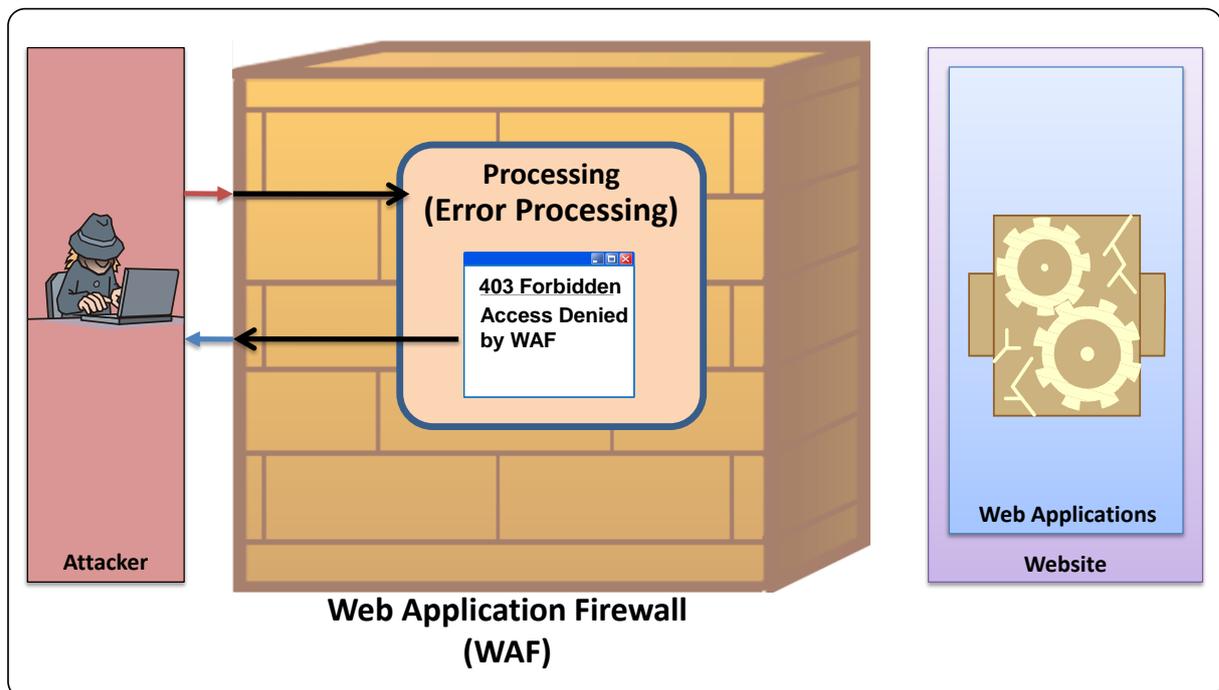


Figure 3-4 WAF Behavior (Error Processing)

◆ **Blocking**

Blocking is an action where the WAF intentionally discards the malicious HTTP packets detected in the inspection (Figure 3-5). When the WAF breaks off a HTTP transaction, it may return a HTTP response that tells the user or website that the connection has been lost or it may not respond at all.

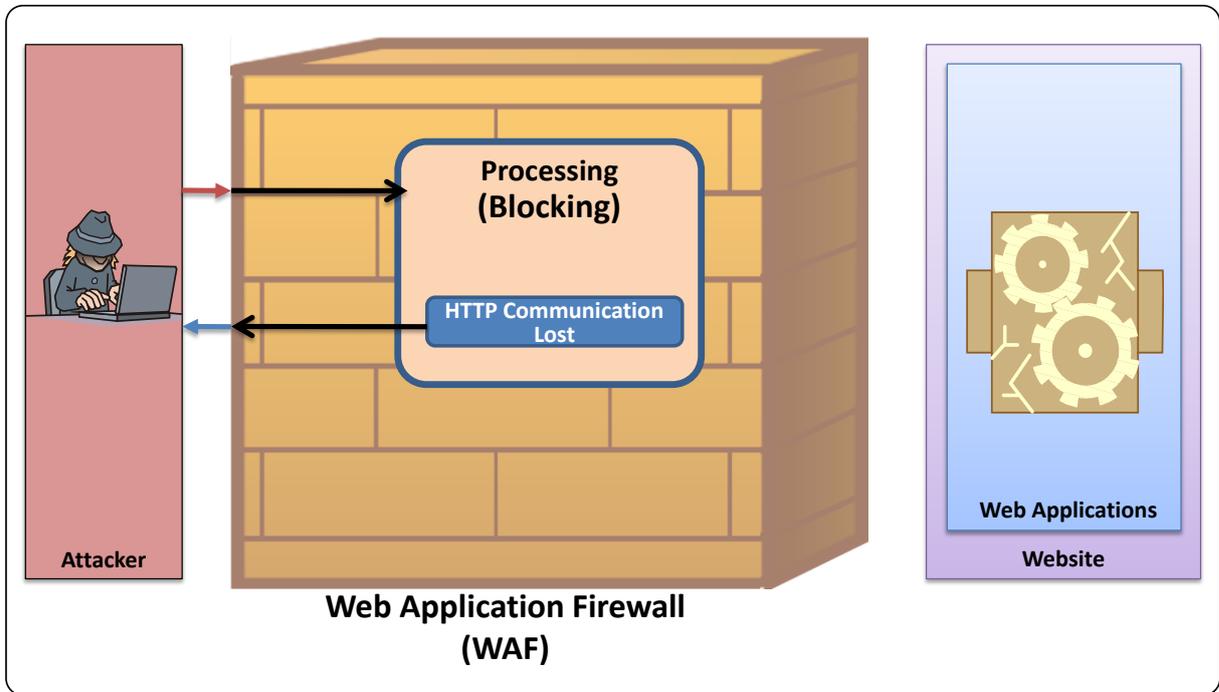


Figure 3-5 WAF Behavior (Blocking)

Some WAFs may have the following features as well.

◆ **Rewriting**

Rewriting is an action where the WAF alters part of the content of the HTTP transaction detected in the inspection and forward to the user or website (Figure 3-6). This mode is used to keep the HTTP connection open even if a suspicious character string is detected in the HTTP transaction, for example, an offending HTTP request that contains a cross-site scripting (XSS) attack or SQL injection attack, or an HTTP response that contains confidential information or unnecessary information.

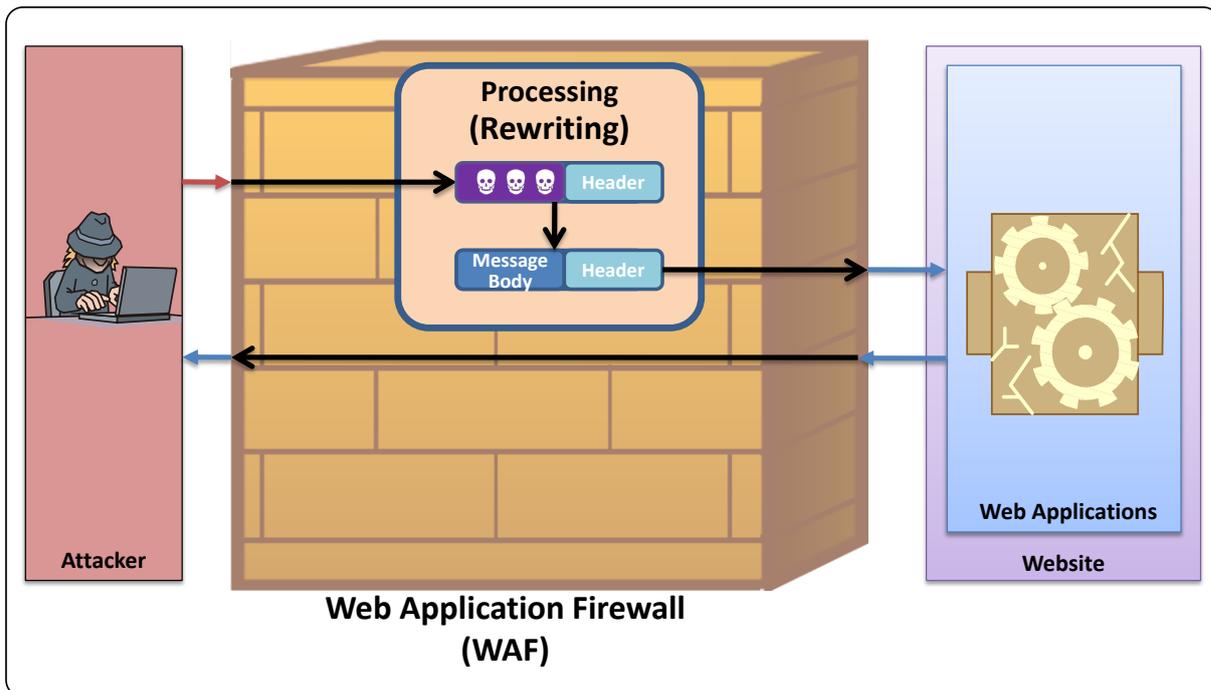


Figure 3-6 WAF Behavior (Rewriting)

## ■ **Log Function**

The log function is a function that records the malicious HTTP transactions detected by the “inspection function” and the behavior of the WAF (hereafter referred as “log”). In general, the WAF logs are stored as a file or in the database. There are 2 types of logs depending on what to log.

### ◆ **Audit**

The audit log records the malicious HTTP transactions detected by the “inspection function” and the actions taken against those HTTP transactions. The content of the audit log includes the date and time of the detection, action taken, source IP address, destination URL, the field of the packet that triggered the detection (such as HTTP request header, HTTP message body) and the WAF rule applied.

The audit log is used by the website operator to check up the malicious HTTP transactions or to generate a WAF management report (as discussed hereinafter).

### ◆ **Debug Log**

The debug log records the behavior and error information of the WAF. The content of the debug log includes the date and time of the occurrence, the behavioral information of the WAF such as start up, stop/shutdown and change of the configuration settings and error information.

The debug log is used by the website operator to check if everything is working okay with the WAF.

### 3.2.2. Advanced Features

---

In addition to the “basic features,” there are the “advanced features” uniquely implemented in each WAF<sup>34</sup>. The advanced features include the following.

- The HTTP Connection Validity Checking Function<sup>35</sup>: check the validity of the HTTP connection<sup>36</sup>.
- The Management Function: improve the convenience to use the WAF.

#### ■ **HTTP Connection Validity Checking Function**

The HTTP connection validity checking function is a function that confirms the validity of the HTTP session parameters and HTTP requests. Since the “inspection function” mainly inspects the content of each HTTP transaction, it is difficult to prevent the attacks that exploit the vulnerabilities in the session management. This feature prevents those attacks in 3 ways.

#### ◆ **Checking Parameters in Session**<sup>37</sup>

“Checking the parameters in sessions” means to see if the parameters used for the session management are altered.

A HTTP request made from a browser to a website is something the user can freely create and send. In the case where a web application stores the session-specific information in the particular parameters<sup>38</sup> and sends back and forth with the user, an attacker can change the session information as he or she pleases by altering the value of the parameters.

With session parameter checking, the WAF prevents the parameter modification by temporarily storing the values of the particular parameters of the HTTP response and checking if the parameter values in the following HTTP request match with the stored values.

The particular parameters include the cookie of the HTTP header and hidden parameter. Some WAFs encrypt the parameter values before sending them to the user.

---

<sup>34</sup> Not all WAFs have these features.

<sup>35</sup> The term, the “HTTP connections validity checking feature” is used locally in this guide for convenience and is not an established term.

<sup>36</sup> A session is a series of interactions between the user and the website that occur during the span of an HTTP connections.

<sup>37</sup> The term, “checking the parameters in sessions” is used locally in this guide for convenience and is not an established term.

<sup>38</sup> Include the Cookie header and the POST data in the HTTP request.

### ◆ Checking Validity of HTTP Request<sup>39</sup>

"Checking the validity of the HTTP requests" means to see if an HTTP request sent from the user during the session is indeed a legitimate request intended by the user.

If the web application has a Cross-Site Request Forgeries (CSRF) vulnerability<sup>40</sup>, an attacker can trick the user and have the user execute unintended actions.

With validity checking of the HTTP request, the WAF prevents CSRF attacks by adding secret information that is unguessable to the third party to the HTTP response and checking the existence of the secret in the following HTTP request.

### ◆ Checking Web Page Transition<sup>41</sup>

"Checking the web page transition" means to see if a web page requested in the HTTP request sent from the user during the session is a proper, expected one.

Among the attacks that try to exploit the vulnerabilities in web applications, a variation of luring attacks tricks and lures the user to visit a malicious web page set up by an attacker<sup>42</sup>. In this case, the transition of web pages is different from the legitimate order and it will show that the user moved from an unknown web page (a malicious web page set up by the attacker) to a legitimate web page provided by the web application.

With web page transition checking, the WAF prevents the said luring attack by predefining the expected order of web page transition and checking if the previous website recorded in the Referer header of the HTTP request matches with the expected order. If not, the WAF judges the HTTP transaction as malicious.

By implementing this technique, the following issues may arise.

- The availability of the website may decrease.
- It may have a negative effect on the rank in search engine results.

---

<sup>39</sup> The term, the "checking the validity of HTTP request" is used locally in this guide for convenience and is not an established term.

<sup>40</sup> For more information about CSRF vulnerability, visit the following websites.

"How to Secure Your Web Site": <http://www.ipa.go.jp/security/english/third.html#websecurity>

"What is Vulnerability?": [http://www.ipa.go.jp/security/vuln/vuln\\_contents/index.html](http://www.ipa.go.jp/security/vuln/vuln_contents/index.html) (Japanese)

<sup>41</sup> The term, the "checking web page transition" is used locally in this guide for convenience and is not an established term.

<sup>42</sup> The CSRF attack for one.

## ■ **Management Function**

The “management function” is a function that gives and improves the convenience to use the WAF. The feature allows the website operator to check up the number of malicious HTTP transactions by analyzing the audit log entries obtained by the log feature, and be saved from the burden of continually updating the WAF rules for the “inspection function”. The management function includes the following.

### ◆ **Generating Reports**

Generating reports means that it outputs a statistics report analyzed from the audit log entries recorded by the “log feature”. A report is typically generated periodically (for example, weekly, monthly or yearly). The report summarizes the events, such as the number of the HTTP transactions deemed malicious per source IP address and what rules they have violated.

### ◆ **Notifying Operator**

Notifying the website operator means that it sends a notification to a predefined operator or operators when a malicious HTTP transaction is detected by the “inspection function” and “HTTP connection validity checking function.” Usually, the feature gives the operator a heads-up when a malicious HTTP transaction is detected.

### ◆ **Generating Whitelist Automatically**

Generating the whitelist automatically means that it automatically creates a whitelist based on the HTTP transactions that pass through the WAF. The whitelist approach has a disadvantage that it requires the preparation of a whitelist for every single web application. Automatic generation will help solve the problem.

### ◆ **Updating Blacklist Automatically**

Updating the blacklist automatically means that it automatically updates the blacklist. Since the blacklist approach depends on the premise that the attacking methods are known in advance, it has a disadvantage that the list must be updated as soon as a new attacking method is reported. Automatic update will help solve the problem.

### 3.3. Points to Remember with WAF Features

Remember that the use of the WAF cannot prevent all the attacks that exploit known and unknown vulnerabilities of every single web application in sophisticated and creative ways. It is also possible that the WAF may block the legitimate HTTP transactions and decrease the availability of the website. When considering the use of the WAF, it is important for the website administrator to understand the following points and make an informed decision.

#### 3.3.1. False Positives and False Negatives in Inspection

Because the WAF performs the filtering mechanically based on the rules when inspecting the HTTP transactions between the users and website, sometimes filtering errors, false positives and false negatives, may occur.

A false positive and false negative are an error a malicious HTTP transaction (positive) and legitimate HTTP transaction (negative) end up in being filtered as the opposite.

A false positive is the case where a legitimate HTTP transaction is deemed malicious. If a false positive occurs, a user's legitimate HTTP transaction may be blocked by the WAF.

A false negative is the case where a malicious HTTP transaction is deemed legitimate. If a false negative occurs, an attack that tries to exploit the vulnerability in the web application may pass through the WAF.

The cause that will lead to a false positive or false negative depends on which approach, blacklist or whitelist, is adopted by the "inspection function". When introducing a WAF, it is important to use it with the understanding of the blacklist and whitelist, including their impact.

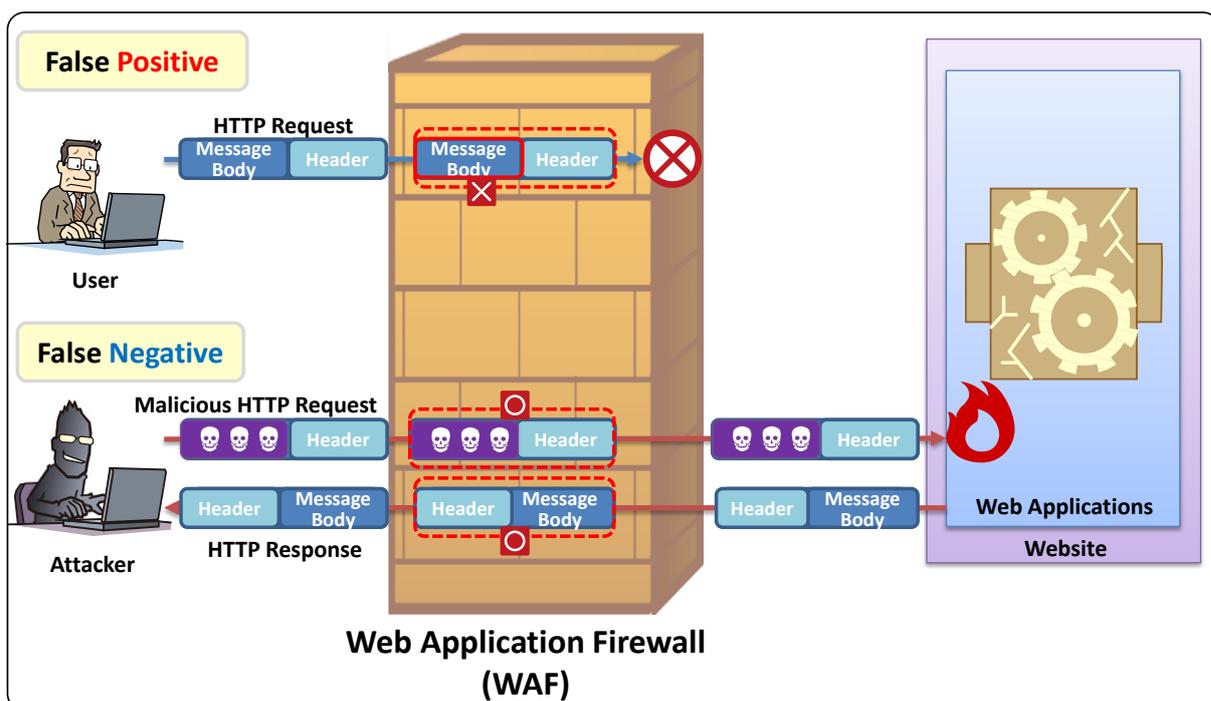


Figure 3-7 False Positive and False Negative in Inspection

### ■ **Points to Remember with Blacklist**

A blacklist can be shared by any web applications. Usually, the WAF developer provides a blacklist to prevent known attacks. If all rules on the blacklist are applied, however, false positives may occur. If a legitimate HTTP transaction is blocked, there is no choice but to disable the rule that the blocked transaction is supposed to have violated.

### ■ **Points to Remember with Whitelist**

If a web application provides an input form where a user can type in freely, it is difficult to define what the legitimate input values are. In this case, the rules for the said input form cannot be defined on the whitelist, and thus the attacks may not be detected.

In addition, if the web application's specification is changed, the whitelist needs to be updated as well. Unless there is some way to reflect the change in the web application's specification to the whitelist, false positives and false negatives may occur.

### **3.3.2. HTTP Transaction That WAF Cannot Prevent**

---

Depending on the vulnerabilities in web applications, the WAF may not prevent the attacks that exploit them.

For example, if a web application has an access control vulnerability which allows unauthorized users to use the features that are supposed to be restricted to the authorized users, the HTTP transactions themselves will be regarded legitimate even if they have been made by the unauthorized users.

# 4. WAF Introduction

In this chapter, to help understand the key points at the introduction of WAF, things to be considered and noted through each of three phases of the WAF introduction, “decision of introduction”, “introduction” and “operation” are explained.

When introducing a WAF, the first thing to do for the website operator at the phase of “decision of introduction” is to see if a WAF will be implemented as an operational security measure to mitigate the effects of attacks. The key point at this phase is to evaluate the cost effectiveness of WAF introduction for the website assuming that the website operator has also considered taking fundamental actions like secure computing. For more information, see “4.1 Decision of Introduction”.

After the website operator decides to introduce a WAF, at the phase of “introduction”, the website operator should create a plan to introduce and operate the WAF from day to day and implement it as planned. The key points at this phase is how well to coordinate works to be done with the parties relevant to WAF introduction and operation and to plan verification testing. For more information, see “4.2 Introduction”.

Completing introducing the WAF is not the end of WAF introduction but a start. At the phase of “operation”, use the WAF as planned by the operation plan. The key point at this phase is to revise the operation procedures as needed. For more information, see “4.3 Operation”.

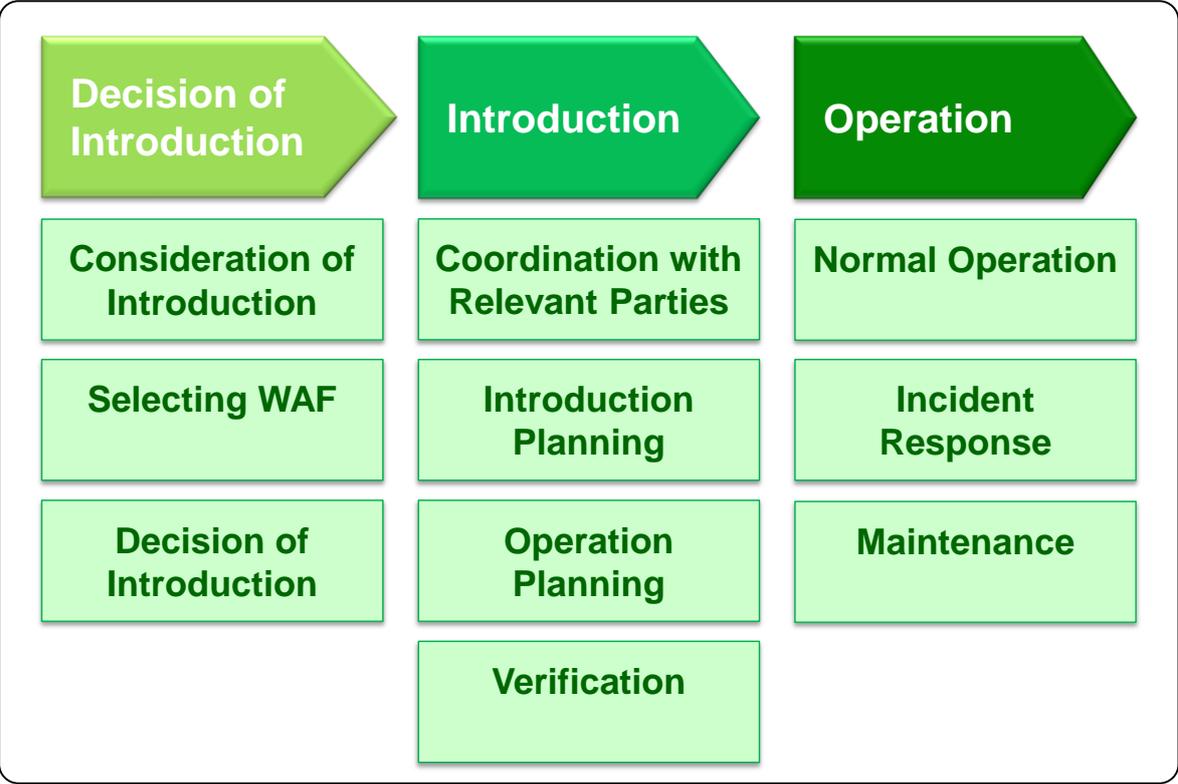


Figure 4-1 Three Phases of WAF Introduction

## 4.1. Decision of Introduction

---

At the phase of “decision of introduction”, the website operator should see if a WAF will be implemented as an operational counter measure to mitigate the effects of attacks. This phase is completed when the decision is made.

In this section, the key points at this phase are explained in 3 actions: “consideration on introduction”, “selecting WAF” and “decision of introduction”. Here, it is supposed that the website operator will take the actions in the order from “consideration of WAF introduction”, “selecting WAF” to “decision of introduction”.

### 4.1.1. Consideration of WAF Introduction

---

First, the website operator should evaluate whether or not introducing a WAF is effective as a vulnerability countermeasure for the website he or she operates.

If the web applications have vulnerabilities, they could be attacked and suffer damage through those vulnerabilities. To avoid that, it is most desirable to fix the vulnerabilities. As explained in “2.4 Situations Where WAF is Effective”. There are the cases where introducing a WAF works well against attacks that exploit vulnerability.

In these cases, consider introducing a WAF based on the following points.

#### ■ **Check If WAF Can Prevent Expected Attacks**

Depending on the types of attacks that exploit vulnerabilities, there are possibilities that a WAF cannot prevent the attacks (See “3.3 Points to Remember with WAF Features”). When considering WAF introduction, check if the WAF can prevent the attacks the website operator wants it to prevent.

If it is difficult to judge whether the WAF can prevent the expected attacks, ask the WAF vendors in advance.



Check if the WAF can prevent the attacks the website operator wants it to prevent in advance by, for example, asking the WAF vendors in advance.

### 4.1.2. Selecting WAF

---

After the website operator has confirmed that the WAF can prevent the attacks that he or she wants it to prevent, select a WAF suitable for the website he or she operates from a standpoint, such as the “impact on the website’s system configuration”, the “impact on the website’s performance” or the “effect on the operation of the website.”

## ■ Impact on Website's System Configuration

For where to place the WAF, see 3.1

By introducing a WAF, the configuration of the website system (such as the network configuration and software configuration of the web server) will change. When selecting a WAF, consider the impact on the website's system configuration.

For example, if the website's network configuration cannot be changed, the website operator can use a server-based WAF to be installed into the server or a WAF service provided by the businesses. On the other hand, if the web server's configuration cannot be changed, the website operator can use a network-based WAF.



When selecting a WAF, check the website's system configuration.

## ■ Impact on Website's Performance

For where to place the WAF, see 3.1

Any WAF will have an impact on the website's performance more or less. When selecting a WAF, consider the impact the WAF will impose on the website's performance and select the WAF that will satisfy the performance requirements after WAF introduction.

For example, if introducing a network-based WAF, it will impact the availability of the website. On the other hand, if introducing a server-based WAF, it will impact the performance of the website.



When selecting a WAF, consider the impact on the website performance.

## ■ Impact on Website Operation

For the features of the WAF, see 3.2

To select a WAF suitable for the website's operational policy, compare the features of the WAFs. The main features of the WAF to compare are listed below.

- Ease of introduction, configuration change and operation (log functions and management functions including user interface)
- Impact of the WAF's malfunction to the services (tolerance for failure)



Select a WAF suitable for the operational policy of the website in consideration of operation.

### **4.1.3. Decision of Introduction**

---

After the website operator have confirmed that the WAF can prevent the attacks that he or she wants it to prevent and selected a WAF suitable for the operational policy of the website, make a decision whether or not to introduce the WAF.

#### **(1) Estimate Cost of WAF Introduction and Operation**

Based on the selection, estimate the cost of introducing and operating the WAF.

The WAF introduction cost includes the price of the WAF product, the cost to install the WAF (such as the labor cost to configure and validate the WAF). The WAF operational cost includes the maintenance cost for the WAF products, the cost to operate the WAF (the labor cost to check the logs, update the WAF and rules).

Introducing a WAF will cost not only the initial cost but also the operational cost. Estimate the operational cost in consideration of a long-time use of the WAF. The operational cost differs depending on which WAF to use. When using a commercial WAF product or service, ask the WAF vendor.

#### **(2) Decide Based on WAF's Cost Performance**

In general, the website operator should compare the WAF introduction cost (the total sum of the initial and operational cost) and the effectiveness of the WAF to the cost to take the fundamental vulnerability countermeasures, and decide to introduce the WAF when the cost effectiveness of introducing the WAF is greater than taking the fundamental countermeasures.

Let's look at the case in (b)-1 in "2.4 Situations Where WAF is Effective". If the same level of effectiveness will be earned from the "Fix Web Applications" and "Introduce WAF", the website operator should compare the cost to implement each measure (Figure 4-2). If "Introduce WAF" is the most cost-effective as the result of comparison, "Introduce WAF" is the leading measure. After examining the shortcomings<sup>43</sup> of the measure to be implemented, decide whether or not to introduce the WAF.

On the other hand, let's take a look at the case (c) in "2.4 Situations Where WAF is Effective. In this case, the cost of introducing the WAF may be higher than other measures. Under this circumstance, however, what is most important is to prevent the attacks that exploit the vulnerabilities in the web applications urgently. There is a case where the website operator decides to introduce a WAF considering the damage you are suffering from the attacks at that point.

---

<sup>43</sup> In this example, even if the vulnerabilities exist, they will not be fixed.

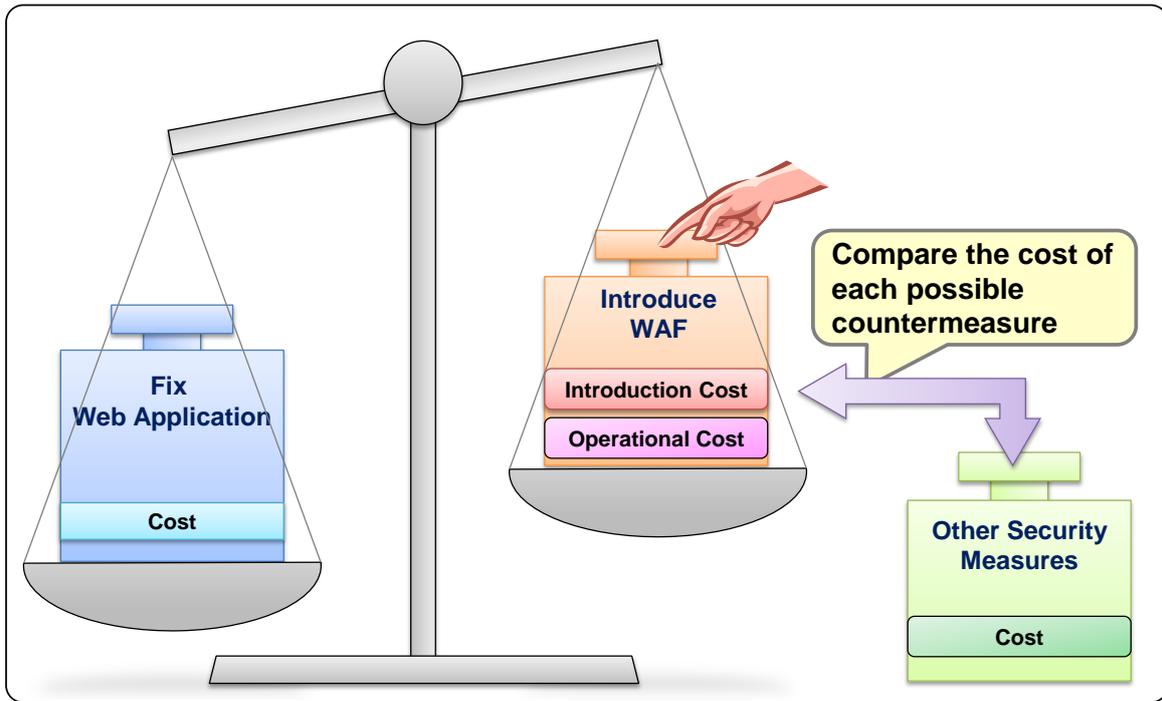


Figure 4-2 Compare Cost of “WAF Introduction” and Other Measures

### (3) **Get Budget**

If WAF introduction is judged most desirable as the result of cost-effectiveness estimation, get the budget to introduce the WAF. Make sure to get the budget not only for the initial cost but for the operational cost.

**POINT** Make sure to get the budget not only for the initial cost but for the operational cost.

## 4.2. Introduction

---

At the phase of “Introduction”, the website operator plans introduction and daily operation of the WAF and implement it as planned. This phase is completed when completing the introduction of the WAF and starting its operation.

In this section, the key points at this phase are explained in 4 actions: “coordination with relevant parties”, “introduction planning”, “operation planning” and “verification”. Here, it is supposed that “verification” is done at last and “coordination with relevant parties”, “introduction planning” and “operation planning” may be done in parallel in some cases.

### 4.2.1. Coordination with Relevant Parties

---

When introducing a WAF, the website operator had better explain the people who are affected by the introduction about what they will be expected to do. This section first explains who will be affected using 2 case studies, and then lists the issues the website operator should work with them.



Properly explaining things to the relevant parties will help the website operator closely cooperate with them to smoothly respond troubles while introducing and operating the WAF, should it is needed.

### Case Where Website Operator Introduce WAF

---

#### ■ Case Where Website Operator Sets Up Commercial WAF on Network

In the case where the website operator sets up a commercial WAF on the network, the “network administrator”, the “web application developer<sup>44</sup>” and the “WAF vender” will be those whom the website operator should work with in advance.

---

<sup>44</sup> In this example, the web application developer develops and maintains a WAF, but in reality, development and maintenance may be done by the different people/parties.

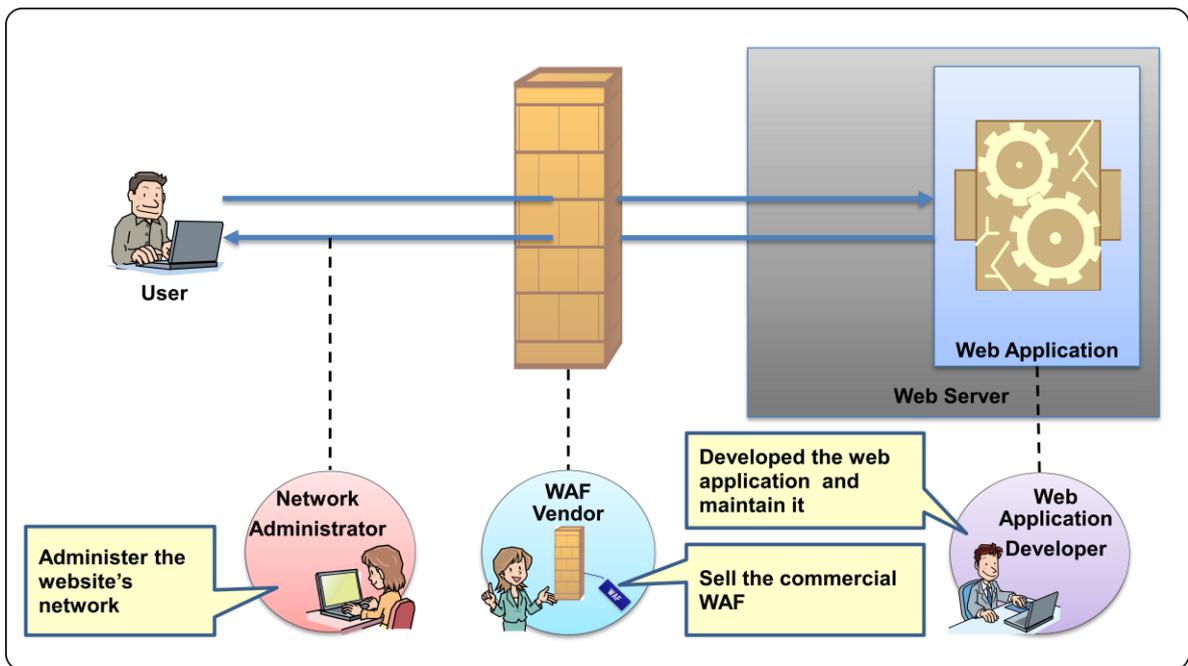


Figure 4-3 Parties Relevant to Introducing Commercial WAF on Network

■ **Case Where Website Operator Set Up Open Source WAF on Web Server**

In the case where the website operator sets up an open source WAF on the web server, the "web server administrator" and the "web application developer" will be those whom the website operator should work with in advance.

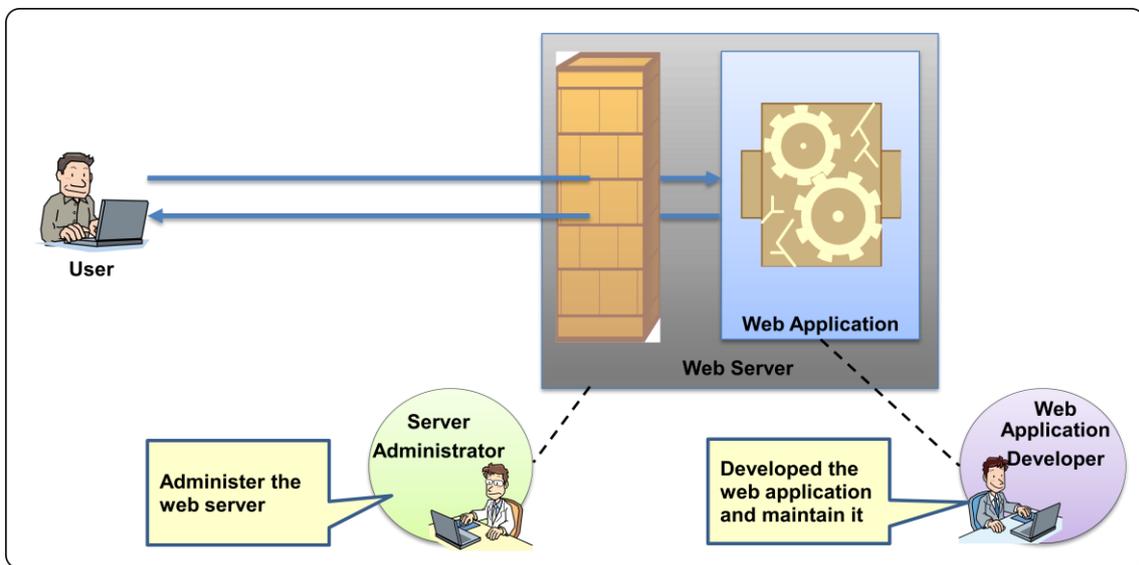


Figure 4-4 Parties Relevant to Introducing Open Source WAF on Web Server

## Relevant Parties and Issues to Work with Them

---

### ■ Network Administrator



**Network Administrator**

- The impact on the network configuration
- The impact on the network throughput (Number of packets processed per unit time)
- The impact on the network when the WAF fails

More or less, any network-based WAF will have an impact on the network configuration. For that, the website operator should better explain how the WAF introduction will affect the network in advance.

In addition, regardless of whether it is network-based or server-based, the WAF will communicate with the external servers depending on how it updates things such as itself and blacklist. To enable them, depending on the network configuration, the website operator needs to ask the network administrator for some work like changing the firewall settings.

### ■ Server Administrator



**Server Administrator**

- The impact on the server resources (e.g. CPU usage, memory usage, hard disk usage)
- The impact on the server behavior and operation
- The impact on the server when the WAF fails

More or Less, any server-based WAF will have an impact on the resources of the web server to be installed with the WAF. For that, the website operator should better explain how the WAF introduction will affect the network in advance.

In addition, there is a possibility that it will affect the web server configuration and may be necessary to install other software or change the settings of the server software. It could become a cause of web service failures. It is recommended to work with them and establish a framework for who will respond to the failure and what to do to what extent in advance.

### ■ Web Application Developer



**Web Application Developer**

- The impact on the support contract for the web application
- Collaboration to create and maintain the whitelist
- Collaboration to prepare for possible incidents (e.g. the occurrence of false positives)

There is a possibility that introducing a WAF may affect the support contract for the web applications and they may go out of the support by the developer. The website operator should ask the WAF vendor about the impact of WAF introduction on the support contract and such in advance.

Since the whitelist used by the WAF is defined based on the design of the web applications, it is important to have help of the web application developer to create a whitelist. Also, checking if there are the WAF rules that should not be on the blacklist to the developer may help introducing the WAF smoothly.

In addition, when a problem regarding the website availability after WAF introduction arises, an investigation of the cause, whether it is the WAF or web application, may be required. In such a case, it is possible that help of the web application developer will speed up the investigation.

■ **WAF Vendor**

An icon for 'WAF Vendor' showing a woman pointing to a stack of blocks and a blue tag labeled 'WAF'.

- The contact information for support
- Scope of support and responsibility

To be able to ask the support desk early on in case of finding out a problem such as the WAF failure, it is important to make sure to have the active contact information. Besides the contact information, also make sure that to what extent the WAF vendor will provide support. It may make it easier to respond to the problem.

## 4.2.2. Introduction Planning

---

To introduce the WAF smoothly, the website operator should create an introduction plan beforehand. Figure 4-5 shows the issues to be addressed in the introduction plan. By discussing these issues and establish a plan, the website operator can introduce the WAF and move to its operation smoothly.

**POINT**  To prepare for the WAF's configuration change or troubles after introduction, it is recommended to document the things that have been discussed during the planning as an introduction procedure.

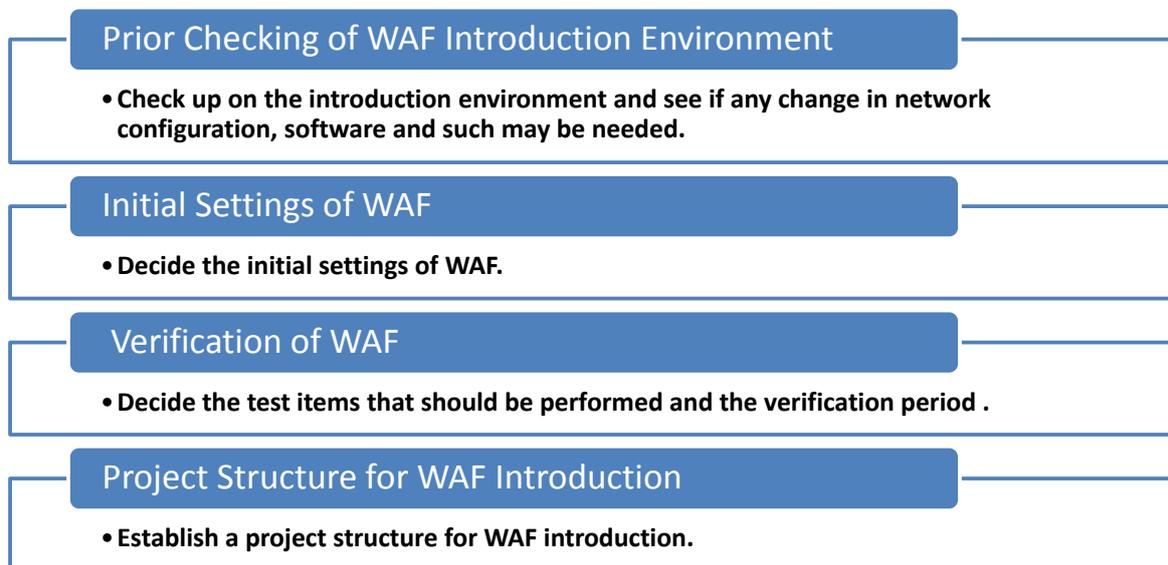


Figure 4-5 Issues to Be Addressed in WAF Introduction Plan

The key points of each issue listed in Figure 4-5 is summarized below.

### ■ **Prior Checking of WAF Introduction Environment**

When creating a WAF introduction plan, first the website operator should check up on the website and web server environment. For example, if setting up a network-based WAF, since it will have an impact on the network configuration more or less, the website operator should see like if any changes other than the settings to use the WAF may be needed. If setting up a server-based WAF, see like if there is additional software that should be installed to use the WAF.

## ■ Initial Settings of WAF

The issues that should be discussed in advance regarding the settings of the WAF at the time of introduction are addressed below.

### ◆ Scope of Inspection

Depending on the website's system configuration and where to deploy the WAF, the WAF can protect multiple web applications. Even so, in some cases, a website operator may want to protect only particular web application. In other cases, since the WAF may have a false positive, a website operator may set up the WAF to prevent only the specific attacks and not all attacks.

The website operator should list up the attacks to prevent and web applications to protect and examine the settings of the inspection functions and advanced functions.

### ◆ Logs

As discussed in the section 3.2.2, the WAF outputs multiple logs. Since not all logs are required for operation, it is recommended to decide which logs to get as well as where to output the logs and how long to store them.

## ■ Verification of WAF

As explained in the section 3.3, introducing a WAF does not mean it can prevent the all attacks that exploit vulnerabilities in all web applications. In addition, by introducing a WAF, it is possible that the WAF may block the legitimate HTTP connections and decrease the availability of the website. To prevent that, it is recommended not to put the WAF in operation right after introducing it and to have a period for operation verification testing.

At this phase, it is desirable to discuss things like what kind of tests should be done, what kind of results will be satisfactory and until when the verification testing should be done. For the details to be considered, see "4.2.4 Verification".

## ■ Project Structure for WAF Introduction

For WAF introduction, it is good to establish a collaborative project structure where the WAF introduction staff can cooperate with the persons in the relevant departments and organizations and obtain useful help and support. Also, it is desirable to clarify to whom to report a trouble, should it arise.

### 4.2.3. Operation Planning

---

Even if the WAF is effectively configured at the start of operation, it does not mean the WAF can keep preventing all attacks. To keep its effectiveness, it is necessary to periodically update the rules or update the WAF in some cases. To ease the operation, it is desirable to create an operation plan in advance.



To avoid the troubles after it is put in operation, it is recommended to document the things that have been discussed during the planning as an operating procedure.

## ■ **Operating Structure**

For WAF operation, it is good to establish and clarify a collaborative operating structure with which the WAF operation staff can cooperate with the persons in the relevant departments and organizations and obtain useful help and support. For the update of the rules that require the change of the WAF settings or the update of the WAF, establish a change approval flow that enables the change management.

Also, it is desirable to clarify to whom to report a trouble, should it arise.

## ■ **Operational Policy**

The WAF operation calls for various tasks (Figure 4-6). Besides the main staff, introducing and operating a WAF will require a lot of people's involvement WAF and it tends to make who is responsible for each work and the scope of responsibility blur. Thus, it is recommended to list up the tasks to be done in the operation plan and establish an operational policy discussing who is responsible for what and how to do it until when. In particular, make sure to decide who is responsible for each task in the operational policy.

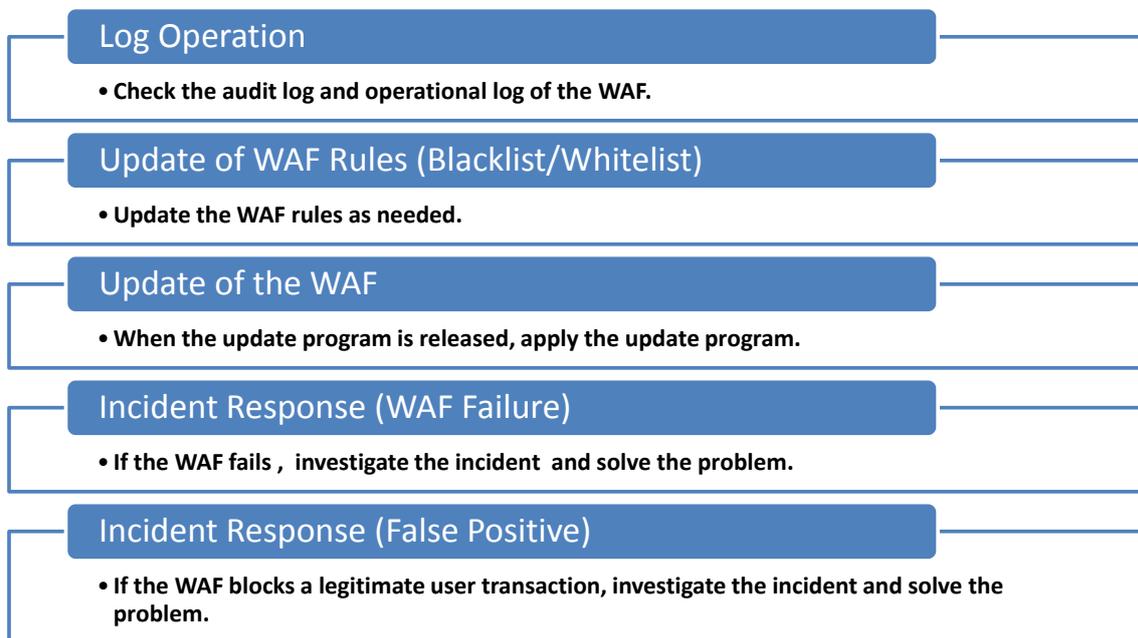


Figure 4-6 Tasks to Be Done for Operating WAF

The key points of the tasks for operating a WAF are summarized below.

### ◆ **Log Operation**

It is desirable to consider how often and for what purpose to obtain the logs in advance. In case of checking up on the occurrence of false positives or the tendency of the attacks, consider to analyze the audit logs frequently (for example, every several hours or, once a day). On the

other hand, if wanting to make sure that the WAF is working correctly, check the operational log”.

#### ◆ **Update of WAF Rules (Blacklist/Whitelist)**

It is important to consider how often and how to update the WAF rules and the possible effect of the update on the website.

The frequency and method to update the WAF rules differ depending on whether it employs the blacklist or whitelist approach. In the case of the WAF that employs the blacklist approach, update the rules as soon as the rule for the new attack becomes available. On the other hand, in the case of the WAF that employs the whitelist approach, revise the rules when changes are made to the web applications in use.

By updating the WAF rules, the users’ HTTP connection may be temporarily dropped. Also, by applying the new rules, false positives may occur. For that, it is recommended to establish the verification and update procedure for the new rules beforehand.

#### ◆ **Update of the WAF**

It is also important to decide how often to check up on the availability of the update program for the WAF, establish the procedure to apply the update program and analyze the impact it may have on the website. By updating the WAF, there is a possibility that the users cannot use the web services. For that, it is recommended to establish the update procedure and the update approval flow to authorize the update beforehand.

#### ◆ **Incident Response (WAF Failure)**

The website operator should establish the incident response structure and procedure. In case where the WAF fails and stops working, the users may become unable to use the web services. Especially if it is a hardware WAF and set up on the gateway of the web server and a trouble such as a hardware failure occurs to the WAF device, there is fear that other web services running on the same website will be also affected. The website operator should consider the extent of the impact of the WAF failure and under what incident response structure to respond to a failure should it happen.

#### ◆ **Incident Response (False Positive)**

The website operator should establish the workaround and incident response structure in case of the occurrence of a false positive. When it occurs, the users may become unable to use the web services. The website operator should consider under what incident response structure to respond to a false positive should it happen.

## 4.2.4. Verification

Before putting the WAF in operation, the website operator should set up a verification period to test its behavior and features. By testing it sufficiently, the website operator can avoid the troubles after the start of operation. In this section, the verification process is explained following the flow shown in Figure 4-7.

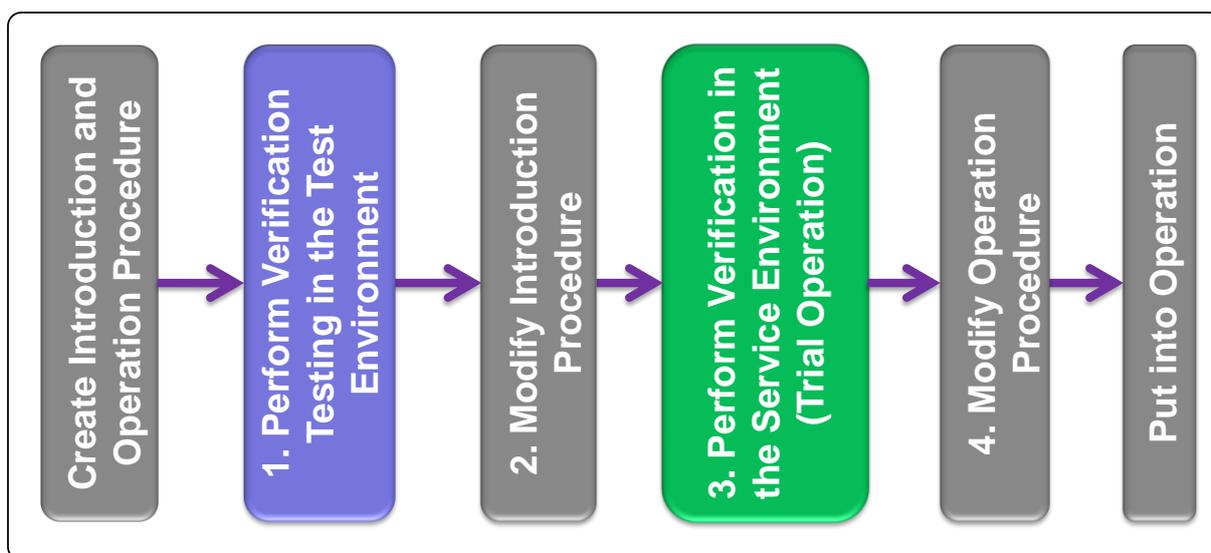


Figure 4-7 Flow of Verification

When verifying a WAF, firstly a test environment is used to avoid causing an impact on the web services. Here, using the test environment that is identical to the service environment will make the verification easier. If a test environment cannot be built by any possibility, start with “3 Verification in Service Environment (Trial Operation)”. In that case, there are the risks that the users may become unable to use the web services because of the unexpected events uncovered by the introduction procedure or errors in the introduction procedure.

After the verification in the test environment is completed, put the WAF in the service environment and start the trial operation. The website operator employs the passing mode (see “3.2.1 Basic Features”) not to affect the HTTP connections established by the users and check the behaviors and logs.

If no problem is discovered after all verification tests, the WAF is put into operation.

The key points in the verification are summarized below.

### (1) **Perform Verification Testing in the Test Environment**

The verification tests to be performed in the test environment are addressed below.

#### ◆ **Verification of Introduction Procedure**

By introducing the WAF into the test environment along the introduction procedure, verify the validity of the procedure. Regardless of where a WAF is deployed, the WAF stands

between the users and the web server. Thus, if there are errors in the introduction procedure, the web services may be shutdown. It is critical to use a test environment and verify that the introduction procedure does not have errors.

#### ◆ **Verification of False Positives**

The website operator needs to check that the WAF rules do not cause false positives. To prevent the website from becoming unable to provide the web services to its users as supposed to, test as much as needed to verify that the false positives do not occur. One method of verification is to manually access the website through a web browser and click all links to confirm that the accesses are not blocked.

If the website operator introduces the WAF, he or she needs to verify the false positives on his or her own. On the other hand, if the website operator has the WAF vendor introduce a WAF, ask the WAF vendor for help about verification for the false positives.

#### ◆ **Verification of False Negatives**

The website operator needs to check that the WAF rules do not cause false negatives due to the errors in the WAF rules. For example, if the rules are based on the blacklist approach, send a typical attack pattern that should be detected by the WAF to the website for each type of the attacks (such as SQL injection), and confirm that the attacks that exploit the vulnerabilities and should be prevented by the WAF are indeed detected by the WAF. On the other hand, if the rules are based on the whitelist approach, input a specific pattern that is unexpected by the web application to the web pages and the parameters that should be protected, and confirm that the WAF indeed detects them as unpermitted transactions.

If the website operator introduces the WAF, he or she needs to verify the false negatives on his or her own as well. On the other hand, if the website operator has the WAF vendor introduce a WAF, ask the WAF vendor for help about verification for the false negatives.

#### ◆ **Verification of Impact on Performance**

As needed, the website operator should measure the impact of WAF introduction on the website's performance from the following viewpoint.

- Response Time and Turnaround Time  
Measure the response time and turnaround time from the viewpoint of the users and confirm that WAF introduction is not affecting the use of the website.
- Throughput  
As for a network-based WAF, measure the number of HTTP requests that can be processed per unit time and confirm that the availability requirement continues to be satisfied after WAF introduction.
- Resource Consumption

As for a server-based WAF, measure the CPU usage, memory usage and hard disk usage of the web server installed with the WAF and confirm that the performance requirements continue to be satisfied after WAF introduction.

## **(2) Modify Introduction Procedure**

If the changes in the initial settings have been made or the errors have been found through verification in the test environment, review the introduction procedure. If the verification procedure has been changed through verification for the false positive or false negatives, review the introduction procedure as well.

## **(3) Perform Verification in the Service Environment (Trial Operation)**

The website operator introduces the WAF in the service environment and starts the trial operation. The tests in the test environment may not have covered all possible attempts the users may make. Therefore, if the WAF is put into operation right away, the troubles that affect the web services, such as a false positive blocking the user's connection, may occur. Unless the website is in urgent need of a counterattack like it is under the attacks and suffering damage, it is most recommended to have a trial operation period.

In the trial operation, the website operator can use the passing mode and check up on the availability of the website and the occurrence of false positives and false negatives. If the problems that could affect the operation of the website are found, solve them, get back to “2 Modify Introduction Procedure” and proceed with the verification.

Other operations such as “log operation”, “update WAF rules” and “update the WAF” should be performed following the operation procedure and make sure that the procedure has no errors.

## **(4) Modify Operation Procedure**

During the verification in the service environment, if problems are found in the operation process based on the operation procedure, review the operation procedure.

## 4.3. Operation

---

At the phase of “operation”, the website operator operates the WAF following the operation procedure. It is important to reaffirm that this phase does not end until the WAF is removed or the website is closed.

In this section, the key points at this phase are explained in 3 actions: “normal operation”, “incident response” and “maintenance”. It is assumed that each action may be done in parallel in some cases.

### 4.3.1. Normal Operation

---

To use the WAF effectively, it is important to perform the tasks addressed below.

#### ■ Update the WAF and Rules

Sometimes, the WAF vendors release the update programs for their WAF products. By applying those update programs, the features are improved and added, and in some cases, vulnerability in the WAF is fixed.

As for the WAF that employs the blacklist approach, the WAF vendor may also release an update of the blacklist that supports the new attacks. On the other hand, as for the WAF that employs the whitelist approach, it is needed to revise the WAF rules when changes are made to the web applications in use. By updating the WAF rules, the WAF can continue to protect the web applications from the attacks that exploit the vulnerabilities in them.

The website operator follows the operation procedure about whether or not to apply the updates of the WAF and rules. When updating the WAF and rules, it is important to verify them in the test environment first, as done in the introduction phase. If no problem is found in the test environment, apply the update in the service environment. If problems are found in the verification in the test environment, consider revising the operation procedure as well.



Perform the verification for each update. It is important to revise the operation procedure if the need arises (see Figure 4-8).

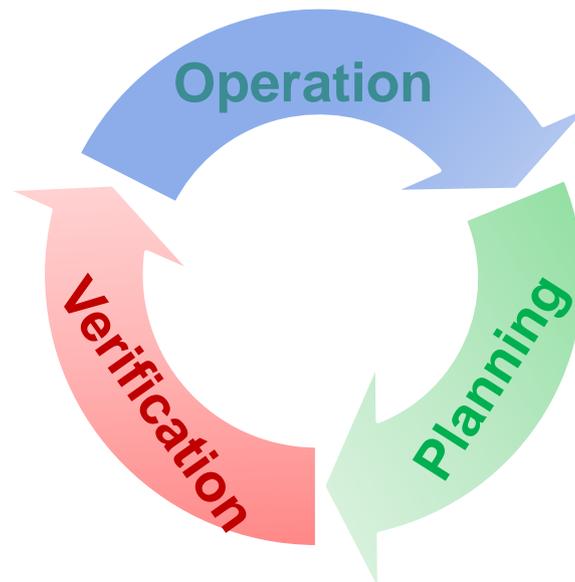


Figure 4-8 Update Operation Cycle

### ■ **Periodic Log Monitoring**

Following the operation procedure, the website operator check up on the WAF logs periodically.

By checking up on the audit logs, the occurrence of false positives and false negatives will be discovered. If false the positives have been identified, perform “4.3.2 Incident Response. Moreover, by analyzing the audit logs, it is possible to assess the current trends in attack techniques and use it as a guideline to improve the website security.

When analyzing the operational logs, look for any predictor that may hint a possible WAF failure. If the operational logs suggest a possible failure, take possible preventive measures (such as arranging the spare parts).

### **4.3.2. Incident Response**

---

Incident response may not occur frequently but once it is called for, it has a profound effect. It is recommended to have a periodic exercise expecting the following events.

#### ■ **WAF Failure**

If the WAF suffers a failure, act as defined in the operation procedure promptly.

#### ■ **False Positives**

Since the false positives have a big impact on the web services, .it is necessary to solve following the operation procedure as fast as possible.

### **4.3.3. Maintenance**

---

When using a commercial WAF, make sure to have a maintenance contract.

#### **■ Hardware Maintenance**

When using a commercial, hardware WAF, it is necessary to have a hardware maintenance contract. When using a server-based WAF, make sure to have a maintenance contract for the server hardware to which the software WAF is installed.

If the hardware is not supported by a maintenance contract, the WAF may not be able to work due to the hardware failure, and as a result, the website may become vulnerable to the attacks that exploit the vulnerabilities in the web applications.

#### **■ Software Maintenance**

To be able to use the update for the WAF and blacklist, it is necessary to have a software maintenance contract.

If the software is not supported by a maintenance contract, there is a possibility that the WAF may not be able to prevent the latest attacks because it cannot update the blacklist or fix the vulnerability in the WAF.

# 5. WAF Introduction Case Study at IPA

In Chapter 4, the key points for WAF introduction are explained. This chapter presents a case study of introducing and operating an open source WAF, “ModSecurity” at IPA. The know-how IPA has learned from its experience is also shared.

## 5.1. Introduction

In 2010, IPA has host a security seminar for website operators. At the seminar, IPA learned that the website operators have been having a hard time in finding the WAF related documents written in Japanese or they did not know what to do to introduce a WAF since there were little case studies available.

To answer their need, IPA has introduced a WAF, accumulated the knowledge about the WAF, like what it can do and what it cannot do, and publishes what IPA has learned as a case study.

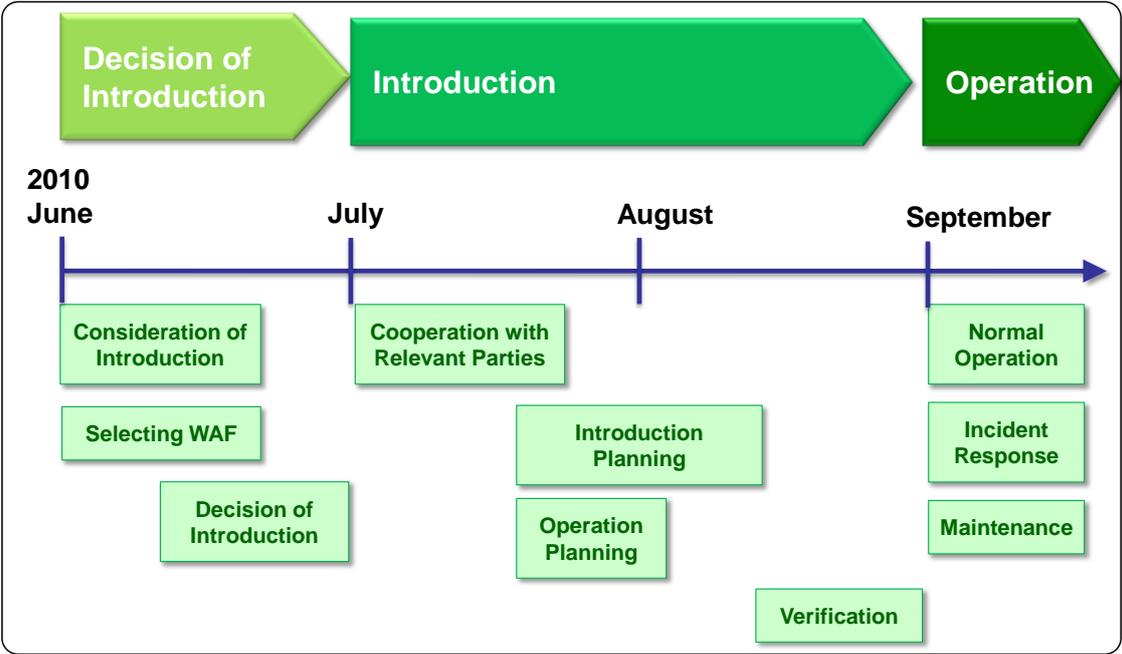


Figure 5-1 IPA WAF Introduction Schedule IPA

IPA has followed the “3 phases for introducing a WAF and putting it into operation” described in Chapter 4 (“decision of introduction”, “introduction” and “operation”) by the schedule shown in Figure 5-1. It took about 2 months. At the time of this writing, it has been for about 4 months since the start of its operation.

In the following sections, the real things like “what IPA has discussed over” and “what exactly IPA did” at each of 3 phases, are presented.

This time, IPA has introduced a WAF in the “JVN iPedia”<sup>45</sup>, a vulnerability countermeasure information database that collects and stores the vulnerability and countermeasure information about the software products (such as OS, applications, libraries, embedded systems) widely used in Japan.

Note that this chapter is explained from the viewpoint of the WAF Promotion Project staffs (hereinafter referred to as “we”) shown in Figure 5-2.

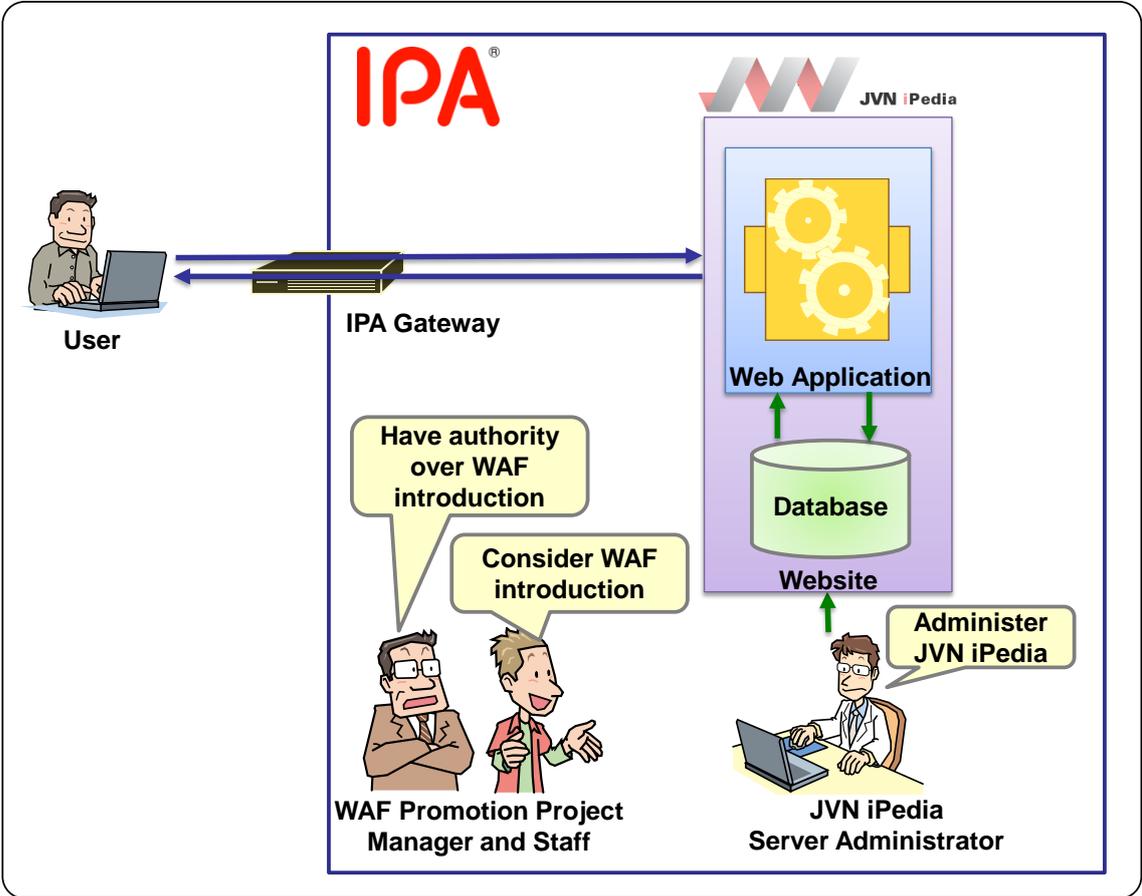


Figure 5-2 Parties Known to Be Relevant to Operation of JVN iPedia at the Time of Project Initiation

<sup>45</sup> <http://jvdb.jvn.jp/>

## **5.2. Decision of Introduction**

---

In this section, the phase of “decision of introduction” - the first of 3 overall phases – is presented. The key points at this phase are addressed in Chapter 4. Reading the previous chapter together will help understand the contents better.

### **5.2.1. Consideration of WAF Introduction [Case Study]**

---

For the key points in “Consideration of WAF Introduction”, see 4.1.1

For the reason mentioned in 5.1, IPA has introduced a WAF as a test project. No vulnerability was identified in JVN iPedia and there was no need to introduce a WAF in JVN iPedia.

In a normal situation, the website operator should consider whether or not to introduce a WAF following “2.4 Situations Where WAF is Effective” and “4.1.1 Consideration of WAF”.

### **5.2.2. Selecting WAF [Case Study]**

---

For the key points in “Selecting WAF”, see 4.1.2

To select a WAF, we discussed with the “WAF Promotion Project manager<sup>46</sup>” who had authority over whether or not to introduce a WAF. We also consulted with the administrator of the JVN iPedia server about its configuration.

#### **■ Discussion with WAF Promotion Project Manager**

First, we discussed about WAF introduction with the WAF Promotion Project manager (hereinafter referred to as “the Manager”). At this point, we explained the “purpose of introducing a WAF” and the “merits and demerits of introducing a WAF”, and the request to introduce a WAF in JVN iPedia was approved.

Next, we talked with the Manager about what kind of WAF to introduce and concluded that they should use an open source WAF instead of a commercial one.

The reason to employ an open source WAF was that since it was a test case, by using an open source WAF instead of a commercial WAF that was feature-rich and came with a solid support by the vendor, the project would face more challenges and could accumulate better knowledge from hands-on experience as a result.

---

<sup>46</sup> It is most desirable that the person also has authority over the budget for the web server operation.

## ■ Consultation with the JVN iPedia Server Administrator

We consulted with the administrator of the JVN iPedia server (hereinafter referred to as "the Server Administrator") about its configuration. The configuration of the JVN iPedia server learned from the Server Administrator was below.

- The OS on JVN iPedia is Linux.
- The web server software on JVN iPedia is Apache.

In addition, from the talk with the Server Administrator, we learned that the JVN iPedia web server acted as a reverse proxy for MyJVN<sup>47</sup>. MyJVN had a lot of dynamic contents, such as APIs. If the WAF inspected the transactions sent to MyJVN, there was a high risk of false positives and the MyJVN service could suffer negative effect. For that, we learned that "it was necessary to exempt the connections to MyJVN from the scope of the WAF inspection and blocking<sup>48</sup> when introducing a WAF". (Figure 5-3).

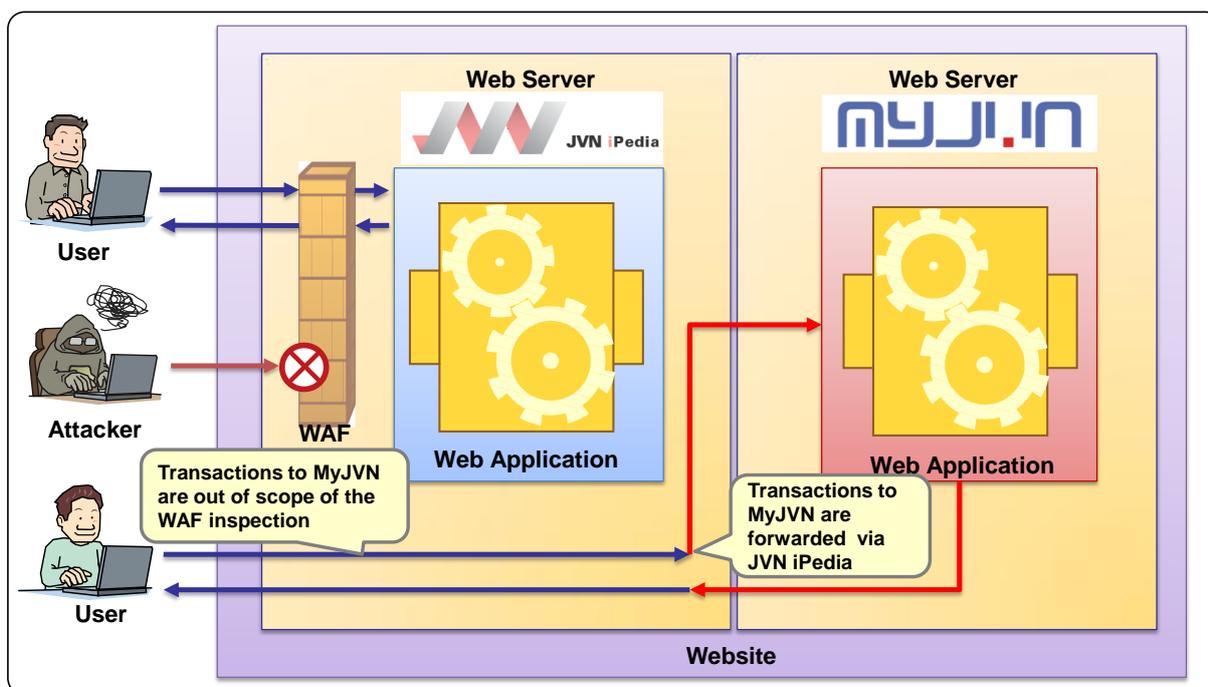


Figure 5-3 Relation between JVN iPedia and MyJVN

In addition, we asked the Server Administrator about who were involved with the operation of JVN iPedia and the role of them. The newly identified parties are listed below. The relationship among the relevant parties identified so far is presented in Figure 5-4.

- IPA's network administrator (the network is managed by a different section).
- The vendor that developed JVN iPedia

<sup>47</sup> <http://jvndb.jvn.jp/apis/myjvn/>

<sup>48</sup> The settings including the exemptions are addressed in "5.3.2 Introduction Planning [Case Study]".

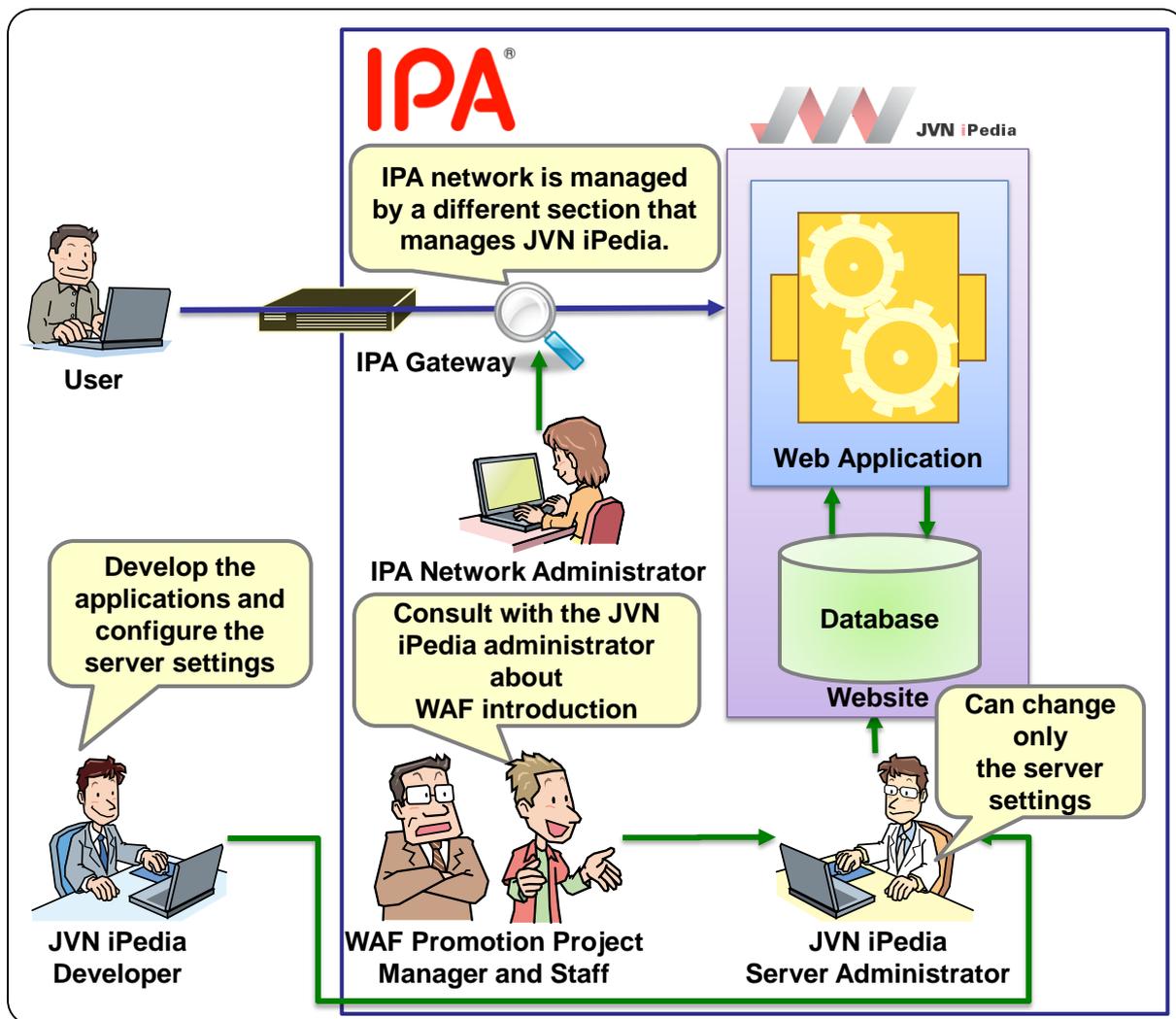


Figure 5-4 Real Parties Relevant to Operation of JVN iPedia



With pre-consulting, we could obtain the information needed to select a WAF.



With pre-consulting, we could obtain the information he should be careful about, such as MyJVN's involvement in the picture, when introducing a WAF.



With pre-consulting, we could find out the additional persons he also needs to talk to when introducing a WAF.

Based on the discussion with the Manager and consultation with the Server Administrator, we selected a WAF for this project.

The requirements for a WAF identified are below.

- In a normal condition, it is needed to estimate the total cost and draw up a requirement definition. However, it is a test case and to face more challenges and accumulate better knowledge from hands-on experience, using an open source WAF is desirable.
- It is difficult to place a burden on other section, like imposing changes in the network configuration, for the test case. Avoid the configuration change as much as possible.
- JVN iPedia is Linux-based and uses Apache as its web server software. Avoid making changes to the environment as much as possible.

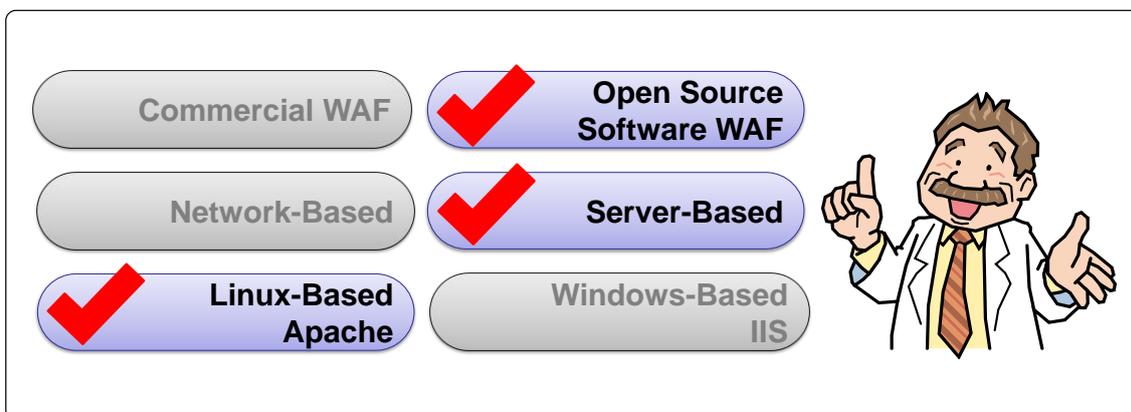


Figure 5-5 Requirements for IPA WAF

As the result of the consideration (Figure 5-5), we decided to use the following WAF.

**Result of Selection: Open Source Software WAF**

**ModSecurity<sup>49</sup>**

### 5.2.3. Decision of Introduction [Case Study]

Upon selection, we explained the process and result of the selection to the Manager. He also presented the initial and operational estimate cost for the selected WAF. After that, we received the final approval to introduce a WAF.

<sup>49</sup> In this guide, the details on ModSecurity are spared. As for the installation procedure, see “Appendix A: Open Source WAF”.

### 5.3. Introduction

In this section, the phase of “introduction” - the second of 3 overall phases – is presented. The key points at this phase are addressed in 4.2. Reading the previous chapter together will help understand the contents better.

#### 5.3.1. Coordination with Relevant Parties [Case Study]

For the key points for “Coordination with Relevant Parties”, see 4.2.1

We explained the plan to introduce a WAF to the persons identified in the consultation with the Server Administrator. The parties involved in this project and those with whom we made contact were shown in Figure 5-6.

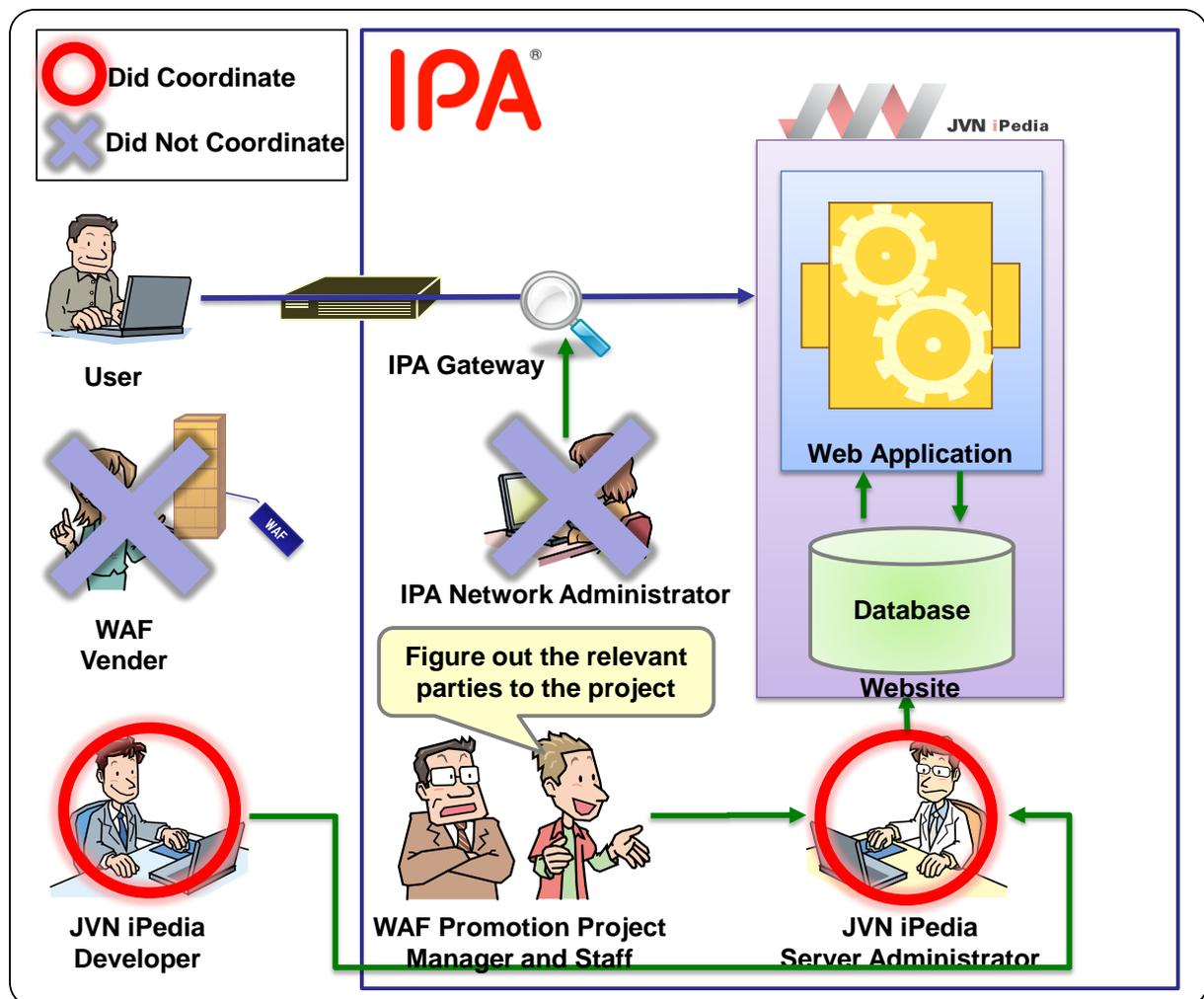


Figure 5-6 Relative Parties with Whom the Staff Made Contact Beforehand

In this section, the issues we had worked with them and the outcome of the talk in detail.

## ■ Coordination with IPA's Network Administrator

Because ModSecurity selected for this project was a server-based WAF<sup>50</sup>, no network configuration change was required. Therefore, there was no need for coordination with IPA's network administrator.



## ■ Coordination with the JVN iPedia Server Administrator

ModSecurity is a kind of WAF that works as an Apache module. We explained the Server Administrator about the possible risks associated with WAF introduction, such as that JVN iPedia would become temporarily unavailable for the users when installing and updating the WAF due to the reboot of Apache.



As discussed in 3.3, there is a possibility that errors like false positives could block the harmless connections made by the users. We explained the Server Administrator about those risks regarding the operation of JVN iPedia as well.

In addition, we also explained the Server Administrator that the project was giving careful consideration to the operation verification testing and the elimination of false positives before putting the WAF into operation to mitigate the risk as much as possible.

As a result, we received the approval from the Server Administrator.

We had a talk with the Server Administrator before at the phase of “decision of introduction”. Of course, we briefly told him about the purpose of the project, the merits and risks at that time. One reason why we explained them again was the fact that the product had been selected unlike the last time. Since the details of the WAF became clear, we reviewed its characteristics and risks and explained them in detail.

## ■ Coordination with JVN iPedia Developer

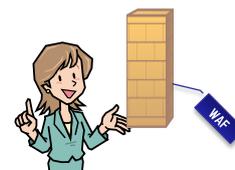
JVN iPedia is a web application developed by a vendor selected under a competitive bidding method. We consulted with the vendor if there was a possibility that introducing ModSecurity would affect the web applications for JVN iPedia.



As a result, we could make sure that WAF introduction would not affect the web applications for JVN iPedia.

## ■ Coordination with WAF Vendor

ModSecurity is an open source WAF and does not offer a support service. Therefore, there was no need for coordination with the WAF vendor.



---

<sup>50</sup> For the characteristics of server-based WAF, see 3.1.2.

### 5.3.2. Introduction Planning [Case Study]

For the key points for “Introduction Planning”, see 4.2.2

After the coordination with the parties relevant to WAF introduction was done, we created an introduction plan. IPA planned to introduce the WAF in the order shown in Figure 5-7.

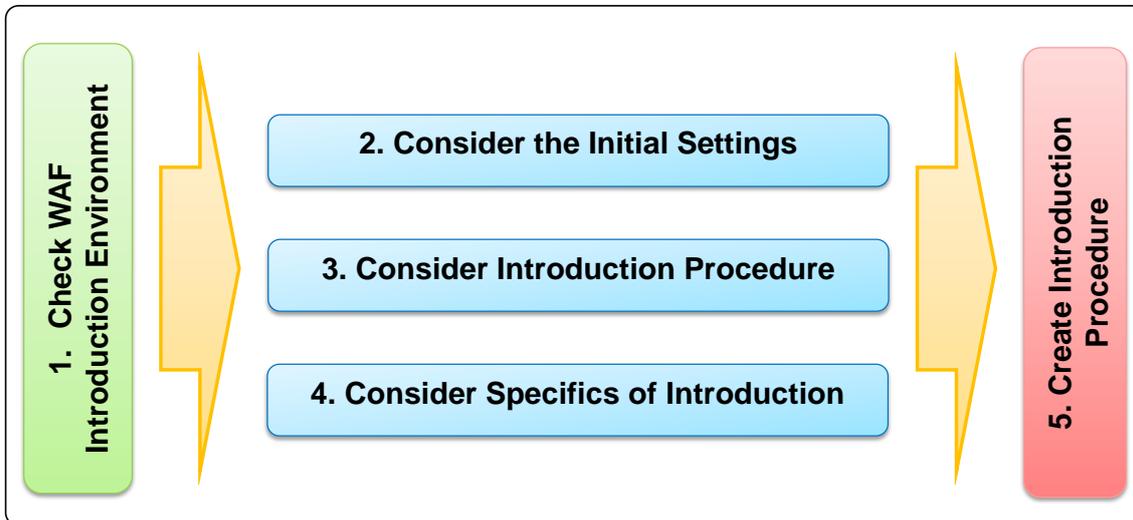


Figure 5-7 IPA’s Introduction Plan

In the following sections, the issues we had worked on and the result of consideration in the introduction planning phase are introduced along each step in Figure 5-7.

#### (1) **Check WAF Introduction Environment**

To create an introduction plan, we looked into the following things about the details on the JVN iPedia web server environment in advance.

##### ◆ **Hardware Configuration**

When looking into the hardware spec of the web server to be installed with the WAF, we mainly checked up on the remaining hard disk space. This was because the amount of overall log files outputted to the hard disk would increase by installing ModSecurity.

It turned out that the JVN iPedia web server did not have a sufficient disk space. As a result, we decided to revise and adjust the amount of the log output while planning the specifics (the detail is showed later in " (2) Consider the Initial Settings").

##### ◆ **Software Configuration**

To install ModSecurity, some prerequisite software must be installed beforehand<sup>51</sup>. We checked if those prerequisite software were installed on the JVN iPedia web server.

According to the result, some software must be installed when installing ModSecurity.

<sup>51</sup> For more information, see "Appendix A. Open Source Software WAF"

## (2) Consider the Initial Settings

### ◆ Decide the rules for ModSecurity

Since this was a test case, one of the purposes of this project was to evaluate the effectiveness of the “Core Rule Set”<sup>52</sup> bundled with ModSecurity.

However, if all the rules included in the Core Rule Set were applied, there would be a high possibility of errors such as false positives to happen and could affect the web service’s operation. Thus, IPA decided to enable only the rule to detect SQL injection attacks which would inflict serious damage should they succeed.

The rule in the Core Rule Set enabled is the following one.

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

### ◆ Decide the Version of ModSecurity and Core Rule Set

The latest version of ModSecurity and the Core Rule Set were installed.

The latest version at the time of introduction was listed below.

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

### ◆ Decide whether or not to output log files and Period of Retention

ModSecurity outputs 3 kinds of log files: filtering log<sup>53</sup>, audit log and operation log. IPA discussed over whether each of these log files were needed.

The purpose of introducing a WAF in this project was to confirm what a WAF could do and what could not. Thus, we decided to output all log files and analyze the behavior of the WAF. However, as mentioned in " ( 1 ) Check WAF Introduction Environment", the JVN iPedia web server did not have sufficient hard disk space. Therefore, we settled that we would retain the audit log and operation log for a shorter period than the access log of the web server.

Retention Period of Audit Log and Operation Log

- 5 Generations (for 5 weeks)

### ◆ Decide Other Settings

As presented in 5.2.2, JVN iPedia acts as a reverse proxy for MyJVN. Since MyJVN provided a lot of dynamic contents, such as APIs, we estimated that the possibility of false positives to happen would be high. The occurrence of false positives would have a big impact on the web service, we decided that ModSecurity would not inspect the transactions to MyJVN.

Here, we present an example of configuration where the transactions to MyJVN are out of scope of inspection. We specified the following setting in the Apache configuration file. This

---

<sup>52</sup> “Core Rule Set” is the rules developed by OWASP. For more information about OWASP and “Core Rule Set”, see “1.3.2 OWASP”.

<sup>53</sup> ModSecurity can outputs the result of detection to the error log for Apache separately from the audit log.

setting specifies that ModSecurity will not inspect the HTTP transactions whose accessing URI are “/en/apis/myjvn” or “/apis/myjvn”.

(Example) The Setting That Exempt HTTP Transaction to [MyJVN] from Inspection

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

### (3) Consider Introduction Procedure

If the JVN iPedia service shuts down due to the introduction of ModSecurity, the users cannot use JVN iPedia. Thus, we decided to perform an operation verification testing and the test to eliminate false positives in the test environment (virtual environment) before introducing ModSecurity into the service environment.

The introduction procedure employed by IPA is shown in Figure 5-8.

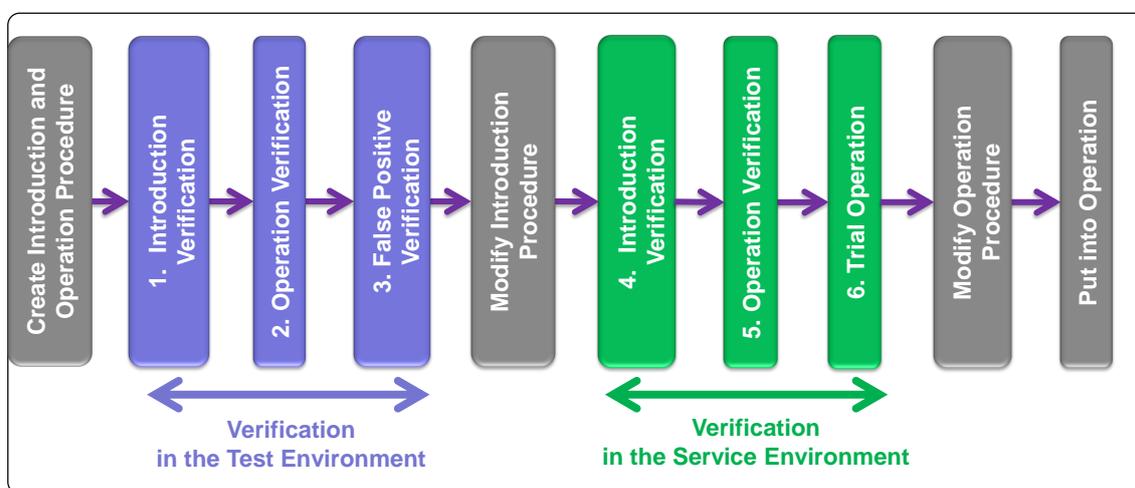


Figure 5-8 Introduction Procedure Taken by IPA

The system configuration of the test environment built in the virtual environment, including software applications and their version, was about the same as that of the service environment. What to do in the verification testing were also considered at this point. The detail is showed late in 5.3.4.

### (4) Consider Specifics of Introduction

Here, we considered non-technical details of how to introduce ModSecurity in JVN iPedia.

#### ◆ Decide Target Day and Notify Users

When introducing ModSecurity, the JVN iPedia service would become temporarily unavailable. The operational policy for JVN iPedia requires that IPA must notify the users when it becomes unavailable for maintenance. Thus, we set the target date to introduce ModSecurity and arranged with the administrator of JVN iPedia about the date when to put the notification on the web page.

## ◆ Clarify Project Schedule and Contact List

We draw up and clarified the project schedule. To be specific, in addition to the main staff working for WAF introduction, we also made those who had authority over the continuation of the introduction at the time of emergency available when their presence was required.

Also, we clarified the contact list of people we would need to contact should something happen, such as the service shutdown due to the introduction.

## (5) Write Up Introduction Procedure

For 2 purposes listed below, we wrote up the introduction procedure based on the result of the consideration in (1)~(4).

- To reduce operation errors such as typos during the introduction as much as possible.
- To enable anyone to introduce just by following it in case of staff rotation.

The introduction procedure for our project includes the following contents (Figure 5-9).

**Introduction Procedure**

1. ModSecurity Introduction Procedure
  - 1.1. Installation Procedure for Prerequisite Software
  - 1.2. Installation Procedure
  - 1.3. Verification Procedure
2. ModSecurity Configuration Change Procedure
  - 2.1. Procedure for Changing Log File Management Setting
  - 2.2. Procedure for Changing the Rules (Signatures)
  - 2.3. ModSecurity Update Procedure
  - ⋮

**TIPS**

- A. Procedure for Changing ModSecurity Operation Mode
- B. Procedure to Remove ModSecurity
- ⋮

I see...these procedures are written in the introduction procedure.

Figure 5-9 Contents of Introduction Procedure for IPA Project

### 5.3.3. Operation Planning [Case Study]

For the key points for “Operation Planning”, see 4.2.3

At the same time of planning the introduction, we also planned the operation of the WAF. In this section, the issues we had worked on and the result of consideration in the operation planning phase are presented.

#### (1) **Establish Operational Policy**

We established an operational policy that set down who is responsible for what and how to do it until when. We introduce the real operational policy used by IPA. The “who” mentioned here were the parties presented in Figure 5-4.

##### ◆ **Update Policy of ModSecurity**

To update ModSecurity, it is necessary to reboot Apache and the JVN iPedia service may be temporarily halted. Therefore, we decided that we would not always update the WAF when the new version of ModSecurity was released but update it only when necessary.

IPA defined the “update policy for ModSecurity” as below (Figure 5-10).

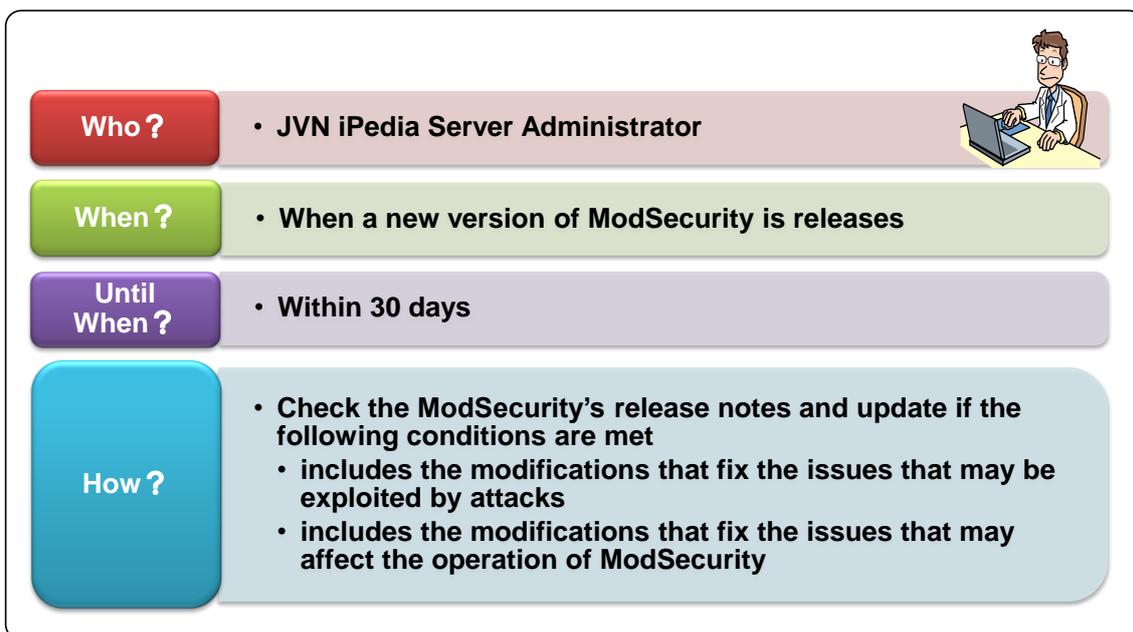


Figure 5-10 Update Policy for ModSecurity

##### ◆ **Update Policy for Core Rule Set**

To update the Core Rule Set, it is also necessary to reboot Apache as with ModSecurity. Therefore, we decided that we would not always update the Core Rule Set when the new one was released but update it only when necessary.

IPA defined the “update policy for the Core Rule Set” as below (Figure 5-11).

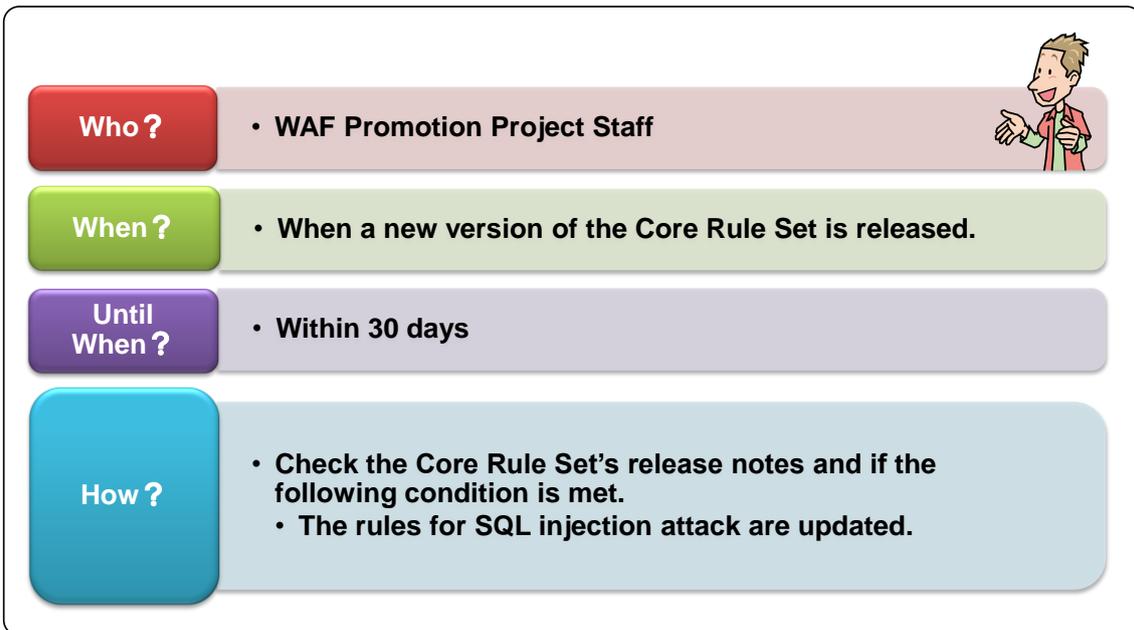


Figure 5-11 Update Policy for Core Rule Set

◆ **Log Checking Policy**

Considering the impact of false positives on the service, we have set up a system to find false positives early on by checking the ModSecurity log files daily.

In the early days after the ModSecurity introduction, IPA defined the 'log monitoring policy' as below (Figure 5-12).

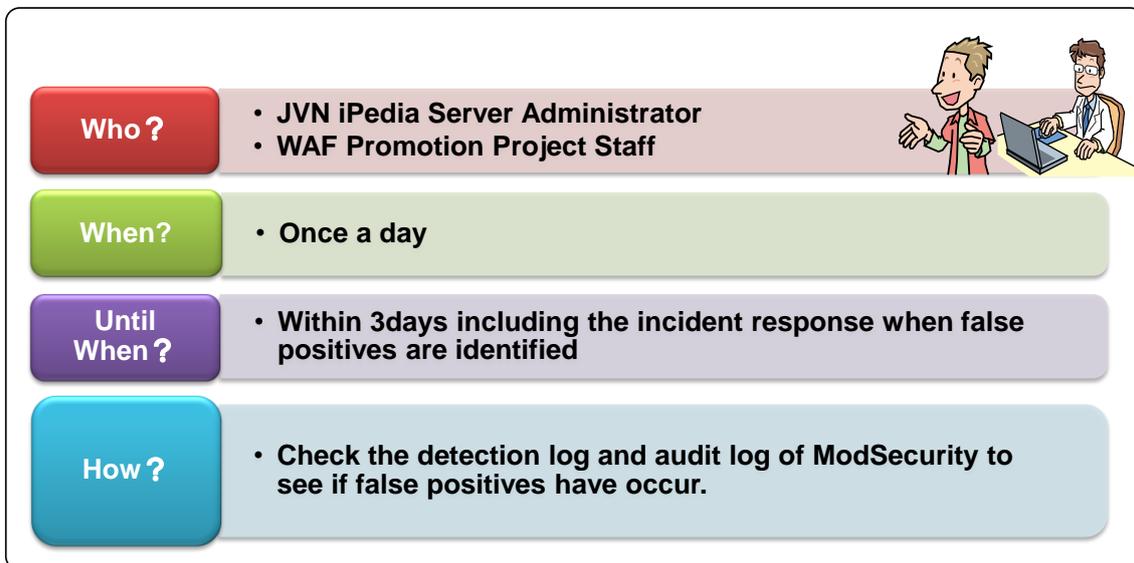


Figure 5-12 Log Monitoring Policy for ModSecurity Log Files

From the start of the operation to early January of 2011, no false positives that affected the JVN iPedia service have been identified. Therefore, the frequency of log monitoring has been changed to once a week at the time of this writing.

### ◆ **False Positive Response Policy**

We thought that if a false positive was found, it was important to deal with it as soon as possible just like log monitoring.

IPA defined the “false positive response policy” as below (Figure 5-13).

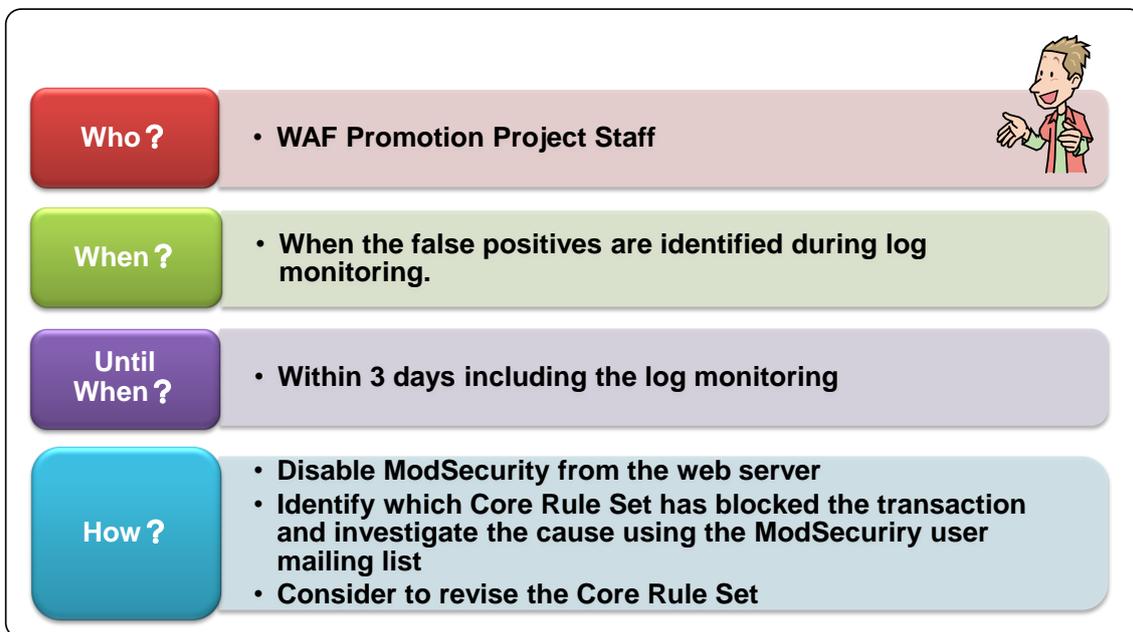


Figure 5-13 False Positive Response Policy

### ◆ **ModSecurity Incident Response Policy**

If ModSecurity suffers a failure, it may affect the operation of JVN iPedia, such as through the shutdown of Apache. Since a ModSecurity failure would have a big impact on the services, we decided to respond to this kind of incidents immediately.

IPA defined the “ModSecurity incident response policy” as below (Figure 5-14).

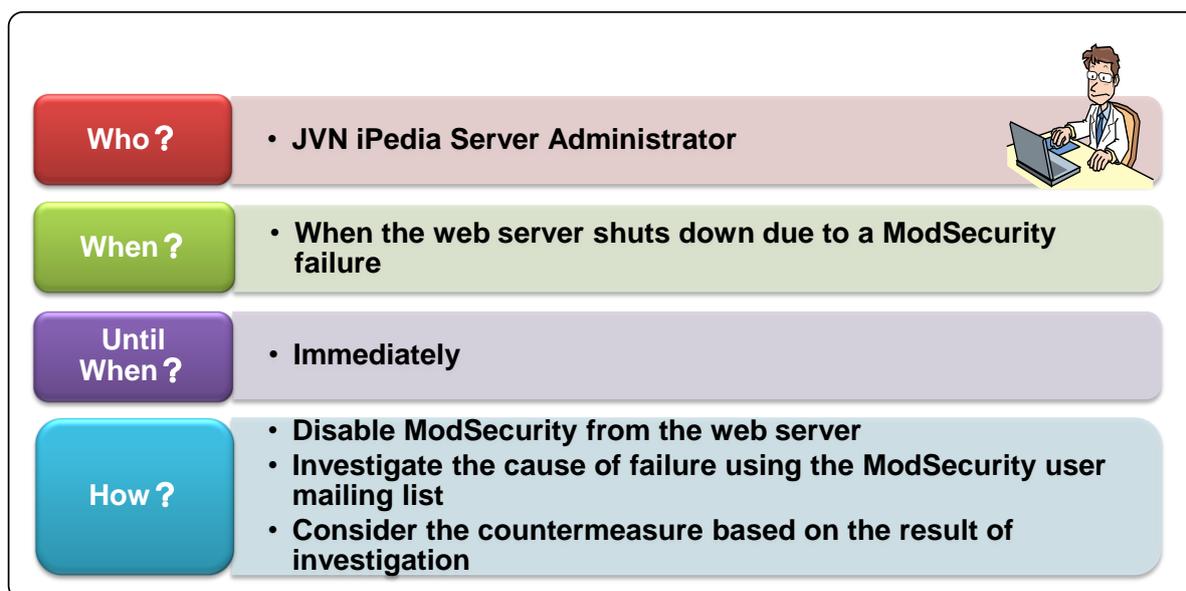


Figure 5-14 ModSecurity Incident Response Policy

## (2) Write Up Operation Procedure

For the same purposes of documenting the introduction procedure, we wrote up the operation procedure based on the operation policies during operation planning.

The introduction procedure for our project includes the following contents (Figure 5-15).

**Operation Procedure**

1. Overview of Operation
  - 1.1. List of Regular Tasks
  - 1.2. List of Non-Regular Tasks
2. Work Flows and Procedures
  - 2.1. ModSecurity Update
  - 2.2. OWASP Core Rule Set Update
  - 2.3. ModSecurity Log Monitoring
  - 2.3. Incident Response Procedure (False Positives)
  - 2.4. Incident Response Procedure (ModSecurity Failure)
  - ⋮

Anyone could do it using this operation procedure.

Figure 5-15 Contents of Operation Procedure for IPA Project

### 5.3.4. Verification [Case Study]

For the key points for “Verification”, see 4.2.4

After developing the introduction plan and operation plan, we proceeded to introduce the WAF and moved to verify it in the test environment. The flow and the result of the verification testing were shown in (Figure 5-16).

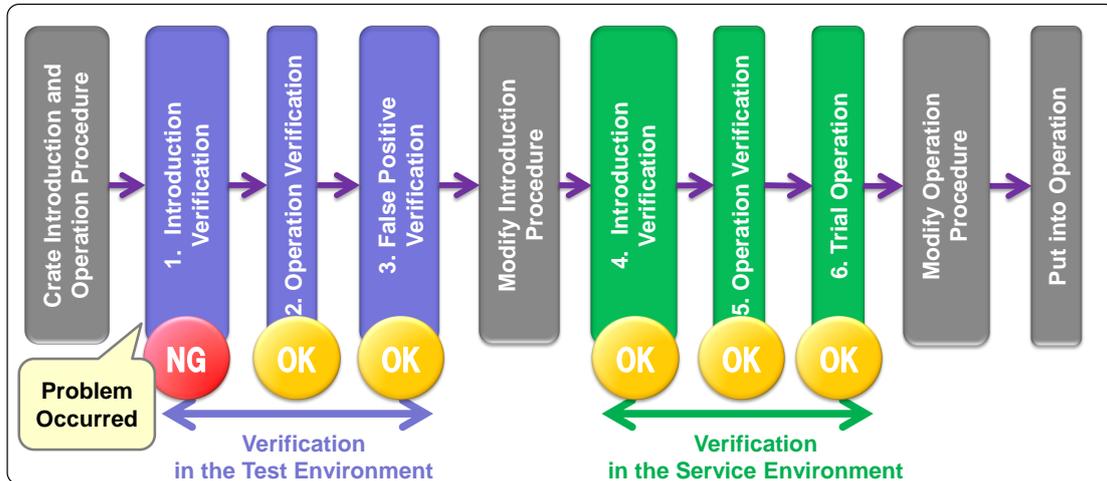


Figure 5-16 Flow and Result of IPA's Verification Testing

First, we have performed 3 tests from 1 to 3 in the test environment. We investigate the cause of the problems found in the Test 1 and revised the introduction procedure before the verification testing in the service environment. Next, we have performed 3 issues test from 4 to 6 in the service environment. We made sure that the WAF worked as expected in the test 6 and put the WAF into operation.

In the following sections, the test items for each test in Figure 5-16 and the result of them are presented. As for the Test 1, where the problems were identified, how we dealt with it is also presented.

## (1) Introduction Verifivation [Test Environment]

### ◆ Test Items

First, install ModSecurity to JVN iPedia set up in the test environment (virtual environment) using the introduction procedure. After installation, reboot Apache and confirm that Apache starts up.

### ◆ Test Results

After we installed ModSecurity and rebooted Apache, Apache did not start up.

### ◆ Countermeasures

We looked into the ModSecurity mailing list archive<sup>54</sup> and found out that the same issue had been reported. We took the countermeasure given on the list<sup>55</sup> and confirmed that Apache did start up.



## (2) Operation Verification [Test Environment]

### ◆ Test Items

Enter the character strings that should be detected by the Core Rule Set configured for ModSecurity (for example, @@version) in the search form of JVN iPedia and send the search requests. Confirm that the search results are not returned and the connections are blocked<sup>56</sup>. Confirm that the detection log entries and audit log entries are outputted as well.

### ◆ Test Results

We entered the character strings that should be detected by ModSecurity in the search form of JVN iPedia and confirmed that the search results were not returned and the connections were blocked. Also, we confirmed that the filtering log entries and audit log entries were outputted.



## (3) False Positive Verification [Test Environment]

### ◆ Test Items

Recreate the HTTP requests of the JVN iPedia users and send them to JVN iPedia. Check the filtering log and confirm that they are not blocked by ModSecurity. If there is a blocked case, check the audit log for its contents.

<sup>54</sup> [http://sourceforge.net/mailarchive/forum.php?forum\\_name=mod-security-users](http://sourceforge.net/mailarchive/forum.php?forum_name=mod-security-users)

<sup>55</sup> We changed the compiler option of ModSecurity. It is now covered in the ModSecurity manuals.

<sup>56</sup> When the result is returned, the web server replies with an HTTP response code 200. This time, we set up the web server to reply with an HTTP response code 403 if Modsecurity blocks the request. For this test item, we confirmed that we got the HTTP response code 403 to our requests.

To recreate the users' HTTP requests, we have used the real access log of JVN iPedia and iLogScanner this time. To be specific, first we identified the requests that were deemed an attack in the access log using iLogScanner. Then, by removing them from the access log, we extracted the legitimate requests by the users. We recreated HTTP requests of the users based on those log entries and sent to JVN iPedia set up in the test environment.

Note that we could use the access log to recreate the HTTP request because all requests to JVN iPedia were fetched by the GET method. If the web application uses the POST method as well, the web operator needs to recreate the requests made through the POST method which cannot be recreated from the access log when testing for false positives.

#### ◆ **Test Results**

We recreated the HTTP requests of the users from the one-month access log of JVN iPedia and tested for false positives and **confirmed that nothing was recorded to the filtering log**. Therefore, we judged that the Core Rule Set used in the verification testing would cause no false positives inflict the normal use of JVN iPedia by the legitimate users.



#### ■ **[Additional Info]:False Negative Verification**

During “ ( 3 ) False Positive Verification [Test Environment]”, we also tested for false negatives.

As a result, we have identified 4 cases of false negatives. We assume that it was attributed to the difference between iLogScanner and the Core Rule Set. We judged that there were attacks that the Core Rule Set used in the verification testing could not prevent.

As for those attacks that we could not prevent, updating the Core Rule Set solved the problem (see 5.4.1. ( 2 ) ).

Just like the false positive verification, we used the real access log of JVN iPedia and iLogScanner for the false negative verification. To be concrete, we identified the requests that were deemed an attack in the access log using iLogScanner. Then, we recreated malicious HTTP requests that exploit the vulnerabilities based on them and sent to JVN iPedia.

#### ( 4 ) **Introduction Verification [Service Environment]**

##### ◆ **Test Items**

As in the test environment (virtual environment), install ModSecurity to JVN iPedia in the service environment using the introduction procedure. However, unlike the verification testing in the test environment (virtual environment), change the ModSecurity setting<sup>57</sup> not to block the connection when it detects an attack in consideration of the impact that the

---

<sup>57</sup> We changed the SecRuleEngine setting of ModSecurity to DetectionOnly. For more information on this setting, see “Appendix A. Open Source Software WAF”.

occurrence of false positives may have on the service. After installation, reboot Apache and confirm that Apache starts up.

#### ◆ **Test Results**

After we installed ModSecurity and rebooted Apache, we confirmed that Apache started up.



### (5) **Operation Verification [Service Environment]**

#### ◆ **Test Items**

Enter the character strings that should be detected by the Core Rule Set (for example, @@version) in the search form of JVN iPedia and send the search requests. Confirm that the filtering log entries and audit log entries are outputted. Unlike the verification testing in the test environment, Confirm that the search results are returned<sup>58</sup>.

#### ◆ **Test Results**

We entered the character strings that should be detected by ModSecurity in the search form of JVN iPedia and confirmed that the filtering log entries and audit log entries were outputted. Also, we confirmed that the search results were returned.



### (6) **Trial Operation [Service Environment]**

#### ◆ **Test Items**

Leaving the setting not to block the attacks, use ModSecurity for about 2 weeks. During the trial operation, monitor the logs daily as defined in the operation procedure and confirm that no false positives are occurring and ModSecurity has been detecting the attack requests.

#### ◆ **Test Results**

As the result of the two-week trial operation, we confirmed that no false positives occurred while 152 cases of SQL injection attacks were detected.

Based on the result of the trial operation, we changed the setting back to blocking the connection when it detects an attack, and put ModSecurity into operation.



By performing the verification in the test environment, we could avoid the installation problem identified in the Test 1 in the service environment. Thus, the JVN iPedia service has escaped a service shutdown.

<sup>58</sup> In this test, unlike the verification in the test environment, since ModSecurity is set not to block the requests, the HTTP response code 200 is returned even if an attack is detected.

## 5.4. Operation

---

In this section, the phase of “operation” - the first of 3 overall phases – is presented based on our experience in having introduced and operated ModSecurity.

### 5.4.1. Normal Operation [Case Study]

---

For the key points for “Normal Operation”, see 4.3.1

In this section, we introduce the normal operation at IPA at the time of this writing.

#### (1) **Periodic Log Checking**

Based on the operation procedure developed in operation planning, we periodically check up on the filtering log outputted by ModSecurity. Here, we introduce how we monitor the filtering log and past performance at the time of this writing.

##### ◆ **Checking Method**

Using iLogScanner, IPA monitors the trends in attacks against JVN iPedia from the filtering log ModSecurity outputs. iLogScanner V3, a tool to detect the attacks against the websites released in August 2008, has a new feature that analyses the filtering result of ModSecurity based on the error log of Apache. It is handy to use this feature of iLogScanner for monitoring the log files.

Also, we check the result of iLogScanner against the audit log to make sure that ModSecurity has not been blocking the HTTP requests by mistake (false positives). The attack trends learned from log analysis can be used as a guide to come up with the security countermeasures. It is recommended to check up on the log files as a routine to find false positives as early as possible or knowing attacks against the web server.

##### ◆ **Past Performance**

The number of detection of SQL injection attack and occurrence of false positives at the time of this writing is shown below (Table 5-1).

Table 5-1 Number of Detection of SQL Injection Attacks and Occurrence of False Positives

Month	Detection of SQL Injection Attacks	Occurrence of False Positives
Sep. 2010	149 Cases	0 Cases
Oct.	14 Cases	0 Cases
Nov.	27 Cases	0 Cases
Dec.	125 Cases	1 Case <sup>59</sup>

---

<sup>59</sup> It was not a false positive that would have affected the operation.

Also, we learned that ModSecurity can detect the attacks hidden in the data sent through the POST method, which cannot be detected with iLogScanner.



Like being able to inspect the data sent through the POST method, we could confirm that the unique features of the WAF are working effectively.

## (2) **Update of ModSecurity and Core Rule Set**

Based on the operation procedure developed in operation planning, we periodically check whether a new version of ModSecurity or the Core Rule Set has been released. If a new version is available, we look into what has been changed and create an update plan. Before we update, we put it through verification in the test environment<sup>60</sup>.

In the following section, we introduce IPA's update works for ModSecurity and Core Rule Set up until the time of this writing of the second edition.

### ◆ **Update to Core Rule Set v2.0.8**

During the trial operation of ModSecurity, the Core Rule Set v2.0.8 was released on August 27, 2010. When we looked into what had been changed, the rules for SQL injection attack were updated. Thus, we considered the update to the Core Rule Set v2.0.8 based on the operational policy defined in the operation procedure.

First, we have tested the Core Rule Set v2.0.8 in the test environment based on the introduction procedure. With the verification testing based on the procedure, we could make sure that the update would not cause a problem, we updated the Core Rule Set used in the service environment to v2.0.8.

As a result, we have become able to detect and block some SQL injection attacks that we could not detect with the earlier version of the Core Rule Set.



By updating the rules, the capability of the WAF 's attack detection capability.

### ◆ **Consideration of Update to Core Rule Set v2.0.10**

The Core Rule Set v2.0.9 and Core Rule Set v2.0.10 were released in a row on November 18 and 20, 2010, respectively. In both versions, the rules for SQL injection attack were updated. Thus, we considered the update to the latest version based on the operational policy defined in the operation procedure.

When we tested the Core Rule Set v2.0.10 in the test environment based on the introduction procedure, we found that using the Core Rule Set v2.0.10 as it is may cause false positives. To be specific, say there exist the products named "XXX and YYY" and if a user puts

---

<sup>60</sup> Since ModSecurity is already in operation, we did not go through the trial operation described in "5.3.4 Verification [Case Study]".

a search for those names on JVN iPedia, ModSecurity would block the request. For this reason, we have not updated the Core Rule Set to the latest version at the time of this writing.

#### ◆ **Consideration of Update to ModSecurity v2.5.13**

On November 29, 2010, the ModSecurity v2.5.13 was released. We confirmed that the update included the changes that would have affected the behavior of ModSecurity. Thus, we considered the update to the ModSecurity v2.5.13 based on the operational policy defined in the operation procedure.

When we verified the installation procedure of the ModSecurity v2.5.13 in the test environment based on the introduction procedure, errors occurred and the installation failed. At the time of this writing, we have not yet been able to find workarounds to this problem in the ModSecurity mailing list or any other way. For this reason, we put off the update to the ModSecurity v2.5.13.



There is a possibility that the update of ModSecurity could fail or cause a trouble, and that the update of the Core Rule Set may cause false positives.

#### **5.4.2. Incident Response [Case Study]**

---

For the key points for “incident response”, see 4.3.2

We expect 2 patterns of incident response.

- Failure of ModSecurity
- Occurrence of False Positives

Until the time of this writing, no incident response had been called for. However, how to respond an incident is defined in the operation procedure (showed in 5.3.3) and we are to follow it should a situation arise.

#### **5.4.3. Maintenance [Case Study]**

---

For the key points for “maintenance”, see 4.3.3

Since ModSecurity is an open software WAF, its user cannot have a maintenance contract with the WAF vendor. Thus, if something is wrong with ModSecurity, the almost only place the user can turn to for help is the user mailing list. In fact, when IPA faced the problem where Apache did not start up in the verification testing of ModSecurity (see “5.3.4 Verification [Case Study]”), we checked out the mailing list and solved the problem.

The problem IPA ran into was lucky enough to solve by searching through the mailing lists since other users had also been experiencing the same problem, but some problems may not

be able to solve. When considering the use of an open source WAF, the website administrator needs to think about this issue carefully.



With no maintenance contract, we had to investigate the cause on our own.

## 5.5. Summary of WAF Introduction and Operation

Before ending this chapter, we explain what IPA thinks is the value of the WAF based on our experience of having introduced and operated an open source WAF “ModSecurity”. We also present what IPA thinks are the challenges of WAF introduction based on our experience, from a perspective of “general challenges when introducing a WAF” and “the challenges when introducing an open source WAF”.

### ■ Value of WAF Learned from Experience

This time, we have introduced an open source WAF “ModSecurity” and started its operation enabling only the rules for SQL injection attacks. As a result, we have been detecting and preventing dozens of attacks per month. We did confirm that a WAF is effective to protect the web applications from attacks that exploit vulnerability.

In addition, like a feature of analyzing and inspecting the malicious request sent through the POST method, the WAF’s unique features are also effective (see 5.4.1 (1)).

### ■ Challenges in Introducing WAF Learned from Experience<sup>61</sup>

#### ◆ General Challenges When Introducing a WAF<sup>62</sup>

We confirmed the problems like that the installation may fail or that using the rules as they are could cause false positives (5.4.1 (2)). We could prevent these problems from occurring in the service environment by doing the tests for the installation of ModSecurity false positive in advance.

We learned firsthand that how important to do the testing beforehand.

#### ◆ Challenges When Introducing an Open Source WAF

Since we have used ModSecurity, an open source WAF, that did not offer a user support, when we failed the installation and identified the false positives, we investigated the cause and countermeasure on our own utilizing the mailing list and other available resources (5.4.3).

We learned that the introduction and operation would take manpower cost.



WAF is effective as a vulnerability countermeasure.



WAF requires manpower cost not only for the introduction but also for the operation.

<sup>61</sup> These challenges are based on IPA’s experience in having introduced and operated ModSecurity that employs the blacklist approach for its WAF rules. Note that if the website administrator uses a WAF that employs the whitelist approach, these challenges may not apply.

<sup>62</sup> These challenges may not apply for a commercial WAF.

# Appendix A. Open Source Software WAF

---

The appendix A gives an overview and installation example of two open source software WAF: ModSecurity and WebKnight.



The procedures and settings used in the installation examples showed here are for our evaluation environment. Note that they differ depending on the operational environment of the system.

## ModSecurity

---

### Overview

---

ModSecurity is open source software provided by Trustwave<sup>63</sup> under the GPLv2 license. ModSecurity works as a module of the Apache web server software. The latest version at the time of this writing is the Version 2.5.13. For its operational environment, see Table A- 1.

Table A- 1 Operating Environment for ModSecurity

Components	Supported Products
OS	Unix, Windows
Web Server	Apache 2.x <sup>64</sup>

---

<sup>63</sup> <http://www.trustware.com/>

<sup>64</sup> ModSecurity 2.x does not support Apache 1.x.

## Installation Example

---

This section introduces the procedures to install ModSecurity 2.5.13 in the environment shown in Table A- 2. For more detailed procedures and settings for ModSecurity, see the documents provided by the developer<sup>65</sup>.

Table A- 2 Test Environment for ModSecurity

Components	Used Products
OS	CentOS release 5.5 (Final) Kernel 2.6.18-194.26.1.el5
Web Server	Apache 2.2.17

### (1) **Download**

ModSecurity is available for download at:  
<http://www.modsecurity.org/download/>

### (2) **Installation**

Compile ModSecurity from source code and install it.

To install ModSecurity, first, installation of the prerequisite software is required. We will spare the steps to install those software.

- Prerequisite Software
  - mod\_unique\_id
  - libapr
  - libapr-util
  - libpcre
  - libxml2
  - liblua 5.1.x
  - libcurl 7.15.1 or later

---

<sup>65</sup> <http://www.modsecurity.org/documentation/index.html>

After installing the prerequisite software, install ModSecurity<sup>66</sup>. For the sake of convenience, the installation is done as the root user here.

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

Next, change the Apache configuration to use ModSecurity.

Add the following lines to the Apache configuration file (httpd.conf).

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

### (3) **Configuration**

Once the installation is complete, configure ModSecurity.

First, set up the rules to use ModSecurity. Here, we use the Core Rule Set available for free. The Core Rule Set is bundled with ModSecurity. Copy it and place it to an appropriate directory (here, we place it under /usr/local/modsecurity2).

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

Next, open the ModSecurity configuration file for Apache (httpd-modsecurity.conf) and add the following lines. This time, we will use only the rules for SQL injection Attacks.

```
# 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
```

Also, open the ModSecurity configuration file (modsecurity\_crs\_10\_config.conf) and add the following lines. Then reboot Apache.

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

---

<sup>66</sup> Depending on the version, the use of the “--with-pcre=” option may be required when configuring Apache. For more information, see the installation manual for ModSecurity.

```

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

```

For references, the major configuration items in the ModSecurity configuration file are shown in Table A- 3. For more information, see the documents provided by the developer.

Table A- 3 Major Configuration Items for ModSecurity

Items	Description	Additional Info
SecRuleEngine	Operation mode	On - process rules (block) Off - do not process rules (disabled) DetectionOnly - detect but not block
SecDefaultAction	Default action	Define the default action ModSecurity takes on a rule match. The individual setting for each rule (SecRule) has priority.
SecAuditEngine	Audit logging operation mode	On - log all transactions Off - disabled RelevantOnly - log transactions that have a status code specified in the SecAuditLogRelevantStatus
SecAuditLogRelevantStatus	Server response codes to be recorded in the audit log	Used when the SecAuditEngine is set to RelevantOnly
SecAuditLog	Path to log file (incl. file name)	-
SecAuditLogType	Type of audit logging mechanism	Serial - stored in one file Concurrent - stored in separate files per session
SecAuditLogStorageDir	Directory to store concurrent audit log entries	Used when the SecAuditLogType is set to Concurrent When set to Serial, it is commented out
SecAuditLogParts	Part of each transaction to be recorded in audit log (A and Z are mandatory)	A - AuditLog header B - Request header C - Request body D - Reserved E - Response body F - Response header G - Reserved H - Additional information. For a rule match transaction, a tag is set here I - Request body that does not contain the

		information about the files J - Reserved K - a full list of every rule that matched Z - final boundary
SecDebugLog	Path to debug log file (incl. file name)	-
SecDebugLogLevel	Level of logging recorded to debug log	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.

Create the log files to which ModSecurity will output the log entries in advance.

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

As for these log files, configuring the log management attributes makes things convenient. Here, the log management settings used by IPA in “5 WAF Introduction Case Study at IPA” are shown below as an example.

```
# 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) **Verification**

Before putting ModSecurity into operation, first set up ModSecurity not to block the HTTP transactions matched with the rules and verify its behavior carefully. Change the ModSecurity configuration file (mod\_security\_crs\_10.config.conf) as below and restart Apache.

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

By configuring this setting, ModSecurity detects based on the rules but not blocks the transactions.

To make sure that the rules are applied, access the following URL from the browser to the web server installed with ModSecurity. With this URL, the character string "and 1=1;--" is given for the parameter *id*, which is often used in an attack that tries to exploit the SQL injection vulnerability. This character string should be detected by the rules set up in (3).

```
http://web server IP address/example.html?id=and 1=1;--
```

ModSecurity will detect this access, judge it as an attack that exploits a SQL injection vulnerability, and output a log entry. At the ModSecurity default settings, the log entries are recorded in the following log file. Check the file and see if the use of " and 1=1;--" has been detected and logged.

```
# tail /var/log/httpd/error_log
[Thu Dec 09 19:44:36 2010] [error] [client 192.168.0.1] ModSecurity: Warning. Pattern match
"%%b(%%d+) ?(?:=<>|<=>|<>|=) ?%%1%%b[%%'"%% %%%xc2%%xb4%%%%xe2%%x80%%x99%%%%xe2%%x80%%x98](%%d+)[
%%'"%% %%%xc2%%xb4%%%%xe2%%x80%%x99%%%%xe2%%x80%%x98] ?(?:=<>|<=>|<>|=) ?[%%'"%% %%%xc2%%xb4%%%%xe
2%%x80%%x99%%%%xe2%%x80%%x98]%%2%%b[%%'"%% %%%xc2%%xb4%%%%xe2%%x80%%x98](%%w+)[%%'"%% %%%xc2%%xb4%%
%%xe2%%x80%%x99%%%%xe2%%x80%%x98] ?(?:=<>|<=>|<>|=) ?[%%'"%% %%%xc2%%xb4%%%%xe2%%x80%%x99%%%%xe2%%x8
0%%x98]%%3%%b|[%%'"%% %%%xc2%%xb4%%%%xe2%%x80%%x99%%%%xe2%%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"]
```

Not that this is one of examples. Check the HTTP transactions logged in the log files to see whether the legitimate HTTP transactions that should be passed through are blocked (false positives) or the malicious HTTP transactions that should be blocked are passed through (false negatives). If wrong behaviors are occurring, change the ModSecurity configuration

settings and the rules and make adjustments. Repeat this process until everything works supposedly and no false positives or false negatives occur.

## (5) **Operation**

After confirming that everything work correctly, put ModSecurity in service. To start its operation, change the ModSecurity configuration file as below and restart Apache.

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

By changing this setting, ModSecurity starts blocking the attacks against the web application based on the rules.

## (6) **Uninstallation**

The ModSecurity manuals do not provide how to uninstall ModSecurity. Here, we introduce how to disable ModSecurity.

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

By configuring this setting, ModSecurity will not block nor detect the HTTP transactions, and this the impact on the web server can be minimized.

Or, it can be disabled by commenting out the configuration file like below.

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

## (7) **TIPS: Combined Use with iLogScanner V3.0**

As explained in (4), the filtering result of ModSecurity needs to be visually confirmed going over the log file. iLogScanner V3.0 provided by IPA can analyze the ModSecurity's detect/block data based on the error log of Apache with only a small change to the Core Rule Set.

All it takes to analyze the detect/block data of ModSecurity with iLogScanner is to add a “tag” that corresponds to vulnerability to the rules of the Core Rule Set. For example, the “tag” name for SQL injection is the following.

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



# WebKnight

---

## Overview

---

WebKnight is open source software provided by AQTRONIX under the GPL license. WebKnight works as an ISAPI filter of Windows Internet Information Service (IIS). The latest version as of translation of this guide is WebKnight 2.4. For its operational environment, see Table A- 4.

Table A- 4 Operating Environment for WebKnight

Components	Supported Products
OS	Windows
Web Server	IIS 5.0, 6.0, 7.0 <sup>67</sup> Web servers that support the ISAPI filter

---

<sup>67</sup> Installation of the ISAPI filter is required.

## Installation Example

---

This section introduces the procedures to install WebKnight 2.2 in the environment shown in Table A- 5. For more detailed procedures and settings for WebKnight, see the documents provided by the developer<sup>68</sup>.

Table A- 5 Test Environment for WebKnight

Components	Used Products
OS	Windows Server 2003
Web Server	IIS 6.0
Note	Use IIS in IIS5.0 process isolation mode

### (1) **Download**

WebKnight is available for download at the AQTRONIX website:  
<http://www.aqtronix.com/?PageID=99#Download>

---

<sup>68</sup> <http://www.aqtronix.com/?PageID=99>

## (2) Installation

There are 3 methods to install WebKnight.

- Windows installer "WebKnight.msi"
- VB Script installer "install.vbs"
- Manual installation

Here, we use Windows installer and install WebKnight as a global filter. To install WebKnight, double click "WebKnight.msi" among the files downloaded in (1) (Figure A-1). Start IIS after installation is completed normally and you will see that WebKnight is added to the ISAPI filters and works properly (Figure A- 2).



Figure A- 1 Windows Installer

WebKnight.msi



Figure A- 2 IIS ISAPI Filter

## (3) Configuration

To configure WebKnight, use the WebKnight Configuration tool.

From the Windows Start menu, select "All Programs", then "AQTRONIX WebKnight" and click "WebKnight Configuration". After the Open Configuration window is popped up, select "WebKnight.xml" and click the OK button (Figure A- 3).



Figure A- 3 Open Configuration Window

Now, the WebKnight Configuration tool starts up (Figure A- 4). For reference, Table A- 6 summarizes the configuration items of WebKnight Configuration.

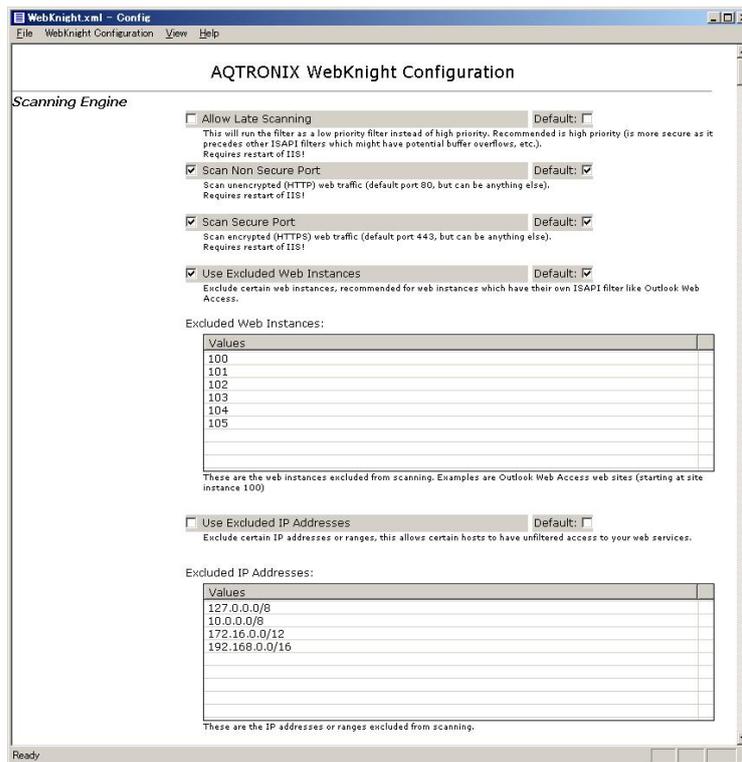


Figure A- 4 WebKnight Configuration Tool

Table A- 6 WebKnight Configuration Tool Configuration Items

Category	Items	Note
Scanning Engine	Configure things such as whether or not to scan HTTP/HTTPS, and specify IP address to be excluded	
Incident Handling Response	Configure things such as how to respond to the unauthorized access	
Logging	Configure things such as whether or not to enable logging, what to log	
Connection	Configure things such as IP addresses to be monitored or blocked	
Authentication	Configure authentication related issues including the countermeasures against brute force attack	
Request Limits	Configure the length of requests such as contents, URL, query strings	
URL Scanning	Configure the character strings and such to be denied for URL	Check the "URL Denied Sequences" box and make sure that the access to its own website is not mistakenly denied.
Mapped Path	Configure the character strings to be denied and paths to be allowed access in the attacks like the directory traversal attack	For "Allowed Paths", see the remarks below.
Requested File	Configure things such as the character strings and the extensions used in the file name to be denied	Make sure that the access to its own files is not mistakenly denied.

Robots	Configure how to handle request from the bots	
Headers	Configure the issues about the server header	
ContentType	Configure things such as whether or not to check the Content-Type header of the requests	
Cookie	Configure issues about the cookie	
User Agent	Configure when to deny the user agents such as browsers.	
Referrer	Configure things such as whether or not to scan Referrer and whether or not to limit hotlinking from specific domains	
Methods	Configure the methods to be allowed and denied	
Querystring	Configure things such as the character string in query strings to be denied	
Global Filter Capabilities	Configure whether or not to apply as global filter and when to deny about POST data	
SQL Injection	Configure keywords to check for SQL injection	
Web Applications	Configure the settings of the web applications	

When configuring the settings, be careful about the following.

- Specify the path of the website in the Allowed Paths setting in the Mapped Path category. If it is not included, access to the website is blocked.
- Do not forget to restart IIS after changing the settings that has the "Requires restart of IIS" remark.
- Inspection of the Post data is disabled at the default settings. If needed, enable the Postdata settings in the Global Filter Capabilities category.

#### (4) **Verification**

See if false positives and false negatives are occurring. A sample procedure is shown below.

##### ① Configure the Response Log Only setting

To see if the WebKnight configuration is causing problems, set the Response Log Only mode not to block the communications. With this setting, the communications that are deemed malicious are logged but not blocked.

To configure the Response Log Only mode, check the "Response Log Only" box in the "Incident Response Handling" category of WebKnight Configuration.

② Check the WebKnight Log

Check the WebKnight logs for the malicious connections and see if there are the legitimate connections mistakenly detected by WebKnight. There are 2 ways to check up the logs.

- Use a WebKnight tool "Log Analysis" (Figure A- 5).

To start "Log Analysis", from the Windows Start menu, select "All Programs", then "AQTRONIX WebKnight", and click "Log Analysis".

- View the WebKnight log file directly

At the default settings, the log files are created daily and stored at C:\Program Files\AQTRONIX WebKnight\LogFiles.

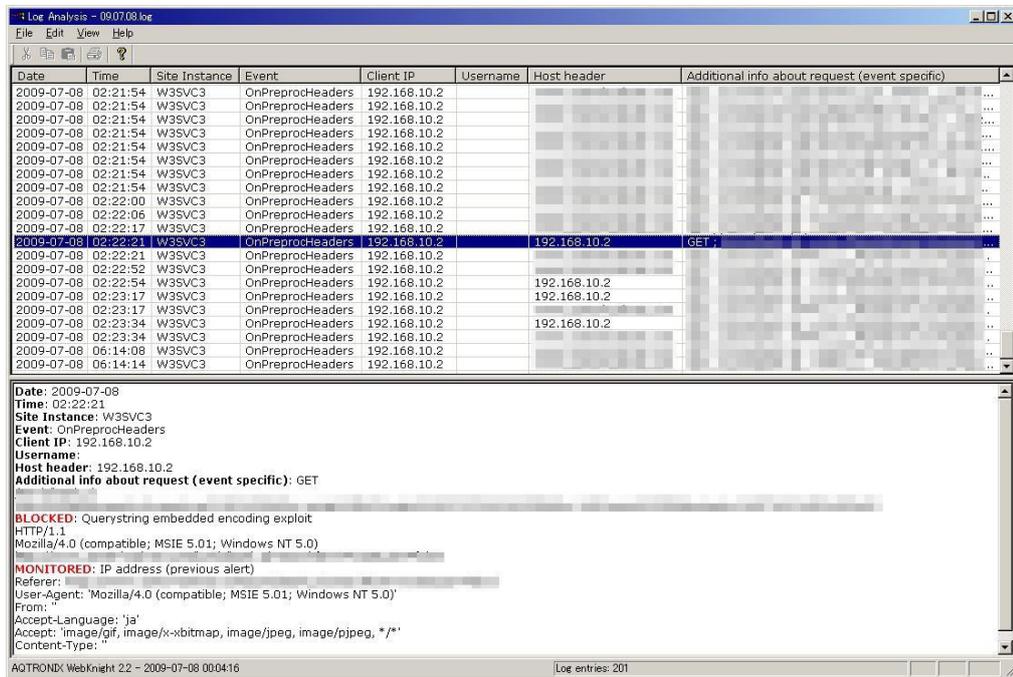


Figure A- 5 Log Analysis

③ Check the IIS Log

Check the IIS logs and see if there are the malicious connections and attacks that have not been detected by WebKnight.

As the result of ② and ③, if false positives and false negatives occurred, return to the procedure (3) and review the configuration settings. Repeat the process until everything works supposedly and no false positives or false negatives occur.

## (5) **Operation**

After confirming that everything work correctly in (4), put WebKnight in service. Before that, decide what to do with the malicious connections after detection. Here, we disable the Response Log Only settings configured in (4) and change the setting to return an Error Page when detected a malicious connection. It can be customized or a default WebKnight error page is available for use.

### ① Disable the Response Log Only setting

Remove the check from the Response Log Only box in the Incident Response Handling category of WebKnight Configuration.

### ② Customize the Error Page

At the default settings of WebKnight<sup>69</sup>, WebKnight returns an error page shown in Figure A- 6 to the browser directly when detecting a malicious transaction.



Figure A- 6 WebKnight Error Page at Default Settings

To customize an error page, disable the Response Directly setting and enable the Response Redirect setting in the Incident Response Handling category of WebKnight Configuration, and specify the path for the customized error page in the Response Redirect URL filed (Figure A- 7).

Note that the path to the customized error page specified in the Response Redirect setting needs to be allowed in the Allowed Paths setting in the Mapped Path category of WebKnight Configuration as well.

<sup>69</sup> The settings where the Response Directly setting in the Incident Response Handling category of WebKnight Configuration is enabled.

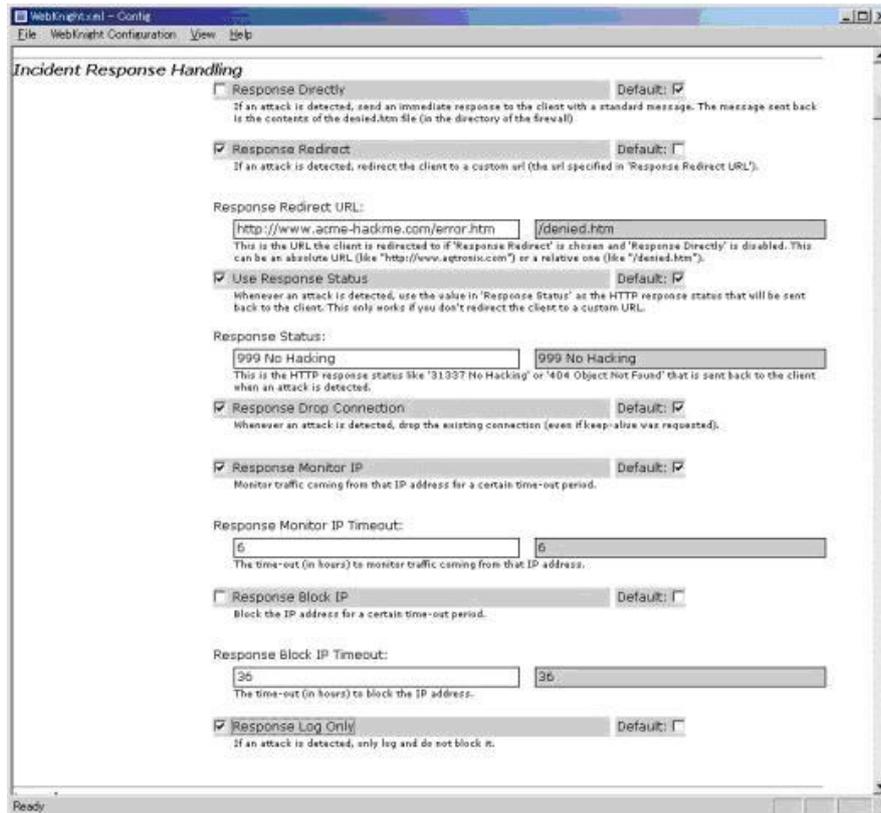


Figure A- 7 the Response Redirect Setting

③ Check the IIS Log

When the Response Log Only setting is disabled, the content of the IIS log differs depending on how to handle the errors.

Below is a sample log for the attack that tried to exploit the SQL injection vulnerability. As seen in the log, an SQL statement is recorded in the parameter "param".

**In the case where the Response Directly setting is enabled**

WebKnight Log

```
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 Log

None

## In the case where the Response Redirect setting is enabled

### WebKnight Log

```
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 Log

```
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) Uninstallation

Just like installation, there are 3 ways to uninstall WebKnight. Here, we explain one that use an OS function. Remember to restart IIS after uninstallation.

### ◆ Uninstall using the Windows “Add or Remove Programs”

To use the OS function "Add or Remove Programs", select "AQTRONIX WebKnight 2.2". Click "Remove" will execute the uninstallation.

Even after the uninstallation of WebKinght, the log file remains (at the default settings, located at C:\Program Files\AQTRONIX WebKnight\LogFiles). If unnecessary, delete the folder after rebooting the OS.

## Appendix B. Commercial WAF

To promote the use of the WAF products and services, Appendix B lists the commercial WAF products and services provided by the vendors contributed to this guide in the Japanese alphabetical order. For more information on each product or service, please contact the respective product developer or service provider.

### Developers and Service Providers listed in the 1<sup>st</sup> Edition, 1<sup>st</sup> Printing

	
Product Developer	株式会社ジェイピー・セキュア(Japanese)
URL	<a href="http://www.jp-secure.com/cont/products/">http://www.jp-secure.com/cont/products/</a> (Japanese)

	
Service Provider	株式会社セキュアスカイ・テクノロジー(Japanese)
URL	<a href="http://www.scutum.jp/index.html">http://www.scutum.jp/index.html</a> (Japanese)

	
Product Developer	株式会社ソリトンシステムズ(Japanese)
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> (Japanese)

	
Product Developer	日本電気株式会社(Japanese)
URL	<a href="http://www.nec.co.jp/soft/siteshell/">http://www.nec.co.jp/soft/siteshell/</a> (Japanese)



Product Developer	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> (Japanese)



**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**

Service Provider	株式会社日立システムズ (Japanese)
URL	<a href="http://www.hitachi-systems.com/solution/s005/web/index.html">http://www.hitachi-systems.com/solution/s005/web/index.html</a> (Japanese)

### **Developers and Service Providers listed in the 1st Edition, 2<sup>nd</sup> Printing**



Product Developer	Imperva Inc.
URL	<a href="http://www.imperva.com/products/wsc_web-application-firewall.html">http://www.imperva.com/products/wsc_web-application-firewall.html</a> <a href="http://www.imperva.jp/products/web_application_security.asp">http://www.imperva.jp/products/web_application_security.asp</a> (Japanese)



Product Developer	Citrix Systems, Inc
URL	<a href="http://www.citrix.com/English/ps2/products/subfeature.asp?contentID=2300448">http://www.citrix.com/English/ps2/products/subfeature.asp?contentID=2300448</a> <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> (Japanese)

### **Developers and Service Providers listed in the 1<sup>st</sup> Edition, 3<sup>rd</sup> Printing**



Product Developer	株式会社ネットファイア (Japanese)
URL	<a href="http://www.netfire.jp/product.html">http://www.netfire.jp/product.html</a> (Japanese)

## Developers and Service Providers listed in the 2<sup>nd</sup> Edition, 2<sup>nd</sup> Printing



IT agility. Your way.

開発元企業	F5 Networks, Inc.
URL	<a href="http://www.f5.com/products/big-ip/application-security-manager.html">http://www.f5.com/products/big-ip/application-security-manager.html</a> <a href="http://www.f5networks.co.jp/product/bigip/asm/index.html">http://www.f5networks.co.jp/product/bigip/asm/index.html</a> (Japanese)



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

# Terminology

---

## **HTTP (Hypertext Transfer Protocol)**

A protocol used for the web server and browser to transfer data.

## **HTTPS Transaction**

In this guide, it means an HTTP transaction encrypted by SSL (Secure Socket Layer) or TLS (Transport Layer Security).

## **SSL (Secure Socket Layer)**

A protocol to encrypt and transfer data over the Internet. Using SSL, confidential data such as personal information and credit card numbers can be transferred safely. Used for online banking.

## **SQL (Structured Query Language)**

A computer query language to define the data schema and manage data in the relational database (RDB). There are several types of SQL statements, such as DDL (Data Definition Language) like a CREATE statement to define the database structure, and DML (Data Manipulation Language) like a SELECT, UPDATE or GRANT statement to manage data within schema objects.

## **Web Application**

A computer system running a website. Unusually developed in Java, ASP, PHP or Perl, and provides dynamic web pages to the website visitors.

## **Website**

A collection of components that make up a specific domain (e.g <http://www.ipa.go.jp/>). The components of a website include web pages, web applications, web servers and database servers.

## **Web Server**

A software or physical server on which web pages and web applications run. In this guide, a web server means a physical server unless otherwise noted.

## **Open Source**

A software whose source code is publically available and free for redistribution.

## **Vulnerability**

A security flaw in systems such as web applications. It may disrupt the functionality and capability of a web application by the attacks, such as unauthorized access and computer virus that exploit the flaw. Vulnerability could include a system environment where the security of the web applications is not maintained, for example, personal information is not managed with proper access control, due to the web administrator's improper management.

## **Protocol**

A convention for communications between two entities to transfer data over network.

[ Produced and Copyrighted by ] Information-technology Promoting Agency, Japan  
(IPA)

[ Editor ] Hideaki Kobayashi

[ Author ] Naoto Katsumi  
Isao Kaine  
Katsutoshi Nakamura NEC システムテクノロジー株式会社 (Japanese)

[ Advisor ] Kazuo Saito JP-Secure Inc.  
Hiromitsu Takagi National Institute of Advanced Industrial Science and  
Technology (AIST)  
Tadashi Yamagishi Hitachi, Ltd.  
Shingo Ootani LAC: Little eArth Corporation Co., Ltd.  
Hiroshi Tokumaru  
Hiroyuki Itabashi Masashi Ohmori Yu Kisoda  
Chisato Konno Motokuni Soma Shunsuke Taniguchi  
Yukinobu Nagayasu Takahito Watanabe

[ Collaborator ] 株式会社 Imperva Japan (Japanese)  
NEC システムテクノロジー株式会社 (Japanese)  
JP-Secure Inc.  
株式会社セキュアスカイ・テクノロジー (Japanese)  
株式会社ソリトンシステムズ (Japanese)  
株式会社ネットファイア (Japanese)  
株式会社日立システムズ (Japanese)  
マクニカネットワークス株式会社 (Japanese)

\*Affiliation omitted for the personnel of IPA

## Web Application Firewall (WAF) Guide, 2<sup>nd</sup> Edition

---

[ Publication ]	Feb. 16, 2010	First Edition, First Printing
	May 12, 2010	First Edition, Second Printing
	Oct. 8, 2010	First Edition, Third Printing
	Feb. 28, 2011	Second Edition, First Printing
	May 12, 2011	Second Edition, Second Printing
	Dec 28, 2011	Second Edition, Third Printing

[ Produced and Copyrighted by ] IT Security Center,  
Information-technology Promoting Agency, Japan

# How to Report Information Security Issues to IPA

Designated by the Ministry of Economy, Trade and Industry, IPA IT Security Center collects information on the discovery of computer viruses and vulnerabilities, and the security incidents of virus infection and unauthorized access.

Make a report via web form or email. For more detail, please visit the web site:

URL: <http://www.ipa.go.jp/security/todoke/> (Japanese only)

## Computer Viruses

When you discover computer viruses or notice that your computers have been infected by viruses, please report to IPA.

## Unauthorized Access

When you detect unauthorized access to your computers via network (e.g. the Internet, LANs, WANs and PC communications), please report to IPA.

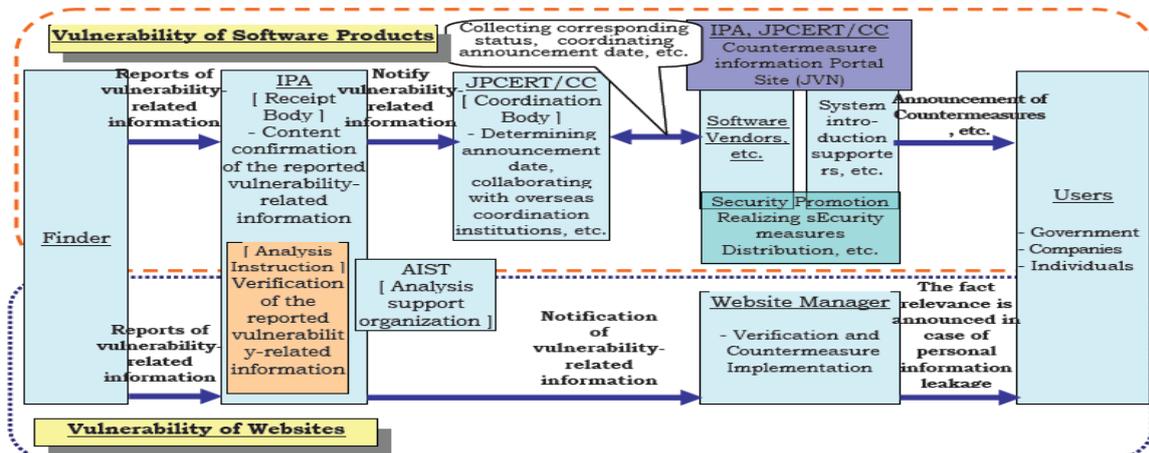
## Software Vulnerability and Related Information

When you discover vulnerabilities in client software (e.g. OS and browser), server software (e.g. web server) and software embedded into hardware (e.g. printer and IC card), please report to IPA.

## Web Application Vulnerability and Related Information

When you discover vulnerabilities in systems that provide their customized services to the public, such as websites, please report to IPA.

## Framework for Handling Vulnerability-Related Information ~ Information Security Early Warning Partnership ~



JPCERT/CC: Japan Computer Emergency Response Team Coordination Center, AIIST: National Institute of Advanced Industrial Science and technology

# IPA

INFORMATION-TECHNOLOGY PROMOTION AGENCY, JAPAN  
 2-28-8 Honkomagome, Bunkyo-ku, Tokyo 113-6591 JAPAN  
<http://www.ipa.go.jp/index-e.html>

IT SECURITY CENTER  
 Tel: +81-3-5978-7527 FAX: +81-3-5978-7518  
<http://www.ipa.go.jp/security/english/>