



**Barracuda Web Application Firewall Administrator's  
Guide**

**Version 7.3**

Barracuda Networks Inc.  
3175 S. Winchester Blvd.  
Campbell, CA 95008  
<http://www.barracuda.com>

## **Copyright Notice**

Copyright 2009, Barracuda Networks  
www.barracuda.com  
090723-73v0015-01-0723

All rights reserved. Use of this product and this manual is subject to license. Information in this document is subject to change without notice.

## **Trademarks**

Barracuda Web Application Firewall is a trademark of Barracuda Networks. All other brand and product names mentioned in this document are registered trademarks or trademarks of their respective holders.

## **Chapter 1 – Introduction . . . . . 11**

Overview . . . . .	12
Features of the Barracuda Web Application Firewall . . . . .	14
Application Firewall . . . . .	14
Adaptive Security . . . . .	14
Load Balancing . . . . .	15
Application Acceleration . . . . .	15
Application Access Control . . . . .	15
Reporting and Alerting . . . . .	15
High Availability . . . . .	15
Role Based Administration . . . . .	16
Energize Updates . . . . .	16
Technical Support . . . . .	16
The Barracuda Web Application Firewall Models . . . . .	16

## **Chapter 2 – Web Application Firewall Concepts . . . . . 19**

Attacks and Terminology . . . . .	20
Basic Terminology . . . . .	23
Web Application Firewall Concepts . . . . .	24
Negative Security Model . . . . .	24
Positive Security Model . . . . .	24
Adaptive Profiling . . . . .	24
False Positives . . . . .	24
Exception Profiling . . . . .	24
Cloaking . . . . .	24
Allow/Deny Rules . . . . .	24
Input Validation . . . . .	25
Web Site Profiles . . . . .	25
Request Matching (Regex) Rules . . . . .	25
Active and Passive modes . . . . .	25
Policy Tuner . . . . .	25
Web Service Firewall . . . . .	25
XML Attacks and Prevention . . . . .	26
SOAP Validation . . . . .	26
WS-I Basic Profile Assertions . . . . .	26
Authentication, Authorization and Access Control . . . . .	26
Internal and External Authentication . . . . .	26
Authorization . . . . .	26
Single Sign-On (SSO) . . . . .	26
Application Acceleration and Assurance . . . . .	27
Application Acceleration . . . . .	27
Content Routing . . . . .	27
Caching . . . . .	27
Compression . . . . .	27
Connection Pooling . . . . .	27

Load Balancing . . . . .	27
Session Persistence . . . . .	27
Server Monitoring . . . . .	28
Load Balancing Algorithm . . . . .	28
High Availability . . . . .	28
Ethernet Hard Bypass . . . . .	28

## **Chapter 3 – Getting Started . . . . .29**

Deployment Modes for the Barracuda Web Application Firewall . . . . .	30
Reverse Proxy (Recommended) . . . . .	32
Bridge-Path . . . . .	33
One-armed . . . . .	33
Best Practise . . . . .	34
Initial Setup . . . . .	36
Prepare for the Installation . . . . .	36
Connect Barracuda Web Application Firewall to Network. . . . .	37
Configure IP Address and Network Settings . . . . .	37
Configure the Barracuda Web Application Firewall . . . . .	38
Activate Your Subscription Status . . . . .	39
Update the Barracuda Web Application Firewall Firmware . . . . .	40
Update Attack Definitions, Virus Definitions and Security Definitions . . . . .	40

## **Chapter 4 – Securing a Web Site . . . . .41**

Creating Services . . . . .	42
Deploying HTTP service . . . . .	42
Deploying HTTPS service . . . . .	43
Securing an HTTP Web Site with HTTPS. . . . .	43
Creating a Redirect service . . . . .	44
Creating a Custom Service . . . . .	45
Creating a Custom SSL Service . . . . .	45
Creating a FTP Service . . . . .	46
Configuring FTP Attack Prevention . . . . .	46
Editing Basic Security for a Service . . . . .	47
Load Balance . . . . .	47
Configuring SSL . . . . .	49
Configuring a Backup Server . . . . .	49
Passive versus Active Mode . . . . .	50
Security Policy. . . . .	50
Allowing/Denying Specific URLs . . . . .	51
Allowing/Denying Specific Headers . . . . .	52

## **Chapter 5 – Customized Security for Web sites . . . . .53**

Security Policies . . . . .	54
Request Limits . . . . .	54
Cookie Security . . . . .	55
Functioning of Cookie Security . . . . .	55
Encrypting Cookies . . . . .	56
Signing Cookies . . . . .	56

Cookie Security interaction with other Security features . . . . .	56
Configuring Cookie Security . . . . .	56
URL Protection . . . . .	57
Attacks via HTTP Headers and Contents . . . . .	57
Attacks by Injecting various Commands in Parameters . . . . .	58
Attacks by Injecting Buffer Overflow in Parameters . . . . .	58
Parameter Protection . . . . .	59
Cloaking . . . . .	59
Data Theft Protection . . . . .	60
Protected Data Types . . . . .	61
URL Normalization . . . . .	61
Global ACLs . . . . .	62
Existing Global ACLs . . . . .	63
Action Policy . . . . .	63
Creating a New Security Policy . . . . .	64
Web Site Profiles . . . . .	65
URL profile . . . . .	65
Parameter profile . . . . .	66
Web Site Translations . . . . .	68
Configuring URL Translation . . . . .	68
Configuring Request Rewrite . . . . .	68
Request Rewrite Condition . . . . .	69
Configuring Response Rewrite . . . . .	70
Response Rewrite Condition . . . . .	71
Configuring Response Body Rewrite . . . . .	72
Trusted Hosts . . . . .	73

## **Chapter 6 – Adaptive Security . . . . .75**

Overview . . . . .	76
Layout of Adaptive Security . . . . .	77
Working with Adaptive Profiling . . . . .	78
Configuring Adaptive Profiling . . . . .	78
Edit Service Adaptive Profiling . . . . .	78
Add Adaptive Profiling Rules . . . . .	78
To Start Adaptive Profiling . . . . .	78
To Stop Adaptive Profiling . . . . .	78
Working with Navigation parameters . . . . .	78
Configuring URLs to be Excluded from Adaptive Profiling . . . . .	79
Understanding Request and Response Learning . . . . .	79
Response Learning . . . . .	81
Request Learning . . . . .	81
Viewing Newly Generated Profiles . . . . .	82
Enforcing Learned Profiles . . . . .	82
Using Strict Profile Checking during Learning . . . . .	83
Working with Exception Profiling . . . . .	84
Configuring Exception Profiling . . . . .	84
Exception Profiling Level . . . . .	84
Learning from Trusted Hosts . . . . .	84
Learning Concurrently from Trusted and Non Trusted Traffic . . . . .	84
Pending Recommendations . . . . .	84
Exception Heuristics . . . . .	86

Working with Exception Profiling Levels . . . . .	86
Web Site Profiles . . . . .	87
Search . . . . .	87
Directories . . . . .	87
Stop learning a directory path . . . . .	87
Things you can do from this page . . . . .	87
Recommended way to use the Adaptive Security feature . . . . .	88

## **Chapter 7 – Traffic Management . . . . .89**

Traffic Management. . . . .	90
Content Rule . . . . .	90
Example1: Content Rule for Images . . . . .	90
Configuring Load Balance for a Content Rule (only in Proxy mode) . . . . .	92
Configuring Caching. . . . .	92
Content Rules and Dynamic Pages . . . . .	93
Object Freshness . . . . .	93
Configuring Compression . . . . .	94

## **Chapter 8 – Keys and Certificates . . . . .97**

Overview . . . . .	98
Types of Certificates. . . . .	98
X.509 Certificate . . . . .	99
Certificate Usage . . . . .	99
Certificate Components . . . . .	99
Key Pair . . . . .	99
Distinguished Name (DN) . . . . .	100
Token . . . . .	100
CA Certificate . . . . .	100
Creating a Test Certificate . . . . .	101
Saved Certificates . . . . .	102
CSR . . . . .	102
Certificate . . . . .	102
Extracting the key from the Certificate . . . . .	102
Uploading a Certificate . . . . .	104
Signed Certificate . . . . .	104
Certificate Key. . . . .	104
Intermediary Certificates . . . . .	104
Uploading a Trusted Certificate . . . . .	104

## **Chapter 9 – User Access Control . . . . .105**

Overview . . . . .	106
Steps to Configure Access Control to different parts of your Web site . . . . .	107
Creating an Authentication Policy . . . . .	107
Single Sign-On (SSO). . . . .	108
Single domain SSO . . . . .	109
Logout in Single domain Single Sign-On Environment . . . . .	109
Multi-domain SSO . . . . .	109
Multi-domain Single Sign-On Functionality . . . . .	110

Chained Logout in a Multi-domain Single Sign-On Session . . . . .	111
Creating an Authorization Policy . . . . .	112
Creating new Authentication Services . . . . .	113
LDAP . . . . .	114
RADIUS . . . . .	115
Creating Local Users/Groups . . . . .	115
Allowing/Denying Client Certificates . . . . .	116

## Chapter 10 – Monitoring, Logging and Reporting . . . . .128

Monitoring Barracuda Web Application Firewall . . . . .	129
Viewing Performance Statistics . . . . .	129
Health Indicators for Services and Servers . . . . .	129
Monitoring the Health of the Server . . . . .	130
In-Band Health Checks . . . . .	130
Out Of Band Health Checks . . . . .	130
Application Layer Health Check . . . . .	130
Viewing System Tasks . . . . .	131
Understanding the Indicator Lights . . . . .	131
Logs. . . . .	132
Web Firewall Logs. . . . .	132
Access Logs. . . . .	133
Audit Logs. . . . .	133
Search Logs. . . . .	134
Export Logs . . . . .	134
Syslog . . . . .	134
FTP Web Logs . . . . .	135
Reports . . . . .	136
Security Reports . . . . .	136
Audit Reports . . . . .	136
Config Summary Reports . . . . .	136
PCI Reports . . . . .	136
Generating Reports . . . . .	136

## Chapter 11 – Advanced Concepts . . . . .138

Deployment . . . . .	139
Multiple IP Address Configuration. . . . .	139
Static Routes . . . . .	139
Interface Routes. . . . .	140
VLAN (Virtual Local Area Network) . . . . .	140
Overview . . . . .	140
VLAN Configuration . . . . .	140
Routing to Multiple VLANs over an Interface . . . . .	141
Bridge Mode . . . . .	141
Security . . . . .	142
Creating Identity Theft Patterns . . . . .	142
Using Custom Patterns . . . . .	143
Creating Attack Types. . . . .	143
Creating Input Types . . . . .	144
Creating Custom Parameter Class . . . . .	145
Creating a Customized Response Page for Errors . . . . .	145

Creating Session Identifiers . . . . .	146
Creating Rate Control Pool . . . . .	146
Before you set up a Rate Control Pool . . . . .	146
After you set up a Rate Control Pool . . . . .	147
Scheduling algorithm for Rate Control Pool . . . . .	147
Virus Protection for File Uploads . . . . .	148
View Internal Patterns . . . . .	150
Copy and Modify a Predefined Pattern Group . . . . .	150
Configuring URL Policy . . . . .	150
Bruteforce Prevention . . . . .	151
Interaction between Rate Control Pool and Bruteforce Prevention . . . . .	152
Configuring FTP Security . . . . .	152
Configuring Session Tracking . . . . .	153
Policy Tuner . . . . .	153

## **Chapter 12 – High Availability . . . . .156**

Creating a High Availability (HA) Environment . . . . .	157
Evaluating System Status . . . . .	158
Failover . . . . .	159
Failback . . . . .	159
Data Propagated to Linked Systems . . . . .	159
Updating Redundant Barracuda Web Application Firewalls . . . . .	160
Removing units from a cluster . . . . .	160
Removing the units for RMA . . . . .	161

## **Chapter 13 – Administrating the Barracuda Web Application Firewall . . . . .164**

Administrative Settings . . . . .	165
Controlling Access to the Administration Interface . . . . .	165
Customizing the Appearance of the Web Interface . . . . .	165
Setting the Time Zone of the System . . . . .	165
Enabling and Disabling Virus Protection . . . . .	165
Enabling SSL for Administration . . . . .	166
Adding new Administrators . . . . .	167
Receiving Trap Messages and System Alerts. . . . .	168
Maintaining the Barracuda Web Application Firewall . . . . .	170
Backing up and Restoring your System Configuration . . . . .	170
Updating the Firmware of your Barracuda Web Application Firewall . . . . .	170
Updating the Attack, Virus and Security Definitions. . . . .	171
Replacing a Failed System . . . . .	171
Reloading, Restarting, and Shutting Down the System . . . . .	172
Using the Built-in Troubleshooting Tools . . . . .	172
Using the Task Manager . . . . .	172
Setting the System Configuration . . . . .	173
Rebooting the System in Recovery Mode. . . . .	173
Reboot Options . . . . .	174



## **Chapter 14 – XML Firewall . . . . .176**

Web Services . . . . .	177
Web Services Implementation. . . . .	177
WSDL . . . . .	178
Web Service Vulnerabilities . . . . .	179
Web Service Protections . . . . .	179
XML Firewall . . . . .	182
Tasks to enforce XML Firewall . . . . .	182
XML Validations. . . . .	183
Import Schema/WSDL . . . . .	183
Protected URLs . . . . .	184
XML Protections . . . . .	185
XML Validation Settings . . . . .	185
WS-I Basic Profile Assertions . . . . .	185
SOAP Validations . . . . .	185

## **Chapter 15 – Role Based Administration . . . . .188**

Overview . . . . .	189
Roles . . . . .	189
Predefined Roles . . . . .	189
New Roles . . . . .	190
Users . . . . .	191
Local Users . . . . .	191
External Users . . . . .	191
Privileges . . . . .	191
Object Privileges . . . . .	191
Screen and Operation Privileges . . . . .	192
Creating New Role . . . . .	192
Creating External Authentication Service . . . . .	193
Creating New Local Administrator Account . . . . .	193

## **Chapter 16 – Network Firewall . . . . .196**

Network Firewall Overview . . . . .	197
Configuring NAT for LAN Servers . . . . .	198
Configuring Source Network Address Translations (SNATs) . . . . .	199
Configuring Access Control Lists (ACLs) . . . . .	201

## **Appendix A – Barracuda Web Application Firewall Hardware 202**

Barracuda Web Application Firewall Front Panel. . . . .	203
Barracuda Web Application Firewall Back Panel . . . . .	204
Hardware Compliance . . . . .	205
Notice for the USA . . . . .	205
Notice for Canada . . . . .	205
Notice for Europe (CE Mark) . . . . .	205

## **Appendix B – WSDL Files 206**

Elements of a WSDL file . . . . .	206
WSDL 1.2 Syntax . . . . .	206
Sample WSDL Code . . . . .	208

## **Appendix C – Extended Match and Condition Expressions 214**

Quick reference . . . . .	214
Structure of an Extended Match Expression . . . . .	214
Operators . . . . .	215
Elements . . . . .	215
Joins. . . . .	216
Combining. . . . .	216
Escaping . . . . .	217

## **Appendix D – Usage Guidelines 218**

Macro Definitions . . . . .	218
No Name Parameters . . . . .	220
Available Statistics from SNMP GET Command . . . . .	220

## **Appendix E – Limited Warranty and License 222**

Limited Warranty . . . . .	222
Exclusive Remedy. . . . .	222
Exclusions and Restrictions . . . . .	223
Software License . . . . .	223
Energize Update Software License . . . . .	224
Open Source Licensing . . . . .	228

# Chapter 1

## Introduction

---

This chapter provides an overview of the Barracuda Web Application Firewall and includes the following topics:

- *Overview* on page 12
- *Features of the Barracuda Web Application Firewall* on page 14

# Overview

---

The Barracuda Web Application Firewall is an integrated hardware and software solution that offers comprehensive Web application and Web services security, load balancing and application acceleration for Web based applications. It offers every capability needed to deliver, secure and manage enterprise Web applications from a single appliance through an intuitive, real-time user interface.

Advanced Web firewall features provide instant and comprehensive security to HTTP based Web Applications, XML based Web Services and SSL-enabled applications. It is deployed in the network as a filtering gateway behind the network firewall and performs deep inspection of all requests and responses to and from the Web servers for any malicious content.

*Figure 1.1: Standard Barracuda Web Application Firewall Deployment*



At a higher level, the following security is provided by the Barracuda Web Application Firewall:

- Protection against known attacks, including the OWASP Top Ten. For example, SQL Injection, Cross Site Scripting, Remote File Inclusion, Buffer Overflows, Forceful Browsing, etc.
- Protection from unknown, Zero Day attacks by determining allowable requests and responses and denying everything else.

The Barracuda Web Application Firewall is designed to easily fit into any existing data center environment, and rapidly secure and accelerate new and existing Web applications out-of-the-box. Deployment options include inline as well as offline modes.

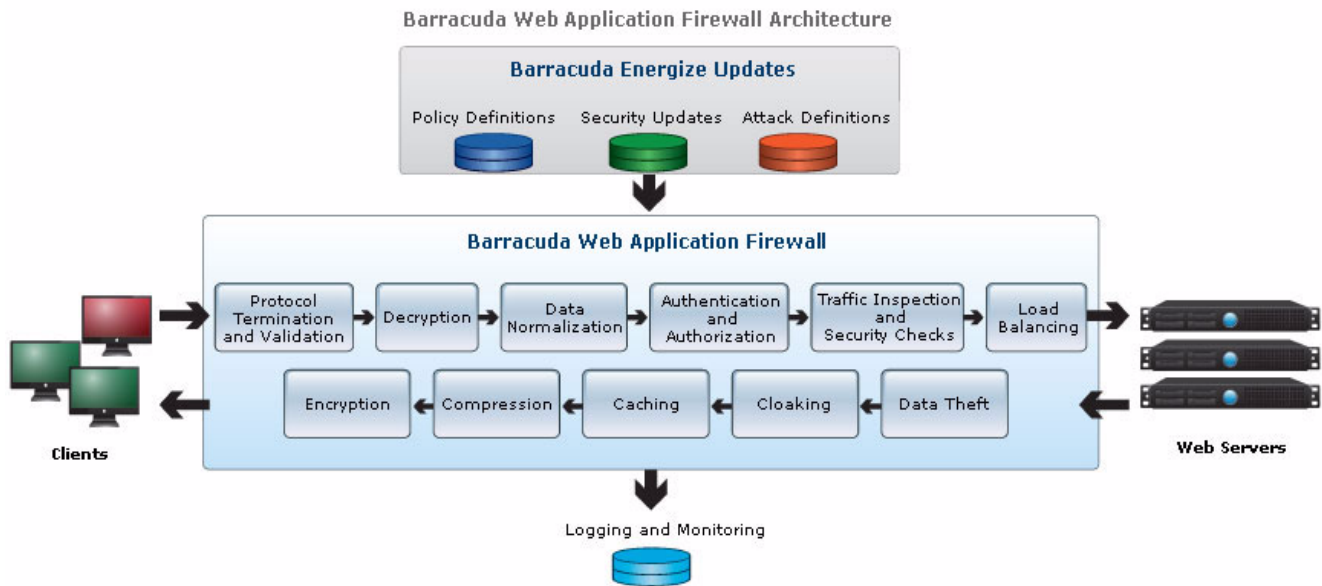
On the incoming path, it terminates application layer protocols for maximum visibility, security and control, decrypts SSL traffic, normalizes the data to handle multiple encoding formats and to detect malicious obfuscations, applies Web site user access control checks and then deep inspects the application layer traffic for any vulnerabilities. It denies malicious traffic and optionally load balances the safe traffic between a set of backend Web servers.

Outbound data is inspected for data leak prevention, such as sensitive information, credit cards, etc. and cloaked to hide server specific information to negate hackers conducting reconnaissance of the Web server resources. Data can be optionally cached and compressed to accelerate the application delivery. Finally, the data is encrypted and sent to the clients of the Web application.

The system logs all the data and actions taken and provides a rich set of real-time reporting and alerting features based on the logs, actions and system state. Security and policies are kept updated on the system by automatically downloading the latest definitions from the Barracuda Energize Update Servers.

The following figure illustrates at a high level, the major modules involved in the inbound and outbound data path.

Figure 1.2: Barracuda Web Application Firewall Architecture



# Features of the Barracuda Web Application Firewall

---

## Application Firewall

---

- **HTTP, HTTPS and FTP protocol compliance:** The Barracuda Web Application Firewall ensures that all inbound requests comply with the HTTP, HTTPS and FTP specifications respectively.
- **Protection against common, high-visibility attacks:** Hackers can take advantage of vulnerabilities in your online Web forms to attack your applications. The Barracuda Web Application Firewall protects your Web applications against all such injection attacks, e.g. SQL injections, OS command injections and cross-site scripting.
- **Protection against attacks based on session state:** The Barracuda Web Application Firewall protects your Web applications against any attacks based on session state, such as forms tampering or cookie tampering.
- **Outbound data theft protection:** All server responses are deep inspected for leakage of sensitive information like credit card data and social security numbers. Users can also specify custom patterns for data leak prevention.
- **Web site cloaking:** Cloaking strips identifying banners of Web server software and version numbers and provides customizable HTTP error handling which defeats server fingerprinting attacks.
- **Application denial of service (DoS) protection:** Implements strict content length checks in individual requests and allows for controlling application session and request rates to prevent malicious users from subverting your Web applications.
- **XML Firewall:** Deep inspection of all XML data provides XML validation and protection from XML based attacks in Web Services, generic XML and Web 2.0 applications.
- **Integrated Anti Virus:** All file uploads to the Web application can be scanned for embedded viruses and malware using the integrated antivirus engine of the Barracuda Web Application Firewall.
- **Brute Force Attack Prevention:** Prevents hackers from guessing passwords by brute force strategies using readily available password dictionaries.
- **Fine-grained control:** The Barracuda Web Application Firewall features fine-grain rules creation based on both HTTP requests and responses down to the level of individual HTML elements.
- **Full Regex Support:** Powerful PCRE regular-expression support allows you to fine tune your security policies by matching custom patterns in the HTTP requests and responses.

## Adaptive Security

---

- **Adaptive Profiling:** Employs a positive security model to provide zero-day protection from Forceful Browsing attacks that access unauthorized resources and tamper with hidden application contexts.
- **Exception Profiling:** Reduces false positives by automatically creating or recommendation policy changes by observing the request and response traffic.

## Load Balancing

---

- **Persistence:** Routes a user session traffic to the same Web server in a server farm
- **Application Content Routing:** Request traffic can be partitioned and routed to different load-balanced Web servers based on the content types
- **Choice of Scheduling Algorithms**

## Application Acceleration

---

- **Caching:** The Barracuda Web Application Firewall can reduce load on back-end Web servers and increase performance by caching Web content and avoiding repeated requests to back-end Web servers.
- **Compression:** To reduce network traffic requirements, the Barracuda Web Application Firewall can automatically apply GZIP compression to renderable HTML content to be decompressed by the browser.
- **Connection pooling:** Reduces the back-end server TCP overhead, by automatically pooling multiple front-end TCP connections into a single back-end connection.
- **SSL offloading:** Streamlines the encryption and decryption of SSL traffic to quickly process secure online transactions without additional burden on any servers.

## Application Access Control

---

- **External Authentication:** Acts as a single point of user authentication by integrating with corporate LDAP and Radius servers, or using client certificates across multiple back-end applications.
- **Granular Authorization:** Access policies can be customized across different parts of the Web application for authenticated users based on their classification.
- **Single Sign-On:** Allows the users of your Web applications to seamlessly browse through multiple application domains without having to log in multiple times.

## Reporting and Alerting

---

- **PCI reports:** Provides snapshots of common application attacks, critical for securing credit card information and providing compliance to PCI DSS requirements.
- **Comprehensive logging:** Maintains a rich set of exportable logs, including system activity, Web Firewall activity, Web services activity, network firewall activity and traditional Web logs.

## High Availability

---

The Barracuda Web Application Firewall can be installed in a redundant pair configuration, providing real-time application state replication so that security and user sessions will not be compromised during a failover event.

## Role Based Administration

You can limit access to system components to privileged users, assigning them the least privileges necessary and commensurate with their job classification and function.

## Energize Updates

Energize Updates provides the Barracuda Web Application Firewall with protection against the latest Internet threats by automatically delivering firmware updates around the clock, including attack and security updates, content categories and virus definitions. This feature also includes basic support via email and phone.

## Technical Support

To contact Barracuda Networks Technical Support:

- By phone: call 1-408-342-5400, or if you are in the United States, (888) Anti-Spam, or (888) 268-4772
- By email: use [support@barracuda.com](mailto:support@barracuda.com)
- Online: visit <http://www.barracuda.com/support> and click on the Support Case Creation link.

There is also a Barracuda Networks Support Forum available, where users can post and answer other users' questions. Register and log in at <http://forum.barracuda.com>.

## The Barracuda Web Application Firewall Models

*Table 1.1: Barracuda Web Application Firewall Models*

Feature	Model 360	Model 460	Model 660	Model 860	Model 960
<b>Web Application Firewall</b>					
HTTP/HTTPS/FTP Protocol Validation	X	X	X	X	X
Protection Against Common Attacks	X	X	X	X	X
Form Field Meta Data Validation	X	X	X	X	X
Web Site Cloaking	X	X	X	X	X
Response Control	X	X	X	X	X
Outbound Data Theft Protection	X	X	X	X	X
Granular Policies to HTML Elements	X	X	X	X	X
Protocol Limit Checks	X	X	X	X	X



Feature	Model 360	Model 460	Model 660	Model 860	Model 960
File Upload Control	X	X	X	X	X
Logging, Monitoring and Reporting	X	X	X	X	X
Brute Force Attack Protection	X	X	X	X	X
Session Tracking		X	X	X	X
Web Address Translation		X	X	X	X
Rate Control			X	X	X
XML Firewall			X	X	X
Adaptive Profiling			X	X	X
Anti Virus Checks			X	X	X
<b>Application Delivery and Acceleration</b>					
High Availability	X	X	X	X	X
SSL Offloading	X	X	X	X	X
Application Content Routing		X	X	X	X
Response Caching		X	X	X	X
Response Compression		X	X	X	X
Connection Pooling		X	X	X	X
Load Balancing		X	X	X	X
<b>User Access Control</b>					
Authentication and Authorization	X	X	X	X	X
Single Sign-On	X	X	X	X	X
LDAP/RADIUS Integration		X	X	X	X
<b>Management</b>					
Web Based User Interface	X	X	X	X	X
XML RPC API	X	X	X	X	X
Role Based Access		X	X	X	X



# Web Application Firewall Concepts

---

This chapter provides an overview of the Barracuda Web Application Firewall and includes the following topics:

- *Attacks and Terminology* on page 20
- *Web Application Firewall Concepts* on page 24
- *Application Acceleration and Assurance* on page 27

# Attacks and Terminology

The following table describes some of the common Web application attack techniques and the corresponding protection employed by the Barracuda Web Application Firewall.

*Table 2.1: Protection against different methods of attack*

Technique	Description	Protection provided by Barracuda Web Application Firewall
<b>Injection Attacks</b>		
SQL Injection	An SQL injection attack is insertion of a SQL query via the input data from the client to the application. A successful SQL injection attack can read sensitive data from the database, modify database data, or shutdown the server.	Protects against all SQL injection vulnerabilities by inspecting application traffic and blocking all methods of inserting dangerous database commands into URLs, headers, and forms.
Cross-Site Scripting	Cross-site scripting takes advantage of a vulnerable Web site to attack clients who visit that Web site. The most frequent goal is to steal the credentials of users who visit the site.	Protects against cross-site scripting vulnerabilities by inspecting application traffic and blocking all methods of inserting malicious scripts into URLs, headers, and forms.
Remote File Injection	Remote File Inclusion attacks allow malicious users to run their own PHP code on a vulnerable website to access anything that the PHP program could: databases, password files, etc.	Protects against all remote file injection vulnerabilities by inspecting application traffic and blocking all methods of inserting dangerous database commands into URLs, headers, and forms.
Command Injection	Operating system and platform commands can often be used to give attackers access to data and escalate privileges on back-end servers.	Protects against all command injection vulnerabilities by inspecting application traffic and blocking all methods of inserting dangerous operating system and platform commands into URLs, headers, and forms.
Meta-character Injection	Meta-character injection attack is used to exploit Web sites by sending in meta-characters, which have special meaning to programming languages, operating system commands, individual program procedures, database queries, etc. These special characters can adversely alter the behavior of a Web application.	Protects against all meta-character injection vulnerabilities by inspecting application traffic and blocking all methods of inserting dangerous database commands into URLs, headers, and forms.
<b>Session Attacks</b>		
Cookie/Session Poisoning	Cookies are often used to transmit sensitive credentials, and they can be modified to escalate access or assume another user's identity.	Digitally encrypts, signs, and time-stamps cookies, protecting their content from tampering.
Cookie Snooping	Cookies are commonly used to transmit user credentials and are often encoded only with simple encoding methods like Base64. This can lead to disclosure of login credentials.	Digitally encrypts, signs, and time-stamps cookies, protecting their content from tampering.

*Table 2.1: Protection against different methods of attack*

Technique	Description	Protection provided by Barracuda Web Application Firewall
Session Hijacking	First the attacker uses a sniffer to capture a valid token session called "Session ID", then he uses the valid token session to gain unauthorized access to the Web Server.	Digitally encrypts, signs, and time-stamps cookies, protecting their content from tampering. Also ties session cookies to the originating client and if it arrives from another source, it is denied.
Cross-Site Request Forgery (CSRF) Attack	CSRF attack tricks the victim into loading a page that contains a malicious request and makes the victim perform actions that they didn't intend to, such as logout, purchase item, change account information, retrieve account information, or any other function provided by the vulnerable website.	Digitally encrypts, signs, and time-stamps cookies, protecting their content from tampering. Also ties session cookies to the originating client and if it arrives from another source, it is denied.
<b>Denial of Service (DoS) Attacks</b>		
Length Attack	Requests with lengths greater than the defaults are potential buffer overflow attack which is the most common kind of DoS attack.	Protects against length attacks by restricting the number, name and value lengths for cookies, HTTP request URL and HTTP headers.
Rate Control Attack	When the number of connections allowed from any specific IP address goes over the Rate Control threshold, it leads to a DoS attack.	Protects against rate control attacks by limiting the number of connections allowed from any specific IP address. When the number goes over the Rate Control threshold, the Barracuda Web Application Firewall blocks further connections.
Session based Attack	When the number of sessions originating from a particular client IP address in a given interval of time increases outside a given limit, it leads to session-based DoS attack.	Protects against session based attacks by limiting the number of sessions originating from a particular client IP address in a given interval of time.
<b>Other Attacks</b>		
Reconnaissance	Hacking-related reconnaissance is to improve the probability that an attack against a target network will be successful and to improve the attackers odds of successfully masking their identity. For example, in a game of chess, a player might perform the "mental walk-through" prior to executing a chess move.	Protects against reconnaissance by unintended information disclosure from error messages, using URL and Parameter profiles for an application and allowing only user requests that match the legitimate profile and specifying the rules to allow or deny a particular URL.
Bruteforce Attack	A bruteforce attack consists of trying every possible code, combination, or password until you find the right one. For example, imagine a system which only allows 4 digit PIN codes. This means that there are a maximum of 10,000 possible PIN combinations that the attacker can try out to gain access to the system.	Protect against bruteforce attacks by counting requests which result in server error and blocks the attacker once a set number of invalid attempts are made within a given time. It also protects against lots of malicious requests in quick succession trying to get access.

*Table 2.1: Protection against different methods of attack*

Technique	Description	Protection provided by Barracuda Web Application Firewall
Cryptographic Interception	Hackers seldom attempt to break strong encryption like SSL. Instead, they attack sensitive hand-off points where data is temporarily unprotected. The use of multiple devices for managing cryptography and encryption makes cryptographic interception far more likely.	Has extensive SSL security capabilities and can ensure that no unencrypted traffic traverses the network in any circumstance. Combining all critical DMZ functionality into a single device also reduces the risk of exposure.
Forceful Browsing/Directory Traversal	Forceful browsing is an attempt to access files and directories in a Web service without using the Web service to provide the links to the files and directories. An attacker who attempts to execute a forceful browsing attack would see a directory such as <code>http://someapp.com/guests/welcome.html</code> and attempt to go to <code>http://someapp.com/members/welcome.html</code> . In this way, the attacker gains access to the members area of the website without supplying proper credentials.	Prevents the access of unpublished Web pages by using application profiles and blocking requests with path traversal metacharacters and enforcing access to only those pages that the application was designed to expose.
Log Tampering	Erasing and tampering with transaction logs allows an attacker to cover their tracks or alter Web transaction records.	Centralizes the collection of all back-end server logs, then digitally signs and encrypts them to prevent tampering. As with all its features, secure logs can be generated on a per-application basis.
Error Message Interception	Information in error messages are often rich with site-specific information, allowing an attacker to learn private application architectures.	The Website cloaking feature prevents unintended information disclosure from error messages.
Attack Obfuscation	Hackers frequently disguise attacks by encoding their requests with methods like URL encoding or Unicode.	Fully decodes URL, Unicode, and polymorphic encoding before inspection.
Application Platform Exploits	Well-known exploits can often be addressed through a patch, but patching is not always timely.	Allows for blocking of well-known attacks, effectively buying time for proper patch management.
Security Management Exploits	Sophisticated attackers may target security management systems in an attempt to modify or turn off security enforcement. (These could be either network or application layer.)	Has all management functions securely firewalled from production traffic and is operated through dedicated, secure management channels.
Parameter/Form Tampering	Parameters used in URLs, HTTP headers, and forms are often used to control and validate access to sensitive information.	Protects against parameter tampering by using parameter profiles for all application parameters and allowing only user requests that match the legitimate profile.
Buffer Overflow	Attackers attempt to flood vulnerable back-end servers with excess requests. If successful, attackers can often execute commands directly on the compromised server.	Automatically enforces legitimate buffer limits at the perimeter, ensuring that even vulnerable servers cannot be compromised.

## Basic Terminology

---

The following is a list of some of the terms used by the Barracuda Web Application Firewall.

*Table 2.2: Basic terminology*

Term	Description
Real Server	Identifies the server (IP, port) that hosts the Web application that will be protected by the Barracuda Web Application Firewall.
Virtual IP (VIP)	The user-defined IP address on which the Barracuda Web Application Firewall accepts traffic for a configured Web application. In a redundant configuration it is a virtual address that applies regardless of which Barracuda Web Application Firewall is managing the application at any given time.
Services	A user-designed entry point for controlled access to the Web site. A service sets the front-end interface (VIP) and a variety of possible controls (such as SSL encryption, authentication, load balancing, and caching policies) for the Web site.
Bridge Mode	In Bridge mode, the Barracuda Web Application Firewall uses the same IP address for the VIP and back-end server, and hence does not require changes to the existing network infrastructure.
One-arm Mode	In One-armed mode, WAN port is used for both external and internal traffic passing through the Barracuda Web Application Firewall. The network throughput is less as only one port (WAN) is used.
Full Proxy Mode	In Full proxy, the Barracuda Web Application Firewall is deployed in-line, using both the physical ports (WAN and LAN) of the device. This is the recommended configuration as it provides the best security.

# Web Application Firewall Concepts

---

## Negative Security Model

The negative security model relies on a set of signature patterns to determine malicious content. Anything not matching these patterns is allowed by default. Legacy network security solutions like IPS and IDS exclusively employed such a model. Such a model suffers from maintaining and regularly updating the signatures and decreased performance when the list gets long. Moreover, signatures are added reactively, that is, after the attacks have already happened somewhere. Thus, relying solely on such a model may not protect against “zero day attacks”.

## Positive Security Model

A positive security model (also known as "whitelist") is one that explicitly defines what is allowed, and rejects everything else. This provides a strong security model for neutralizing “zero day attacks” not anticipated in advance. For example if an entry in a FORM representing a person's age is restricted to the range 1-120 only, it will always block new injection attacks without having to update signatures.

## Adaptive Profiling

Adaptive Profiling generates a positive security profile for your applications over time while providing instant post-deployment protection using a negative security model. The positive security profile is generated over time by observing successful requests and responses. Multiple configurable heuristics determine that anomalous traffic is not used for generating the profile.

## False Positives

False positives are requests, or responses, that are deemed as attacks and denied, even though they are safe and no malicious intent is involved.

## Exception Profiling

Exception Profiling assists you in automating false positive reductions. A tight security policy may sometimes deny genuine requests. You can turn on exception profiling for desired Web site sections which identifies such false positives and automatically refines the security policy rules for the respective site sections. For critical site sections, the profiler can be instructed to issue recommendations rather than automatically modifying the policy. A heuristics screen allows you to configure the exception profiling criteria for different violations.

## Cloaking

The Barracuda Web Application Firewall masks or hides URL return codes, HTTP headers and content in error responses, and back-end IP addresses, making it less likely that hackers or worms will be able to launch a successful attack. For more information refer *Cloaking* on page 59.

## Allow/Deny Rules

Allow/Deny Rules allows you to define access restrictions on an individual URL basis. These rules override the firewall policies. If a request's URL matches an allow rule, then the firewall rules are not applied and the request is let through. On matching a deny rule, the request is immediately dropped.



For more information refer *Allowing/Denying Specific URLs* on page 51 and *Allowing/Denying Specific Headers* on page 52.

## Input Validation

Failure to validate URL and FORM parameter input values leads to almost all of the major injection vulnerabilities in applications, such as SQL Injection, cross site scripting, remote file inclusion, OS command injection, buffer overflow, denial of service, etc. You can validate the input using positive security, for example allowing only integers for a parameter representing age, or using negative security, for example by defining disallowed metacharacters and patterns in the input elements. The default security policies also provide length based restrictions across all user input elements.

## Web Site Profiles

Web site profiles consists of profiles for URLs and the parameters contained in those URLs. The URL profile defines a list of allowed parameters like HTTP methods, names and types of each parameter, query strings, length based restrictions, etc. The Parameter profile defines the allowed format of the parameter using either a negative or positive security model as well as length restrictions. Web site profiles can be defined manually or automatically generated by using Adaptive Security. For more information refer *Web Site Profiles* on page 65.

## Request Matching (Regex) Rules

You can use Perl Compatible Regular Expressions (PCRE) in various security policy elements to specify request/response and data patterns matching. Different elements of the request can be matched and can be combined in a very flexible manner to identify a request and apply security measures on those requests alone. Regex patterns can also be used for specifying input validation rules, data theft prevention, web address translation rules amongst others. For more information refer *Extended Match and Condition Expressions* on page 214.

## Active and Passive modes

Passive mode only logs attacks whereas active mode blocks them as well. This mode can be applied at the Service level or more granularly at the security policy elements within the Service such as Allow Deny rules, URL profiles etc. Consider using the passive mode after initial deployment to reduce the impact of false positives. Then, incrementally turn the policy to active mode as and when the incidence of false positives reduces to an acceptable minimum for each policy element.

## Policy Tuner

Policy Tuner assists you in fine tuning your security policies by integrating with the Web Firewall log entries. For each entry in the logs, it provides a recommendation which can be applied automatically to the policy. When applied, the changed policy rule ensures that the violation that generated the log entry will not be considered an attack in the future. For more information refer *Policy Tuner* on page 153.

---

## Web Service Firewall

XML data is increasingly being used over HTTP in integrating systems using the SOAP messaging framework and the new breed of interactive web 2.0 applications that use XML intensively. XML based applications are susceptible to the same attacks as normal Web applications and introduce

additional vectors such as XML entity expansion. The XML Firewall module validates and defends against attacks embedded in SOAP messages and XML data.

### **XML Attacks and Prevention**

XML based applications can be targeted with denial of service attacks and attacks targeted on the XML processing framework itself, for example recursive entity expansion or other repeated processing that exhausts server resources before security policies can even be applied to the XML data. The system defends your XML applications against such attacks by performing full security checks and blocking malicious requests before they can reach the servers. You can also customize XML security, for example define maximum tree elements, child nodes, blocking processing instructions etc. For more information, refer *Table 14.1*.

### **SOAP Validation**

By importing your Web Service's WSDL and schema files, you can validate incoming and outgoing SOAP messages against them. Malicious attack patterns found in the messages get denied.

### **WS-I Basic Profile Assertions**

These are published by the Web Services Interoperability Organization for validating Web Services. The Barracuda Web Application Firewall performs these tests during run time validation for SOAP messages. There are forty two test case parameters, which are all set to 'Yes' by default. You can edit the existing values. For more information, refer *XML Firewall* on page 176.

## **Authentication, Authorization and Access Control**

---

### **Internal and External Authentication**

The Barracuda Web Application Firewall can authenticate your application users before allowing their requests to reach the servers. Authentication can be done by integrating with corporate authentication databases such as LDAP or Radius, or by creating an internal user database on the Barracuda Web Application Firewall itself. This lets you consolidate the authentication configuration and maintenance in a single centralized device front-ending multiple backend applications.

### **Authorization**

Authorization features allow you to define the level of access to authenticated users for different site sections. For example, only business partners may be allowed access to the partner portal section of your corporate Web site.

### **Single Sign-On (SSO)**

SSO provides single sign-on functionality across single and multiple cookie domains i.e. a single set of user credentials is used for authentication and authorization to access multiple applications across different Web servers and platforms, without having to re-authenticate. The Barracuda Web Application Firewall SSO supports both single domain and multi-domain SSO. For more information, refer *Single Sign-On (SSO)* on page 108.

# Application Acceleration and Assurance

---

## Application Acceleration

---

The following features accelerate the application delivery thereby reducing the page load times for your Web site visitors and creating a pleasant browsing experience.

### Content Routing

The Barracuda Web Application Firewall allows you to route application layer (Layer 7) content based on the content type. This allows you to partition your server content by content type and process the requests more efficiently by directing them to the relevant server. For example, all image content can be hosted on a separate server optimized for image delivery and then have the image requests routed to it. A server optimized for CGI scripts execution can be set up to handle all script requests (e.g. \*.cgi). For more information, refer *Content Rule* on page 90.

### Caching

Static outgoing content can be cached by the system in local memory and served directly in subsequent requests, relieving the backend servers to perform other tasks e.g. generating and delivering dynamic content. For more information, refer *Configuring Caching* on page 92.

### Compression

The system can dynamically compress outgoing content which can significantly reduce the transmission times, especially over slow network connections. Compression is also found very useful in verbose protocols over HTTP, specifically XML which is increasingly being used in new applications. For more information, refer *Configuring Compression* on page 94.

### Connection Pooling

Connection Pooling reuses the backend server connections from a pool of established connections which decreases the connection set up and tear down overhead, thereby delivering requests faster. It also reduces the server load and frees up resources for handling other important tasks. For more information refer *Creating Rate Control Pool* on page 146.

## Load Balancing

---

The load balancing module allows adding or removing servers for a protected Web site, without interrupting the existing traffic. It allows for scaling up your Web site performance as well as adding redundancy in case one of the servers fails.

### Session Persistence

Session Persistence directs requests from a particular client to the same server. This is required when the server maintains session information, for example shopping cart contents. Layer 7 session persistence is based on either the client IP or by actively inserting or monitoring server cookies to track clients. You can employ session persistence for your HTTPS services as well.

## Server Monitoring

To ensure optimum load balancing performance, the system allows you to monitor the server's health. If a server is found to be not responding, it is marked as down, its persistence requests may be directed to another healthy server and future requests are not sent to the server which is down. Monitoring may be done either inline of the user traffic, by examining the server response codes, or out-of-band, by generating independent requests.

## Load Balancing Algorithm

You can choose round robin, weighted round robin or least requests algorithms to determine the load balancer decisions.

## High Availability

---

Enabling High Availability (HA) on your Barracuda Web Application Firewall allows you to connect it to a backup Barracuda Web Application Firewall to act as a fail-over server in case your primary system fails for any reason. For more information refer *Creating a High Availability (HA) Environment* on page 157.

## Ethernet Hard Bypass

---

The hardware bypass on failure is implemented using a hardware bypass card in the system. On the event of any hardware or software failure, the bypass gets activated. In such a scenario, your client traffic is bridged to the backend sites and there is no downtime of your Web site. When the system is restored, the bypass gets disabled.

## Chapter 3

# Getting Started

---

This chapter provides general instructions for deploying and installing the Barracuda Web Application Firewall. This chapter covers the following topics:

- *Deployment Modes for the Barracuda Web Application Firewall* on page 30
- *Initial Setup* on page 36

# Deployment Modes for the Barracuda Web Application Firewall

Before deploying the Barracuda Web Application Firewall, user must get acquainted with the interfaces present on the Barracuda Web Application Firewall. They are:

- **WAN** - The WAN interface is connected to the client side or Internet facing network.
- **LAN** - The LAN interface is connected to the server side or Application facing network.
- **MGMT (Management)** - The MGMT interface is connected to a separate out-of-band network for securely managing the Barracuda Web Application Firewall.
  - By default the configuration happens through the WAN port
  - But it is highly recommended that the configuration happen through the MGMT port
- **Virtual IP (VIP)** - This is a per service IP address configured for each service on the Barracuda Web Application Firewall. Depending on the mode of deployment, these can be same as or different from the Real Server IP addresses. In Bridge-path mode, they are the same while in Reverse Proxy and One Arm mode they are different.

**Note**



WAN IP and Management IP should be in different subnets.

Services on the Barracuda Web Application Firewall can be deployed in the following three modes:

<i>Reverse Proxy (Recommended)</i> .....	32
<i>Bridge-Path</i> .....	33
<i>One-armed</i> .....	33

The deployment mode chosen is usually dependent on the type of network configuration that currently exists at your site as well as on the types of services that you want from the Barracuda Web Application Firewall. The Bridge-path is recommended for initial deployment, as it requires the least amount of invasive changes to your existing network configuration. For this reason the Barracuda Web Application Firewall is shipped with the bridge mode setting. Depending on your need, you can choose the mode of operation. To change the mode of operation, go to **BASIC > IP Configuration** page, select the appropriate radio button under **Operation Mode** and click **Save Changes**.

**Note**



When deploying in either One-armed or Reverse Proxy mode, the **Operation Mode** should be set to **Proxy**.

*Table 3.1* The following table lists the criteria to consider when deciding which deployment strategy is optimal for your environment.

*Table 3.1: Deployment Options*

Criteria	Reverse Proxy	Bridge-Path	One-Armed
Maximize network bandwidth (use both ports)	X	X	

Table 3.1: Deployment Options

Criteria	Reverse Proxy	Bridge-Path	One-Armed
Create secure path to Web servers	X		
Minimize change to existing network infrastructure			X
Integrate with existing enterprise load balancers			X
Establish multiple paths to servers for testing			X*
Cannot change existing IP addresses.		X	X**

**Note:**

\* - Clients can reach the website either (i) via the Barracuda Web Application Firewall VIP (secure) or (ii) directly through the server host IP (insecure).

\*\* - Server host can retain IP address, but DNS will have to be changed to point to the VIP for the website on the Barracuda Web Application Firewall. Also accessing directly via server IP will be insecure since it bypasses the Barracuda Web Application Firewall.

## Reverse Proxy (Recommended)

In Reverse Proxy, the Barracuda Web Application Firewall is deployed in-line, using both the physical ports (WAN and LAN) of the device. This is the recommended configuration and/as it provides the best security, but it requires changes to the existing network infrastructure.

With reverse proxy, the WAN and LAN interface of the Barracuda Web Application Firewall must be on separate logical networks.

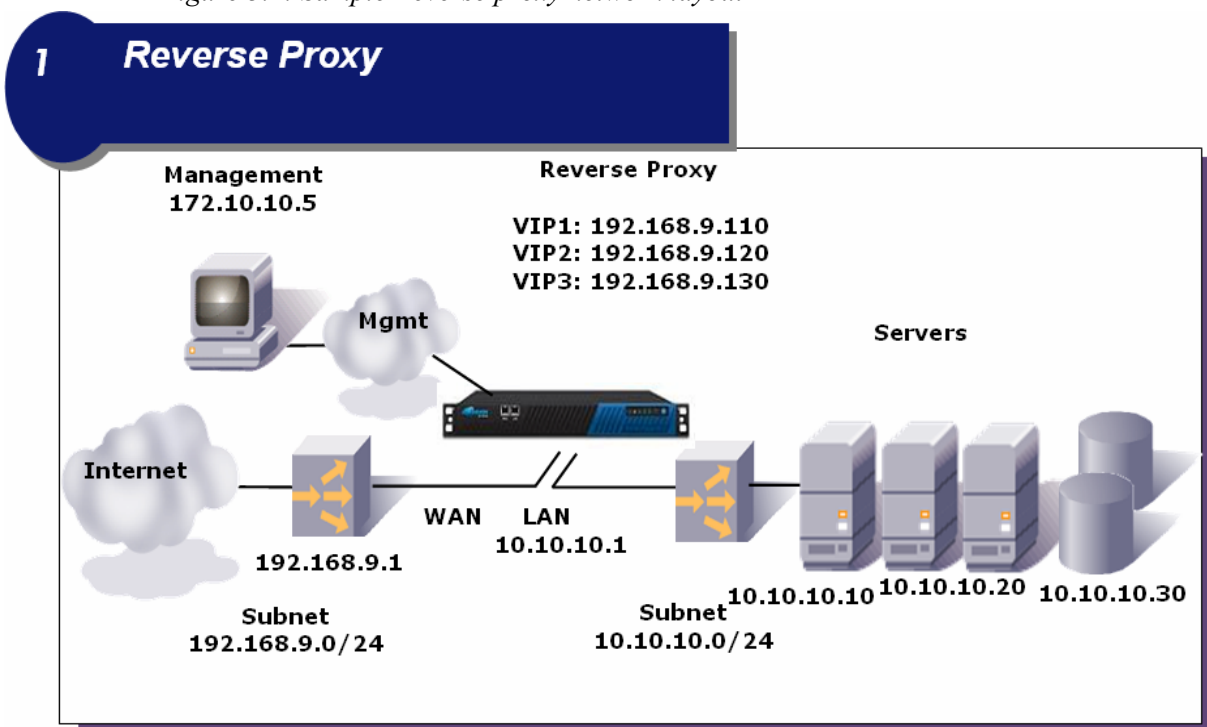
- The servers are moved to a private network connected through a switch on the LAN port.
- The WAN port connects to a switch to which the publicly accessible IP addresses will be routed from the internet.

Each server in the private network gets a virtual IP on the Barracuda Web Application Firewall. The virtual IP addresses should be accessible from the internet and should be routed to the WAN port via the switch connected to it. When a request is received by the Barracuda Web Application Firewall on a VIP advertised through the WAN port, it inspects it and redirects it to the real server on the private network via the LAN port.

The following table describes the advantages and disadvantages of deploying your Barracuda Web Application Firewall in *Reverse proxy* mode.

Advantages	Disadvantages
Full feature availability including Load Balancing and Instant SSL	Network changes such as Server IP addresses and DNS mappings are required
Most Secure Deployment Scheme since backend servers are completely isolated	Backing out requires undo of all the changes
Fast High Availability failover	Deployment requires cutover of live services

Figure 3.1: Sample Reverse proxy network layout





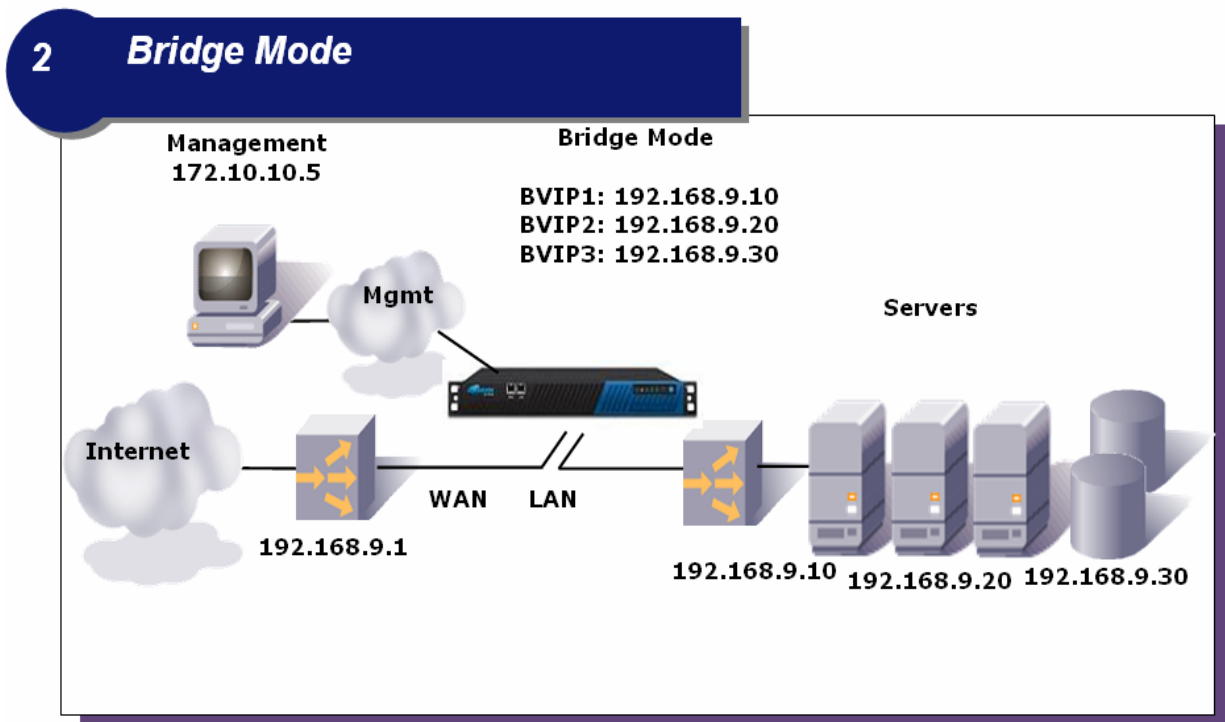
## Bridge-Path

Bridge-Path provides an easy configuration scenario by using the same IP address for the VIP and back-end server. It does not use any extra IP address. Also changes to the server IP addresses or DNS mappings are not required. Users may place the Barracuda Web Application Firewall inline with their existing IP infrastructure, and add servers as required without changing IP addresses. With Bridge-Path deployment, the WAN and LAN interfaces must be on physically separate networks and the LAN interface must be on the same logical switch as the servers.

The following table describes the advantages and disadvantages of deploying your Barracuda Web Application Firewall in **Bridge-Path** mode.

Advantages	Disadvantages
Minimal network changes since the existing IP infrastructure is reused	Sensitive to broadcast storms and other errors related to loops in a Spanning Tree protocol.
Real Servers keep their existing IP addresses	Less resilient to network misconfigurations
	Features like Load balancing, Instant SSL and TCP pooling are not available

Figure 3.2: Sample Bridge-Path network layout



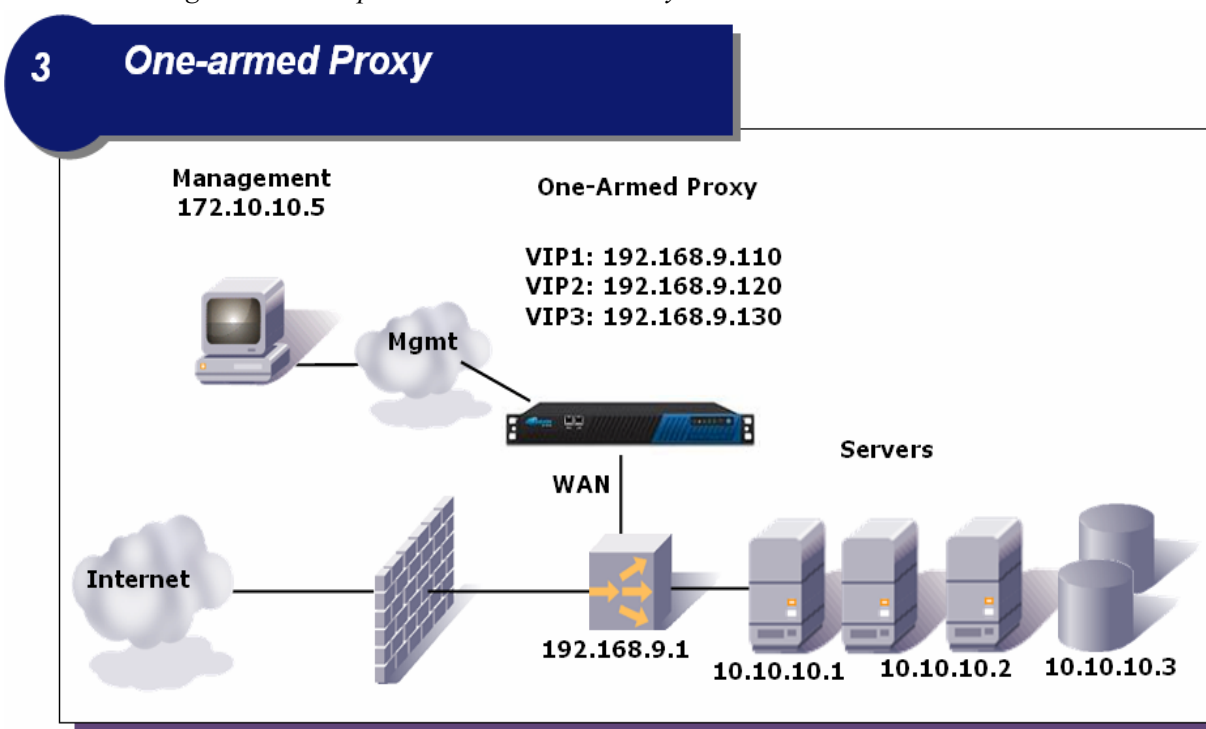
## One-armed

One-armed deployment minimizes changes to the existing infrastructure. This option uses WAN port for both external and internal traffic passing through the Barracuda Web Application Firewall. The network throughput is less as only one port (WAN) is used.

The following table describes the advantages and disadvantages of deploying your Barracuda Web Application Firewall in *One-Armed* mode.

Advantages	Disadvantages
Easier Deployment compared to Reverse Proxy. Network infrastructure and partitioning does not need to be changed.	Requires DNS, IP changes as in Reverse Proxy.
Helps establish multiple path to servers for testing.	Lower throughput since only one port (WAN) is used.
Easier integration with existing enterprise load balancers.	Not as secure as Reverse Proxy since server can be accessed directly.

Figure 3.3: Sample One-armed network layout



## Best Practise

Barracuda realizes that some customers don't want to change their IP addressing schemes and that certain implementation modes might be easier on the migration path to Reverse-Proxy mode. The following conclusion can be drawn from the above three modes of deployment:

- Using one interface, the One-Armed mode is by far the most transparent and easiest way to plug into a network, without affecting any existing traffic in the network.
- Using Bridge Mode, no renumbering of IP addresses on either for the Real Servers or for the client facing IP address for the Service is required. The bridge is transparent, so no existing services are disrupted, disconnecting it from the network is easy, and all proxy-based security features at the application layer are supported.
- Reverse-Proxy is the most secure of all topologies and results in a complete security barrier between the internet and the Web Services.

In addition, services in each of the modes can be run in **Passive** or **Active** mode.

- Passive mode simply logs offending traffic and doesn't block
- Active mode performs full blocking of threats

So the general practise would be to initially run the service in **Passive** mode as the application is new, there is not much traffic on it and then gradually change it to **Active** mode when the traffic starts coming in.

# Initial Setup

---

These are the general steps to set up your Barracuda Web Application Firewall. For more detailed instructions for each step, see the following reference pages.

<i>Prepare for the Installation .....</i>	<i>36</i>
<i>Connect Barracuda Web Application Firewall to Network .....</i>	<i>37</i>
<i>Configure IP Address and Network Settings .....</i>	<i>37</i>
<i>Configure the Barracuda Web Application Firewall .....</i>	<i>38</i>
<i>Activate Your Subscription Status .....</i>	<i>39</i>
<i>Update the Barracuda Web Application Firewall Firmware .....</i>	<i>40</i>
<i>Update Attack Definitions, Virus Definitions and Security Definitions .....</i>	<i>40</i>

## Prepare for the Installation

---

Before installing your Barracuda Web Application Firewall, complete the following tasks:

- Decide which type of deployment is most suitable to your network. For more information on the deployment options, see *Deployment Modes for the Barracuda Web Application Firewall* on page 30.
- To install the Barracuda Web Application Firewall certain changes might be required to your existing network. The exact nature of the changes depends upon your existing network configuration and the mode in which you deploy the Barracuda Web Application Firewall. Network Changes can be classified as:
  - **Hardware changes** - Changes related to cabling, switches, routers, network interfaces, etc.
  - **Configuration changes** - Changes related to DNS databases, IP addresses of hosts and services, router configuration etc.
- *(Reverse proxy deployment only)* Re-configure the Real Servers with a new private network and set the Real Servers' default gateway to an unused IP address in this subnet. This IP address will be assigned to the LAN IP address of the Barracuda Web Application Firewall in step 3a of *Configure the Barracuda Web Application Firewall*.
- Identify the TCP port numbers used by the services/applications running on the real servers that you want to protect.
- Verify you have the necessary equipment:
  - Barracuda Web Application Firewall (check that you have received the correct model)
  - AC power cord
  - Ethernet cables
  - Mounting rails (model 660 and higher) and screws
  - VGA monitor (recommended)
  - PS2 keyboard (recommended)

## Connect Barracuda Web Application Firewall to Network

---

1. Fasten the Barracuda Web Application Firewall to a standard 19-inch rack or other stable location.

### Caution



Do not block the cooling vents located on the front and rear of the unit.

2. If using **Reverse proxy**, then the network switch referenced in the steps below may be the same physical switch. If using **Bridge-Path**, however, then *separate* switches on different Layer 2 networks *must* be used.
  - 2a. Connect a CAT5 Ethernet cable from the **WAN** interface on the Barracuda Web Application Firewall to the network switch where the *VIPs* reside.
  - 2b. Connect a CAT5 Ethernet cable from the **LAN** interface on the Barracuda Web Application Firewall to the network switch where the *Real Servers* reside.

### Note



It is recommended to connect MGMT port located on the back panel of the unit to the network switch where the *VIPs* reside.

3. Connect the following to your Barracuda Web Application Firewall:
  - Power cord
  - VGA monitor
  - PS2 keyboard

After you connect the AC power cord, the Barracuda Web Application Firewall may power on for a few seconds and then power off. This behavior is normal.

4. Press the **Power** button located on the front of the unit.

The login prompt for the administrative console displays on the monitor, and the power light on the front of the Barracuda Web Application Firewall turns on. For a description of each indicator light, refer to *Understanding the Indicator Lights* on page 131.

## Configure IP Address and Network Settings

---

The Barracuda Web Application Firewall is assigned a default IP address of 192.168.200.200. You can change the address using the administrative console or by pressing and holding the **RESET** button on the front panel.

Holding **RESET** for eight seconds changes the default IP address to 192.168.1.200. Holding the button for 12 seconds changes the IP address to 10.1.1.200.

### To set a new IP address from the administrative console:

1. Connect your keyboard and monitor directly to the Barracuda Web Application Firewall.
2. At the `barracuda` login prompt, enter **admin** for the login and **admin** for the password.

The User Confirmation Requested window displays the current IP configuration of the Barracuda Web Application Firewall.

3. Using your Tab key, select **Change** and press **Enter** to change the IP configuration.
4. Enter the new IP address, netmask, and default gateway for your Barracuda Web Application Firewall. Select **Save** to enter your changes. (The Primary and Secondary DNS fields are optional at this time, but if not entered at this step then they must be entered in step 3c. of *To configure the Barracuda Web Application Firewall* on page 38). Select **Exit**.

The new IP address and network settings are applied to your Barracuda Web Application Firewall.

## Configure the Barracuda Web Application Firewall

---

After specifying the IP address of the Barracuda Web Application Firewall and opening the necessary ports on your corporate firewall, configure the Barracuda Web Application Firewall from the Web administration interface. Make sure the system being used to access the Web interface is connected to the same network as the Barracuda Web Application Firewall, and that the appropriate routing is in place to allow connection to the Barracuda Web Application Firewall's IP address via a Web browser.

### To configure the Barracuda Web Application Firewall

1. From a Web browser, enter the IP address of the Barracuda Web Application Firewall followed by port 8000.

For example: `http://192.168.200.200:8000`.

2. To log into the administration interface, enter **admin** for the username and **admin** for the password.
3. Select **BASIC > IP Configuration**, and perform the following steps:

- 3a. Enter the following information in the LAN IP Configuration section:

- **LAN IP Address** - The address that connects the Barracuda Web Application Firewall to the Real Server network.

*When in Reverse proxy mode, the LAN interface provides the default gateway for the Real Servers. All Real Server IP addresses need to be in the same subnet as the LAN IP address because they will need to use this IP as their default gateway.*

- **LAN Netmask** - The subnet mask tied to the LAN.

- 3b. Enter the IP address of your primary and secondary DNS servers (if these have not yet been set up).
- 3c. Enter the default hostname and default domain name of the Barracuda Web Application Firewall.
- 3d. Click **Save Changes**.

#### Note



When you **reconfigure** the WAN IP address of the Barracuda Web Application Firewall on the **IP Configuration** page, you will be disconnected from the administration interface. Please log in again using the new IP address.

4. Select **BASIC > Administration**, and perform the following steps:
  - 4a. Assign a new administration password to the Barracuda Web Application Firewall (optional). This step is highly recommended.
  - 4b. Make sure the local time zone is set correctly.

Time on the Barracuda Web Application Firewall is automatically updated via NTP (Network Time Protocol). It requires that port 123 is opened for outbound UDP (User Datagram Protocol) traffic on your firewall (if the Barracuda Web Application Firewall is located behind one).

It is important that the time zone is set correctly because this information is used to coordinate traffic distribution and in all logs and reports.

- 4c. If desired, change the port number used to access the Barracuda Web Application Firewall administration interface. The default port is 8000.
- 4d. Enter the amount of time for the session expiration length (in minutes) of your Web administration interface session.

At expiration, you are required to log back into the administration interface.

- 4e. (Optional) Specify your local SMTP server. Enter the email address for your Administrator to receive system and threat email alerts and notifications.
- 4f. Click **Save Changes**.

## Activate Your Subscription Status

---

After installation, your Energize Updates and other optional subscriptions must be activated for the Barracuda Web Application Firewall to be fully enabled, and continue to receive the latest updates to all virus, attack, and security definitions from Barracuda Central. The Energize Updates service is responsible for downloading these updates to your Barracuda Web Application Firewall.

### To activate your subscription status:

1. At the top of every page, you may see the following warning:

**Error: Activation has not been completed. Please activate your Barracuda Spam & Virus Firewall to enable functionality. ([Click here to activate](#))**

2. Click on the designated link to open up the **Product Activation** page in a new browser window.
3. On the **Product Activation** page, fill in the required fields and click **Activate**. A confirmation page opens to display the terms of your subscription.
  - 3a. If your Barracuda Web Application Firewall is not able to communicate directly to Barracuda Central servers, then an Activation Code will be displayed as well which you will need to enter in the next step.
4. Return to the Barracuda Web Application Firewall administration interface and navigate to the **BASIC > Status** page. In the **Subscription Status** section, verify that the word *Current* appears next to **Energize Updates**, **Instant Replacement Service** (if purchased), and **Premium Support** (if purchased).
  - 4a. If you had received an Activation Code above then there will also be an **Activation Code** area in this section, where you must first enter that Code and click **Activate** in order to activate your Barracuda Web Application Firewall.
5. There may be a slight delay of a few minutes for the display to reflect your updated subscription status. If the status is still showing as unactivated, click **Refresh** in the **Subscription Status** section.

#### Note



If your subscription status does not change to *Current*, or if you have trouble filling out the **Product Activation** page, call your Barracuda Networks sales representative.

## Update the Barracuda Web Application Firewall Firmware

---

**To update the firmware on the Barracuda Web Application Firewall:**

1. Select **Advanced > Firmware Update**.
2. Read the release notes to learn about the latest features and fixes provided in the new firmware version.
3. Click **Download Now** next to Latest Version. Click **OK** on the download duration window.  
Updating the firmware may take several minutes. Do not turn off the unit during this process.  
**Download Now** is disabled if the Barracuda Web Application Firewall is already up-to-date with the latest firmware version.  
The Barracuda Web Application Firewall begins downloading the latest firmware version. You can view the download status by clicking **Refresh**. A “Firmware downloaded” message displays once the download is complete.
4. Click **Apply Now** when the download completes.
5. Click **OK** when prompted to reboot the Barracuda Web Application Firewall.  
A Status page displays the progress of the reboot. Once the reboot is complete, the login page appears.

## Update Attack Definitions, Virus Definitions and Security Definitions

---

**To apply latest attack, virus and security definitions.**

1. Select **Advanced > Energize Updates**.
2. Select **Hourly** or **Daily** for **Automatically Update** parameter. The recommended setting is **Hourly** for Attack Definition Updates and **Daily** for Virus Definition Updates.
3. Check to see if the current version is the same as the latest general release. If the rules are up-to-date, proceed to the next section. If the rules are not up-to-date, continue to the next step.
4. Click **Update** to download and install the latest available attack definitions, virus definitions or security definitions onto the Barracuda Web Application Firewall.
5. Click **Save Changes**.



## Chapter 4

# Securing a Web Site

---

This chapter describes the configuration and monitoring tasks you can perform from the Web interface. The following topic is covered:

- *Creating Services* on page 42

### Note



For more detailed information about a specific page in the Web interface, view the online help by clicking the question mark icon on the right side of the interface.

# Creating Services

---

The **BASIC > Services** page allows you to add a new service that specifies the parameters for configuring a service and server(s) to be protected by the Barracuda Web Application Firewall. The following types of services can be created under Proxy mode:

- HTTP service
- HTTPS service
- Redirect service
- Instant SSL service
- Custom service
- Custom SSL service
- FTP service

## Note



In **Bridge All Traffic** mode, only HTTP, HTTPS, Custom and Custom SSL services can be created. To change the operation mode, go to the **BASIC > IP Configuration** page, set the **Mode of Operation** to **Proxy** under **Operation Mode** section.

By default, passive mode is enabled for a service. For more information refer *Passive versus Active Mode* on page 50. To change it to active mode, click **Edit** next to the service and select “**No**” for **Passive Mode** under **Basic Security** section. Click **Save Changes**.

## Note



In **Bridge All Traffic** mode, the same IP address is used for both the VIP (service) and the server. Therefore **Real Server** field is not available. To change the operation mode, go to the **BASIC > IP Configuration** page, set the **Mode of Operation** to **Proxy** under **Operation Mode** section.

## Deploying HTTP service

---

An **HTTP service** acts as a front-end for a non-encrypted HTTP application on the server.

### To create an HTTP service:

1. Specify values for the following fields:
  - **Service Name** - Name used to identify this specific service.
  - **IP Address** - The IP address used to reach this service.
  - **Port** - Specify the specific TCP port for the service to listens on.
  - **Type** - Select HTTP from the drop down list.
  - **Real Server** - Specify the IP Address of the server that hosts the service that will be protected by the Barracuda Web Application Firewall.
2. Click **Add**. The service appears on the **BASIC > Services** page with a green, orange, or red health indicator next to it. For more information, refer to the next section.
3. To configure advanced settings for a service, click **Edit** next to the service.

## Deploying HTTPS service

---

An **HTTPS service** acts as a front-end for an encrypted HTTPS application on the server. This allows all transactions to the clients to be authenticated via SSL.

### To manually create an HTTPS service:

1. Specify values for the following fields:
  - **Service Name** - Name used to identify this specific service.
  - **IP Address** - The IP address used to reach this service.
  - **Port** - Specify the specific TCP port for the service to listens on.
  - **Type** - Select HTTPS from the drop down list.
  - **Real Server** - Specify the IP Address of the server that hosts the service that will be protected by the Barracuda Web Application Firewall.
  - **Certificate** - Select a certificate from the drop-down list. This is the certificate presented by the service when authenticating itself to a browser or some other client. The list of certificates available is based on the certificates that are created or imported in the **BASIC > Certificates** page.
2. Click **Add**. The service appears on the **BASIC > Services** page with a green, orange, or red health indicator next to it.
3. This creates an HTTPS service that encrypts only the contents between the Barracuda Web Application Firewall and the client browser. To use SSL encryption between the Barracuda Web Application Firewall and the server, click **Edit** next to the server and select “**Yes**” for **Server uses SSL** under **SSL (Server)** section.
4. To configure advanced settings for a service, click **Edit** next to the service.

## Securing an HTTP Web Site with HTTPS

---

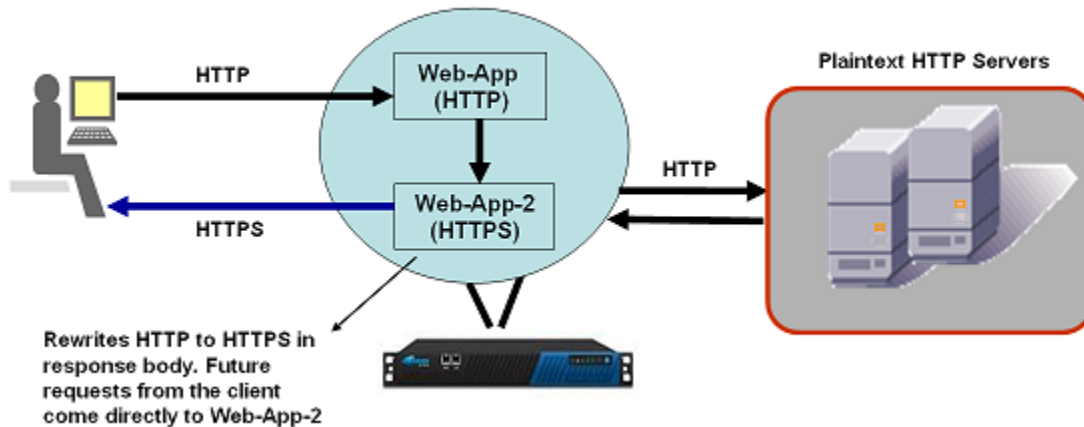
An **Instant SSL service** redirects an HTTP connection to an HTTPS service. This creates two services with the same IP address. An HTTPS service with port 443 and a redirect service with port 80. The redirect service is a non-SSL service that redirects all the requests to the HTTPS service. At least one secured site domain should be specified to determine which links in the response will be converted from 'http' to 'https'. For example refer *Figure 4.1*; if *http://www.barracuda.com* is specified, all such links found in the outgoing response will be rewritten to *https://www.barracuda.com*. After adding, you can edit the HTTPS service to add more domains which must be rewritten in the response. On receiving the request, the redirect service does the redirection to the service on port 443/HTTPS which in turn sends it to the servers. The HTTPS service rewrites an “http:...” request into an “https:...” in the response content.

### SharePoint Rewrite

The Instant SSL also supports SharePoint Rewrite. Microsoft’s SharePoint provides an enterprise business solution that integrates information from various systems into one solution through single sign-on and enterprise application integration capabilities, with flexible deployment options and management tools. Users can find relevant information quickly through customization and personalization of portal content and layout, as well as by audience targeting, scan target information, programs, and updates to audiences based on their role, team membership, interest, security group, or any other membership criteria that can be defined.

Use the **BASIC > Services** and click **Edit** against a HTTPS service to turn ‘On’ **SharePoint Rewrite Support**, which is not recommended. By default SharePoint Rewrite is disabled.

Figure 4.1: Instant SSL Service



#### Note



Instant SSL works only in **Proxy** mode.

#### To manually create an Instant SSL service:

1. Specify values for the following fields:
  - **Service Name** - Name used to identify this specific service.
  - **IP Address** - The IP address used to reach this service.
  - **Port** - Specify the specific TCP/UDP port for the service.
  - **Type** - Select Instant SSL from the drop down list.
  - **Real Server** - Specify the IP Address of the server that hosts the service that will be protected by the Barracuda Web Application Firewall.
  - **Domain** - Specify the main domain which identifies links to be rewritten from 'http' to 'https'.
  - **Certificate** - Select a certificate from the drop-down list. This is the certificate presented by the service when authenticating itself to a browser or some other client. The list of certificates available is based on the certificates that are created or imported in the [BASIC > Certificates](#) page.
2. Click **Add**. The service appears on the [BASIC > Services](#) page with a green, orange, or red health indicator next to it.
3. To configure advanced settings for a service, click **Edit** next to the service.

## Creating a Redirect service

The redirect service is a non-SSL service that redirects all the HTTP requests to an existing HTTPS service. Since the only purpose of this service is to redirect to an existing service, the Real Server IP cannot be specified. The redirect service is useful for allowing client requests on port 80 even when the server is only serving SSL requests on port 443.

### To manually create a Redirect service:

1. Specify values for the following fields:
  - **Service Name** - Name used to identify this redirect service.
  - **IP Address** - The IP address used to reach this service.
  - **Port** - Specify the specific TCP/UDP port for the service.
  - **Type** - Select Redirect Service from the drop down list.
2. Click **Add**. The service appears on the **BASIC > Services** page with a green, orange, or red health indicator next to it.
3. To configure advanced settings for a service, click **Edit** next to the service.

## Creating a Custom Service

---

A custom service allows the Barracuda Web Application Firewall to process any application layer protocol over TCP. The data sent by the client to a custom service is forwarded to the back-end servers without analysis. The Barracuda Web Application Firewall does not validate the incoming requests or outgoing responses.

### To manually create a Custom service:

1. Specify values for the following fields:
  - **Service Name** - Name used to identify this custom service.
  - **IP Address** - The IP address used to reach this service.
  - **Port** - Specify the specific TCP/UDP port for the service.
  - **Type** - Select Custom Service from the drop down list.
  - **Real Server** - Specify the IP Address of the server that hosts the service that will be protected by the Barracuda Web Application Firewall.
2. Click **Add**. The service appears on the **BASIC > Services** page with a green, orange, or red health indicator next to it.
3. To configure advanced settings for a service, click **Edit** next to the service.

## Creating a Custom SSL Service

---

A custom SSL service is used to encrypt traffic sent to a custom application.

### To manually create a Custom SSL service:

1. Specify values for the following fields:
  - **Service Name** - Name used to identify this custom SSL service.
  - **IP Address** - The IP address used to reach this service.
  - **Port** - Specify the specific TCP/UDP port for the service.
  - **Type** - Select Custom SSL Service from the drop down list.
  - **Real Server** - Specify the IP Address of the server that hosts the service that will be protected by the Barracuda Web Application Firewall.
  - **Certificate** - Select a certificate from the drop-down list. This is the certificate presented by the service when authenticating itself to a browser or some other client. The list of certificates available is based on the certificates that are created or imported in the **BASIC > Certificates** page.

2. Click **Add**. The service appears on the **BASIC > Services** page with a green, orange, or red health indicator next to it.
3. To configure advanced settings for a service, click **Edit** next to the service.

## Creating a FTP Service

---

An FTP service allows the Barracuda Web Application Firewall to process FTP traffic from the clients to the servers. An FTP client connects to a FTP server to manipulate files on that server.

### Note



FTP service is available only in **Proxy** mode.

### To manually create a FTP service:

1. Specify values for the following fields:
  - **Service Name** - Name used to identify this FTP service.
  - **IP Address** - The IP address used to reach this service.
  - **Port** - Specify the specific TCP/UDP port for the service.
  - **Type** - Select FTP Service from the drop down list.
  - **Real Server** - Specify the IP Address of the server that hosts the service that will be protected by the Barracuda Web Application Firewall.
2. Click **Add**. The service appears on the **BASIC > Services** page with a green, orange, or red health indicator next to it.
3. To configure advanced settings for a service, click **Edit** next to the service.

## Configuring FTP Attack Prevention

The Barracuda Web Application Firewall allows you to specify which FTP commands should not be allowed by enabling FTP Attack Prevention and also to specify the IP address for PASV mode. The PASV mode is a secure mode of operation for FTP protocol, because it does not require the client to open a port, which could potentially be used for attacks. PASV mode opens up a random port for the data connection and uses the VIP for the IP address.

### To configure FTP Attack Prevention:

1. Click **Edit** under **Actions** against the created FTP service.
2. Under **FTP** section, specify values for the following fields:
  - **FTP Attack Prevention Status** - Set to “Yes” to enable attack prevention for this FTP service.
  - **PASV IP Address** - Enter the IP address for the PASV mode data connection. If an IP address is not specified, the PASV response uses the VIP.
3. Click **Save Changes** to save the above settings.

## Editing Basic Security for a Service

---

When you create a service, a basic set of Web Firewall features are activated automatically. By clicking **Edit** for a service, you can change the basic security for a service which include, security policy and log levels. These default configuration features provide adequate amount of protection from the majority of web attacks like injection attacks, denial of service attacks, data theft protection, etc. In addition, there are several other parameters that can be set to expand and increase service attack prevention.

### To edit Basic Security settings:

1. Specify values for the following fields:

- **Web Firewall Policy** - Select the web firewall policy used for this service. By default all services use 'default' as the policy. Other policies that can be selected are sharepoint, owa and oracle.
- **Web Firewall Log Level** - Select the level of logging events for a module. A lower level signifies lesser information logging. Most of the attacks fall under 1-Alert. Few cookie related logs and cloaking logs fall under 4-Warning, so these would be logged automatically with 4-Warning.
- **Rate Control Status** - Set the status to **On** to enable the Rate Control Pool for this service.
- **Rate Control Pool** - Select a Rate Control Pool for this service. The Rate Control Pool enables you to throttle client requests. Rate Control Pools can be specified on the [ADVANCED > Libraries](#) page. For more information, refer *Creating Rate Control Pool* on page 146.
- **Mode** - Select the mode for the service. If set to **Passive**, logs the intrusions but allows traffic to pass through. If set to **Active**, logs and blocks the intrusions.
- **Trusted Hosts Action** - Select any special action that needs to be taken for a set of hosts (trusted) accessing the service. If the action is set to either Allow or Passive, all requests, including those that are possible attacks are allowed. In Allow mode, no logs are generated, whereas in Passive mode, logs are generated. When set to Default, no special action is taken for any trusted host, and the **Mode** selected above applies to the Trusted Hosts as well.
- **Trusted Hosts Group** - Select the trusted hosts group to apply any special action to. This setting is required only if the parameter "Trusted Hosts Action" is set to either Allow or Passive.
- **Ignore case** - Select how the URLs are matched to rules like URL ACLs and URL Profiles. If this is set "Yes", the case of the URL is ignored when matching with any Barracuda Web Application Firewall rule.

2. Click **Save Changes** to save the above settings.

## Load Balance

---

A load balancer is a networking device that distributes traffic to servers so that the demand on the servers is evenly distributed. This ensures that one server is not overburdened and Web traffic is sent faster to its intended destination.

The Barracuda Web Application Firewall has the capability to act as a stand-alone load balancer or in conjunction with other load balancers. It can be situated in front of a set of back-end servers and distribute incoming traffic across the servers based on an algorithm you choose. The Barracuda Web Application Firewall supports load balancing in all types of applications (Web, FTP, and custom).

The Barracuda Web Application Firewall includes the following load-balancing features:

- Sends traffic requests to a collection of back-end servers according to a user-configured algorithm.
- Automatically identifies the status of a server for appropriate routing of traffic.
- Add and removes servers without interrupting network traffic.
- Provides persistence support that allows a user to maintain connection integrity between a client and a Web service.
- Provides the ability to specify a backup server which is only used in the event when all the other servers (being load balanced) are out-of-service.

Load balancing can be configured at two levels:

- General (all application types)
- Content Rule (Web service only)

The general policy applies to all requests, while the content rule policy applies to the content rule requests only. However, the general and content rule configuration procedures are identical. There are three steps to configure load balancing on an Barracuda Web Application Firewall:

- Configure the load balancing method and other general parameters.
- Configure a persistence method to maintain the integrity of a connection.
- Configure a failover method to serve a request to a server which is down.

#### Note



The **Load Balance** feature is available for Barracuda Web Application Firewall **model 460 and above**.

#### To edit Load Balancer settings:

1. Click **Edit** under **Actions** against the created service.
2. Under **Load Balance** section, specify values for the following fields:
  - **Algorithm** - Select the algorithm to be used for load balancing.
  - **Persistence Method** - Select the persistence method of load balancing. Persistence maintains a connection between a client and the first server that it connects to. When the system is load balancing traffic, subsequent requests from that client are always sent to the same server. This is useful when the server requires to maintain state information about every client. For example, in an E-commerce application this policy maintains the connection from the time an online customer begins filling a shopping cart until that customer purchases the cart contents and completes the transaction.
  - **Failover Method** - Select the failover method for responding to a request which is persistent, but the server that must serve the request set to "out-of-service".
  - **Source IP Netmask** - Enter the netmask for Source IP persistence method. The IP plus netmask results in a network identifier which is used to identify a client. A more specific netmask (such as 255.255.255.255) will track each client independently, and may cause a higher memory load on the Web Application Firewall. Whereas as a less specific netmask (such as 255.255.0.0) will group multiple clients under the same network identifier and connect them all to the same server.



3. Click **Save Changes** to save the above settings.

**Note**



Apart from load balancing a set of servers, an additional server can be added to a service and set as **Backup Server**. When all the servers being load balanced fail, requests are sent to the **Backup server**. For more information, refer *Configuring a Backup Server* on page 49.

## Configuring SSL

---

SSL enables encryption between the client and the service on the Barracuda Web Application Firewall. SSL needs a digital certificate that authenticates the server of the service.

**Note**



**SSL** can be only configured for HTTPS, Instant SSL and Custom SSL services.

**To edit SSL settings:**

1. Click **Edit** under **Actions** against the created service.
2. Under **SSL** section, specify values for the following fields:
  - **Certificate** - Select from the drop-down list, the certificate presented by the service when authenticating itself to a browser or some other client.
  - **Enable Client Authentication** - If set to “Yes”, the users connecting to this site must present their certificate which will be validated using one of the Trusted Certificates.
  - **Enforce Client Certificate** - If set to “Yes”, causes the SSL handshake to be immediately terminated if the client did not return a certificate.
  - **Trusted Certificates** - Select one or more certificates that are trusted. Only those client certificates that are signed by one of these trusted certificates will be allowed access. Trusted Certificates can be uploaded on the **BASIC > Certificates** page.
3. Click **Save Changes** to save the above settings.

**Note**



If **Enforce Client Certificate** is set to **No** and **Enable Client Authentication** is set to **Yes**, then a client request without a certificate is serviced using the authorization policy, configured on the **ACCESS CONTROL > Authorization** page.  
If **Enable Client Authentication** is set to **No**, then the client request is denied.

## Configuring a Backup Server

---

An optional backup Web server can be added and used as a backup when all other load balanced servers fail.

**To add a Backup server:**

1. Click **Add** under **Actions** against the created service. The **Real Server** dialog box appears.
2. Specify values for the following fields:

- **IP** - Enter the new server IP address.
- **Port** - Enter the new server port number.
- **Backup server** - If “Yes”, sets this server as a backup server.

3. Click **Add** to add the above settings.

## Passive versus Active Mode

---

Active mode blocks any request when an anomaly or intrusion is observed. Passive mode observes the traffic, logs all observed anomalies and intrusions, and allows the traffic to pass through the Barracuda Web Application Firewall. Active mode might block some legal traffic resulting in false positives if the policy is not configured properly. Consider enabling a **Service** and or individual **URL Policy** in passive mode initially. Observe the Web firewall logs for potential intrusions, and make appropriate adjustments to security policies before enabling active mode.

If the **Passive Mode** is set to “Yes” for a service, all request with attacks on that service would be logged but allowed to pass the Barracuda Web Application Firewall. This setting overrides the passive mode setting in the **URL Policies** and **URL Profiles** for that service.

When the **Passive Mode** is set to “No”, an attack is allowed or denied based on the **URL Match** settings in **URL Policy** and **URL Profile**.

Also, if the service is in passive mode, the deny/process ACLs under **WEBSITES > Allow/Deny > URL ACLs** for that service will be logged but not enforced.

## Security Policy

---

The Barracuda Web Application Firewall provides a range of security policies for your Web sites and Web services. These policies determine what actions to take when one or more of the rules match the request. All policies are global and they can be shared among multiple services configured on the Barracuda Web Application Firewall. Some commonly used policies are defined by default. They are:

- Default
- Sharepoint
- OWA
- Oracle

Apart from these default policies, you can create customized policies. Each policy is a collection of 9 sub-policies. They are; Request Limits, Cookie Security, URL Protection, Action Policy, Global ACLs, URL Normalization, Cloaking, Parameter Protection and Data Theft Protection.

To create a new policy, go to **SECURITY POLICIES > Policy Manager** enter a name for the new policy in **Policy Name** field and click the Add button. This creates a new policy with default values. To modify a particular policy, go to the desired sub-policy page and select the policy from the **Policy Name** drop-down list and change the value of the parameter(s) and click the **Save Changes** button.

When a service is created, a basic set of Web Firewall features are activated automatically. By default all services use “default” as the policy. Based on your requirement you can change the policy for a service. To change the policy for a service, go to the **BASIC > Services** page, click **Edit** under **Actions**. The service page opens, select the desired policy from the **Web Firewall Policy** drop-down list under **Basic Security**. Click the **Save Changes** button to save and activate the new setting.

## Allowing/Denying Specific URLs

---

The **WEBSITES > Allow/Deny** allows you to define strict allow/deny rules for a Web Site. URL ACLs allow you to customize access to your site for a variety of specific conditions such as the following:

- Partition a Web site into security zones and configure different security policies on each zone.
- Configure explicit deny ACLs for known or observed attacks.

A security zone is a partition of a site that can be specified using a URL ACL key. The key is comprised of the URL, host and an optional extended match rule (that is, a value or expression for the header parameter). The matching URL ACL is determined by a best match algorithm using the host, URL and extended match fields in specified sequential order. In most cases, host and URL may only be used to specify an ACL. The Barracuda Web Application Firewall optimizes the search for the most common case by implementing a parallel search algorithm on all ACLs. The best matching ACL is the ACL with longest matching host and URL keys. To configure a more complex ACL based on certain fields in the request such as a client IP or an HTTP header, use extended match rules. Not all extended match rules are considered for evaluation. Only the ones specified for the matching host and URL are used for evaluation. If more than one such rule is specified, they are evaluated based on the specified extended match sequence.

There are two ways of redirecting a request using the URL ACL:

1. Set the **Action** parameter to **Redirect**, and specify the **Redirect URL**.
2. Set the **Action** parameter to **Deny**, set the **Deny Response** to **Redirect** and specify the **Redirect URL**.

The first case is not considered as an attack, therefore:

- It is logged at a lesser severity.
- Passive mode has no effect on it.

Whereas the second case is a suspected attack, therefore:

- It is logged at a higher severity.
- Passive mode is applied on it so that the request is not denied.

### To manually create/edit an URL ACL:

1. From **WEBSITES > Allow/Deny** page, click **Add** under **Options**. The **Create ACL** page opens.
2. Specify values for the following fields:
  - **URL ACL Name** - Enter the name for the URL ACL.
  - **Enable URL ACL** - Enables access controls policy for all services.
  - **Host Match** - Enter the matching criterion for host field in the Request Header. This is either a specific host match or a wildcard host match with a single “\*” anywhere in the URL. You can enter a partial domain with wildcard (for example: \*.abc.com) but you can not use multiple asterisks. (Examples: \*, \*.abc.com, www.abc.com)
  - **URL Match** - Enter the matching criterion for URL field in the Request Header. The URL should start with a “/” and can have only one “\*” anywhere in the URL. A value of /\* means that the ACL applies for all URLs in that domain. (Examples: /, /index.html, /public/index.html)
  - **Extended Match** - Enter an expression that consists of a combination of HTTP headers and/or query string parameters. Use “\*” to denote “any request”, that is, do not apply the Extended Match condition. To build an expression, click the *Edit* image button that appears next to this field, and specify the values for the following fields:
    - **Header Expression**: Specify a valid header expression.

- **Element Type:** Select the element type from the drop-down list.
  - **Operation:** Select the operation from the drop-down list.
  - **Value:** Specify a valid expression.
  - **Concatenate:** Select 'and' radio-button to add some more expressions to the existing match sequence. Select 'or' radio-button to replace the existing match sequence.
- Click **Insert** and then click **Apply** or click **Cancel**. For more on how to write extended match expressions, refer *Extended Match and Condition Expressions*.
  - Extended Match Sequence** - Enter an order for matching the extended match rule when a request matches multiple rules with the same URL Match.
  - Action** - Select the action to be taken on the request matching this URL from the drop-down list.
  - Deny Response** - Select the response to be sent to the client if the request is denied. A deny response is used when the “Action” is set to “Deny”.
  - Response Page** - Select the response page to be sent to the client, if the deny response parameter is set to “Custom Response”.
  - Redirect URL** - Enter the URL to be used to redirect the client if the “Deny Response” is set to “Redirect”.
- Click **Add** to add the above configurations.

## Allowing/Denying Specific Headers

---

The **WEBSITES > Allow/Deny** allows you to define strict allow/deny rules for a Web Site. A header ACL is created to define strict limitations on incoming headers intended for a Web site or Web service. It is used to sanitize HTTP headers that carry sensitive information identifying the client and some application-specific state information passed as one or more HTTP headers. A header ACL can be configured to protect against attack types and potentially malicious metacharacters and keywords that are placed in a header.

### To manually create/edit a Header ACL:

- From **WEBSITES > Allow/Deny** page, click **Add** under **Options**. The **Create Header ACL** page opens.
- Specify values for the following fields:
  - **Header ACL Name** - Enter the name for the Header ACL.
  - **Header Name** - Enter the header whose value should be validated. Enter either a single name or an asterisk (\*), which means apply this ACL to all headers.
  - **Status** - Select whether to enable or disable this policy for all services.
  - **Mode** - Select whether to enable this ACL in Active (blocking) or Passive (monitoring and logging only) mode.
  - **Max Header Value Length** - Enter the maximum allowable length in bytes for the header.
  - **Denied Metacharacters** - Enter the metacharacters to be denied.
  - **Blocked Attack Types** - Select the attack types to block from the default set of attack types.
  - **Custom Blocked Attack Types** - Specify your own attack patterns to enable as a Blocked Attack Type. These custom attack types can be defined on the **ADVANCED > Libraries** page as a set of regular expression patterns.
- Click **Add** to add the above configurations.

# Customized Security for Web sites

---

This chapter describes the configuration and monitoring tasks you can perform from the Web interface. The following topics are covered:

- *Security Policies* on page 54
- *Creating a New Security Policy* on page 64
- *Web Site Profiles* on page 65
- *Web Site Translations* on page 68
- *Trusted Hosts* on page 73

### Note



For more detailed information about a specific page in the Web interface, view the online help by clicking the question mark icon on the right side of the interface.

# Security Policies

---

A Web service is configured with several security policies. These policies define the processing of HTTP requests destined to the IP address and port configured on the Web service. Barracuda Web Application Firewall allows you to define strict checks to a Web Site and Web services. It is a shareable policy that can be used among multiple Web services configured on a Barracuda Web Application Firewall. Some commonly used policies, defined by default are:

- default
- sharepoint
- owa
- oracle

Apart from these default policies, you can create customized policies. Each policy is a collection of nine sub-policies. You can modify a policy, by visiting the desired sub-policy page and changing the value of the parameter(s). The following topics discuss these nine sub-policies in detail:

<i>Request Limits</i> .....	54
<i>Cookie Security</i> .....	55
<i>URL Protection</i> .....	57
<i>Parameter Protection</i> .....	59
<i>Cloaking</i> .....	59
<i>Data Theft Protection</i> .....	60
<i>URL Normalization</i> .....	61
<i>Global ACLs</i> .....	62
<i>Action Policy</i> .....	63

## Request Limits

---

The **SECURITY POLICIES > Request Limit** page displays the parameters to configure Request Limits.

Request limits define the validation criterion for incoming requests by enforcing size limits on HTTP request header fields. The request that have fields larger than the defined lengths are dropped. Proper configuration of limits helps mitigate buffer overflow exploits that lead to Denial of Service (DoS) attacks.

Request limits are enabled by default, and the default limit values are chosen with the assumption that any requests with lengths greater than the defaults are potential buffer overflow attacks. The defaults are normally appropriate, but you might choose to change one or more of the default values under certain conditions.

### To configure Request Limits

1. Select the policy from the **Policy Name** drop-down list.
2. Specify values for the following fields:
  - **Enable Request Limits** - Select whether to enable size limit checks on requests.
  - **Max Request Length** - Enter the maximum allowable request length. This includes the Request Line and all HTTP request headers.
  - **Max Request Line Length** - Enter the maximum allowable length for the request line. The request line consists of the method, the URL (including any query strings) and the HTTP version.
  - **Max URL Length** - Enter the maximum allowable URL length including the query string portion of the URL.

- **Max Query Length** - Enter the maximum allowable length for the query string portion of the URL.
- **Max Number of Cookies** - Enter the maximum number of cookies to be allowed.
- **Max Cookie Name Length** - Enter the maximum allowable length for cookie name.
- **Max Cookie Value Length** - Enter the maximum allowable length for a cookie value. Requests with Cookie values that are larger than the defined setting are denied.
- **Max Number of Headers** - Enter the maximum number of headers in a request. If there are more headers than this limit in the request, the request is denied.
- **Max Header Name Length** - Specifies the maximum allowable length for header name.
- **Max Header Value Length** - Enter the maximum allowable length for any request header. A request header could either be a HTTP protocol header such as "Host," "User-Agent" and so on, or a custom header such as "IIS Translate" header. A request may contain any number of these headers.

3. Click **Save Changes** to save the above settings.

## Cookie Security

---

The **SECURITY POLICIES > Cookie Security** page displays the parameters to configure Cookie Security.

Cookies provide a mechanism to store service state information on a client's navigation platforms, such as browsers and other user agents. Cookies are used to store user preferences, shopping cart items, and sometimes very sensitive information such as registration and login information. If the structure of the cookie can be revealed, the user's information is vulnerable to attack.

A Web server can send a cookie, which is a packet of whatever information the server chooses to send (such as information to authenticate or identify a user), to maintain state between otherwise stateless HTTP transactions. Because cookies are simple ASCII name value pairs, they can easily be altered and then used to launch a Web attack. Cookies can also be stolen and sensitive information, such as client information, can be obtained from the message.

You have the option to apply security features to the cookies sent from the Web servers to the Web users using the cookie security. This enables the security features for HTTP cookies. The Barracuda Web Application Firewall's cookie security policy provides two cryptographic techniques, *Encrypted* that encrypts the cookie data and *Signed* attaches a digital signature to the server generated cookie.

### Functioning of Cookie Security

The cookie security features of the Barracuda Web Application Firewall are transparent to the back-end server. When the server inserts a cookie, the Barracuda Web Application Firewall intercepts the response (encrypts or signs) the cookie before delivering it to the client browser. When a subsequent request from the client returns this cookie, the Barracuda Web Application Firewall intercepts the request, decrypts or verifies its signature. If the cookie is intact, the Barracuda Web Application Firewall forwards the decrypted or unsigned cookie to the server.

If the cookie is tampered, decryption or signature verification fails, and the Barracuda Web Application Firewall removes the cookie and forwards the request to the server without the cookie.

When the cookie security features (*Encrypted* or *Signed*) are not enabled, the cookies pass through the Barracuda Web Application Firewall without cryptographic checking.



## Encrypting Cookies

The Barracuda Web Application Firewall performs cookie encryption. In this mode, when the server returns a response message that contains a cookie, the Barracuda Web Application Firewall intercepts the response, encrypts the cookie before delivering it to the client browser. When a subsequent request from the client returns this cookie, the Barracuda Web Application Firewall intercepts the request, decrypts the cookie and forwards the request to the server (origin server) that generated the cookie.

If the cookie is tampered, decryption fails, and the Barracuda Web Application Firewall removes the cookie and forwards the request to the server without the cookie.

Encryption method prevents attackers from being able to view the content of a cookie and secures cookie from tampering. This is useful when client side scripts need not have to access the cookie value.

## Signing Cookies

The Barracuda Web Application Firewall attaches a digital signature to the original server generated cookie. In this mode, when the server returns a response message that contains a cookie, the Barracuda Web Application Firewall intercepts the response, attaches a digital signature to the original server generated cookie, and forwards two cookies to the client browser, one plain text cookie and the other signed cookie. When a subsequent request from the client returns the cookies, the Barracuda Web Application Firewall intercepts the request, verifies the signature and forwards the request to the server that generated the cookie. The signed cookie is always stripped out by the Barracuda Web Application Firewall before forwarding a request to the origin server.

If either of the cookies is tampered, signature verification fails, and the Barracuda Web Application Firewall removes the cookies and forwards the request to the server without the cookie.

This is useful when client side scripts need to look at the cookie value to make some decisions.

## Cookie Security interaction with other Security features

When a cookie is encrypted it may change the length of the cookie, but the number of headers in the message remains unchanged. When a cookie is signed, it changes the length of the cookie and appends one or more header to the forwarded message. If the **SECURITY POLICIES > Request Limits** configuration specifies constraints on the number or length of HTTP headers, a signed or encrypted cookie may result in unwanted rejection of messages. The rejected messages are logged as ‘*Cloak*’ under **Action** on **BASIC > Web Firewall Logs** page.

## Configuring Cookie Security

To encrypt or sign cookies and reject tampered cookies, follow the steps below:

**Step 1:** From **SECURITY POLICIES** tab, navigate to **Cookie Security** page.

**Step 2:** Select a policy from the **Policy Name** drop-down list. Note that you are enabling cookie security for this policy.

**Step 3:** Under **Cookie Security** section, select **Tamper Proof Mode** as **Encrypted** or **Signed**.

**Note:** *Encrypted* makes cookie data illegible before forwarding it to the client. *Signed* attaches a digital signature to the server generated cookie before forwarding it to the client. If required change the value of other parameter(s):

- **Cookie Max Age** - Enter the maximum age for session cookies.
- **Cookie Replay Protection Type** - Select the type of protection to be used to prevent the cookie replay attacks from the drop-down list.



- **Custom Headers** - Enter the custom headers to be used in the cookie if the parameter "Cookie Replay Protection Type" is set to "Custom Headers" or "IP and Custom Headers" and click **Add**.
- **Secure Cookie** - Select whether the cookies need to be returned for HTTPS requests only. This parameter directs the user agents to send this cookie back only when they make secure HTTPS connection to the origin server.
- **HTTP Only** - Select whether the cookie security feature should be enabled for HTTP cookies.
- **Allow Unrecognized Cookies** - Select an action that has to be taken on an unrecognized cookie.
- **Cookies Exempted** - Enter the cookies to be exempted from the policy and click **Add**.

**Step 4:** Click **Save Changes** to save the above settings.

**Step 5:** Now, you have to bind this policy with a service for which you want to enable cookie security.

**Step 6:** Go to **BASIC** tab, and navigate to **Services** page. Click **Edit** against the service for which you want to bind this policy.

**Step 7:** Under **Basic Security** section, select the policy from **Web Firewall Policy** drop-down list. Set the **Mode** as Active and click **Save Changes**.

## URL Protection

---

The **SECURITY POLICIES > URL Protection** page displays the parameters to configure URL Protection. URL Protection protects the service against Web attacks in the absence of a URL Profile. The parameter section provides protection for parameters embedded in the URL itself.

For example:

```
http://www.example.com/sharepoint/default.aspx/%22);}if(true){alert(%22qwertytis
```

In the above example, malicious script has been added in the URL itself.

When submitting URL requests, malicious users can carry out various attacks via the HTTP protocol and form parameters. These attacks can take the following forms:

### Attacks via HTTP Headers and Contents

While GET and POST are by far the most common methods that are used to access information provided by a web server, the Hypertext Transfer Protocol (HTTP) allows several other (and somewhat less known) methods. RFC 2616 describes the following HTTP methods in detail:

- HEAD
- GET
- POST
- PUT
- DELETE
- TRACE
- OPTIONS
- CONNECT

It is easy for a malicious user to use the OPTIONS command to find out which of these methods are supported by the Web server. Some of these methods can potentially pose a security risk for a web application, as they allow an attacker to modify the files stored on the web server and, in some scenarios, steal the credentials of legitimate users. Many web application frameworks sometimes allow well chosen and/or arbitrary HTTP methods to bypass an environment level access control check. PUT, DELETE, and TRACE should normally not be allowed. The allowed content-types in a request should also be carefully restricted in order to prevent similar security threats.

### Attacks by Injecting various Commands in Parameters

When a web application does not properly sanitize user-supplied input before using it within application code, it may be possible to trick the application into executing certain commands such as Operating System, database commands, etc. Users may also inject cross site scripting or remote file inclusion attacks inside parameters name/value. The executed commands will run with the same permissions of the component that executed the command (e.g. Database server, Web application server, Web server, etc.).

### Attacks by Injecting Buffer Overflow in Parameters

Buffer Overflow exploits are attacks that alter the flow of an application by overwriting parts of memory. A Buffer Overflow can be used as a Denial of Service attack when memory is corrupted, resulting in software failure. Even more critical is the ability of a Buffer Overflow attack to alter application flow and force unintended actions.

Restricting the size and content of parameter ensures that web applications are protected against the above attacks.

#### To configure URL Protection

1. Select the policy from the **Policy Name** drop-down list.
2. Specify values for the following fields:
  - **Enable URL Protection** - Select whether to enable or disable URL protection
  - **Allowed Methods** - Enter the list of allowable methods in the request and click **Add**.
  - **Allowed Content Types** - Enter the list of allowable content types in the POST body for a URL and click **Add**.
  - **Max Content Length** - Enter the maximum allowable length of the content, that is, the request body.
  - **Max Parameters** - Enter the maximum number of parameters allowed in a request.
  - **Max Upload Files** - Enter the maximum number of files that can be of file-upload type in one request.
  - **Max Parameter Name Length** - Enter the maximum length of any parameter name in the request.
  - **Blocked Attack Types** - Select the default set of attack types to block.
  - **Custom Blocked Attack Types** - Select the custom attack types to block. These custom attack types are defined under **ADVANCED > Libraries** as a set of regular expression.
3. Click **Save Changes** to save the above settings.

## Parameter Protection

---

The **SECURITY POLICIES > Parameter Protection** page setting protects the service against attacks based on parameter values in the absence of a parameter profile. These security settings protect against attacks on the URL query string parameters as well as on the POST parameters in forms.

Special characters such as " ' ", " ; " or ' ' are used to embed SQL expressions in parameter values. SQL keywords such as "OR," "SELECT," "UNION" can be embedded in parameter values to exploit vulnerabilities. Special characters such as '<' or keywords such as "<script>," "<img" are used to embed html tags in parameter values in the case of Cross Site scripting attacks. Keywords such as "xp\_cmdshell" are used in System Command Injection attacks. If a parameter matches any of the attack patterns configured in the parameter protection, then the Barracuda Web Application Firewall does not process the request / response.

### To configure Parameter Protection

1. Select the policy from the **Policy Name** drop-down list.
2. Specify values for the following fields:
  - **Enable Parameter Protection** - Select whether to enable or disable parameter protection.
  - **Denied Metacharacters** - Enter the metacharacters to be denied in this parameter value.
  - **Max Param Value Length** - Enter the maximum allowable length of the value of a parameter.
  - **File Upload Extensions** - Enter the extensions that are allowed while uploading the files. '.' is a special extension which indicates no extension, and \* indicates any extension and click **Add**.
  - **Max Upload File Size** - Enter the maximum size for individual files that can be of file-upload type in one request.
  - **Blocked Attack Types** - Select the check box(es) for malicious patterns in the parameter value. An intrusion is detected when the value of a parameter matches one of the selected Attack Types.
  - **Custom Blocked Attack Types** - Select the check box(es) to define your own attack patterns to extend the parameter protection. These custom attack types can be defined in the **Advanced > Libraries** page as a set of regular expression patterns.
  - **Ignore Parameters** - Enter the list of parameters to exempt from all validations and click **Add**.
3. Click **Save Changes** to save the above settings.

## Cloaking

---

The **SECURITY POLICIES > Cloaking** page is used to enable security policies to cloak a Web site or Web service.

It does this by removing HTTP headers and return codes before sending a response to a client. This prevents hackers from gleaning information about services that could be used to launch attacks. (In the industry, the term Web site cloaking is sometimes used to refer to the technique of delivering one version of a Web page to a user while delivering a different version of the same page to a search engine, but that is not the meaning in this case.)

This policy offers the following cloaking features:

- Remove specified headers such as Server from responses.

- Suppress client error (status code 4xx) and server error (status code 5xx) messages from responses.

All Website Cloaking parameters are enabled by default. There is rarely a reason to change any of the default values, except to add headers to be filtered.

### To configure Cloaking

1. Select the policy from the **Policy Name** drop-down list.
2. Specify values for the following fields:
  - **Suppress Return Code** - Select whether to enable or disable the blocking of the return of an HTTP status code in a response header.
  - **Filter Response Header** - Select whether to enable or disable removing the HTTP headers in responses. Configure the list of HTTP headers to remove using the "Headers to Filter" parameter.
  - **Headers to Filter** - Enter a list of headers that are removed from a response before serving it to a client and click **Add**.
3. Click **Save Changes** to save the above settings.

## Data Theft Protection

---

Data theft refers to the unauthorized copying of confidential personal or business information such as social security numbers, credit card information, and other privileged personal or corporate information. Unintended exposure of such data can lead to identity theft.

The **SECURITY POLICIES > Data Theft Protection** page is used to identify such data in responses sent by the server and protect it against exposure. Enabling data theft protection on an ACL increases the processing overhead per page. To optimize performance, enable it only for parts of the site that are known to carry such sensitive information.

### Note



You can implement the data theft element configured on this page only if you set the parameter **Enable Data Theft Protection** to **Yes** on the **WEBSITES > Advanced Security** page for a Service bound to the Security Policy. Once the data theft protection is enabled, all URL policies for that service will prevent theft of the data type elements in server response pages as configured.

### To configure Data Theft Protection

1. Select the policy from the **Policy Name** drop-down list.
2. Specify values for the following fields:
  - **Data Theft Element Name** - Enter the name for this data theft element.
  - **Enabled** - Select whether the data theft element is used. When this set to "Yes", all URL policies which have **Data Theft Protection Status** set to "On" will look for this data type in server response pages.
  - **Identity Theft Type** - Select the data type that the element refers to from the drop-down list. It is a binding to data type defined under **ADVANCED > View Internal Patterns > Identity Theft Patterns**.
  - **Custom Identity Theft Type** - Select the customized identity theft type to be used, if the parameter "Identity Theft Type" is set to "<CUSTOM>" from the drop-down list. Refer *Creating Identity Theft Patterns* on page 142 for more information.
  - **Action** - Select whether to block any page sent by server containing this data type or to cloak (part of the data overwritten with "X"s) from the drop-down list.

- **Initial Characters to keep** - Enter the number of initial characters are to be displayed to the user when the data of this data type is identified in a server page.
- **Trailing Characters to keep** - Enter the number of trailing characters are to be displayed to the user when the data of this data type is identified in a server page.

3. Click **Save Changes** to save the above settings.

## Protected Data Types

This table lists all the default and created data theft elements. You can edit these data theft elements.

## URL Normalization

The **SECURITY POLICIES > URL Normalization** page displays the parameters to configure URL Normalization.

The Barracuda Web Application Firewall normalizes all traffic into a standard or "canonical" form before applying any security policy string matches. In the HTTP world, this means decoding Unicode, UTF, or Hex to base text. Otherwise, hackers can disguise attacks within different encoding formats that the firewall might not detect using a string match.

Normalization (converting a URL into a canonical form) is always enabled if the Barracuda Web Application Firewall is active. The **Default Character set** parameter specifies the character set encoding type for incoming requests. It is set to ASCII by default; to specify an alternate type, simply select a different type such as Shift-JIS for Japanese characters.

Additional checks to prevent path traversal and path disclosure attacks are set in URL Normalization.

There are situations where multiple character set encoding is needed. For example, a Japanese language site might need both Shift-JIS and EUC-JP encoding. You have the option of setting the Barracuda Web Application Firewall to automatically add character set encoding as needed. To configure this, set the **Detect Response Charset** parameter to **Yes**. (It does this by searching all response headers for a META tag that specifies the character set encoding type and dynamically adding any supported types listed in the META tags.)

Double encoding is the re-encoding of the encoded data. For example: The UTF-8 escape for the backslash character is %5C, which is a combination of three characters i.e. %, 5, and C. So the Double encoding is the re-encoding either one or all the 3 characters by using their corresponding UTF-8 escapes as %25, %35, and %63.

The following table describes double-encoding variations of the \ character.

*Table 5.1: Double encoding variations of the \ character.*

Escape	Description
%5C	%5C Normal UTF-8 escape of the backslash character.
%255C	%25, the escape for % followed by 5C.
%%35%63	The % character followed by %35, the escape for 5, and %63, the escape for C.
%25%35%63	The individual escapes for %, 5, and C.

### To configure URL Normalization

1. Select the policy from the **Policy Name** drop-down list.
2. Specify values for the following fields:
  - **Default Character Set** - Select the character set encoding type to be used for incoming requests from the drop-down list.
  - **Detect Response Charset** - Defines whether or not the Barracuda Web Application Firewall will detect the character set encoding from the response. When it cannot determine the character set encoding, it will default to one specified for the 'charset' parameter.
  - **Parameter Separators** - Select the URL-encoded parameter separator to be used from the drop-down list.
  - **Double Encoding** - Select whether to re-encode the encoded data.
3. Click **Save Changes** to save the above settings.

## Global ACLs

---

The **SECURITY POLICIES > Global ACLs** page allows you to define strict allow/deny rules for all the services configured on the Barracuda Web Application Firewall. It is a shareable policy that can be used among multiple services. You can add a new URL ACL or modify the existing URL ACL.

### To configure Global ACLs

1. Select the policy from the **Policy Name** drop-down list.
2. Specify values for the following fields:
  - **URL ACL Name** - Enter the name for the URL ACL.
  - **Enable URL ACL** - Enables access controls policy for all services.
  - **URL Match** - Enter the matching criterion for URL field in the Request Header. The URL should start with a "/" and can have only one "\*" anywhere in the URL. A value of /\* means that the ACL applies for all URLs in that domain.
  - **Extended Match** - Enter an expression that consists of a combination of HTTP headers and/or query string parameters. Use '\*' to denote "any request", that is, do not apply the Extended Match condition. To build an expression, click the *Edit* image button that appears next to this field, and specify the values for the following fields:
    - **Header Expression**: Specify a valid header expression.
    - **Element Type**: Select the element type from the drop-down list.
    - **Operation**: Select the operation from the drop-down list.
    - **Value**: Specify a valid expression.
    - **Concatenate**: Select 'and' radio-button to add some more expressions to the existing match sequence. Select 'or' radio-button to replace the existing match sequence.
    - Click **Insert** and then click **Apply** or click **Cancel**. For more on how to write extended match expressions, refer *Extended Match and Condition Expressions*.
  - **Extended Match Sequence** - Enter an order for matching the extended match rule when a request matches multiple rules with the same URL Match.
  - **Action** - Select the action to be taken on the request matching this URL from the drop-down list.
  - **Redirect URL** - Enter the URL to be used to redirect the client if the parameter "Deny Response" is set to "Redirect".
3. Click **Add** to add the above configurations.

## Existing Global ACLs

This table lists all the default and created global URL ACLs. You can edit these URL ACLs.

## Action Policy

---

The **SECURITY POLICIES > Action Policy** page specifies the action to be taken for a particular type of Web attack. Action policy is a collection of settings that decide what action to be taken when a violation occurs. It consists of a set of attack groups, which in turn contains a set of attack actions. The following attack groups are available:

The following attack groups are available:

- advanced-policy-violations
- application-profile-violations
- param-profile-violations
- protocol-violations
- request-policy-violations
- response-violations
- url-profile-violations

### To edit an Action Policy

1. Select the policy from the **Policy Name** drop-down list.
2. Click **Edit** against the default attack action.
3. Specify values for the following fields:
  - **Action** - Select the action to be taken for an invalid request from the drop-down list.
  - **Deny Response** - Select the response from the drop-down list to be sent to the client if the request is denied. A deny response is used when the parameter "Action" is set to "Protect".
  - **Redirect URL** - Enter the URL to be used to redirect the request if the deny response is set to "Redirect".
  - **Response Page** - Select the response page to be sent to the client, if the parameter "Deny Response" is set to "Send Response" from the drop-down list.
  - **Follow Up Action** - Select the follow up action to be taken from the drop-down list if the request is denied.
  - **Follow Up Action Time** - Enter the time in seconds to block the client IP, if the parameter "Follow Up Action" is set to "Block Client IP".
4. Click **Save Changes** to save the above settings.

# Creating a New Security Policy

---

Apart from the default policies, you can create customized policies. The new policy is created with default values. You can change the default values by visiting the individual sub-policy pages and changing the values.

## To create a customized Security Policy:

1. From the **SECURITY POLICIES > Policy Manager** page, enter a name of the new policy under **Create New Policy** and click **Add**.
2. The new policy with the default values is created and added to the list of policies under **Policy Overview**.
3. To modify a policy, go to the desired sub-policy page and select the policy from the **Policy Name** drop-down list.
4. Change the value of the parameter(s) and click **Save Changes** to save and activate the new settings.
5. Click **Delete** to remove a policy. All the policies can be removed except the default policy.

By default all services use the default policy. Based on your requirement you can change the policy for a service.

## To change the policy for a service

1. From the **BASIC > Services** page, click **Edit** under **Actions**. The service page opens.
2. Select the desired policy from the **Web Firewall Policy** drop-down list under **Basic Security**.
3. Change the value of the parameter(s) and click **Save Changes** to save and activate the new settings.



# Web Site Profiles

---

The **WEBSITES > Web Site Profiles** page uses the URL profiles and Parameter profiles created for a service to validate the requests coming for that service. When a new service is added, a Web Site profile is created by default and is enabled for that service. You can modify the default settings, which overrides the default policies. If the parameter "Use Profile" is set to "Yes", then URL Profiles and Parameter Profiles must be created for validating the requests coming for that service. This falls in accordance with the positive security nodes, which denies any request for which there is no URL or Param Profile.

## To edit a Web Site Profile

1. Click **Edit** against the created Web Site Profile. The **Edit Web Site Profile** dialog box appears.
2. Specify values for the following fields:
  - **Use Profile** - Select whether to use URL profiles and parameter profiles for validating the requests coming for this service.
  - **Strict Profile Check** - Specifies whether to enforce strict profile checks thereby denying requests which do not match any profile. If set to "No", then the service's default web firewall policy will be applied to those requests which do not have a profile.
  - **Allowed Domains** - Enter the domain attribute of the session cookie and click **Add**.
  - **Exclude URL Patterns** - Enter the list of URL patterns to exclude the URL profile validations and click **Add**. (Examples: \*.html, \*.htm, \*.jpg, \*.gif, \*.css, \*.js).
  - **Include URL Patterns** - Enter the list of URL patterns to be included in the URL profile validations in spite of being listed in "Exclude URL Patterns" parameter and click **Add**.
  - **Mode** - Select the mode of the Web Site profile for this service.
  - **Session Cookie Domain** - Enter the cookie domain to be used to allow the browser to send the cookie back to the Barracuda Web Application Firewall.
  - **Session Cookie Timeout** - Enter the time-out for session cookie after the successful login. 0 indicates no time-out, the session lives forever.
3. Click **Save Changes** to save the above settings.

## URL profile

URL Profiles are used to validate the requests coming to the service as per the settings in parameter "State" in the Web Site Profile. You can add more than one URL profiles for a service.

## To add/edit a URL Profile

1. Click **Add URL** under the **URL Profile** section. The **Create URL Profile** dialog box appears.
2. Specify values for the following fields:
  - **URL Profile Name** - Enter the name for this URL profile.
  - **Status** - Select whether to enable this URL profile.
  - **URL** - Enter the matching criterion for URL field in the Request Header. The URL should start with a "/" and can have only one " \*" anywhere in the URL. A value of /\* means that the ACL applies for all URLs in that domain.
  - **Extended Match** - Enter an expression that consists of a combination of HTTP headers and/or query string parameters. Use '\*' to denote "any request", that is, do not apply the Extended Match condition. To build an expression, click the *Edit* image button that appears next to this field, and specify the values for the following fields:
    - **Header Expression**: Specify a valid header expression.

- **Element Type:** Select the element type from the drop-down list.
  - **Operation:** Select the operation from the drop-down list.
  - **Value:** Specify a valid expression.
  - **Concatenate:** Select '**and**' radio-button to add some more expressions to the existing match sequence. Select '**or**' radio-button to replace the existing match sequence.
  - Click **Insert** and then click **Apply** or click **Cancel**. For more on how to write extended match expressions, refer *Extended Match and Condition Expressions*.
  - **Extended Match Sequence** - Enter an order for matching the extended match rule when a request matches multiple rules with the same Host Match and URL Match.
  - **Mode** - Select the mode for this URL profile from the drop-down list. It can be either “Active” or “Passive”.
  - **Allow Methods** - Add the list of allowable methods in the request.
  - **Allow Query String** - Select whether to allow query string in URL or not.
  - **Allow Content Types** - Add the list of allowable content-types in the POST body for a URL.
  - **Hidden Parameter Protection** - Select the protection for hidden parameters in the forms and URLs from the drop-down list.
  - **CSRF Prevention** - Select the cross-site request forging prevention for the forms and URLs from the drop-down list. This CSRF protection is not applicable when there is no parameter profile.
  - **Max Content Length** - Enter the maximum allowable request header length excluding the request body.
  - **Max Parameter Name Length** - Enter the maximum length of the parameter name.
  - **Max Upload Files** - Enter the maximum number of files that can be of file-upload type in one request.
  - **Blocked Attack Types** - Select the attack types to block from the default set of attack types.
  - **Custom Blocked Attack Types** - Specify your own attack patterns to enable as a Blocked Attack Type. These custom attack types can be defined on the **ADVANCED > Libraries** page as a set of regular expression patterns.
3. Click **Add** to add the above configurations.

## Parameter profile

Parameter Profiles is used to validate the requests coming for this service as per the settings for the parameter "State" under URL profile. You can add more than one parameter profiles for a service.

### To add/edit a Parameter Profile

1. Click **Add Param** under the **Parameters for:** section for the selected URL profile. The **Create Parameter Profile** dialog box appears.
2. Specify values for the following fields:
  - **Parameter Profile Name** - Enter the name for this parameter profile.
  - **Status** - Select whether to enable or disable this parameter profile.
  - **Parameter** - Enter the name of the parameter expected in the requests. No name parameter (&name-param) is also supported. The parameter name with the special characters like &pathinfo and &sessionid and wildcard (\*) should be manually specified.
  - **Type** - Select the type of parameter to be validated in the requests from the drop-down list.
  - **Values** - Enter a fixed set of strings to match against the parameter's value, if the parameter "Type" is set to "Global Choice".

- **Parameter Class** - Select the parameter class to be used from the drop-down list.
- **Custom Parameter Class** - Select the customized parameter class to be used, if the parameter "Parameter Class" is set to "<Custom>" from the drop-down list. Refer *Creating Custom Parameter Class* on page 145 for more information.
- **Max Value Length** - Enter the maximum allowable length for the value of the parameter.
- **Required** - Select whether this parameter is required to be present always in the request, or can it be skipped.
- **Ignore** - Select whether to ignore this parameter completely, that is, not to validate the value of this parameter at all.
- **Max Instances** - Enter the maximum number of times this parameter is allowed.
- **File Upload Extensions** - Enter the list of extensions that are allowed in file uploads. '.' is a special extension which indicates no extension, and \* indicates any extension is allowed.
- **Allowed Metacharacters** - Enter the list of metacharacters to be allowed in spite of it being marked as denied in the parameter class.

3. Click **Add** to add the above configurations.

For more information, refer *Web Site Profiles* on page 87.

# Web Site Translations

---

The **WEBSITE > Web Site Translations** page is used to set a variety of address translation rules for application-specific packets sent through the Barracuda Web Application Firewall. It translates the internal codes, headers, and cookies so that the actual message is concealed to the external users. It has features for Web site cloaking and translation of URLs and headers in the requests and responses.

## Note



Web Site Translation feature is only available with 460 and above models.

## Configuring URL Translation

URL translation is a method for mapping the prefix, domain and response body of an internal URL to a different URL that is displayed to external users. Translation in this case refers to modifying the URL, not to translating it to a different language. When a URL is returned from a Web server, sensitive information about the Web server could be displayed in the URL. The rewritten external URLs are used in ACLs and Content Rules. This information can be used to launch a variety of Web attacks against the server. URL translation hides the internal names and prevents such attacks.

A primary reason for URL translation is to externalize internal applications with links to internal servers that are not defined in the external DNS name space. For example, Company ABC has an internal application registered in the internal DNS as *finance.abc*. They would like to make this application available to external partners behind a common public domain (*www.companyabc.com*) without exposing the internal name space. Through URL translation, Company ABC can map different internal and external prefixes such that the internal application is available on the public Internet as *www.companyabc.com/finance.abc*.

### To configure URL Translation

1. Select the Web service from the **Web Site** drop-down list.
2. Specify values for the following fields:
  - **Rule Name** - Enter a name for the URL translation rule.
  - **Outside Prefix** - Enter the external URL for the application.
  - **Outside Domain** - Enter the external domain name, IP address and port number for the application.
  - **Inside Prefix** - Enter the internal URL match rule for the application.
  - **Inside Domain** - Enter the internal domain name, IP address or port number for the application.
3. Click **Add** to add the above configurations.

## Configuring Request Rewrite

This policy sets rewrite rules for inbound requests. It specifies the parameters to modify incoming request headers. It allows you to add, delete, or rewrite headers and rewrite or redirect the URL. Request Rewrites are used for specific purposes. For example, the Barracuda Web Application Firewall fully terminates the TCP/IP session for the original request and creates a new request using the private interface (PIF) as the new source IP address. If you have a need for the original source IP address, you could add a header that stores that address.

It is used to relay the actual client information back to the back-end resource. Enabling this parameter allows a back-end resource to know exactly who is the client and from where the client is requesting the information.

This feature is available only with the purchase of Barracuda Web Application Firewall model 460 and higher.

### To configure Request Rewrite

1. Select the Web service from the **Web Site** drop-down list.
2. Specify values for the following fields:
  - **Rule Name** - Enter the name for the request rewrite rule.
  - **Sequence Number** - Enter the sequence number of this request rewrite policy. The sequence number sets the order of execution for multiple configured policies from highest (1) to lowest (64).
  - **Action** - Select the request rewrite action for this policy from the drop-down list.
  - **Header Name** - Enter the header name to match for this policy. This is required if the action is to a header (rather than an URL).
  - **Old value** - Enter the old value of a header or URL path to be rewritten.
  - **Rewrite Value** - Enter the new value of a header or URL path to rewrite. This is required if the action is to rewrite (or redirect) a header or URL.
  - **Rewrite Condition** - Enter the condition under which the rewrite should occur. An asterisk (\*) indicates there are no conditions (applies to all). Refer *Request Rewrite Condition* on page 69 for more information.
  - **Continue Processing** - Select whether to check against other (higher sequence number) policies or stop here. This is relevant only when additional policies have been configured.
3. Click **Add** to add the above configurations.

### Request Rewrite Condition

Request rewrite condition is an expression that consists of a combination of HTTP headers and/or query string parameters. For HTTP headers, word "Header" should be prefixed before the header expression and for Non-HTTP headers "Header" should not be prefixed. Define the header type (for example, user agent or accept) for which you want an action to occur or add a wildcard to accept any type of header.

The following are the possible operations that can be given in the expression:

- contains, CONTAINS, co, CO - checks if the operand contains the given value.
- ncontains, nCONTAINS, nco, nCO - checks if the operand does not contain the given value.
- rcontains, rCONTAINS, rco, rCO - checks if the operand contain the given value. The given value is interpreted as a regular expression.
- equals, EQUALS, eq, EQ - checks if the operand is equal to the given value.
- nequals, nEQUALS, neq, nEQ - checks if the operand is not equal to the given value.
- requals, rEQUALS, req, rEQ - checks if the operand is equal to the given value. The given value is interpreted as a regular expression.
- exists, EXISTS, ex, EX - checks if the operand exists. It does not require any given value.
- nexists, nEXISTS, nex, nEX - checks if the operand does not exist. It does not require any given value.

Each expression can be joined with another expression by using either of the following tokens:

- or, OR, || - This checks for either of the expressions are true.

- and, AND, && - This checks if both the expressions are true.

More than one expressions can be grouped together by using parenthesis '(' and ')'.  
The expression consists of an operation being carried out on one of the following tokens. Each of the following tokens are case insensitive.

### **Header**

This refers to an HTTP header on the request path. The term "Header" should be followed by the name of the header on which the action is to be applied.

Example: Header Accept co soap or Header Soap-Action ex

### **Client IP**

This refers to the IP address of the client sending the request. The IP address can be either host IP address or subnet IP address specified by a mask. Only the following operations are possible for this token:

"EQUAL" and "NOT EQUAL"

Using any other operations are not permitted.

Example: Client-IP eq 192.168.1.0/24 (subnet IP address containing the mask)

Client-IP eq 192.168.1.10 (host IP address)

### **Uri**

The URI is a Uniform Resource Identifier and identifies the resource upon which to apply the request.

Example: URI rco /abc\*.html

### **Method**

This refers to HTTP method in the request.

Example: Method eq GET

### **Http Version**

This refers to the version of the HTTP protocol of the request.

Example: HTTP-Version eq HTTP/1.1

### **Parameter**

This refers to query part of the URL which is passed to the servers as a name-value pair. In addition, the word "\$NONAME\_PARAM" is used to refer to the case where the parameter name is absent.

Example: Parameter sid eq 1234, Parameter \$NONAME\_PARAM co abcd

### **Pathinfo**

This refers to the portion of URL which contains extra information about the path of the resource on the server.

Example: pathinfo rco abc\*

## **Configuring Response Rewrite**

This policy sets rewrite rules for outbound responses. It allows you to add, delete, or rewrite headers. Response Rewrites are used for specific purposes. For example, if responses include a header that lists the source IP address you could delete that header. This would prevent users from seeing the actual IP address of a server.

## To configure Response Rewrite

1. Select the Web service from the **Web Site** drop-down list.
2. Specify values for the following fields:
  - **Rule Name** - Enter the name for the response rewrite rule.
  - **Sequence Number** - Enter the sequence number of this request rewrite policy. The sequence number sets the order of execution for multiple configured policies from highest (1) to lowest (64).
  - **Action** - Select the request rewrite action for this policy from the drop-down list.
  - **Header Name** - Enter the header name to match for this policy. This is required if the action is to a header (rather than an URL).
  - **Old value** - Enter the old value of a header or URL path to be rewritten. This is required if the action is to rewrite (or redirect) a header or URL.
  - **Rewrite Value** - Enter the new value of a header or URL path to rewrite. This is required if the action is to rewrite (or redirect) a header or URL.
  - **Rewrite Condition** - Enter the condition under which the rewrite should occur. An asterisk (\*) indicates there are no conditions (applies to all). Refer *Response Rewrite Condition* on page 71 for more information.
  - **Continue Processing** - Select whether to check against other (higher sequence number) policies or stop here. This is relevant only when additional policies have been configured.
3. Click **Add** to add the above configurations.

## Response Rewrite Condition

Response rewrite condition is an expression that consists of a combination of HTTP headers and/or query string parameters. For HTTP headers, word "Header" should be prefixed before the header expression and for Non-HTTP headers "Header" should not be prefixed. Define the header type (for example, user agent or accept) for which you want an action to occur or add a wildcard to accept any type of header.

The following are the possible operations that can be given in the expression.

- contains, CONTAINS, co, CO - checks if the operand contains the given value.
- ncontains, nCONTAINS, nco, nCO - checks if the operand does not contain the given value.
- rcontains, rCONTAINS, rco, rCO - checks if the operand contain the given value. The given value is interpreted as a regular expression.
- equals, EQUALS, eq, EQ - checks if the operand is equal to the given value.
- nequals, nEQUALS, neq, nEQ - checks if the operand is not equal to the given value.
- requals, rEQUALS, req, rEQ - checks if the operand is equal to the given value. The given value is interpreted as a regular expression.
- exists, EXISTS, ex, EX - checks if the operand exists. It does not require any given value.
- nexists, nEXISTS, nex, nEX - checks if the operand does not exist. It does not require any given value.

Each expression can be joined with another expression by using either of the following tokens:

- or, OR, || - This checks for either of the expressions are true.
- and, AND, && - This checks if both the expressions are true.

More than one expressions can be grouped together by using parenthesis '(' and ')'.

The expression consists of an operation being carried out on one of the following tokens. Each of the following tokens are case insensitive.

## Header

This refers to an HTTP header on the request path. The term "Header" should be followed by the name of the header on which the action is to be applied.

Example: Header Accept co soap or Header Soap-Action ex

## Response Header

This refers to an HTTP header on the response path. The term "Response-Header" should be followed by the name of the header on which the action is to be applied.

Example: Response-Header Set-Cookie co sessionid

## Status Code

This refers to the status code of the response returned by the servers.

Example: Status-Code eq 200

## Configuring Response Body Rewrite

This policy sets the rule for searching and replacing any text string in the response body. The responses whose content-type begins with text/ can only be searched. That is only text/html, text/plain, text/javascript, text/css, text/xml can be searched and not flash or applet content.

The search and replace strings should be text and not a regex. You cannot use metacharacters such as \r or \n in either search or replace. This means that you cannot search and replace any multi-byte charset strings.

### To configure Response Body Rewrite

1. Select the Web service from the [Web Site](#) drop-down list.
2. Specify values for the following fields:
  - **Rule Name** - Enter a name for the response body rewrite rule.
  - **Sequence number** - Enter the sequence number of this response body rewrite policy. The sequence number sets the order of execution for multiple configured policies from lowest (1) to highest (128).
  - **Host Match** - Enter the matching criterion for host field in the Request Header. This is either a specific host match or a wildcard host match with a single " \* " anywhere in the URL. You can enter a partial domain with wildcard (for example: \*.abc.com) but you can not use multiple asterisks.
  - **URL Match** - Enter the matching criterion for URL field in the Request Header. The URL should start with a "/" and can have only one " \* " anywhere in the URL. A value of /\* means that the ACL applies for all URLs in that domain.
  - **Search String** - Enter the text string to be searched in the response body.
  - **Replace String** - Enter the replacement text.
3. Click [Add](#) to add the above configurations.



# Trusted Hosts

---

The **WEBSITES > Trusted Hosts** page allows you to designate a trusted host, that is, to specify an IP address for which authentication is not necessary. In this case, traffic coming from trusted hosts is assumed to be "safe" as very few security checks are applied.

## To configure a Trusted Host

1. Enter a name in **Trusted Host Group Name** field and click **Add**.
2. Click **Add Host** under **Actions** to add a trusted host under the created Trusted Host Group. The **Create Trusted Host** dialog box appears.
3. Specify values for the following fields:
  - **Trusted Host** - Enter the trusted host name for which you want to exempt the security checks.
  - **IP Address** - Enter the IP address for the trusted host.
  - **Mask** - Enter the associated netmask for the trusted host.
4. Click **Add** to add the above configurations.





## Chapter 6

# Adaptive Security

---

This chapter explains you how to effectively use the Adaptive Security feature of the Barracuda Web Application Firewall. The following are the topics covered in this chapter:

- *Overview* on page 76
- *Working with Adaptive Profiling* on page 78
- *Working with Exception Profiling* on page 84
- *Exception Heuristics* on page 86
- *Web Site Profiles* on page 87
- *Recommended way to use the Adaptive Security feature* on page 88

### Note



For more detailed information about a specific page in the Web interface, view the online help by clicking the question mark icon on the right side of the interface.

# Overview

---

Adaptive security comprises of:

- Adaptive Profiling
- Exception Profiling

Both of these analyze the request and response traffic to generate or fine tune security policies.

**Adaptive Profiling** learns the intricate structure of an application and then enforces conformance to that structure. It creates detailed security profiles by learning from requests and responses served by a particular Web application. It employs a positive security model by generating a whitelist of valid URLs and parameters which provides protection against Forceful Browsing and Parameter Tampering attacks.

The learned structure of the application is called the profile of the application. The application profile consists of individual URL and Parameter Profiles. These profiles are initially generated using the settings in the default security policy, but over time the profiler refines them to accurately reflect the behavior and security profile for the Web application. When strict profile checking is enforced (described later in this chapter), any request not found in the application profile is discarded.

Hackers commonly probe an application for vulnerabilities using automated vulnerability assessment tools. These tools try to probe the application for known weak files, scripts and directories that may be vulnerable to attack or may contain sensitive content. Such attacks are immediately denied when adaptive profiling is in use, since the requests violate the application profile.

For URL profiles the profiler learns the valid HTTP methods per URL, the allowed content types, maximum content, parameter lengths, maximum file uploads and referrers.

For parameter profiles, the profiler learns the valid parameters per URL, parameter length restrictions according to the user inputs as well as HTML markups in the response body, verification of predefined values in FORM input elements such as radio buttons, check boxes, drop-down lists, protection against tampering of hidden parameters and protection against range and type violations in input parameters.

**Exception Profiling** provides a heuristics based strategy to assist the administrator in reducing false positives. Since Web applications are dynamic and vary widely, a one size fits all security policy might not be adequate across such a spectrum. Patterns deemed to be attack patterns might be acceptable in some parameters in some FORMS. Long length inputs which normally violate the request limits might be required for some applications to work properly.

Exception Profiling essentially creates URL and Parameter profiles which help in relaxing the default security policy or existing URL profiles or Parameter profiles by fine-tuning them. In case a profile does not exist, Exception Profiling creates a new profile relaxing the default security policy, if a profile already exists it is fine tuned further to reflect the exception.

For example, the default setting on the [SECURITY POLICIES > Parameter Protection](#) page for the parameter “Max Parameter Value Length” is “1000”. Your application may have FORM parameters for which larger length user inputs are legitimate. That is, a free form text input representing user comments might need an exception to this “max 1000 character” rule.

Without exception profiling, you would have to create this exception manually using a URL and parameter profile. Chances are, you will be notified of this from upset clients whose requests have been blocked, since it is tedious to review all your application’s URL/FORM parameters manually.

With exception profiling, the system will “learn” this exception and create an appropriate URL and parameter profile to allow such exceptions, by observing the user traffic.

## Layout of Adaptive Security

---

The Adaptive Security module Web user interface is split into 4 pages:

1. Adaptive Profiling (**WEBSITES > Adaptive Profiling**)
2. Exception Profiling (**WEBSITES > Exception Profiling**)
3. Exception Heuristics (**ADVANCED > Exception Heuristics**)
4. Web Site Profiles (**WEBSITES > Web Site Profiles**)

The Learning process depends on the configuration settings of Adaptive Profiling, Exception Profiling and Exception Heuristics pages. Configuration of Adaptive Profiling is done via 1 and 4, whereas configuration of Exception Profiling is done via 2, 3 and 4.

# Working with Adaptive Profiling

---

## Configuring Adaptive Profiling

---

You can configure Adaptive Profiling to learn the application profile from either requests or response or both. The application profiling learn rules can be turned “On” or “Off” on URL spaces, by employing URL and extended matching, which allows you to selectively profile critical application components.

URL spaces to be profiled can be configured from the [WEBSITES > Adaptive Profiling](#) page. For example, you can turn on adaptive profiling for /cgi-bin/\*, /scripts, etc. The application profile developed by Adaptive Profiling is displayed on the [WEBSITES > Web Site Profiles](#) page. The configuration parameters are explained below.

### Edit Service Adaptive Profiling

By default adaptive profiling is configured for each service with predefined settings. You can edit the predefined settings for adaptive profiling by clicking on the **Edit** link under **Actions**.

### Add Adaptive Profiling Rules

You can create one or more learn rules for a URL space by clicking on the **Add** link under **Actions**. The profiling rule defines the rules for learning the specified URL space.

### To Start Adaptive Profiling

Click the **Start Learning** button under the [WEBSITES > Web Site Profiles > Service](#) section.

### To Stop Adaptive Profiling

Click the **Stop Learning** button under the [WEBSITES > Web Site Profiles > Service](#) section.

## Working with Navigation parameters

---

If you want the profiler to distinguish between two requests which have the same URL but different query parameters, then you should specify those parameters as Navigation Parameters. When you do so, a page is uniquely defined using the combination of the request URL and the navigation parameters, rather than just the request URL.

For example, consider the following Barracuda Web user interface URL:

```
http://waf.barracuda.com/cgi-bin/index.cgi?primary_tab=basic&secondary_tab=status
```

Here the value of the query parameters `primary_tab` and `secondary_tab` together determines the page. Different value combinations of the above generate completely different pages, containing different FORM elements and content.

In such a case, `primary_tab` and `secondary_tab` should be specified as Navigation Parameters so that the profiler generates separate profiles for the following:

```
/cgi-bin/index.cgi?primary_tab=basic&secondary_tab=ip_config
```

```
/cgi-bin/index.cgi?primary_tab=basic&secondary_tab=services  
/cgi-bin/index.cgi?primary_tab=advanced&secondary_tab=troubleshooting
```

The default behavior is to consider all parameters as non-navigation parameters.

The **WEBSITES > Web Site Profiles** page indicates the values of all the navigation parameters being used for the URL profile in the **Nav Params** column.

## Configuring URLs to be Excluded from Adaptive Profiling

Certain static content which do not have URL and FORM parameters should be excluded from being profiled. For example, images, style sheets, flash files do not need to be learned. These can be configured on the **WEBSITES > Web Site Profiles** page. A default list including the former is provided. You should customize this list for your Web sites, so that unnecessary profiles are not generated. This keeps the configuration size small and the performance high.

## Understanding Request and Response Learning

The following elements are learned during the learning phase:

*Table 6.1: Elements learned during learning phase*

Profile Type	Elements learned during learning phase
URL Profile	Allowed Query string, Allowed Methods, Allowed Content Types, Max Content Length, Max Parameters, Max Upload Files, Max Parameter Name Length, Referrers.
Parameter Profile	Type, Allowed Meta-characters, Max Parameter Value Length, Required, Ignore, Max Instances, File Upload Extensions, Max Upload File Size.

Where available, the initial values of these elements are taken from the default policy settings under the **SECURITY POLICIES** tab.

For example, when a new URL profile is generated, the parameter “Max Content Length” is set to 32k if the default security policy is being used. If the profiler receives a request with content length 35k, the new value of “Max Content Length” is set to 35k for the request URL.

All other security policy elements for the Service specified using the **WEBSITES** tab (e.g. Advanced Security elements, Allow/Deny rules, etc.) as well as those inherited from the **SECURITY POLICIES** tab (e.g. Cookie Security, Normalization, etc.) continue to apply during the learning phase. This ensures that your application is not vulnerable to attacks during the profile development phase. Whether attacks are blocked or just logged during this phase depends on the Service’s mode setting (Active or Passive).

The following URL and parameter profile configuration elements are not learned by the Adaptive Profiler. They continue to be applied as specified in the Service's security policy, even during the learning phase:

*Table 6.2: Elements not learned during learning phase*

Profile Type	Elements not learned during learning phase
URL Profile	Blocked Attack Types, Custom Blocked Attack Types
Parameter Profile	Blocked Attack Types, Custom Blocked Attack Types

The following table describes the different "Types" for a parameter profile. The top four are automatically learned by the profiler. The last two need to be manually specified.

*Table 6.3: Different Types for a Parameter Profile*

Param Type	FORM Attribute Used for Learning	Description	Allowed Values
Input		The parameter other than File Upload, Global Choice, Read Only, Session Choice, and Session Invariant type is treated as "Input" type.	All values allowed by the regex for "comments" Data Type element on the <a href="#">ADVANCED &gt; View Internal Patterns</a> page.
Read Only	type=hidden	All hidden FORM parameters are learned as "Read Only". This is done by observing the "<type=hidden>" attribute in the HTML response content. The value of the parameter is learned on a per session basis.	When profile is in Active mode: The allowed value for such a parameter in a request should be exactly equal to that learned from the response.  When profile is in Learning mode: If the value is found to be modified in a request during the learning stage, then the type is updated to "Input".
File Upload	type=file	The parameter of the type file upload in FORM is treated as "File Upload" type.	
Global Choice	type=checkbox radio submit	The input type parameters like check boxes, radio buttons and menu parameters in a FORM are treated as "Global Choice" type.	The system constructs an aggregated list of values learned from observing the values in responses across sessions.
Session Choice	NA	Has to be specified manually. It is similar to Global Choice, but the values are learned on a per session basis.	The system constructs an allowed list of values learned from observing the values on a per session basis.
Session Invariant	NA	Has to be specified manually. If the parameter value should be constant across multiple requests from the same session, then it can be set as "Session Invariant" type, for example; <i>session-id</i> .	The unique value learned for such a parameter per session.



## Response Learning

If Response Learning is turned on for a URL space, the system inspects HTTP responses in that space and learns the following from it:

**FORM parameters**—Parameter profiles are created based on their FORM attribute type as described in the table above. The system will also learn the maximum length for the parameter if specified (using the “maxlen” attribute in the HTML). Note that these parameter profiles are created for the action URL specified in the FORM, and not for the request URL which generated this response.

**Embedded URLs**—The profiler parses all the hyperlinks in the response body and generates URL Profiles for them. Note that this is only done for those hyperlinks which match one of the URL profile learn rules specified on the [WEBSITES > Adaptive Profiling](#) page.

**Embedded URL query parameters**—For the embedded URLs found above in the response content, the profiler also generates parameter profiles for the query parameters, if any. By default, these parameters are learned as Read Only parameters. If they are found to be modified in a subsequent request while the URL space is still being learned, their type is changed to Input type.

## Request Learning

When the profiler sees an incoming GET request, it generates profiles for the URL and its query parameters (assuming they do not match any of the navigation parameters for the containing URL space).

For a POST request, for say *url1*, the profiler would probably have learned the FORM and query parameters already from a prior response when the backend specified *url1* as the action URL for a FORM embedded in a response for another URL, say *url2*.

Client side scripting may introduce additional parameters in the POST request for *url1* which were not present in the *url2* response. These are learned as Input type parameters. If the client side scripting modifies a parameter learned as “Read Only” from the response earlier, the profiler will change its type to “Input” if the parameter “Request Learning” is set to “On” for this URL and the profile is in learning state. When “Request Learning” is set to “Off”, and the profile is in learning state, then the request is allowed, but the parameter type is not changed. When the profile is in active (learning is turned Off) state, then the request is blocked.

### Example:

The following example shows a request response scenario with the corresponding profiles generated by the profiler at each step.

1. Initial request for a.html containing two query parameters

**Request:** a.html?q1=abc&q2=def

URL Profile	Parameter Profile
a.html	Query Params {q1, q2}

2. Response for a.html containing an embedded FORM with action URL= b.html

**Response:** a.html?q1=abc&q2=defsd fsdf

```
<FORM action="b.html?q3=userinfo" method="post">
  <INPUT type="text" id="firstname"><BR>
  <INPUT type="text" id="lastname"><BR>
  <INPUT type="checkbox" id="married" value=""> Married<BR>
```

```
<INPUT type="submit" value="Send"> <INPUT type="reset">
</FORM>
```

URL Profile	Parameter Profile
a.html	Query Params {q1, q2}
b.html	Form params: {firstname, lastname, married}
	Query params: {q3}

3. User submits the FORM; Client-side injects additional parameters

**Request:** b.html?q3=userinfo


**Client side javascript:** If Married, inject FORM param: spousename

URL Profile	Parameter Profile
a.html	Query Params {q1, q2}
b.html	Form params: {firstname, lastname, married, spousename}
	Query params: {q3}

## Viewing Newly Generated Profiles

The newly generated profiles from the Adaptive Profiler module are displayed in red color on the **WEBSITES > Web Site Profiles** page. To view, select the application from the drop-down list and view the URL/Parameter profiles in red as shown in the figure below.

Figure 6.1: Newly generated profile

URL Profiles		Page 1 of 6	Filter	More Actions	Add URL	?
<input type="checkbox"/>	URL	Hits	Nav Params	Status	Mode	Action
<input checked="" type="checkbox"/>	 /webgoat/attack	0		On	Learning	Edit

You can filter the list of profiles by using “Profiles not reviewed” or “URLs with Params not reviewed”. If you have viewed a profile, you may mark it as “Read” by selecting it and using the **Mark Read** option from the **More Actions** drop-down list. After doing this, the profile will show up in black the next time the profile is viewed. This mechanism helps the administrator to review the profiles learned by the Barracuda Web Application Firewall.

## Enforcing Learned Profiles

Once you are satisfied with the generated profiles, you can select these profiles, and select the **Lock Profiles** option from the **More Actions** drop-down list to enforce them. After this is done, the system

will consider profile violations as attacks and would not learn from them. Then the Active/Passive mode setting for these URLs determines how attacks violating the profiles are disposed off.

To assist you in making this transition, the system displays the number of successful requests matching a generated URL profile after the last update to the profile. This is displayed as “Hits” for the URL profile on the [WEBSITES > Web Site Profiles](#) page.

## Using Strict Profile Checking during Learning

Enforcing “Strict Profile Checks” denies requests which do not match any profile. If set to "No", then the service's default web firewall policy will be applied to those requests which do not match a profile. Strict Profile checking cannot be edited when Adaptive Profiling is “On”. Also it varies with respect to the “Use Profile” parameter. The “Use Profile” parameter specifies whether to use URL profiles and Parameter profiles for validating the requests coming for a service. The following table describes the “Strict Profile Check” parameter behaviour with respect to the “Use Profile” parameter.

*Table 6.4: Strict Profile Check parameter behaviour*

Use Profile	Strict Profile Check	Behaviour
No	Yes, No	Profiles not used.
Yes	No	A "deny unless allow" strict rule is enforced making sure requests with no matching profiles are dropped right away thus enforcing a positive security model.
Yes	Yes	A negative security model is enforced making sure requests which do not match any profile go through default protections, and exceptions can be added to any default policy violations by just adding the relevant profiles.

# Working with Exception Profiling

---

## Configuring Exception Profiling

---

Exception profiling for a service can be configured on the [WEBSITES > Exception Profiling](#) page. The configuration elements specified here are on a per-service level. By default “Exception Profiling Level” is set to “None” and learning from trusted host is disabled. To learn from trusted hosts and change the “Exception Profiling Level” settings, edit the default settings by clicking on the **Edit** link under **Actions**.

## Exception Profiling Level

---

Depending on the security profile of the Service, choose from the available options (low, medium, high) for the “Exception Profiling Level” parameter. The settings are decreasingly liberal in creating exceptions from low to high. The detailed heuristics for each of these policies are defined on the [ADVANCED > Exception Heuristics](#) page.

## Learning from Trusted Hosts

---

You may designate a set of hosts as trusted and quickly learn exceptions from the traffic originating from these hosts. When learning from trusted hosts, exceptions are created automatically on a single violation.

## Learning Concurrently from Trusted and Non Trusted Traffic

---

The system can concurrently examine different traffic types and apply the appropriate policy. If the traffic originates from trusted hosts, it applies the trusted policy heuristics. If the traffic originates from non-trusted hosts, the selected Exception Profiling policy takes effect.

### Note



1. If **Learn From Trusted Hosts Group** is set to **Yes** and **Exception Profiling Level** is set to either **Low, Medium or High**, then exceptions from trusted hosts are directly learned using trusted hosts heuristics, and at the same time exceptions from non-trusted hosts are learned, based on the low, medium, high exception profiling settings.
2. If **Learn From Trusted Hosts Group** is set to **Yes** and **Exception Profiling Level** is set to **None**, then, only exceptions from trusted host are learned.

## Pending Recommendations

---

Exceptions for attacks whose action is set to “Manual” on the [ADVANCED > Exception Heuristics](#) page are not applied automatically, but show up as recommendations under the Pending Recommendations section on the [WEBSITES > Exception Profiling](#) page. For details on Exception Heuristics, refer *Exception Heuristics* on page 86.

Pending Recommendations displays a table of violations (which the system considers false positives) and the recommended fixes. The user can select single or multiple entries by selecting the check boxes. Click **Apply Fix**, to apply the recommended fix for the attack(s).

By looking at the Pending Recommendations table logs you can decide whether to accept the exception by clicking on the **Apply Fix** button or ignore the logged attack by clicking on the **Ignore** button. Clicking the Ignore button will only remove the selected log from the Pending Recommendations table. The next time if the same attack is encountered the Barracuda Web Application will again log it under Pending Recommendations. If you do not want to click the Ignore button every time it encounters the same attack, you have the option of turning off learning by going to the **ADVANCED > Exception Heuristics** page and setting the parameter “Setting” to “Off” for that particular attack.

For more information, refer the **WEBSITES > Exception Profiling** online help.

# Exception Heuristics

---

Exception Heuristics on the [ADVANCED > Exception Heuristics](#) page, allow you to configure rules to create exceptions on a per attack basis. The violations are divided into:

- Length Violations
- Input Violations
- Header Violations
- Cookie Violations
- Forceful Browsing

For each violation type you can set the following parameters:

**Setting**—Determines if the exception will be created automatically or will need manual approval via the Pending Recommendations table.

**New Value**—Determines the new value of the parameter after learning. This new value is different for different violation types. This new value for most of the violations is decided by selecting an option from the drop-down list. For the rest of the violations, the new value is based on the default option provided.

**Trigger Count**—This is a numeric value, which, when equals the number of times a violation is encountered from unique sources, triggers the exception learning process either automatically or manually. Only unique requests from a client are considered for counting. For example, if the same client generates multiple violations the system will count them as a single violation for the purpose of maintaining the trigger count. This neutralizes a hacker conducting repeated attacks on the Service.

## Working with Exception Profiling Levels

Exception profiling levels determine the exception creation heuristics for the Services to which they are bound. The four levels provided are: Low, Medium, High and Trusted. To view the settings for a profiling level, select the level and click [Show Definition](#). The **Request Violation Handling** module gets populated with the settings for that level. The levels defined here are shareable across multiple Services. Any change made to a level, reflects in the Exception Profiling criteria of a Service bound with this level.

# Web Site Profiles

---

The **WEBSITES > Web Site Profiles** page allows the Barracuda Web Application Firewall to use the URL profiles and Parameter profiles created for a service to validate the requests coming for that service.

In addition to that, you can learn the selected Web application space by setting the parameter “Mode” to “Learning”. The **URL Profiles** and **Parameters for** sections display a list of profiles that are learned from requests and responses served by the Web application.

## Search

---

The Barracuda Web Application Firewall enables you to search services using specific criteria like Web Site Name, IP and Port in the **Select Filter** drop-down list. The **Web Site** drop-down list under Service section displays the services based on the filter selected.

## Directories

---

Displays a tree structure of the URLs of the selected Web Application learned based on the directory path of the URL. Expand the parent directory and click the required leaf directory to display the profiles learned for that leaf directory, under the **URL Profiles** section.

## Stop learning a directory path

---

Select the parent directory check box(es) and choose **Lock Profiles** from the **More Actions** drop-down list. The next time the profiling agent will not learn that particular URL directory path.

## Things you can do from this page

---

- Edit the configurations for the existing URL Profiles and Parameter Profiles.
- Manage and organize large numbers of URL Profiles and Parameter Profiles.
- Manually configure URL Profiles and Parameter Profiles.

For more information, refer the **WEBSITE > Web Site Profiles** online help.

## Recommended way to use the Adaptive Security feature

---

1. From the **WEBSITE > Web Site Profiles** page, click **Start Learning**.
2. Either manually browse through the application (recommended) or crawl the application.
3. Let the "Adaptive Profiling" feature populate the URL and parameter profiles automatically.
4. Visit the created profiles, review them, if found satisfactory, click **Stop Learning** to stop the profiling for the Service. You may also select a subset of the profiles and enforce them by selecting the "Lock Profiles" action.
5. The profiles will be in "Passive" state. Look out for any false positives on the **BASIC > Web Firewall Logs** page. Also examine the "Hits" statistics under the URL Profile section. If found satisfactory, select **Lock all Profiles** from the **More Actions** drop-down list to turn all profiles to "Active".
6. If "Exception Profiling" is enabled, it would take care of any missing URL spaces which went uncovered during "Adaptive Profiling".
7. If possible, manually combine the learned profiles to optimize the configuration.
8. If your backend application or a portion of it has changed, you may relearn the space by choosing **Resume Learning** from the **More Actions** drop-down list on the desired space.

For more information on the different Adaptive Security modules, refer the online helps specific to those modules.



## Chapter 7

# Traffic Management

---

This chapter describes the three major traffic management features like Content Rule, Caching and Compression. The following topics are covered:

- *Content Rule* on page 90
- *Configuring Caching* on page 92
- *Configuring Compression* on page 94

### Note



For more detailed information about a specific page in the Web interface, view the online help by clicking the question mark icon on the right side of the interface.

# Traffic Management

---

The traffic management feature can be used to manage Web site traffic to improve the performance of Web sites. This includes Content Rule, Caching and Compression features for a service.

## Content Rule

---

A Web service can be further partitioned based on content in the HTTP requests by creating a content rule. A content rule is a collection of one or more rules that specify a pattern in the URL or header fields of the request. URLs and header fields are matched against these rules to determine whether the content rule policies apply to that request. The policies configured for the content rule are applied if the request matches any of the rules defined under the content rule. Content rules allow you to manage traffic flow for a Web service.

Configuring a content rule involves the following steps:

- Create the content rule in the target Web service.
- Add one or more rules that define which requests should match this content rule.
- Configure the policies that should be applied to requests that match this content rule.

You can configure policy for a content rule in the following three areas:

- **Load Balancing (only in Proxy mode)** - Sets load balancing policy for the content rule. By default, the parent Web service's load balancing policy is copied into the content rule. Load balancing is tied to a server group, and the Content Rule configurations specifies which server group to use. This allows you to distribute processing based on the content type (see *Example1: Content Rule for Images* on page 90).
- **Caching** - Sets caching policy for the content rule (refer *Configuring Caching* on page 92). This allows you to selectively cache based on the content type.
- **Compression** - Sets compression policy for the content rule. This improves the response time for clients accessing the Web service by compressing web pages with specific content type.

Rules are evaluated based on a key. The key is comprised of the URL, host and an optional extended match rule (that is, a value or expression for the header parameter). For more information refer *Extended Match and Condition Expressions* on page 214. The matching URL ACL is determined by a best match algorithm using the host, URL and extended match fields in specified sequential order. In most cases, host and URL may only be used to specify a rule. The Barracuda Web Application Firewall optimizes the search for the most common case by implementing a parallel search algorithm on all rules. The best matching rule is the rule with the longest matching host and URL keys. To configure a more complex rule based on certain fields in the request such as a client IP or an HTTP header, use extended match rules. Not all extended match rules are considered for evaluation. Only the ones specified for the matching host and URL are used for evaluation. If more than one such rule is specified, they are evaluated in the order of their weights; the lower the weight, the higher the precedence.

### Example1: Content Rule for Images

Assume that requests to a Web service are normally served by servers S1, S2, and S3.

However, assume you want to direct all image requests for content inside the images directory in the web server to a different set of servers say S4, and S5. To accomplish this

- Create a content rule for requests matching the URL `/images/*` as follows:

Figure 7.1: Configuring Content Rule

Content Rules							
Services	IP:Port	Rule	Domain	URL	Server IP	Port	Options
OnlineStore	216.101.241.211:8080	images	*	/images/*			Add Rule Add Server Edit Rename Delete

- Add one or more servers for this content rule (S4 and S5 in this case) using the **Add Server** option. Adding servers for content rules is similar to adding servers for a service. Any future requests matching /images/\* will be now directed to one of the servers added to this content rule (S4, S5) instead of being sent to the servers associated with the parent service (S1, S2, S3).
- By default, the load balancing policy of the parent service is inherited by newly added content groups. To change this, edit the content Rule and customize the load balancing policy to use between the servers added for this content rule.

To configure caching for this newly added content rule, create caching rules (see Figure 7.2). These rules allow you to specify file extensions and size restrictions for objects that should be served from the cache on the Barracuda Web Application Firewall itself in the future rather than fetching the object content from the backend servers.

Figure 7.2: Configuring Caching for Content Rule

Caching								
Services	Rule	Domain	URL	Status	Extensions	Max Size (KB)	Min Size (B)	Options
OnlineStore				Off				Edit
OnlineStore	images	*	/images/*	On	gif, tif, jpg, jpeg, png, bmp, ico, swf	256	256	Edit

Create compression rules for this newly added content rule. Similar to caching, compression rules let you decide what response content should be compressed by the Barracuda Web Application Firewall in order to better utilize network bandwidth.

### To create a Content Rule

1. Click **Add Rule** under **Actions**.
2. Specify values for the following fields:
  - **Content Rule Name** - Enter a name for the content rule.
  - **Status** - Select whether to enable or disable the content rule.
  - **Host Match** - Enter the host match that applies to this rule (www.example.com in this example).
  - **URL Match** - Enter the URL Match that applies to this rule (/index.html/david.gif in this example, which means david.gif file).
  - **Extended Match** - Enter the extended match selection criteria. This can be a specific extended match, an expression, or an asterisk (meaning any extended match). To build an expression, click the *Edit* image button that appears next to this field, and specify the values for the following fields:
    - **Header Expression**: Specify a valid header expression.
    - **Element Type**: Select the element type from the drop-down list.
    - **Operation**: Select the operation from the drop-down list.
    - **Value**: Specify a valid expression.
    - **Concatenate**: Select 'and' radio-button to add some more expressions to the existing match sequence. Select 'or' radio-button to replace the existing match sequence.
    - Click **Insert** and then click **Apply** or click **Cancel**. For more on how to write extended match expressions, refer *Extended Match and Condition Expressions*.
  - **Extended Match Sequence** - Enter a sequence number (0 to 1000) to specify an order for matching the extended-match rules to resolve conflicting URL ACLs that have the same host-match, url-match and extended-match.

3. Click **Add** to add the above settings.

## Configuring Load Balance for a Content Rule (only in Proxy mode)

### To configure Load Balance for a Content Rule

1. Click **Edit** under **Actions** against the created rule.
2. Under **Load Balance** section, specify values for the following fields:
  - **Algorithm** - Select the algorithm to be used for load balancing.
  - **Persistence Method** - Select the persistence method of load balancing. Persistence maintains a connection between a client and the first server that it connects to. When the system is load balancing traffic, subsequent requests from that client are always sent to the same server. This is useful when the server requires to maintain state information about every client. For example, in an E-commerce application this policy maintains the connection from the time an online customer begins filling a shopping cart until that customer purchases the cart contents and completes the transaction.
  - **Failover Method** - Select the failover method for responding to a request which is persistent, but the server that must serve the request set to "out-of-service".
  - **Source IP Netmask** - Enter the netmask for Source IP persistence method. The IP plus netmask results in a network identifier which is used to identify a client. A more specific netmask (such as 255.255.255.255) will track each client independently, and may cause a higher memory load on the Web Application Firewall. Whereas as a less specific netmask (such as 255.255.0.0) will group multiple clients under the same network identifier and connect them all to the same server.
3. Click **Save Changes** to save the above settings.

## Configuring Caching

---

Caching is a process of storing commonly used information in local memory for quick retrieval (rather than sending repeated requests to the Web server for the same information). This can improve performance (sometimes dramatically) and reliability. Caching can store Web pages and commonly used objects such as graphics files. Caching provides the following benefits:

- Reduced latency when retrieving Web content.
- An overall reduction in bandwidth and server load.
- Automatic identification and replication of site content.

### To configure caching

1. Click **Edit** under **Actions**.
2. Specify values for the following fields:
  - **Status** - Select whether to enable or disable caching.
  - **File Extensions** - Enter a file extension in the text box and click **Add**.
  - **Max Object Size** - Enter the maximum size for a cached object in KB (default is 256).
  - **Min Object Size** - Enter the minimum size for a cached object in bytes (default is 256).
  - **Ignore Request Headers** - Select Yes to ignore request cache-control headers.
  - **Ignore Response Headers** - Select Yes to ignore response cache-control headers.
  - **Cache Negative Responses** - Select Yes if you want common negative responses (204, 305, 404, 405, 414, 501, 502, and 504) to be cached.

- **Expiry Age** - Enter the maximum duration in minutes for a cached object.

3. Click **Add** to add the above settings.

## Content Rules and Dynamic Pages

When a request is given to a Web service, the content rule whose rule matches the request is selected and caching in that content rule is checked to see if cache is enabled. When enabled, the response is cached, and subsequent requests for the same object are served from cache. When disabled, each subsequent request is forwarded to the back-end server for the reply.

This works best for responses containing static pages, since they remain unmodified over multiple requests. But caching fails when the response content changes for each request due to different context or conditions. This kind of dynamic response is normal when server side scripting is used to generate context-sensitive response based on URL/form params in requests.

For example, a content rule with a URL of /reports/\* matches all pages under /reports. Hence, if cache is enabled on this content rule, all pages under /reports are cached. If the /reports folder also contains dynamic pages that are not to be cached, the user has the following options:

- Move all dynamic pages into another folder, for example /reports/cgi\_bin, and then create another content rule with a URL of /reports/cgi\_bin\* and disable cache for this content rule.
- Disable cache for the original content rule, and do not cache any page under /reports.

## Object Freshness

The freshness of cached objects determines the life span of objects that are stored locally until retrieval of a newer version from the originating server. To accomplish this task, the object is designated with a freshness algorithm. When an object expires, this algorithm will direct the returning browser to retrieve a new copy from the originating server. Otherwise, the browser will load a cached copy that was stored locally.

If the cached object is stale, it will be retrieved from the originating server. If the object expires, it is still served from cache; however the response will include a Warning 110 (response is stale) header.

The following algorithm is used for calculating object freshness. For this algorithm, age is calculated as follows:

$$\text{age} = (\text{current\_time} - \text{time\_retrieved}) + \text{object\_age}$$

- When both **Ignore Request Headers** and **Ignore Response Headers** are enabled, all objects are considered fresh.
- When **Ignore Request Headers** is enabled:
  - If **Ignore Response Headers** is not set and the age of an object is greater than cached response max-age (if present), the object is considered stale.
  - If **Ignore Response Headers** is set and the age of an object is not greater than cached response max-age (if present), the object is considered fresh.
- When **Ignore Request Headers** is disabled:
  - If age is greater than request max-age header (if present), the object is considered stale.

The following table describes how to determine an object's freshness.

Table 7.1: Object Freshness Calculations

If ...	Then object freshness is calculated as ...
Ignore Response Headers is enabled.	$\text{freshness} = \text{age} - \text{expiry\_age}$
Cached response had an expiration time.	$\text{freshness} = \text{current\_time} - \text{object\_expiration\_time}$
Age of an object is greater than Expiry Age.	$\text{freshness} = \text{age} - \text{expiry\_age}$
Cached response has a time last modified header.	$\text{stale\_age} = \text{time\_object\_retrieved} + \text{object\_age} - \text{time\_object\_last\_modified}$ $\text{stale\_age} = \text{stale\_age} * \text{age\_from\_last\_modified\_percentage} / 100)$ if $\text{age} > \text{stale\_age}$ , $\text{freshness} = \text{age} - \text{stale\_age}$
Cached response does not have a time last modified header:	$\text{freshness} = \text{age} - \text{expiry\_age}$
Staleness < 0, a min-fresh header request is present, and it is set to be greater than the staleness value (positive value of it).	The object is considered stale.
Staleness < 0, and a min-fresh header request is not present.	The object is considered fresh.
A max-stale header request is present, and it is set to be greater than the staleness.	The object will expire.
A max-stale header request is not present.	The object is considered stale.

## Configuring Compression

The Compression policy improves the response time for clients accessing the service through modems. Enabling this feature compresses web pages that use HTML, JavaScript, Java and other text-based languages, which results in enhanced traffic management and significant reduction in download time. This policy can be applied to all the client requests and also to specific client requests that is using Content Rules. Enabling compression for a service applies compression policy to all the requests.

### Note



Cached object, which is already present in the cache in an uncompressed form should be cleared in order to get a compressed object.

### To configure compression

1. Click **Edit** under **Actions**.
2. Specify values for the following fields:
  - **Status** - Select whether to enable or disable compression.

- **Content Types** - Enter the content-type to be compressed. Examples: text/css, text/html, text/js, text/plain.
- **Min Object Size** - Enter the minimum size for the response.
- **Compress Others** - Select whether the Barracuda Web Application Firewall should compress files of unknown content types or not. These unknown file types can be non text content types such as executable binaries, flash content and so on. Unknown content-type does not mean "not in the list of compressible content-types." Content which is missing a content-type header is not compressed.

3. Click **Add** to add the above settings.

#### Note



It is recommended that compression be turned off in the Web server on the backend. Barracuda Web Application Firewall will not uncompress, inspect and recompress such compressed responses originating from the backend servers. Instead users should allow the backend to send uncompressed content, let the Barracuda Web Application Firewall examine it for security violations and then compress it and send it out to their client's browsers.





## Chapter 8

# Keys and Certificates

---

This chapter provides an introduction to the Public Key Infrastructure (PKI) technology. It also includes a system overview of how the Barracuda Web Application Firewall uses PKI encryption to protect traffic:

- *Overview* on page 98
- *Creating a Test Certificate* on page 101
- *Uploading a Certificate* on page 104

### Note



For more detailed information about a specific page in the Web interface, view the online help by clicking the question mark icon on the right side of the interface.

# Overview

---

The Barracuda Web Application Firewall uses PKI objects for Secure Socket Layer (SSL) encryption. This technology encrypts data that is to be transmitted between a client and the Barracuda Web Application Firewall, or between the Barracuda Web Application Firewall and Web servers. SSL encryption is the most effective way to securely send confidential user information over the Internet. The Barracuda Web Application Firewall includes Public Key Infrastructure (PKI) objects like keys and certificates that can be used for SSL encryption.

The PKI technology allows secure exchange of data over the Internet. It allows successful and safe accomplishment of the transactions such as using a credit card to purchase items online or allowing employees' access company's Intranet. Unlike earlier forms of cryptography the public key technology works with a pair of keys for authentication and encryption. This type of cryptography starts with the creation of the two keys: a public key and a private key. The contents of a public key is to be known by everyone, whereas the contents of a private key is secret and is known only by its owner. This key pair is then used to encrypt and then decrypt messages sent by this owner. As the private key remains secret and the public key is known you are able to initiate a secure communication without having to previously share a secret through some other type of medium. Private portion of the key pair confirms the owner's identity.

## Caution



A compromised private key is a security threat! If a private key is exposed, then the public key can be easily derived. However, a private key cannot be derived from an exposed public key. A digital certificate is derived from a key pair. It is an attachment to a sent message and used for security purpose. The certificate helps verify the identity of a user that sends the message. A certificate also provides the receiver the means to encode a reply. The most widely used standard for digital certificates is X.509.

Most browsers require a digital certificate from a trusted third-party certificate authority (CA). In most cases, if a browser does not recognize a certificate, it may deny a transaction or refuse access to a secret portion of their Web site. In this case, an individual wants to send an encrypted message will apply for a digital certificate from a trusted CA, such as VeriSign or Thawte. A CA issues a certificate containing the applicant's public key and other identification information. This digital certificate is installed locally as root certificate. This enables confidential and authenticated communication with other browsers that already have a certificate installed from the same CA.

PKI protects data sent over the Internet in the following ways:

- **Authentication** - An issued digital certificate that is given to a user, organization, or Web site validates the identity of a person and then allows them to access the Web site.
- **Privacy** - A certificate also protects data from being intercepted during transmission.
- **Integrity** - A "signed" digital certificate ensures that the message or document has not been manipulated or corrupted during transmission.
- **Authorization** - Before certificates, users were required to give an ID and password for authorization. This information was sometimes lost or forgotten. Certificates guarantee the authenticity of each user.

Digital certificates that is created within the Barracuda Web Application Firewall are of the standard X.509 format. A test (user) certificate that is created is considered as self-signed.

## Types of Certificates

---

Currently the Barracuda Web Application Firewall only supports X.509 certificate.

## X.509 Certificate

This certificate is used to encrypt a client's session. This is an industry standard digital certificate, which the Barracuda Web Application Firewall uses to negotiate an SSL session to a service. The X.509 digital certificate can be created from scratch within the Barracuda Web Application Firewall. This certificate is one of the most commonly used types of certificate and the International Telecommunication Union (ITU) recommends it. This certificate is a self-signed certificate.

## Certificate Usage

---

The Barracuda Web Application Firewall encrypts the traffic coming from a client to a back-end Web server, and also makes server-side encryption. Standard X.509 certificates are used for the server-side encryption. This type of certificates can either be created on the Barracuda Web Application Firewall or they can be imported from a trusted third-party CA such as VeriSign or Thawte for the SSL encryption.

Besides server-side encryption, the Barracuda Web Application Firewall can also be used for client-authentication. This type of encryption is used to protect services configured on the Barracuda Web Application Firewall. The certificate received from a client's browser acts as trusted certificate and authenticates the client to the Barracuda Web Application Firewall.

The client-server negotiations includes the following:

- The Barracuda Web Application Firewall sends the server's certificate to the client.
- Then the Barracuda Web Application Firewall requests the certificate in return.

The server is verified in the SSL handshake that allows the server and the client to authenticate each other. This allows the alliance of the client and the server in creating symmetric keys that are used for encryption, decryption, and tamper detection during the SSL sessions.

The Barracuda Web Application Firewall sends the server's certificate to the client and requests the certificate. In return, the client sends the user certificate, which is authenticated by Barracuda Web Application Firewall using the trusted certificate.

The traffic between the Barracuda Web Application Firewall and the back-end server can either be encrypted or unencrypted. The actual configuration of the server does not impact the SSL negotiation between the Barracuda Web Application Firewall and the client in any way.

## Certificate Components

---

### Key Pair

The Barracuda Web Application Firewall implements an asymmetric methodology for encryption, where two related keys are used in combination as opposed to symmetric encryption where just one key is used. A key pair consists of a public key and a private key. They work together in such a way that one of the key pair encrypts a message, and the other decrypts the encrypted message.

Even though the public key is known, it is still practically and mathematically impossible to deduce the private key from the public key.

## **Distinguished Name (DN)**

The Distinguished Name (DN) in the certificate uniquely identifies the public key owner who issues the certificate.

## **Token**

A token is a cryptographic item used for secure storage and transfer of private interface and certificate. Currently, the Barracuda Web Application Firewall supports only the PKCS12-type token. The PKCS12 token can be loaded onto the Barracuda Web Application Firewall from a remote system or saved from the Barracuda Web Application Firewall onto a remote system.

## **CA Certificate**

A CA certificate is a third-party certificate that a CA issues and is part of trust between a root CA and a certificate installed in the Barracuda Web Application Firewall. This certificate can be added to a certificate chain, where it is used for encryption and authentication. Some browsers require that a certificate should be from a known source (such as VeriSign), before communication between a client and a server occurs.

# Creating a Test Certificate

---

Another PKI object that can be created from scratch within the Barracuda Web Application Firewall is an X.509 digital certificate. This certificate is one of the most commonly used types of certificate and the International Telecommunication Union (ITU) recommends it. However, it is not defined as the industry standard for certificates. This means that an X.509 certificate generated by the Barracuda Web Application Firewall may or may not be readable by a Web server or a Web browser.

This certificate is a self-signed certificate and is also called as a *user certificate*.

## To create a test certificate do the following:

1. Select **BASIC > Certificates > Certificate Generation**.
2. Click the **Create Certificate** button. The **Certificate Generation** dialog box appears. Enter the following information:
  - 2a. **Certificate Name** - Enter a name to identify this certificate.
  - 2b. **Common Name (CN)** - Enter the domain name (DN) that is used to access the Barracuda Web Application Firewall's Web interface. For example:  
"barracuda.domain.com".
  - 2c. **Country** - Enter the two-letter country code for this new DN.
  - 2d. **State or Province Name** - Enter the state's complete name for this DN.
  - 2e. **Locality Name** - Enter a location's complete name for this DN.
  - 2f. **Organization Name** - Enter the organization's complete name for this DN.
  - 2g. **Organization Unit Name** - Enter the organization's division or unit for this DN.
3. Click the **Generate Certificate** button.

# Saved Certificates

---

All the created (self-signed) certificates and uploaded (trusted) certificates are listed under **BASIC > Certificates > Saved Certificates** section. The certificates created using the Barracuda Web Application Firewall are secure and authentic, some browsers require that an encrypted certificate come from a reliable and known certificate authority (CA). You can request a third-party certificate and then store it locally on the Barracuda Web Application Firewall. The Barracuda Web Application Firewall supports both locally created and third-party created certificates. You can download the CSR file and send it to a trusted CA such as VeriSign or Thwate for signing. The following are the two download options that are available:

## CSR

---

A Certificate Signing Request (CSR) is created each time you generate a certificate using the Barracuda Web Application Firewall. It contains information such as organization name, domain name, locality, country and the public key. This is the file sent to a trusted third-party CA such as VeriSign or Thawte for authorization. A CA administrator verifies the CSR, signs the request and returns a valid certificate to be used for SSL encryption.

### To Download a CSR

1. Under **Saved Certificates** section, identify the certificate that needs to be signed by a third-party CA.
2. Click **CSR** under **Download** option. The pop-up window appears. Select **Save** to save the file to the location you desire. A CSR file is saved with the extension .csr.
3. You can send this CSR file to a trusted Certificate Authority (CA) for signing. A CA verifies the CSR and returns a signed certificate to be used for SSL encryption.

## Certificate

---

Certificate is a digital identity document that enables both server and client to authenticate each other. This is a signed certificate. It can be self-signed or signed by a trusted third-party CA. A certificate contains information such as user name, expiration date, a unique serial number assigned to the certificate by a trusted CA, the public key, and the name of the CA that issued the certificate. You need to extract the key from the certificate, and then install the certificate on the Barracuda Web Application Firewall under **Upload Certificate** section.

### Extracting the key from the Certificate

Once the CSR is signed and returned, the certificate file is replaced by the new certificate. You need to extract the key and install the signed certificate on the Barracuda Web Application Firewall.

### To extract the key from a certificate

1. Click **Certificate** under **Download** option. The **Save Token** pop-up window appears.
2. Enter the pass phrase in **Encryption Password** field and click **Save**. The certificate gets exported as pkcs12 token.
3. Extract the private key from pkcs12 token using the same pass phrase.
4. The openssl command used to extract the key is:

```
openssl pkcs12 -in < pkcs-token > -nocerts -out < key.pem >
```

5. Once you extract the private key, you need to upload the certificate on the Barracuda Web Application Firewall. See *Uploading a Certificate* for more information about uploading a signed certificate.

**Note**



Download options (CSR and Certificate) are valid only for imported and generated certificates. In case of trusted certificates, CSR download is not valid. The user can only download the certificate as PEM.

# Uploading a Certificate

---

The **BASIC > Certificates > Upload Certificate** section allows CA certificates to be uploaded on Barracuda Web Application Firewall.

## Signed Certificate

---

A signed-certificate is a certificate obtained from the third party CA organization (such as VeriSign or Thawte) to be used for SSL encryption. When uploading a signed certificate as a PKCS #12 token, it is mandatory that the file should be with .pfx extension, otherwise it is treated as \*.pem file.

### To upload a Signed Certificate:

1. Specify values for the following fields:
  - **Certificate Name** - Enter the name of the certificate.
  - **Upload Signed Certificate** - Click **Browse** to upload a signed certificate.
  - **Certificate Password** - Enter the password for the certificate only if its in PKCS12 token.
2. Click **Upload Now** to upload the signed certificate.

## Certificate Key

---

A private-key is the secret portion of an encryption/decryption key pair. Only the person exchanging a secure transaction should know it. Click **Browse** to upload an unencrypted private key to accompany a X.509 certificate. This key must be unencrypted and it should be in PEM format.

## Intermediary Certificates

---

Intermediary certificates are the are the CA certificates in PEM format. Click **Browse** to upload an intermediary certificate. You can add a single or a set of intermediary CA certificates by clicking the '+' button.

## Uploading a Trusted Certificate

---

A trusted-certificate is a certificate sent from a CA. Including the CA's certificate as a trusted certificate implies that any entity that has a certificate signed by the CA will be authenticated for the SSL Web services that the Barracuda Web Application Firewall provides.

### To upload a Trusted Certificate:

1. Specify values for the following fields:
  - **Certificate Name** - Enter a name to identify this certificate.
  - **Upload Trusted Certificate** - Click **Browse** to upload the trusted certificate in PEM format, as a \*.pem file.
2. Click **Upload Now** to upload the trusted certificate.



## Chapter 9

# User Access Control

---

This chapter describes how to configure user authentication and access control through the Barracuda Web Application Firewall. The following topics are covered:

- *Overview* on page 106
- *Creating an Authentication Policy* on page 107
- *Creating an Authorization Policy* on page 112
- *Creating new Authentication Services* on page 113
- *Creating Local Users/Groups* on page 115
- *Allowing/Denying Client Certificates* on page 116

### Note



For more detailed information about a specific page in the Web interface, view the online help by clicking the question mark icon on the right side of the interface.

## Overview

---

The Barracuda Web Application Firewall provides features to implement user authentication and access control. You can create a virtual private network (VPN) tunnel to control user access to Web sites. The user-access features allow you to specify who should have access to your Web sites and what level of access each user should have. By combining these with SSL encryption, you can create a secure VPN to your Web sites.

The authentication process requires users to provide a valid name and password to gain access. A validated user has qualified access to the Website; that is, the amount of data and services to which this user has access depends on the user's authorization level. The following figure illustrates the authentication process:

The user accesses a login page (a GET request). This page provides a form for entering a user name and password. (The page can contain other information, but any other entered information is ignored and discarded.) The login form must be accessible to all users. (While the following figure shows the form residing on a back-end server, this is not necessary. Because it is just a form to submit, it can be located anywhere. In addition, the Barracuda Web Application Firewall includes a default login form stored internally which you can use in lieu of creating your own login page.)

The user submits the form (a POST request). The input is authenticated in the Barracuda Web Application Firewall; it is not sent to a back-end server. The Barracuda Web Application Firewall compares the submitted information against an internally or externally located authentication database. Upon a successful authentication, the Barracuda Web Application Firewall sends a cookie back to the user.

If the user is authenticated and allowed access after logging in, he is redirected to a success page. On subsequent requests, the Barracuda Web Application Firewall authenticates the user and (if allowed access) immediately forwards the request to the desired location.

When an authenticated user attempts to access an unauthorized page (that is, a page for which he does not have permission to view), he is redirected to a denied authorization page.

If a user fails authentication, he is redirected to a failed authorization page (not illustrated in the following figure).

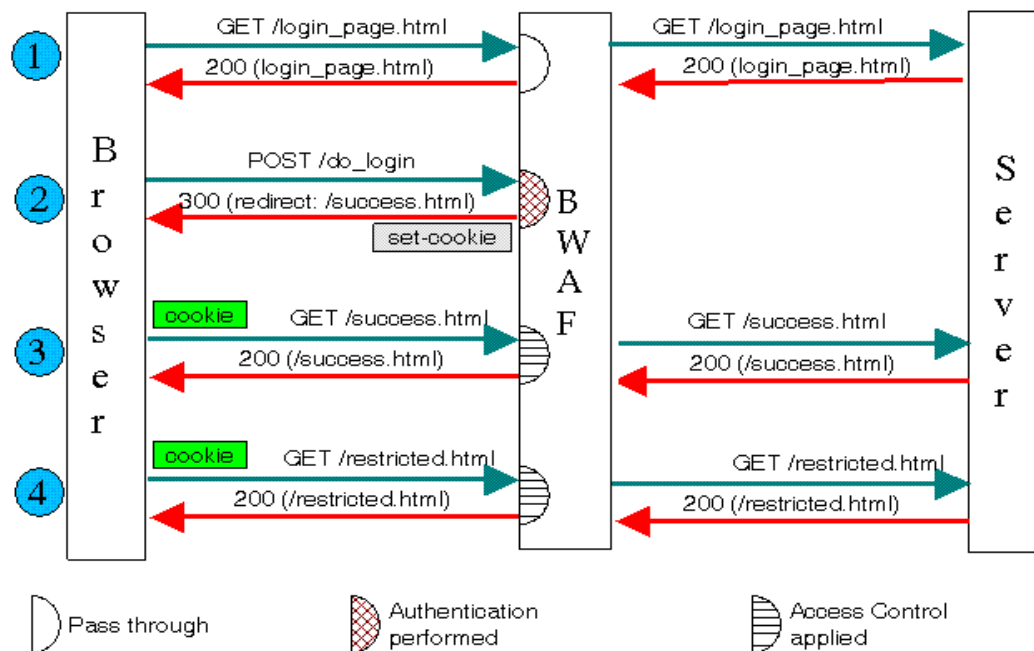


Figure 9.1: Authentication Process Diagram

## Steps to Configure Access Control to different parts of your Web site

1. First set up authentication for your Web site on the **ACCESS CONTROL > Authentication** page.
2. Associate an authentication database with your Web site. This database can be internal or external.
3. If internal authentication database, then set up local users and groups on the **ACCESS CONTROL > Local User/Groups** page.
4. If external authentication database, then set up LDAP/Radius on the **ACCESS CONTROL > Authentication Services** page.
5. After setting up authentication, configure the authorization policy as per your requirements. You can configure the authorization policy for the whole of the Web site, for example, /\*, or only certain part of the Web site, for example, /abc/\* only.

## Creating an Authentication Policy

The **ACCESS CONTROL > Authentication** page allows you to specify the parameters and resources for configuring authentication for a service. For authentication, users are presented with an HTML form with user ID and password fields. This page has unrestricted access and it should be hosted on existing servers as part of your regular services. The Barracuda Web Application Firewall processes the GET and POST request. The user is authenticated based on the username and password presented in the request.

### To edit an Authentication Policy

1. Click **Edit** under **Actions**.
2. Specify values for the following fields:
  - **Service** - Specifies the service to be authenticated.
  - **Status** - Select whether to enable or disable authentication for this service.
  - **Authentication Service** - Select the service to be used for authentication. The list includes all authentication services created in **ACCESS CONTROL > Authentication Service** page.
  - **Auth Success URL** - Specifies the URL to which a user is redirected when authentication succeeds. If this is not specified, Barracuda Web Application Firewall displays a default page indicating authentication succeeded.
  - **Auth Logout Success URL** - Specifies the priority over the default logout page thrown by the Barracuda Web Application Firewall on successful logout.
  - **Auth Failure URL** - Specifies the URL to which the user is redirected when authentication fails. If this is not specified, the Barracuda Web Application Firewall displays a default page indicating authentication failed.
  - **Trusted Hosts Action** - Select the action to be taken for the trusted hosts accessing this service.
  - **Trusted Hosts Group** - Select the trusted hosts group binding for this service.
  - **Idle Timeout** - Specifies the duration in minutes that a user can remain idle without accessing the SSO environment. For example, if the user does not access the SSO environment within the specified time, the user's session becomes idle and the user is challenged to provide login credentials to access the SSO environment again.
  - **SSO Cookie Update Interval** - Specifies the duration in seconds to update the SSO user session cookie with the new timestamps that is, if the user has not been idle, the cookie is updated at the specified interval to prevent the session from timing out. It is recommended that the value for this parameter is lesser than the idle and creation timeout.
3. Click **Save Changes** to save the above settings.

## Single Sign-On (SSO)

---

Single Sign-On (SSO) is a mechanism where a single set of user credentials is used for authentication and authorization to access multiple applications across different Web servers and platforms, without having to re-authenticate.

SSO system acts as a Web gate for all inbound Web traffic. When a user attempts to access the first Web site, the user is challenged to provide login credentials. If successful, the user is authenticated to the system, and a SSO User Session Cookie is generated. SSO User Session Cookie indicates that the user is authenticated for some duration. If not successful, the user's authentication request is rejected.

The SSO environment enables you to define the resources (Web sites and applications) that have to be protected, and provide rules for accessing the resource. The rules are as follows:

- **Authentication:** Authentication verifies that a user is who he or she claims to be. For authentication, the users are presented with an authentication login page, and are challenged to provide user credentials.
- **Authorization:** Authorization determines if a user is granted access to a requested resource. A user may want to see the data that is protected by a policy.

The Barracuda Web Application Firewall supports both single domain and multi-domain SSO.

## Single domain SSO

Single domain SSO takes place within a single domain. For example, consider the domain 'barracuda.com' is hosting several restricted Web sites on several hosts. The user can set up single sign-on to this domain, so that the authenticated users can access all or a subset of these restricted resources by authenticating just once.

### Setting up Single domain Single Sign-On Environment

#### Steps to Configure Single domain SSO

1. Go to **ACCESS CONTROL** > **Authentication** page.
2. Identify the service to which you want to set up single sign-on.
3. Click **Edit** against that service, **Edit Authentication** pop-window appears.
4. Set the **Status** 'On'.
5. Select **Authentication Service** from the drop-down list to be bounded with the service.
6. Set the **Session-Cookie Domain** to have same domain name for the services that you want to configure single domain SSO. For example, service1 and service2 both have '.barracuda.com' as the session cookie domain).
7. Click **Save Changes**.

#### Note



In a Single domain SSO set up, ensure that you configure same **Session Cookie Domain** name for all the services on **ACCESS CONTROL** > **Authentication** page.

### Logout in Single domain Single Sign-On Environment

When a user logs out of a domain, the Barracuda Web Application Firewall removes the user session cookie from the user's browser by expiring it, and the user is automatically logged out of other corresponding domains. For example, consider that you are logged into 'host1.bc.com', 'host2.bc.com' and 'host3.bc.com' that contain 'bc.com' as their cookie domain. If you perform a logout in host1.bc.com, the user session cookie is removed from the browser, and you are automatically logged out of host2.bc.com and host3.bc.com.

#### Note



If the user does not access the SSO environment within the specified idle timeout, the user's session becomes idle and the user is challenged to provide login credentials to access the SSO environment again.

## Multi-domain SSO

Multi-domain SSO enables a user authentication to be honored by all the hosts in two or more domains. For example, a set of URLs that reside within the domains 'www.abc.com' and 'www.xyz.com' can be set to single sign-on.

To achieve a multi-domain single sign-on, a master domain is required for authentication. The Barracuda Web Application Firewall multi-domain single sign-on environment can have one master domain and one or more slave domains. The master domain acts as a centralized authentication server that authenticates the users and transfers the SSO User Session Cookie to the slave domains.

In a multi-domain single sign-on environment, each domain is responsible for maintaining and enforcing its own idle timeout. This means the cookie value for different domains might be different. You have to configure the master service and the slave services on the Barracuda Web Application Firewall on [ACCESS CONTROL > Authentication](#) page.

### Multi-domain Single Sign-On Configuration

For a multi-domain SSO environment, you should explicitly specify the master service and master service URL for the domains as explained below:

- **Master Service:** Specifies if the master service URL is handled by this service. When the parameter is set to 'Yes', this service acts as the master domain to the subsequent domains. When the parameter is set to 'No', this service acts as the slave domain that accepts the cookie from the master domain.
- **Master Service URL:** Specifies the URL that provides a cookie. In case of the master domain specify only the URL path, but for the slave domains specify the protocol, host, master domain and URL path.

#### Note



The master service URL path can be any URL that you prefer. For example, /ncso.process, /index.html, etc. This URL is used to identify the master service URL in a multi-domain environment.

For example, consider 'www.abc.com' as the master domain and 'www.xyz.com' as the slave domain. If the master service URL for the master domain is '/ncso.process', then the master service URL for the slave domain is 'http://www.abc.com/ncso.process'.

### Multi-domain Single Sign-On Functionality

If a user attempts to access the master domain first, the user is challenged to provide login credentials. On a successful login, the user gains access to the master domain and to the subsequent domains. But if a user attempts to visit the slave domain first, the Barracuda Web Application Firewall redirects the user to the master service URL for authentication, and is challenged to provide login credentials. If successful, the user gains access and is redirected to the requested domain.

For example, consider 'www.abc.com' as the master domain and 'www.xyz.com' as the slave domain. If a user attempts to access the master domain (www.abc.com) first, the user is challenged to provide login credentials. A SSO User Session Cookie is generated on a successful login. Now, the user gains access and can navigate to the slave domains using the generated session cookie without having to re-authenticate.

In case, if a user attempts to access the slave domain (www.xyz.com) first, the Web application redirects the user to the master service URL for authentication. The user is challenged to provide login credentials. If successful, SSO User Session Cookies are generated for both the domains (master and slave domains) and enables the user to access the slave domain.

### Setting up Multi-domain Single Sign-On Environment

#### Steps to Configure Master domain

1. Go to [ACCESS CONTROL > Authentication](#) page.
2. Identify the service that you want to configure as master domain.
3. Click **Edit** against that service, **Edit Authentication** pop-window appears.
4. Set the **Status** 'On'.

5. Select **Authentication Service** from the drop-down list to be bounded with the service.
6. Click **Save Changes**.
7. Now, under **Single Sign On** section, set the parameter **Master Service** to 'Yes'. This service is identified as the master service that provides cookie for the subsequent slave domains.
8. Specify the URL path in **Master Service URL**. The master service URL path can be any URL that you prefer. For example, /nesso.process, /index.html, etc.
9. Click **Save Changes**.

#### Steps to Configure Slave domain

1. Go to **ACCESS CONTROL > Authentication** page.
2. Identify a service that you want to configure as slave domain.
3. Click **Edit** against that service, **Edit Authentication** pop-window appears.
4. Set the **Status** 'On'.
5. Select **Authentication Service** from the drop-down list to be bounded with the service.
6. Click **Save Changes**.
7. Now, under **Single Sign On** section, set the parameter **Master Service** to 'No'. This service is identified as the slave domain.
8. Specify the protocol, host, master domain and URL path in **Master Service URL**.
9. Click **Save Changes**.

#### Chained Logout in a Multi-domain Single Sign-On Session

If the user performs a logout in the master domain, the Barracuda Web Application Firewall removes the master domain's cookie from the user's browser by expiring it. But if the user performs a logout from the slave domain, it removes the slave domain's cookie from the user's browser by expiring it, redirects the user to the master domain, and informs the master domain to logout (remove the master's cookie) the user.

For example, consider the case where three domains www.xyz.com, www.abc.com and www.def.com are a part of a multi-domain SSO environment, www.xyz.com as the master domain, www.abc.com and www.def.com as the slave domains. When a user performs a logout in the master domain (www.xyz.com), the user session cookie is removed from the browser by expiring it, and automatically logs out of other corresponding domains (www.abc.com and www.def.com).

When a user performs a logout in the slave domain (www.abc.com), the slave domain expires its cookie, redirects to the master domain (www.xyz.com), and requests the master domain to expire its cookie and logout the user. After these steps have been completed successfully, www.abc.com redirects the user to logout from www.def.com. This is achieved by configuring the **Auth Logout Success URL** parameter as 'http://www.def.com/nclogin.submit?f\_method=LOGOUT' under authentication of www.abc.com.

In the above scenario, the **Auth Logout Success URL** in www.abc.com is 'http://www.def.com/nclogin.submit?f\_method=LOGOUT'. This assumes that 'nclogin.submit' is configured as the login-processor-path in www.def.com. Similarly for multiple slave domains; you need to configure the same settings in www.def.com for the corresponding next domain and so on.

#### Steps involved in chained logout:

1. User performs a logout on www.abc.com. www.abc.com expires its cookie, and redirects the user to www.xyz.com.
2. www.xyz.com expires its cookie and redirects the user back to www.abc.com
3. www.abc.com redirects the user to perform a logout on www.def.com

4. www.def.com expires its cookie and redirects the user to www.xyz.com
5. www.xyz.com simply redirects the user back to www.def.com, since www.xyz.com's cookie has been expired in step 2.
6. The SSO User Session cookie of all the three domains has been removed from the user's browser.

This process can be extended for more slave domains by simply chaining the logout-success-url configuration in authentication container.

#### Note



If the user does not access the SSO environment within the specified idle timeout, the user's session becomes idle and the user is challenged to provide login credentials to access the SSO environment again.

## Creating an Authorization Policy

---

The **ACCESS CONTROL > Authorization** page allows you to provide custom access across your Web site. This custom access specifies which user/group has access to a specific service.

Access control for a service is configured per URL policy. You configure the access control for the URL key of a service informing which user/group can access that service. This allows you to customize access based on the category of the user/group.

Based on the user identity, the Barracuda Web Application Firewall determines the groups of the user. The groups of the user are determined from the authentication realm specified in the Authentication page. Based on the groups of a user, the Barracuda Web Application Firewall applies access restrictions to services.

### To create an Authorization Policy

1. Specify values for the following fields:
  - **Service** - Specifies the service for which access control to be configured.
  - **Access Control URL Name** - Specifies the name for this access control URL.
  - **Status** - Enables authorization for this service.
  - **URL Match** - This is used to specify the matching criterion for URL field in the Request Header. The URL should start with a "/" and can have only one "\*" anywhere in the URL. A value of /\* means that the ACL applies for all URLs in that domain.
  - **Host Match** - This is used to specify the matching criterion for host field in the Request Header. This is either a specific host match or a wildcard host match with a single "\*" anywhere in the URL. You can enter a partial domain with wildcard (for example: \*.abc.com) but you can not use multiple asterisks.
  - **Extended Match** - Specifies an expression that consists of a combination of HTTP headers and/or query string parameters. Use '\*' to denote "any request", that is, do not apply the Extended Match condition. To build an expression, click the Edit button that appears next to this field and, specify the values for the following fields:
    - **Header Expression**: Specify a valid header expression.
    - **Element Type**: Select the element type from the drop-down list.
    - **Operation**: Select the operation from the drop-down list.
    - **Value**: Specify a valid expression.



- **Concatenate:** Select '**and**' radio-button to add some more expressions to the existing match sequence. Select '**or**' radio-button to replace the existing match sequence.
  - Click **Insert** and then click **Apply** or click **Cancel**. For more on how to write extended match expressions, refer *Extended Match and Condition Expressions*.
  - **Extended Match Sequence** - This parameter is used to specify an order for matching the extended match rule.
  - **Login Method** - Specifies the login method to be used for authenticating the user.
  - **Allow any Authenticated User** - Specifies whether to allow any authenticated user or not. This parameter specifies whether the authenticated user has access to further pages or only those specified in "Allowed Users" or "Allowed Groups."
  - **Allowed Users** - Specifies the list of allowed users to access the URL. **Note:** To get access the URL, the user must be included either in "Allowed Users" or "Allowed Groups".
  - **Allowed Groups** - Specifies the list of allowed groups to access the URL. These are specified as comma (,) separated list.
  - **Auth Not Done URL** - Specifies the URL to which a user is redirected, if the user tries to access a protected URL before being authenticated. If this parameter is not supplied with the URL to redirect, then the user is redirected to a login page generated by the Barracuda Web Application Firewall.
  - **Access Denied URL** - Specifies the URL to which a user is redirected, if an authenticated user does not have access to a requested URL. If this parameter is not supplied with the URL to redirect, then the user is redirected to a login page generated by the Barracuda Web Application Firewall. **Note:** The redirect URLs need not reside in the same service. Also, these pages must be hosted outside the Barracuda Web Application Firewall, typically in the server of the application. The internal Barracuda Web Application Firewall pages cannot be customized.
  - **Send Basic Authentication** - If this parameter is set to 'Yes', the user credentials are converted to HTTP Basic Authentication header and every request that is forwarded to the server contains this header. It uses RFC 2617 to send the credentials. This is useful when using "HTML Form" as "Login Method" and when the server needs to know the user credentials.
2. Click **Add** to add the above settings.
  3. Click **Edit** against the created Authorization policy to modify the settings.
  4. Click **Delete** against the created Authorization policy to delete it.

Typical cases of the server requiring to know the user credentials are:

- To implement single sign on, such that the user is not asked to login once again into the service. The server may require customizing to process the Basic Authentication header, extract the user ID and password, and perform any authentication or authorization required by the service.
- To personalize the home page, the server requires to know the user ID.  
**Note:** HTTP Basic Authentication Headers are sent in clear text, and is not a secure means of exchanging user credentials. The user ID and password are visible in the data packets transmitted to the server. It is recommended that this option is used only when the traffic to the server is encrypted.

## Creating new Authentication Services

---

The first step is to associate an authentication database with a Web service. This database can be internal or an existing Lightweight Directory Access Protocol (LDAP) server or RADIUS authentication database. LDAP is an industry standard that is used by most databases when storing information. LDAP is an open protocol and is based on the X.500 global directory structure. It

supports TCP/IP and any of the application-specific protocols contained in an IP packet. Authentication can be implemented only in HTTP or HTTPS Web services.

The RADIUS protocol is based on a client/server model. The Barracuda Web Application Firewall can potentially operate as a client of a RADIUS server. The client is responsible for passing user information to a designated RADIUS server and then acting on the response that is returned.

A RADIUS server (or daemon) can provide authentication and accounting services to one or more Barracuda Web Application Firewall devices. RADIUS servers are responsible for receiving user connection requests, authenticating users, and then returning all configuration information necessary for the client to deliver service to the users. A RADIUS server is generally a dedicated workstation connected to the network.

## LDAP

LDAP Authentication service identifies a database server supporting the LDAP protocol, which contains a set Authentication service. It is a unique identifier that identifies a set of users, groups and contains mapping between the groups and the users. Configuration of this page allows the Barracuda Web Application Firewall to communicate with an existing LDAP directory server, and authenticate a user.

### To configure an LDAP Authentication Service

1. Specify values for the following fields:
  - **Realm Name** - Enter the name of a realm. A realm is an LDAP complaint database of authorized user and group records. The realm can be located internally or externally on an LDAP server.
  - **Server IP** - Enter the IP address of an external LDAP server used for authenticating users.
  - **Server Port** - Enter the port address of the external LDAP server used for authenticating users. Port 389 is normally used for LDAP.
  - **Secure Connection Type** - Select the type of secure connection to be used by Barracuda Web Application Firewall when querying the LDAP database for user authentication and role retrieval.
  - **Bind DN** - Enter a Distinguished Name (DN) used to bind a user to the LDAP server.
  - **Base DN** - Enter the base DN of the LDAP database used to specify the scope of any LDAP search.
  - **Bind Password** - Enter the password used for binding to the LDAP server.
  - **Login Attribute** - Enter the attributes of an LDAP object used for identifying the user.
  - **Group Name Attribute** - Enter the attributes of an LDAP object used for identifying the name of a group.
  - **Group Filter** - Enter the LDAP filter used to retrieve the list of groups of a user. The maximum allowable characters is 500.
  - **Query For Group** - Select whether to look for the group or to look for individual user names for authentication. Select 'Yes' to enable this to look for group for authentication.
2. Click **Add** to add the above settings.
3. Click **Edit** against the created LDAP service to modify the settings.
4. Click **Delete** against the created LDAP service to delete it.

## RADIUS

RADIUS Authentication service identifies a database server supporting the RADIUS protocol that contains a set of users, groups and mapping between groups and users. This container allows the user to configure the Barracuda Web Application Firewall, to communicate to an existing RADIUS directory server, for authenticating a user.

### To configure an RADIUS Authentication Service

1. Specify values for the following fields:
  - **Realm Name** - Enter the name of a realm. A realm is a RADIUS complaint database of authorized user and group records. The realm can be located internally or externally on a RADIUS server.
  - **Server IP** - Enter the IP address of a server used for authenticating users.
  - **Server Port** - Enter the port address of the RADIUS server used for authenticating users. Port 1812 is normally used for RADIUS.
  - **Shared Secret** - Enter the secret key which is shared between the Barracuda Web Application Firewall and RADIUS server. Minimum value of the key is 6.
  - **Timeout** - Enter the wait timeout in seconds before deciding and re-sending the packet.
  - **Retries** - Enter the number of retrials to send packet before giving up.
2. Click **Add** to add the above settings.
3. Click **Edit** against the created RADUIS service to modify the settings.
4. Click **Delete** against the created RADIUS service to delete it.

## Creating Local Users/Groups

---

The **ACCESS CONTROL > Local Users/Groups** page allows you to create users and groups for internal authentication services. One or more users can be added to each group. One user can belong to multiple groups.

### To add a group

1. Specify values for the following fields:
  - **New Group Name** - Enter a name for the group.
2. Click **Add**.
3. Click **Delete** against the created group to delete it.

### To add a user

1. Specify values for the following fields:
  - **New user name** - Enter a name for the new user. This user will be added to the internal database.
  - **Password** - Enter a password for the user.
  - **User Groups** - Select a group from the right hand side list and click **Add**. Similarly repeat the same to add more groups. To remove a group, select the group and click **Remove**.
2. Click **Delete** against the created user to delete it.

## Allowing/Denying Client Certificates

---

The **ACCESS CONTROL > Client Certificates** page allows you to define allow/deny rules based on Client Certificates. For these settings to take effect, SSL Client Authentication must be enabled on the **Basic > Services** page.

When Client Authentication is turned on, all clients are required to present a certificate to be able to access the Web Site. The certificate is first checked for validity. It must be a valid certificate, must have not expired, and must be signed by a CA which is listed as one of the Trusted Certificates in the service.

Further, even if a certificate is valid and signed by one of the trusted CA certificates, you can reject it based on the certificate attributes. This is useful when you like to revoke an already issued certificate.

Each Allow/Deny rule has the following important attributes:

- A sequence, or order, in which to evaluate the rule.
- A set of attribute matches (like the Certificate Serial number). The attribute can either be a wildcard match (\*, to indicate match any value), or it can be a specific value, matching the certificate's corresponding attribute exactly.
- An action to take when a rule matches the certificate in the request.

When a request is received, the Client certificate is matched one by one against all the Allow/Deny rules. The rules are matched in order of the sequence number, starting from the lowest sequence number. Each attribute in the rule that is not a '\*' is compared against the same attribute in the certificate. When a match is found, the corresponding action (Allow or Deny) is taken, and further comparison with other rules is stopped.

When no rule matches the Client Certificate in the request, the request is allowed by default.

If you require that every Client Certificate be explicitly mentioned before it is allowed, you must create a default Deny rule, with a high sequence number, say, 10000, which has \* for all attributes and the action is deny. This will ensure that every certificate will eventually match this rule the action for which is Deny. Those certificates which must be allowed should be added as Allow rules with lower sequence numbers.

Complex rules can be built by manipulating the Allow/Deny rules. For example, to deny all certificates from the Sales department except one that is identified by the serial number, you would create the following two rules:

- Sequence = 1; Action = Allow; Organizational Unit = Sales; Serial Number = 12345
- Sequence = 2; Action = Deny; Organizational Unit = Sales

Although such complex rules can be built, it is recommended that you allow all certificates that are signed by the trusted CA's certificate, and use the Allow/Deny list only to revoke rights to certificates that have been issued but no longer must be allowed access. Since the serial number is unique among all certificates issued by a single CA, it can be reliably used to identify a certificate. The Common Name (CN) is also usually a good field to identify a certificate that must be revoked.

### To add or edit Allow/Deny Certificate

1. Click **Add** under **Actions**. The **Add Allow/Deny Certificate** page opens.
2. Specify values for the following fields:
  - **Rule Name** - Enter the name that identifies the Allow/Deny certificate rule in the Web Application Firewall.
  - **Status** - Select whether to enable or disable the rule.

- **Sequence** - Enter a number between 1 and 10000, to indicate the order of processing when matching rules. Rules with a lower number are matched first, and attempts to match will stop at the first rule that matches. A rule with sequence number 1 will be matched first, and a rule with sequence number 10000 will be matched last.
  - **Action** - Select the action to take when the rule matches a Client Certificate. Valid values are Allow and Deny.
  - **Country** - Enter the value of the Country field in the certificate's subject. This is a two-letter country code. Use a '\*' to ignore this field while matching.
  - **State** - Enter the value of the State field in the certificate's subject. Use a \* to ignore this field while matching.
  - **Locality** - Enter the value of the Locality field in the certificate's subject. Use a \* to ignore this field while matching.
  - **Organization** - Enter the value of the Organization field in the certificate's subject. Use a \* to ignore this field while matching.
  - **Organizational Unit** - Enter the value of the Organizational Unit field in the certificate's subject. Use a \* to ignore this field while matching.
  - **Common Name** - Enter the value of the Common Name field in the certificate's subject. Use a \* to ignore this field while matching.
  - **Certificate Serial Number** - Enter the value of the Serial Number in the certificate. Use a \* to ignore the serial number while matching. Simple serial numbers as integers as well as in the Hex format are accepted.
3. Click **Add** to add the above configurations.
  4. Click **Edit** against the created rule to modify the settings.
  5. Click **Delete** against the created rule to delete it.



# Monitoring, Logging and Reporting

---

This chapter provides information about the various types of logs and reports available for the Barracuda Web Application Firewall. The following topics are covered:

- *Monitoring Barracuda Web Application Firewall* on page 129
- *Logs* on page 132
- *Reports* on page 136

### Note



For more detailed information about a specific page in the Web interface, view the online help by clicking the question mark icon on the right side of the interface.

# Monitoring Barracuda Web Application Firewall

This section describes the monitoring tasks you can perform from the Web administration interface and from the front panel of the Barracuda Web Application Firewall. This section covers the following topics:

<i>Viewing Performance Statistics .....</i>	<i>129</i>
<i>Health Indicators for Services and Servers.....</i>	<i>129</i>
<i>Monitoring the Health of the Server.....</i>	<i>130</i>
<i>Viewing System Tasks.....</i>	<i>131</i>
<i>Understanding the Indicator Lights.....</i>	<i>131</i>

## Viewing Performance Statistics

The **BASIC > Status** provides an overview of the health and performance of your Barracuda Web Application Firewall, including:

- Traffic statistics, which shows the number of requests for various types of traffic since the last system reset.
- Subscription status of Energize Updates.
- Performance statistics, such as CPU temperature and system load. Performance statistics displayed in red signify that the value exceeds the normal threshold.
- Hourly and daily traffic statistics.

The following table describes the various health indicators displayed for the system:

Performance Indicator	Description
Green	The component is working fine.
Orange	Indicates warning. The component configuration is more or less than the standard limit.
Red	Indicates danger. Needs immediate attention.

## Health Indicators for Services and Servers

The **Basic > Services** page display the health of your Services and Servers.

The following table describes the various health indicators displayed for each of the services and the servers:

Service and Server Health Indicators	Description
Green dot	Service is up; Server is responding.
Orange dot	If multiple servers are configured for a service, the orange dot indicates that at least one of the Servers is up and the Service is running.
Red dot	Service is down; Server is not responding.



## Monitoring the Health of the Server

---

The health of the servers configured on the Barracuda Web Application Firewall can be monitored using the In-Band, Out-of-Band and Application layer health checks.

For detailed descriptions, see the online help by clicking [Edit](#) for the server on the [BASIC > Services](#) page.

### In-Band Health Checks

The Barracuda Web Application Firewall monitors the health of data transmissions of a server. In-Band refers to the user traffic connections. The In-Band Health check policy specifies the In-band server health check parameters. These parameters specify the layer 4 and layer 7 error thresholds. The server connections and responses are monitored for errors. When the errors go over the specified thresholds, the server is marked out-of-service.

Servers in the out-of-service state are disregarded as potential servers for serving content. If other servers are defined to load balance requests, traffic will be routed to the other servers. If only one server is defined, and it is in the out-of-service state, it will result in an error response to the browser.

In-band monitoring is always enabled, and default parameters are provided, so that you need not configure the In-Band health check parameters. If you want, you can modify the default settings.

### Out Of Band Health Checks

The Barracuda Web Application Firewall periodically monitors server health independent of the data transmissions. Out-of-Band refers to those connections made outside of the user-traffic connections. The Out-of-Band health check parameters specify the layer 4 and layer 7 server monitoring.

If the server health check fails, the server is marked as out-of-service. The server continues to be monitored, and when the server health check succeeds, the server's status is reverted to in-service. This is unlike In-Band Health checks, where the server can only be placed in the out-of-service status, but cannot be reverted back as no further user traffic is directed towards an out-of-service server.

#### Note



Once the server is set offline by **Out of Band Health Checks**, it is always fetched back online after the interval (by default 10 seconds) if the server probes succeed.

### Application Layer Health Check

Application Layer Health Check involves making a HTTP request to see if the server is responding correctly. If the server responds correctly, the server is said to be healthy. Otherwise, the server is marked as out-of-service. The settings for Application Layer determine what kind of HTTP request is made (URL, Method, Headers), and how to determine if the response was a good response (Status Code and Match Content String).

#### Note



Application Layer Health Check is valid only if the parameter **Enable OOB Health Checks** is set to **Yes** under the **Out Of Band Health Checks** section.

## Viewing System Tasks

The **Advanced > Task Manager** page provides a list of tasks that are in the process of being performed and also displays any errors encountered when performing these tasks.

Some of the tasks that the Barracuda Web Application Firewall tracks include:

- Cluster setup
- Configuration restoration

If a task takes a long time to complete, you can click the **Cancel** link next to the task name and then run the task at a later time when the system is less busy.

The Task Errors section will list an error until you manually remove it from the list. The errors are not phased out over time.

## Understanding the Indicator Lights

The Barracuda Web Application Firewall has five indicator lights on the front panel that blink when the system processes any traffic.

Figure 10.1 displays the location of each of the lights.

Figure 10.1: Indicator Lights



Table 10.1 describes each indicator light.

Table 10.1: Description of the Indicator Lights

Light	Color	Description
Attack Activity	Red	Blinks when the Barracuda Web Application Firewall blocks possible attacks.
Virus Download	Yellow	Blinks when the Barracuda Web Application Firewall blocks malicious software or virus being downloaded.
Traffic	Green	Blinks when the Barracuda Web Application Firewall processes traffic.
Data I/O	Green	Blinks during data transfer.
Power	Green	Displays a solid green light when the system is powered on.

# Logs

---

The Barracuda Web Application Firewall has comprehensive logging feature to record occurrence of significant events. Events related to http traffic, action of web firewall and user actions are captured and made available to the administrator through the user interface. These log messages enable the system administrators to:

- obtain information about the Barracuda Web Application Firewall traffic and performance
- analyze logs for suspicious activity
- troubleshoot the problems

The following types of logs are available in Barracuda Web Application Firewall:

- **Web Firewall Logs** - All the actions/events on the web firewall are logged under web firewall logs. These logs help the administrator to analyze the traffic for suspicious activity and also fine tune the web firewall policies.
- **Access logs** - All web traffic activities are logged under the access logs. These logs helps the administrator to obtain information about the web site traffic and performance.
- **Audit logs** - All administration and configuration activities of the administrator are captured under the audit logs. This information can be used for audit purpose.

Every log in the Barracuda Web Application Firewall has a level associated with it, which indicates the severity of the logs. An administrator can configure what level of logs should be recorded and control the volume of logs been persisted.

Apart from persisting the logs in the internal storage, Barracuda Web Application Firewall also has the capability to configure an external syslog server, for persistent external storage.

## Web Firewall Logs

---

The **BASIC > Web Firewall Logs** page allows you to view the generated log messages stored in a syslog server or the buffer. Use the built-in filters to quickly locate specific types of log entries. Click **Preferences** to set the number of messages to be displayed per page.

### Note



The processing of log requests takes longer in the following two cases:

1. If invalid filter criteria is entered.
2. If the log matching the filter criteria, lies at the end of the log database.

The processing of log requests takes longer in the following two cases:

- If invalid filter criteria is entered.
- If the log matching the filter criteria, lies at the end of the log database.

Click **Back** or **Forward** to toggle between different pages of log entries. The logging of Web Firewall Logs are enabled by default.

Click **Export to CSV** to save all the logs in .csv format to your computer.

### To configure the log level for Web Firewall Logs

1. From the **BASIC > Services** page, click **Edit** against the service.

2. On the Edit page, under the **Basic Security** section, select the **Web Firewall Log Level** from the drop-down list. A lower level signifies lesser information logging.
3. Click **Save Changes** to save the above settings.

## Access Logs

---

The **BASIC > Access Logs** page allows you to view the generated log messages stored in a syslog server or the buffer. Use the built-in filters to quickly locate specific types of log entries. Click **Preferences** to set the number of messages to be displayed per page.

### Note



The processing of log requests takes longer in the following two cases:

1. If invalid filter criteria is entered.
2. If the log matching the filter criteria, lies at the end of the log database.

Click **Back** or **Forward** to toggle between different pages of log entries. Access Logs are enabled by default. To disable Access Logs, on **BASIC > Services** page, click **Edit** against the service and set the **Enable Access Logs** parameter to **Off**.

Click **Export to CSV** to save all the logs in .csv format to your computer.

### To disable Access Logs

1. From the **BASIC > Services** page, click **Edit** against the service.
2. On the Edit page, under the **Services** section, set the **Enable Access Logs** to “Off”.
3. Click **Save Changes** to save the above settings.

## Audit Logs

---

The **BASIC > Audit Logs** page allows you to view the generated log messages stored in a syslog server or the buffer. Use the built-in filters to quickly locate specific types of log entries. Click **Preferences** to set the number of messages to be displayed per page.

### Note



The processing of log requests takes longer in the following two cases:

1. If invalid filter criteria is entered.
2. If the log matching the filter criteria, lies at the end of the log database.

Click **Back** or **Forward** to toggle between different pages of log entries.

Click **Export to CSV** to save all the logs in .csv format to your computer.

In the following situations there won't be any logout logs in Audit Logs:

- When Barracuda Web Application Firewall is restarted because other critical processes crashed, the current existing sessions won't be logged out. So audit logs will not have corresponding logout logs.
- When maintenance command is executed by user or by Barracuda Web Application Firewall, the current existing sessions won't be logged out. Hence audit logs will not have corresponding logout logs.

In the following situations there will not be any login logs in Audit Logs:

- When maintenance command is executed by user or by Barracuda Web Application Firewall, a new login session will be created in maintenance mode, but it won't be logged.

## Search Logs

---

You can use filters and the search option to quickly locate specific types of log entries.

### To use the search criteria

1. Select the filter column and appropriate operator from the drop-down list and enter the value. Click **Search**, to display the audit logs pertaining to the filter specified.
2. Click '+' to add more search fields and '-' to remove it. Multiple search criterion can be specified, which will result in an 'And' combination of the filters.
3. Specify the complete timestamp while searching for the log messages generated within the specified period.
4. While using multiple search criterion, same fields cannot be specified more than once.
5. Click **Back** or **Forward** to toggle between different pages of log entries.
6. Click **Export to CSV** to save all the logs in .csv format to your computer.
7. You can specify regular expressions (Regexp) for some selected fields.

## Export Logs

---

The **ADVANCED > Export Logs** contains all the parameters and resources for configuring the log policy on a service. The Barracuda Web Application Firewall uses this policy and generates the logs of standard and custom formats, and then exports them to the destined servers.

To export the Web log messages to an FTP server, you need to configure the FTP Web Logs.

### Note



Usually, filtered logs are saved to .csv format. If no filter is applied then all the logs are transported to .csv format.

## Syslog

The **ADVANCED > Export Logs** page is a standard UNIX/Linux tool for sending remote system logs and is available on all UNIX/Linux systems. Syslog servers are also available for Windows platforms from a number of free and premium vendors. Barracuda Networks has tested with a Windows freeware syslog server from Kiwi Enterprises ([www.kiwisyslog.com](http://www.kiwisyslog.com)). Barracuda Networks makes no guarantees that your Barracuda Web Application Firewall will be completely compatible with this syslog server.

### To configure System Logs

1. From the **ADVANCED > Syslog** page, under the **System Logs Syslog Configuration** section, enter the IP address to send syslog data related to generic system events.
2. Click **Save Changes** to save the above settings.

3. Click **Monitor Syslog** to view the system logs.

**Note**



This syslog data appears on the local0 facility with events at various priority levels on the specified syslog server.

**To configure Application Logs**

1. From the **ADVANCED > Export Logs** page, under the **Application Logs Syslog Configuration** section, enter the IP address to send syslog data related to structure Application Logs.
2. Click **Save Changes** to save the above settings.

**Note**



Different facilities are specified to distinguish between the log type as all the log types go to the same syslog server. They are: Web Firewall Logs: local0; Web Logs: local1; Audit Logs: local2.

## FTP Web Logs

FTP Web Logs specifies general information about the server that will host the Web logs and the format of log events being transported.

**To configure FTP Web Logs**

1. Specify values for the following fields:
  - **Server IP** - Enter the IP address of the remote server. This server would collect the Web logs. However, it is mandatory that the FTP server should be running on this machine.
  - **Server Port** - Enter the port on which the FTP server is running.
  - **Server Username** - Enter the user name to login to the FTP server and to export the Web logs. This should be a valid FTP user name that the remote machine accepts for file transfers.
  - **Server Password** - Enter the password associated with the user value entered in the 'Server Username' field.
  - **Destination Directory Path** - Enter the name of the target directory. This specifies where the logs should be stored. This directory should be created manually in the FTP server, before sending the Web logs.
  - **Logs Format** - Select the format in which the logging should occur.
  - **Logs Custom Format** - Enter the values for 'Custom Format'. For Common Log Format, NCSA Extended Format and W3C Extended Format, values are already defined.
2. Click **Save Changes** to save the above configurations.

# Reports

---

This section is intended for users who need to configure and generate reports. It describes the procedures for generating reports on various categories. The generated reports help the system administrators in their day-to-day security management and statistical analysis of the log messages. The reporting feature augments the management capabilities available with the Barracuda Web Application Firewall. Using this module, reports can be generated based on all the logged information.

Barracuda Web Application Firewall reports are broadly classified under four groups based on their functions. Each group contains a predefined set of report types. Users have to select a **Report Group** and a report type from the corresponding **Report Type** drop-down list. These four groups are described as follows:

## Security Reports

These are the reports pertaining to the Web attack activity performed by the Barracuda Web Application Firewall. **Note:** The report types namely: Top Clients by Bandwidth, Top URL by Bandwidth, Top Domains by Bandwidth, Top Services by Bandwidth and Top Entry Pages will not include data corresponding to URLs containing files with extension jpg, png, gif, ico, css, js.

## Audit Reports

These are the reports pertaining to the server details and the login/logout activities performed by different user roles.

## Config Summary Reports

These are the reports pertaining to:

- The performance of the Barracuda Web Application Firewall features such as Load Balancing, Rate Control, Learning, etc.
- The details of the digital certificates like its issuing date, expiry date and the services it is associated with.
- The details of the accounts and the users belonging to those accounts, the privileges assigned to them, the operations they can perform, etc.

## PCI Reports

These are the reports pertaining to PCI (Payment Card Industry) standards and they display:

- The combined details of the PCI attacks such as top attacking Clients and top attacked Services, Domains and URLs.
- The details of the PCI directives and whether the Barracuda Web Application Firewall satisfies those directives.

---

## Generating Reports

Use the **BASIC > Reports** page to choose from 12 different reports that can help you keep track of activity performed by the Barracuda Web Application Firewall. You can either generate a report on demand or configure the Barracuda Web Application Firewall to automatically generate the reports on a daily, weekly or monthly basis and email the reports to specific email addresses.

Reports can be anchored on user-activity, content or actions. For detailed descriptions, see the online help for the [BASIC > Reports](#) page.

**Note**



Reports run immediately can potentially consume too many system resources on the Barracuda Web Application Firewall. Due to this potential hazard, reports over 7 days in length can only be generated through email.



# Chapter 11

## Advanced Concepts

---

This chapter describes the advance security configurations you can perform on the Barracuda Web Application Firewall. The following topics are covered:

- *Deployment* on page 139
- *Security* on page 142

### Note



For more detailed information about a specific page in the Web interface, view the online help by clicking the question mark icon on the right side of the interface.

# Deployment

---

The **ADVANCED > Advanced IP Config** pages allows you to configure advanced IP configurations for the Barracuda Web Application Firewall. This section covers the following topics:

<i>Multiple IP Address Configuration</i> .....	139
<i>Static Routes</i> .....	139
<i>Interface Routes</i> .....	140
<i>VLAN (Virtual Local Area Network)</i> .....	140

## Multiple IP Address Configuration

---

The **Multiple IP Address Configuration** section allows you to add virtual interface(s) to the physical port (WAN or LAN or MGMT) used to communicate with the servers. This interface is a logical exit point that allows traffic to safely and securely travel between the Barracuda Web Application Firewall and the servers. This lists all service IP addresses along with the virtual interfaces.

### To configure Multiple IP Addresses

1. Specify values for the following fields:
  - **IP/Network Address.** Enter an IP address to communicate with the servers.
  - **Netmask.** Enter an associated netmask for this interface.
  - **Network Interface.** Select the interface over which communication will be transmitted. To do this, select either WAN or LAN or MGMT. (Back-end traffic is normally over LAN.)
2. Click **Add** to add the above configurations.

## Static Routes

---

The **Static Routes** section allows you to create a static route to specify the exact route to a remote network. This allows you to route an interface that is located on a different subnet.

### To configure Static Routes

1. Specify values for the following fields:
  - **IP/Network Address** - Enter the IP address of the destination in a route entry. An address of 0.0.0.0 indicates this route applies to any destination IP address.
  - **Netmask** - Enter the mask for the route entry. The destination and the mask together define the set of destinations that can be reached through this route. An address of 0.0.0.0 indicates this route applies to any destination mask.
  - **Gateway Address** - Enter the IP address of the network gateway.
2. Click **Add** to add the above configurations.
3. Click **Bulk Edit** to perform multiple changes to a list of configuration settings in one step. The above values is translated into CSV (Comma-Separated Values) format and displayed in the Bulk Edit window. From there, you can manually add, modify or remove as many rows as you desire and click **Save Changes**.

## Interface Routes

---

The **Interface Routes** section allows you to create an interface route to specify the interface to use for remote network. This is useful in the bridge mode where the service IP address is not owned by the Barracuda Web Application Firewall.

### To configure Interface Routes

1. Specify values for the following fields:
  - **IP/Network Address** - Enter an IP address which has to be routed through the interface.
  - **Netmask** - Enter an associated netmask for this interface route.
  - **Network Interface** - Select the interface over which communication will be transmitted. To do this, select either WAN or LAN or MGMT.
2. Click **Add** to add the above configurations.

## VLAN (Virtual Local Area Network)

---

### Overview

A VLAN (Virtual Local Area Network) is a logical construct, similar to a LAN, which defines a broadcast domain. In a LAN, all hosts belonging to the LAN must be physically connected to the same switch, whereas, in a VLAN, the hosts can be a part of the broadcast domain even when they are not connected to the same switch. Further, the ports on a switch with VLAN abilities can itself be divided into multiple independent broadcast domains. Network reconfiguration can be done through software instead of physically relocating devices.

When a VLAN spans multiple switches, the VLAN traffic is routed over **trunk ports** on the switches. The link between two trunk ports is known as the **trunk link**. Usually a trunk link is implemented between fast switch ports on two different switches using a crossover cable. A VLAN might have 3 ports on one switch, and 7 ports on another, the inter-switch traffic is routed on the trunk ports.

Traffic for multiple VLANs can be transferred across a single trunk link. This is made possible through **VLAN tagging**, which tags Ethernet packets with **VLAN IDs**, denoting the VLAN to which the packet belongs. As against this, VLAN ports are those ports on the VLAN switch which belong to a single VLAN and hence only see the broadcast traffic of that VLAN.

### VLAN Configuration

To be able to route to a VLAN via any one of the interfaces – WAN, LAN or MGMT, a VLAN interface must be added on to that interface. This interface receives the broadcast traffic on the VLAN. It is also used to route VLAN traffic to the VLAN. Adding a VLAN interface involves specifying the VLAN ID, apart from the IP address and subnet mask for the interface. Based on the destination IP address and subnets of network packets, the Barracuda Web Application Firewall routes the packets to the appropriate VLAN interface.

Adding a VLAN interface makes the Barracuda Web Application Firewall VLAN-aware for that VLAN. This enables it to perform explicit VLAN tagging functions for traffic being routed to the VLAN as well as removing VLAN tagging when routing packets received from the VLAN to non-VLAN networks.

For example, if all the Real Servers reside in VLAN 100, then the LAN port may be connected to a port on the VLAN switch belonging to VLAN 100. Correspondingly a VLAN interface must be added

to the LAN interface with VLAN ID 100 and having an available IP address belonging in the VLAN's broadcast domain.

To add a VLAN interface, navigate to the **ADVANCED > Advanced IP Config** page and provide the relevant information in the VLAN Configuration module.

### To configure a VLAN interface

1. Specify values for the following fields:
  - **VLAN Name** - Enter the name of the VLAN.
  - **VLAN ID** - Enter a number in the range 1 to 4094 to uniquely identify the VLAN.
  - **VLAN Interface** - Select the port over which communication will be transmitted. Select either WAN, LAN or MGMT.
  - **VLAN IP Address** - Enter the IP address associated with the VLAN ID.
  - **VLAN IP Subnet** - Enter the associated subnet mask for this interface.
2. Click **Add** to add the above configurations.

You can also configure SNAT and ACLs for the newly created VLAN interface on the **ADVANCED > Network Firewall** page.

### Routing to Multiple VLANs over an Interface

If any interface on the Barracuda Web Application Firewall has to route to multiple VLANs, it must be connected to the VLAN switch via a trunk (or hybrid) link, since multiple VLAN traffic can only be transported over trunk links. In order to route to multiple VLANs via any of the interfaces, a VLAN interface needs to be added to that physical interface for each of the VLAN to which the interface will communicate with. If the Real Servers are distributed across multiple VLANs, say 100, 105, and 111, then the LAN port must be connected to a trunk port on the VLAN switch. A VLAN interface must be added for each of these VLANs on the LAN interface with the corresponding VLAN IDs, 100, 105 and 111. This allows the Barracuda Web Application Firewall to route to the correct VLAN by inserting appropriate VLAN IDs before forwarding on to the trunk link.

### Bridge Mode

In Bridge mode, if VLANs are being used, both the LAN and WAN ports must be on the same VLAN and a corresponding VLAN interface must be added on either the WAN or LAN interface. A configuration in which the LAN and WAN are connected to different VLANs is not currently supported in Bridge mode. If the MGMT port is part of one or more VLANs, then VLAN interfaces must be added on to the MGMT port for the respective VLANs.

# Security

---

The Barracuda Web Application Firewall defragments, normalizes, and decodes all incoming requests; examines them for validity and correct formation; and only allows properly formatted and RFC-compliant requests to pass through. Known data patterns of malicious activity are blocked and invalid input embedded in headers, forms, and URLs is stopped.

The Barracuda Web Application Firewall uses regular expressions (regex) to define the data type patterns. (See *Extended Match and Condition Expressions* on page 214 for guidelines on how to write regular expressions.) The pattern-match engine recognizes the lexical patterns in text and compares the given input against the target data type patterns. For example, the following is the default regex pattern for a Visa credit card:

```
4[[:digit:]]{12}|4[[:digit:]]{15}
```

You can define a new regex pattern by configuring additional data type. For example, if you are looking for <script> in the HTML tag, then your regex is <\script\s>.

The following topics discuss the different data patterns used for advanced security. In addition, a pattern can also be associated with an algorithm. The algorithm is then run on all strings matching the regular expression to determine whether they actually conform to this pattern.

This section covers the following topics:

<i>Creating Identity Theft Patterns .....</i>	<i>142</i>
<i>Using Custom Patterns .....</i>	<i>143</i>
<i>Creating Attack Types .....</i>	<i>143</i>
<i>Creating Input Types .....</i>	<i>144</i>
<i>Creating Custom Parameter Class .....</i>	<i>145</i>
<i>Creating a Customized Response Page for Errors .....</i>	<i>145</i>
<i>Creating Session Identifiers .....</i>	<i>146</i>
<i>Creating Rate Control Pool .....</i>	<i>146</i>
<i>Virus Protection for File Uploads .....</i>	<i>148</i>
<i>View Internal Patterns .....</i>	<i>150</i>
<i>Configuring URL Policy .....</i>	<i>150</i>
<i>Configuring FTP Security .....</i>	<i>152</i>
<i>Configuring Session Tracking .....</i>	<i>153</i>
<i>Policy Tuner .....</i>	<i>153</i>

## Creating Identity Theft Patterns

---

The **ADVANCED > Libraries > Identity Theft Patterns** allows you to add customized Identity Theft data type apart from the default data types that are available under **Advanced > View Internal Patterns** page. One or more "patterns" which define the format of the data type can be added to each group.

Identity theft is the unauthorized collection and use of your personal information like name, address, telephone number, credit card, and so on, for unauthorized and malicious purposes. Your personal data like Social Security number, bank account or credit card number, telephone calling card number, and other valuable identifying data can be used to personally profit them at your expense, if they fall into the wrong hands.

### To create customized Identity Theft Data Type

1. Enter the name for the **New Group** and click **Add**.
2. Against the newly created group click **Add Pattern**.

3. Specify values for the following fields:
  - **Pattern Name** - Enter a name for the pattern.
  - **Status** - Select whether to enable or disable this pattern. Only patterns with status set to "On" are used for actual pattern matching.
  - **Pattern Regex** - Enter the regular expression of the pattern. It recognizes the lexical patterns in text.
  - **Pattern Algorithm** - Select the algorithm for the pattern from the drop-down list.
  - **Case Sensitivity** - Select whether the pattern regular expression is to be treated as case sensitive or case insensitive from the drop-down list.
  - **Pattern Description** - Enter the description for the pattern used. For example, Visa credit card pattern. This indicates that the pattern used here is the visa credit card pattern.
4. Click **Add** to add the above configurations.

The added identity theft pattern becomes available under **SECURITY POLICIES > Data Theft Protection**. Select the **Identity Theft Type** as "<CUSTOM>" and click the **Custom Identity Theft Type**, which lists out all the newly added identity theft patterns.

## Using Custom Patterns

---

A pattern describes a data type format. When a data type is bound to an attack, all user input matching the ACL is scanned for strings corresponding to any enabled pattern. The pattern format is defined with a regular expression. In addition, a pattern can also be associated with an algorithm. The algorithm is then run on all strings matching the regular expression to determine whether they actually conform to this pattern.

Similarly, when a data type is bound to a data theft element, all server pages are scanned for strings matching any enabled pattern in the data type.

## Creating Attack Types

---

The **ADVANCED > Libraries > Attack Types** allows you to add customized Attack data type apart from the default data types that are available under **Advanced > View Internal Patterns** page. One or more "patterns" which define the format of the data type can be added to each group.

Attack is request based while identity theft is response based. You can define the list of items that a request should not contain. If a request contains the patterns described here, then the request will be dropped. For example, the attack can be SQL injection or OS command injection or cross site scripting.

### To create customized Attack Type

1. Enter the name for the **New Group** and click **Add**.
2. Against the newly created group click **Add Pattern**.
3. Specify values for the following fields:
  - **Pattern Name** - Enter a name for the pattern.
  - **Status** - Select whether to enable or disable this pattern. Only patterns with status set to "On" are used for actual pattern matching.
  - **Pattern Regex** - Enter the regular expression of the pattern. It recognizes the lexical patterns in text.

- **Pattern Algorithm** - Select the algorithm for the pattern from the drop-down list.
- **Case Sensitivity** - Select whether the pattern regular expression is to be treated as case sensitive or case insensitive from the drop-down list.
- **Pattern Description** - Enter the description for the pattern used. For example, Visa credit card pattern. This indicates that the pattern used here is the visa credit card pattern.

4. Click **Add** to add the above configurations.

The added attack type pattern becomes available under **ADVANCED > Libraries > Custom Parameter Class > Add Custom Parameter Class > Custom Blocked Attack Types** as check box(es), which is enabled by default. Also it is available under **SECURITY POLICIES > Parameter Protection > Custom Blocked Attack Types** as check box(es), which is to be manually selected.

## Creating Input Types

---

The **ADVANCED > Libraries > Input Types** allows you to add customized input type apart from the default input types that are available under **Advanced > View Internal Patterns** page. One or more "patterns" which define the format of the input type can be added to each group.

Includes a collection of pre-defined and custom input data types, which can be used to validate HTTP Request parameters. This validates the inputs entered in the fields available in the forms used. Most of the attacks can be prevented by properly validating input parameter values against the expected input. Input Type validation enforces the expected value type as opposed to looking for malicious values. Values of configured parameters are validated against the specified Input Type and requests with failed validations are detected as intrusions and blocked.

Input Types are defined using reg-ex patterns. Input Types for alpha-numeric strings, credit card, date and positive-long-integer are provided by default. Custom Input Types can be created and used in the validation.

### To create customized Input Type

1. Enter the name for the **New Group** and click **Add**.
2. Against the newly created group click **Add Pattern**.
3. Specify values for the following fields:
  - **Pattern Name** - Enter a name for the pattern.
  - **Status** - Select whether to enable or disable this pattern. Only patterns with status set to "On" are used for actual pattern matching.
  - **Pattern Regex** - Enter the regular expression of the pattern. It recognizes the lexical patterns in text.
  - **Pattern Algorithm** - Select the algorithm for the pattern from the drop-down list.
  - **Case Sensitivity** - Select whether the pattern regular expression is to be treated as case sensitive or case insensitive from the drop-down list.
  - **Pattern Description** - Enter the description for the pattern used. For example, Visa credit card pattern. This indicates that the pattern used here is the visa credit card pattern.
4. Click **Add** to add the above configurations.

The added input type pattern becomes available under **ADVANCED > Libraries > Custom Parameter Class > Add Custom Parameter Class**. Select the **Input Type Validation** as "<CUSTOM>" and click the **Custom Input Type Validation**, which lists out all the newly added input types.

## Creating Custom Parameter Class

---

The [ADVANCED > Libraries > Custom Parameter Class](#) allows you to add customized parameter classes apart from the internal parameter class that are available under [ADVANCED > View Internal Patterns](#) page. One or more "patterns" which define the format of the data type can be added to each group.

### To create customized Parameter Class

1. Click [Add Custom Parameter Class](#). The **Add Custom Parameter Class** dialog box appears.
2. Specify values for the following fields:
  - **Name** - Enter the name for this custom parameter class.
  - **Input Type Validation** - Select the expected type for the configured parameter from the drop-down list. Most of the attacks could be prevented by properly validating input parameter values against the expected input.
  - **Custom Input Type Validation** - Select the expected custom input data type for the configured parameter. You need to create your own custom types.
  - **Denied Metacharacters** - Enter the metacharacters to be denied in this parameter value.
  - **Blocked Attack Types** - Select the check box(es) to detect malicious patterns in the configured parameter. An intrusion is detected when the value of the configured parameter matches one of the specified Attack Types and the request is blocked.
  - **Custom Blocked Attack Types** - Select the custom attack type check box(es) to be used to detect the intrusions.
3. Click [Add](#) to add the above configurations.

The added custom parameter class becomes available under [WEBSITES > Web Site Profile > URL Profile > Param Profile](#). When you add/edit a param profile, select the [Parameter Class](#) as "<CUSTOM>" and click the [Custom Parameter Class](#), which lists out all the newly added custom parameter classes.

## Creating a Customized Response Page for Errors

---

The [ADVANCED > Libraries > Response Pages](#) allows you to create a customized HTML response page for HTTP requests that violate security policies on the Barracuda Web Application Firewall. You can either edit the available default response page or add customized response pages that can be shared among multiple services.

### To create customized Response Pages

1. Click [Add Response Page](#). The **Add Response Page** dialog box appears.
2. Specify values for the following fields:
  - **Response Page Name** - Enter the name for this response page.
  - **Status Code** - Enter the HTTP status code for this response page.
  - **Headers** - Enter the response headers for this response page.
  - **Body** - Enter the response body for this response page.
3. Click [Add](#) to add the above configurations.

The added custom response page becomes available under [WEBSITES > Allow/Deny](#). When you add a new URL ACL, select the [Deny Response](#) as "Response Page" and click [Response Page](#), which lists out all the newly added response pages. Also it is available under [SECURITY POLICIES > Action](#)



**Policy.** Click **Edit** against any action policy, the **Response Page** lists out all the newly added response pages.

## Creating Session Identifiers

---

Session Identifiers allows the Barracuda Web Application Firewall to recognize the session information from the requests and responses.

### To create customized Session Identifiers

1. Click **Add Session Identifiers**. The **Add Session Identifiers** dialog box appears.
2. Specify values for the following fields:
  - **Session Identifier Name** - Enter the name of the new session identifier.
  - **Session Token Name** - Enter the session token name
  - **Session Token Type** - Select the session token type from the drop-down list
  - **Session Token Start Delimiter** - Enter the start delimiter for the session.
  - **Session Token End Delimiter** - Enter the end delimiter for the session.
3. Click **Add** to add the above configurations.

The added session identifiers becomes available on the **WEBSITES > Advanced Security** page under **Session Tracking**. Click **Edit** against any session tracking policy, the **Session Identifiers** lists out all the newly added session identifiers.

## Creating Rate Control Pool

---

Rate Control allows you to configure the number of connections allowed from any specific IP address. When the number goes over the Rate Control threshold, the Barracuda Web Application Firewall blocks further connections.

The Rate Control policy defines shareable policy for controlling the rate of request to a Web application. A Rate Control Pool specifies the maximum number of Active Requests and Client Backlogs along with a set of Preferred Clients. A Preferred Client specification defines a range of IP addresses and an associated weight. The Barracuda Web Application Firewall uses these weights to perform a weighted round robin scheduling between queues when forwarding requests to the application server from the rate control pool. Weights range from 1-100, 1 being the lowest and 100 being the highest.

### Before you set up a Rate Control Pool

Consider the following scenarios before you set up a rate control policy:

1. What is the maximum simultaneous requests that can be served by the resource being protected. This determines the Max Active Requests setting.
2. What, if any, are the bonafide gateways and mega-proxies that will be accessing the protected resources. If they proxy client requests, assign a suitable weight to the proxy IP and if they relay a set of client IP addresses, then assign a weight to the range of IP addresses.
3. What is the maximum queue that you should allow for IP addresses not defined in **Step 2**. This defines the Max Unconfigured Client Backlog setting.

## After you set up a Rate Control Pool

Setting up a Rate Control policy by considering the above steps, helps in:

1. **Step 3** above helps in throttling back attackers flooding the application with DoS attacks, since their requests get queued (slowed down) for weighted round robin scheduling, though it does not provide the ability to completely block them out.
2. Helps in protecting “Load Sensitive” applications, such as, search, DBMS intensive applications, etc., from application DoS attacks.
3. Helps in allowing bonafide gateways and mega-proxies higher access.

## Scheduling algorithm for Rate Control Pool

The scheduling algorithm between queues is weighted round robin. Implicitly, the weight of each Unconfigured client queue is 1. For example, a Preferred client is defined with weight 5 and at a given time the Barracuda Web Application Firewall has queues for 2 Unconfigured clients with a few requests in each. The Barracuda Web Application Firewall will serve 1 request each from the Unconfigured clients queue and then serve 5 requests from the Preferred client queue and so on.

The rate control policies can be specified per service or per URL policy. Rate Control Pools are defined on the [ADVANCED > Libraries](#) page. These rate control pools are globally shareable among services or among URL policies or both. Once defined, they can be bound to multiple services on the [BASIC > Services](#) page, when you [Edit](#) a service. Also they can be bound to multiple URL policies on the [WEBSITES > Advanced Security](#) page, when you [Edit](#) a URL policy.

### To add a Rate Control Pool

1. From the [ADVANCED > Libraries](#) page, click [Add Rate Control Pool](#) under the **Rate Control Pool** section. The **Add New Rate Control Pool** dialog box appears.
2. Specify values for the following fields:
  - **Rate Control Pool Name** - Enter a name for the new rate control pool.
  - **Maximum Active Requests** - Enter the maximum number of Active Requests processed at a given time by the Barracuda Web Application Firewall. An active request is a request which has not fully completed. For example, the TCP connection that has not been closed down.
  - **Max per client backlog** - Enter the number of requests per client IP that will be queued when the Barracuda Web Application Firewall has reached the Maximum Active Requests limit. For example, if “Max Per Client Backlog” is set to 32 and the Barracuda Web Application Firewall is processing the default 100 “Maximum Active Requests”, then for any given client IP, the Barracuda Web Application Firewall will queue upto 32 requests. Any requests after that will be dropped until a request is deleted from the queue.
  - **Max Unconfigured Clients** - Enter the maximum number of Unconfigured Clients. All clients which are not Preferred Clients are Unconfigured Clients. For each unique client IP, the Barracuda Web Application Firewall will maintain an individual backlog queue. For example, if “Max Unconfigured Clients” is set to 100 and “Max Per Client Backlog” is set to 32, the Barracuda Web Application Firewall will maintain 100 queues each with 32 pending requests, a total of 3200 pending requests.
3. Click [Add](#) to add the above configurations.

Click [Add Preferred Clients](#) against the pool to add the range of IP addresses to that pool. Preferred clients is a list of client IP addresses which will be treated in a preferential manner. If the preferred client queue represents a range of IP addresses, the queue for the preferred client will contain the requests from all the clients falling within that range.

**To create preferred clients:**

1. Click **Add Preferred Clients**, under **Options**. The **Add Preferred Client** dialog box appears.
2. Specify values for the following fields:
  - **Client Weight Name** - Enter the name for the client weight.
  - **Status** - Sets the status of the preference. Enabling this makes the client IP range as preferred list of IP addresses.
  - **Preferred Client IP Range** - Enter the IP address or the range of IP addresses (For example: 10.0.0.1 – 10.0.0.10) which will be treated in a preferential manner. Preferred Client is an IP address or a range of IP addresses with an associated weight. Each Preferred client also gets a separate queue with number of entries equal to that defined in "Max Per Client Backlog", times the weight. For example, if "Max Per Client Backlog" is set to 32 and preferred client "Weight" is set to 5, then the queue size will be 32 x 5.
  - **Weight** - Enter the weight for the range of IP addresses. These IP addresses are evaluated in the order of their weights; the higher the weight the higher the precedence (1 is the lower priority and 100 is the higher priority).
3. Click **Add** to add the above configurations.
4. Click **Edit** against any Rate Control Pool to modify the configuration.
5. Click **Delete** to delete the created Rate Control Pool from the list.

## Virus Protection for File Uploads

---

The Barracuda Web Application Firewall integrates anti-virus software into its Web firewall engine to scan incoming requests for virus signatures, specifiable on a per URL basis. Files containing viruses may be uploaded by malicious users or by non-malicious users who are unaware that the files they are uploading have been infected. Often anti-virus software is either not installed or configured correctly to scan the right folders on Web servers. Sometimes uploaded files may not be stored in a traditional file system, for example, Microsoft SharePoint application stores uploaded Microsoft Office files in a database. Since host based anti-virus software can only work by inspecting files on the file system, it is rendered ineffective in such scenarios. Thus, if users are able to upload infected files on Web servers, the later can unintentionally become a viral proliferation tool. This is especially important for content management systems (CMS) which are used as a repository for sharing documents between an online community or within an organization.

The integration with anti-virus software allows the Barracuda Web Application Firewall to perform virus scanning at the network periphery. Requests containing viruses are blocked from reaching the Web servers. The Barracuda Web Application Firewall updates the virus signature database automatically via the Energize Updates, administrators do not have to concern themselves with manually updating them. Virus definitions are updated on a regular basis (Daily by default).

Downloads are incremental, so network bandwidth is not hogged by the download process. The anti-virus software can detect viral content within popular document formats as given in *Table 11.1*.

*Table 11.1: File formats protected by Barracuda Web Application Firewall*

File Type	Formats
Executables	<ul style="list-style-type: none"> <li>• Aspack (2.12)</li> <li>• UPX (all versions)</li> <li>• FSG (1.3, 1.31, 1.33, 2.0)</li> <li>• Petite (2.x)</li> <li>• PeSpin (1.1)</li> <li>• NsPack</li> <li>• wwpack32 (1.20)</li> <li>• MEW</li> <li>• Upack</li> <li>• Y0da Cryptor (1.3)</li> </ul>
Mail Files	Barracuda Web Application Firewall protects almost every mail file format including attachments.
Archives and Compressed files	<ul style="list-style-type: none"> <li>• Zip (+ SFX)</li> <li>• RAR (+ SFX)</li> <li>• Tar</li> <li>• Gzip</li> <li>• Bzip2</li> <li>• MS OLE2</li> <li>• MS Cabinet Files (+ SFX)</li> <li>• MS CHM (Compiled HTML)</li> <li>• MS SZDD compression format</li> <li>• BinHex</li> <li>• SIS (SymbianOS packages)</li> <li>• Autolt</li> </ul>
Documents	<ul style="list-style-type: none"> <li>• MS Office and MacOffice files</li> <li>• RTF</li> <li>• PDF</li> <li>• HTML</li> </ul>
Others	<ul style="list-style-type: none"> <li>• JPEG (exploit detection)</li> <li>• RIFF (exploit detection)</li> <li>• uuencode</li> <li>• ScrEnc obfuscation</li> <li>• CryptFF</li> </ul>

#### Note



It is recommended that virus checking be only enabled for URLs which allow file uploads, since virus checking is a performance intensive task, involving matching the request contents against many virus signatures.

We recommend that the **Automatically Update** setting for your virus definitions be set to **Daily** on the **ADVANCED > Energize Updates** page so your Barracuda Web Application Firewall receives the latest definitions as soon as new threats are identified by the Barracuda Central.

#### To enable anti-virus on a URL

Consider “<http://www.example.com/cgi/upload.cgi>” is the URL for which you want to enable the anti-virus scan. Do the following:

1. From the **WEBSITES > Advanced Security** page, under **Advanced Security** section, click **Edit** against the URL <http://www.example.com/cgi/upload.cgi>.
2. Set **Enable Virus Scan** parameter to ‘Yes’ and click **Save Changes**.

## View Internal Patterns

---

The **ADVANCED > View Internal Patterns** page displays the details of different regex patterns grouped under Identity Theft Patterns, Attacked Types, Input Types and Parameter Classes. For detailed information on Identity Theft Patterns, Attacked Types, Input Types and Parameter Classes refer *Security* on page 142.

#### Copy and Modify a Predefined Pattern Group

The patterns exhibited under each pattern group are predefined and hence cannot be modified by the user. The Barracuda Web Application Firewall provides Copy function that helps you to copy the predefined patterns of a particular group and modify it as required.

#### To copy and modify a predefined Pattern Group

1. Click **Copy** against the group that you want to copy, the **Copy** pop-window appears.
2. In **New Group** field, specify a new name for the group and click **Paste**.
3. Go to **ADVANCED > Libraries** page.
4. You can see the new pattern group copied and pasted under the group to which it belongs.
5. Click **Edit Pattern** to edit a particular pattern.
6. Click **Delete** to delete a particular pattern.

## Configuring URL Policy

---

URL Policy sets URL access control policies to control traffic at the service level (that is, HTTP and HTTPS). The Barracuda Web Application Firewall employs a positive security policy; that is, by default all requests are denied unless explicitly allowed. To provide initial access, the Barracuda Web Application Firewall automatically creates a default URL policy. This URL policy allows entry to all users attempting to access a service. You can delete, disable (set to passive mode), or redefine the

default URL policy. You can also create additional URL policies to work with the default URL policy to limit user access to a Web site or service.

**Note**



When you enable data theft protection for a Service bound to the Security Policy, you must also set the parameter **Enabled** to **Yes** on the **SECURITY POLICIES > Data Theft Protection** page. Only then you can implement the data theft element configured on the **SECURITY POLICIES > Data Theft Protection** page and all URL policies for that service will prevent theft of the data type elements in server response pages as configured.

### To configure a URL Policy

1. From the **WEBSITES > Advanced Security** page, click **Add** under **Actions**. The **Configure URL Policy** dialog box appears.
2. Specify values for the following fields:
  - **URL Policy Name** - Enter the name for the URL policy.
  - **Status** - Select whether to enable or disable access control policy for all services.
  - **Host Match** - Enter the matching criterion for host field in the Request Header. This is either a specific host match or a wildcard host match with a single " \* " anywhere in the URL. You can enter a partial domain with wildcard (for example: \*.abc.com) but you can not use multiple asterisks.
  - **URL Match** - Enter the matching criterion for URL field in the Request Header. The URL should start with a "/" and can have only one " \* " anywhere in the URL. A value of /\* means that the ACL applies for all URLs in that domain.
  - **Extended Match** - Enter an expression that consists of a combination of HTTP headers and/or query string parameters. Use '\*' to denote "any request", that is, do not apply the Extended Match condition. To build an expression, click the *Edit* image button that appears next to this field, and specify the values for the following fields:
    - **Header Expression**: Specify a valid header expression.
    - **Element Type**: Select the element type from the drop-down list.
    - **Operation**: Select the operation from the drop-down list.
    - **Value**: Specify a valid expression.
    - **Concatenate**: Select 'and' radio-button to add some more expressions to the existing match sequence. Select 'or' radio-button to replace the existing match sequence.
    - Click **Insert** and then click **Apply** or click **Cancel**. For more on how to write extended match expressions, refer *Extended Match and Condition Expressions*.
  - **Extended Match Sequence** - Enter an order for matching the extended match rule when a request matches multiple rules with the same Host Match and URL Match.
  - **Mode** - Select the mode of action for request violations matching this URL Policy.
  - **Data Theft Protection** - Select whether to enable or disable data theft protection for this service.
  - **Bruteforce Prevention** - Select whether to enable or disable the bruteforce attack prevention.
3. Click **Add** to add the above configurations.
4. Click **Edit** against newly created URL Policy to modify the above configurations.

### Bruteforce Prevention

Bruteforce Prevention protects Web applications and Web sites from bruteforce attacks. A bruteforce attack consists of trying every possible code, combination, or password until you find the right one. For example, imagine a system which only allows 4 digit PIN codes. This means that there are a maximum of 10,000 possible PIN combinations that the attacker can try out to gain access to the

system. When a default URL Policy is created, bruteforce prevention is disabled by default. To enable bruteforce prevention, edit the default URL Policy.

### To configure a URL Policy with BruteForce Prevention

1. Modify a URL Policy by clicking **Edit** under **Actions**. The **Edit URL Policy** dialog box appears.
2. Under **Edit BruteForce Prevention**, specify values for the following fields:
  - **BruteForce Prevention** - Select whether to enable or disable the bruteforce attack prevention.
  - **Invalid Status Codes Only** - Select whether to enable or disable the brute force attack prevention for failed pages only.
  - **Count Window (seconds)** - Enter the time in seconds for allowing the maximum number of requests as per the settings in the parameter "Max Allowed Accesses Per IP" or "Max Allowed Accesses From All Sources".
  - **Max Allowed Accesses Per IP** - Enter the maximum number of requests allowed per IP address to access this Web service, if the parameter "Counting Criterion" is set to "Per IP".
  - **Max Allowed Accesses From All Sources** - Enter the maximum number of requests allowed from all sources to access this Web service, if the parameter "Counting Criterion" is set to "All Sources".
  - **Counting Criterion** - Select the criteria for allowing the requests.
  - **Exception Clients** - Enter the IP addresses that should be exempted (not locked out). You can enter a single, or a range of IP addresses, or a combination of both with comma (,) as a delimiter. The range of IP addresses must be separated with a hyphen (-). This makes an exception list of client IPs (permitted users). This list should not have any overlapping IP ranges.
3. Click **Save Changes** to save the above configurations.

### Interaction between Rate Control Pool and BruteForce Prevention

When you configure **BruteForce Prevention** and **Rate Control Pool** for a URL Policy, bruteforce prevention takes precedence over rate control pool.

Consider an example where bruteforce prevention has parameters "Max Allowed Accesses Per IP" set to 10 and "Count Window" set to 60 seconds and rate control pool has parameter "Maximum Per Client Backlog" set to 32. Assume that maximum active requests from the rate control pool are full. Now when the 11th request is served within the 60 seconds time interval it gets dropped by the Barracuda Web Application Firewall and the logs on the **BASIC > Web Firewall Logs** appear as "Brute force from IP". Also if the 11th request is served after the 60 seconds time interval, it gets dropped by the Barracuda Web Application Firewall.

## Configuring FTP Security

---

FTP Security allows you to specify which FTP commands should not be allowed. Command blocking is used to stop commands (verbs) that might be sent in an attempt to attack your FTP service. get and mget are FTP client side directives. They translate to RETR on the control connection. put and mput translate to STOR on the control connection. To block downloads and only allow uploads add RETR to the list of commands to be blocked and STOR should not be a part of blocked commands list. FTP Block Verbs parameter table lists the commands blocked by default. You can add to (or delete from) this list.



### To modify the list of blocked verbs

1. From the **WEBSITES > Advanced Security > FTP Security**, click **Edit** under **Actions**.
2. Specify values for the following fields:
  - **Status** - Select whether to enable or disable command blocking.
  - **FTP Block Verbs** - Enter the FTP verbs to block and click **Add**. **Note:** Even when the PORT command is allowed, sending a PORT command with a port less than 1024 is not allowed.
3. Click the trash icon against the FTP Verb to be deleted.
4. Click **Save Changes** to save the above settings.

## Configuring Session Tracking

---

A Session refers to all the request that a single client makes to a server. A session is specific to the user and for each user a new session is created to track all the request from that user. Every user has a separate session and separate session variable is associated with that session.

Session Tracking enables the Barracuda Web Application Firewall to limit the number of sessions originating from a particular client IP address in a given interval of time. Limiting the session generation rate by client IP helps prevent session-based Denial of Service (DoS) attacks.

### To edit Session Tracking

1. From the **WEBSITES > Advanced Security > Session Tracking**, click **Edit** under **Actions**.
2. Specify values for the following fields:
  - **New Session Count** - Enter the maximum number of new sessions allowed per IP address.
  - **Interval** - Enter the time in seconds for the maximum number of requests to be tracked.
  - **Status** - Select whether to enable or disable session tracking.
  - **Session Identifiers** - Select the token type which is used to recognize sessions.
  - **Exception Clients** - Enter the IP addresses that should be exempted (not locked out). You can enter a single, or a range of IP addresses, or a combination of both with comma (,) as a delimiter. The range of IP addresses must be separated with a hyphen (-). This makes an exception list of client IPs (permitted users). This list should not have any overlapping IP ranges.
3. Click **Save Changes** to save the above settings.

## Policy Tuner

---

Policy Tuner simplifies the human intervention to allow the false- positives (URLs and parameters that should be allowed but are not allowed). This helps manually allowing the URLs/parameters. The user needs to look for the causes in Web firewall logs and manually set the related rules to allow the acceptable URLs. In this method the user applies the recommended fix to allow a URL/parameter.

### To apply a fix to a security policy

1. Search for Web firewall logs by specifying values for the following fields:
  - **Service IP** - Enter the IP address and port of the service that received the request.
  - **Client IP/Mask** - Enter the IP address and netmask of the client that originated the request.



- **Date/Time Range** - Select a date and time range to search for Web firewall logs during that period. **Note:** The difference between the two dates should be less than or equal to 7 days.
2. Click **Get Logs** to search for log entries matching the data specified in the filter.
  3. From the list of Web Firewall logs, select single or multiple logs by selecting the check boxes.
  4. Click **Apply Fix** to apply the recommended fix for the attack.



This chapter describes the configuration and monitoring tasks you can perform from the Web interface. The following topic is covered:

- *Creating a High Availability (HA) Environment* on page 157

### Note



For more detailed information about a specific page in the Web interface, view the online help by clicking the question mark icon on the right side of the interface.

# Creating a High Availability (HA) Environment

The **ADVANCED > High Availability** page allows you to link a second Barracuda Web Application Firewall to act as a backup to your primary as shown in *Figure 12.1*. Both systems must be located on the same network, if the primary unit is down for any reason, the backup unit assumes and inherits the work of the primary unit. This provides continuous network availability. The Barracuda Web Application Firewall uses ports 8001 and 8002 to synchronize configuration between linked systems.

*Figure 12.1: High Availability Pair of Barracuda Web Application Firewall*



Each linked Barracuda Web Application Firewall sends a custom “heartbeat” to the other using UDP, to let each other know that they are up and running. The backup unit automatically becomes active and takes over the services of the primary system, if the primary system fails to send a heartbeat for nine (9) seconds or if it receives a heartbeat with primary unit’s state as “failed”.

Before adding two systems in a cluster, each Barracuda Web Application Firewall must meet the following requirements:

- Be installed on a unique WAN IP address. The Barracuda Web Application Firewalls use the WAN IP address (UDP port) to communicate for HA.
- Be able to ping each other on the WAN interface.
- The WAN interface on both Barracuda Web Application Firewall must be on the same switch (or physical network).

## To link two Barracuda Web Application Firewalls together:

1. Complete the installation process for each system as described in *Chapter 3 Initial Setup*.
2. From the **ADVANCED > Task Manager** page on Barracuda Web Application Firewall 1, verify that no processes are running. Complete this step on Barracuda Web Application Firewall 2 as well. No processes should be running when you link systems together.
3. From the **ADVANCED > High Availability** page on Barracuda Web Application Firewall 1, enter the Cluster Shared Secret password, and click **Save Changes**.
4. From the **ADVANCED > High Availability** page on Barracuda Web Application Firewall 2:
  - 4a. Enter the Cluster Shared Secret password. Both units in a cluster must have the same cluster shared secret to communicate. Click **Save Changes**.
  - 4b. In the Clustered Systems section, enter the WAN IP address of Barracuda Web Application Firewall 1, and click **Join Cluster**. Make sure that the join cluster task is not cancelled when the join is in progress. The unit from which the **Join Cluster** is executed becomes the designated backup unit. That is, Barracuda Web Application Firewall 1 becomes primary and Barracuda Web Application Firewall 2 becomes backup.
5. Refresh the **ADVANCED > High Availability** page, and verify that:
  - Each system’s WAN IP address appears in the **Clustered Systems** list.
  - The status is green for both units. This indicates the communication status.

- The **High Availability Status** can be viewed from **BASIC > Status > Performance Statistics** page. This shows the role and state of that unit.

Figure 12.2 and Figure 12.3 shows how this section would look before and after the second Barracuda Web Application Firewall has been clustered with the primary unit.

Figure 12.2: An unclustered Barracuda Web Application Firewall 2

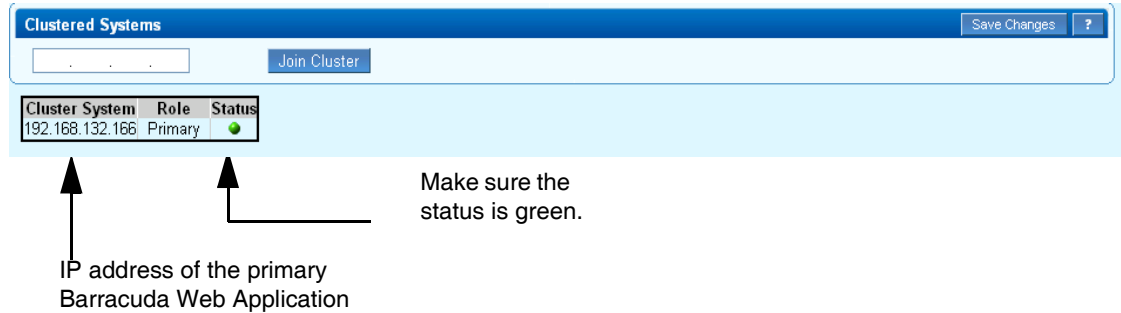
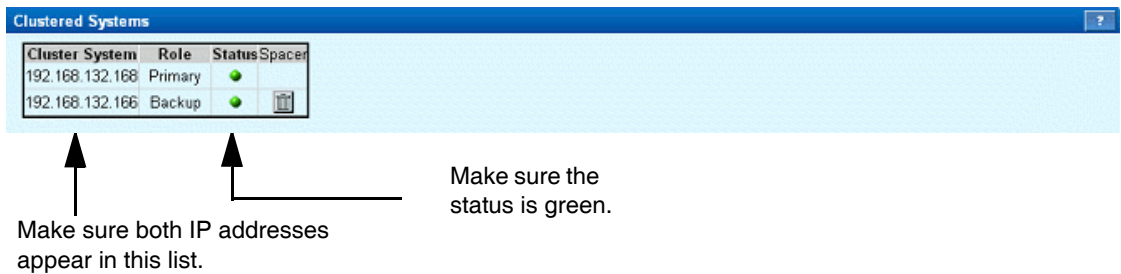


Figure 12.3: Two clustered Barracuda Web Application Firewalls



## Evaluating System Status

A Barracuda Web Application Firewall can be in a number of system states when it is in cluster. Once two Barracuda Web Application Firewalls are configured in redundant mode, you can view their system states under **Performance Statistics** section against **High Avail. Status** in the **BASIC > Status** page. When a single unit is deployed **High Avail. Status** is displayed as **Stand-alone**.

The following table describes the possible system states.

Table 12.1: System Status

State	Description
Active	<ul style="list-style-type: none"> <li>The Barracuda Web Application Firewall is powered up, all processes are running, the hardware is operating properly. The unit is capable of actively serving requests coming for services (if any).</li> </ul>
Standby	<ul style="list-style-type: none"> <li>The Barracuda Web Application Firewall is powered up, all processes are running, the hardware is operating properly. The unit is ready to assume services (if any).</li> </ul>
Failed	<ul style="list-style-type: none"> <li>The Barracuda Web Application Firewall is powered up, but the physical link status of an interface (WAN, LAN or Management) is down or Data Path Process has crashed. If the unit is in failed state, then the services should failover to the other unit, if the other unit is standby or active.</li> </ul>

## Failover

Failover is the process of moving active services from the primary unit to the backup unit when the primary unit is in the failed state. The backup unit should not be in a failed state in order to take over services. On failover, the backup unit assumes the services of the failed unit. It will continue to process the traffic until the failed unit is restored. Failover can occur in three ways:

1. **Link Down** - If the parameter **Monitor link** for **WAN IP Configuration**, **LAN IP Configuration** and **Management IP Configuration** is set to “Yes” in **BASIC > IP Configuration** page and when the link is down for any one of these interfaces, then the system goes into a failed state.
2. **Data Path Process Crash** - If the system had an outage, particularly, if the process has been crashing frequently, the system will be placed in the failed state. The frequency is measured by number of crashes in the last 5 minutes. If 3 crashes occur within the last 5 minutes, the system goes into failed state.
3. **Lost Heartbeat** - When the backup unit does not receive a heartbeat from WAN interface on the primary unit for 9 seconds, it concludes that the primary unit is down or dead and it executes failover.

### Note



In both **Automatic** and **Manual** modes, the Failover process cannot take place in case the back-up unit is also down.

## Failback

Failback is an operation that restores the functioning of services that have failed over from the primary unit to the backup unit. When the primary unit is returned from a failed state to the active state (that is, capable of actively serving application requests), the services will be automatically failed back (released) from the backup unit. The backup unit now goes into standby state.

There are two ways in which a Barracuda Web Application Firewall can failback:

- On the **Advanced > High Availability** page, set the **Failback Mode** parameter to **Automatic** so that it restarts active services automatically.

OR

- On the **Advanced > High Availability** page, set the **Failback Mode** parameter to **Manual**.
- When the primary unit is down, click **Failover** against the primary unit. The backup unit assumes the services of the failed primary unit.
- When the primary unit is up, click **Failback** against the back-up unit. The primary unit now resumes the services from the back-up unit.

## Data Propagated to Linked Systems

Linking systems together not only makes it easier to manage the two Barracuda Web Application Firewalls, but it also provides 100 percent redundant coverage of the propagated data.

Synchronization of the configuration takes place every 5 minutes on both units. *Table 12.2* identifies the data that is propagated when two systems are linked.

*Table 12.2: Data Propagated Between Linked Systems*

Propagated Data	Data Not Propagated
Any configuration changes through the Administration interface.	<ul style="list-style-type: none"><li>• System IP configuration (IP address, netmask, gateway, and DNS server) and Monitor Link information configured on the <a href="#">BASIC &gt; IP Configuration</a> page.</li><li>• System password and time zone as configured on the <a href="#">BASIC &gt; Administration</a> page.</li></ul>

## Updating Redundant Barracuda Web Application Firewalls

Updating the Firmware on a redundant pair of Barracuda Web Application Firewalls can be done without loss of services.

### Do the following to upgrade the Firmware:

1. Download the new version of the Firmware on both units.
2. After downloading, first apply the new version on the backup unit, this reboots the unit. Wait until backup unit comes up.
3. Apply the new version on the primary unit, this reboots the unit. For a small period of time, services may failover to the backup unit (because it may not receive the heartbeat from the primary unit while it is rebooting, and it will assume services). Once the primary has rebooted successfully, it will resume services and the backup will relinquish.

#### Note



For seamless HA functionality, ensure that both the units have the same firmware version. **Step 3** can be postponed to a later time, but the caveats of the database version and the heartbeat version may come into force during the period where there is a mismatch.

## Removing units from a cluster

A Barracuda Web Application Firewall can be removed from a cluster at any time. When removing a unit from a cluster, make sure that none of the units assume the ownership of interfaces and/or services. This can cause IP conflicts and services to go down.

### Do the following to remove the units:

1. Take backups of both the units.
2. Ensure that the primary is active and handling traffic.
3. From the [ADVANCED > High Availability](#) page of the backup unit, remove the cluster by clicking the delete button in the [Clustered Systems](#) list. This clears the backup configuration.
4. From the [ADVANCED > High Availability](#) page of the primary unit, remove the cluster by clicking the delete button in the [Clustered Systems](#) list. This retains the configuration of the primary unit, but removes the peer information from the configuration database.

## Removing the units for RMA

Before attempting RMA, it is required to remove from the cluster. This is because, the serial number is used to identify the peer and communicate between the two units. A replacement unit with a different serial number will not be able to automatically become part of the cluster; it has to go through the clustering procedure.

There are two scenarios for replacing the units:

- when the primary is dead. There are two ways to replace a primary unit:
  - backup unit to become the new primary
  - the new unit assume the role of the primary
- when the backup unit is dead

### To remove the cluster when the primary is dead:

1. To replace backup unit as the new primary unit.
  - 1a. Take a backup on the backup unit.
  - 1b. Ensure that the backup is active and handling traffic.
  - 1c. From the **ADVANCED > High Availability** page of the backup unit, remove the cluster by clicking the delete button in the **Clustered Systems** list.
  - 1d. Configure a new unit with the primary unit's WAN IP address in **BASIC > IP Configuration** page.
  - 1e. From the **ADVANCED > High Availability** page of the new unit, enter the WAN IP address of the new unit, and click **Join Cluster**.
2. To replace the new unit as primary unit, the pre-requisites are:
  - A backup of the primary unit exists before it went down
  - A down-time period is planned
  - 2a. Take a backup on the backup unit.
  - 2b. Ensure that the backup is active and handling traffic.
  - 2c. From the **ADVANCED > High Availability** page of the backup unit, remove the cluster by clicking the delete button in the **Clustered Systems** list.
  - 2d. Configure a new unit with the primary unit's WAN IP address.
  - 2e. Clear all configuration on the backup unit (this is required so that the services are stopped).
  - 2f. Restore the Primary unit's backup on the new unit. Note that this is not the backup taken in **Step 1**. Now, the primary assumes the services.
  - 2g. From the **ADVANCED > High Availability** page of the old backup unit, enter the WAN IP address of the new primary unit, and click **Join Cluster**.

### To remove the cluster when the backup is dead:

1. Ensure that the primary is active and handling traffic.
2. From the **ADVANCED > High Availability** page of the primary unit, remove the cluster by clicking the delete button in the **Clustered Systems** list.
3. Configure the new unit with the old backup unit's WAN IP address in **BASIC > IP Configuration** page.
4. From the **ADVANCED > High Availability** page of the new backup unit, enter the WAN IP address of the new backup unit, and click **Join Cluster**.







# Administering the Barracuda Web Application Firewall

---

This chapter provides general instructions for administering and maintaining the Barracuda Web Application Firewall. This chapter covers the following topics:

- *Administrative Settings* on page 165
- *Maintaining the Barracuda Web Application Firewall* on page 170

### Note



For more detailed information about a specific page in the Web interface, view the online help by clicking the question mark icon on the right side of the interface.

# Administrative Settings

---

This section covers the basic administrative settings for your Barracuda Web Application Firewall.

<i>Controlling Access to the Administration Interface.....</i>	<i>165</i>
<i>Customizing the Appearance of the Web Interface.....</i>	<i>165</i>
<i>Setting the Time Zone of the System.....</i>	<i>165</i>
<i>Enabling and Disabling Virus Protection.....</i>	<i>165</i>
<i>Enabling SSL for Administration.....</i>	<i>166</i>
<i>Adding new Administrators .....</i>	<i>167</i>
<i>Receiving Trap Messages and System Alerts.....</i>	<i>168</i>

## Controlling Access to the Administration Interface

---

The **BASIC > Administration** page allows you to perform the following tasks:

- Change the password of the administration account.
- Specify the IP addresses or netmask of the systems that can access the Web interface. All other systems will be denied access. This is configured in the **Administrator IP/Range** section.
- Change the port used to access the Web administration interface.
- Change the length of time users can be logged into the Web interface (default is 60 minutes).

## Customizing the Appearance of the Web Interface

---

The **ADVANCED > Appearance** page allows you to customize the default images used on the Web interface.

## Setting the Time Zone of the System

---

The **BASIC > Administration** page allows you to set the time zone of your Barracuda Web Application Firewall. The current time on the system is automatically updated via Network Time Protocol (NTP).

It is important that the time zone is set correctly because this information is used to coordinate traffic distribution and in all logs and reports.

**Note:** The Barracuda Web Application Firewall automatically reboots when you change the timezone.

## Enabling and Disabling Virus Protection

---

Use the **WEBSITES > Advanced Security page** to turn on Virus Scan. Under Advanced Security section, click **Edit** against a service to enable or disable Virus Scan. By default, Virus Scan is set to No, and the virus definitions are updated on regular basis (hourly by default) using Energize Updates.

When the Virus Scan is set to "Yes" for a service, all requests passing through the Barracuda Web Application Firewall is scanned for viruses and any traffic that contains virus is blocked. The Barracuda Web Application Firewall logs the request that can be viewed on the **BASIC > Web Firewall Logs**, so the system administrator can take appropriate action.

## Enabling SSL for Administration

The **ADVANCED > Secure Administration** page allows you to configure SSL for the administration Web interface for your Barracuda Web Application Firewall. Click **Save Changes** after making any changes.

SSL not only ensures that your passwords are encrypted, but also ensures that the rest of the data transmitted to and received from the administration interface is encrypted as well. For users who want to only allow secured connection, set up SSL.

### Note



The SSL configuration referred to here is related only for the Web-based administrative interface.

### To enable SSL

1. Select **ADVANCED > Secure Administration**.
2. Select **Yes** to enable HTTPS/SSL access only.
3. Enter the HTTPS port. The default is 443.

The following table describes the fields on the **Advanced > Secure Administration** page

Table 13.1: SSL Fields

Field	Description
<b>Web Interface HTTPS/SSL Configuration</b>	
HTTPS/SSL access only	Select <b>Yes</b> to enable SSL and only allow access to the Web administration interface via SSL. Select <b>No</b> to use standard HTTP access.
Web Interface HTTPS/SSL port	The SSL port used by the Barracuda Web Application Firewall. Default port for SSL is 443.
<b>SSL Certificate Configuration</b>	
Certificate Type	Select one of the following certificates for SSL: <ul style="list-style-type: none"><li>• <b>Default (Barracuda Networks)</b> certificates are free but generate browser alerts. The default certificate is signed by Barracuda Networks and provided free as the default type of certificate.</li><li>• <b>Private (self-signed)</b> certificates provide strong encryption without the cost of purchasing a certificate from a trusted certificate authority (CA). However Web browsers cannot verify the authenticity of the certificate and therefore display a warning every time a user accesses the administration interface.</li><li>• <b>Trusted</b> certificates are issued by trusted Certificate Authorities (CA), which are usually recognized by your Web browser so no additional configuration is required.</li></ul>

Table 13.1: SSL Fields (Continued)

Field	Description
<b>Certificate Generation</b>	
Organization Info	<p>The information stored in your certificates and Certificate Signing Requests. Provide the following information:</p> <ul style="list-style-type: none"> <li>• <b>Common Name</b> is the fully qualified domain name used to access the administration interface. For example: "barracuda.yourdomain.com"</li> <li>• <b>Country</b> is the two-letter country code where your organization is located.</li> <li>• <b>State or Province Name</b> is the full name of the state or province where your organization is located.</li> <li>• <b>Locality Name</b> is the city where your organization is located.</li> <li>• <b>Organization Name</b> is the legal name of your company or organization.</li> <li>• <b>Organization Unit Name</b> is an optional field in which to specify a department or section within your organization.</li> </ul>
<b>Trusted Certificate</b>	
Upload Signed Certificate	<p>After purchasing the certificate, browse to the location of the certificate and click <b>Upload Now</b>. Once you upload the certificate, your Barracuda Web Application Firewall automatically begins using it.</p> <p>Once you have uploaded your signed certificate, make sure <i>Trusted</i> is selected for the Certificate Type (described above).</p>
Upload Backup SSL Private key	<p>After downloading the corresponding private key, browse to the location of the key and click <b>Upload Now</b>.</p>

## Adding new Administrators

The **BASIC > Administrators** page allows you to add a new user with Administrative roles. The new administrator can login using his user ID and password and set new Web security policies.

### To add a new administrator

1. Specify values for the following fields:
  - **User ID** - Enter the username to login to Barracuda Web Application Firewall.
  - **Password** - Enter the password to login to Barracuda Web Application Firewall.
  - **Re-type Password** - Re-enter the password.
  - **Email Address** - Enter your valid email address.
2. Click **Add**. The newly added administrator appears under **Configured Administrators**.
3. Click **Change Password** under **Actions** to modify the password.
4. Click **Delete** to remove the newly added administrator's account.

## Receiving Trap Messages and System Alerts

From the **BASIC > Administration** page under **Trap Receivers**, specify the client's IP address and port number to receive trap messages. It also sends out an alert email to the recipient's email address, if the email is configured under **BASIC > Administration > Email Notifications**.

Trap defines the SNMP trap for generating customized alert for an event. The default alerts and their descriptions are given in the following table:

*Table 13.2: Trap Messages*

Trap Name	Object ID	Description
TempCritical	1.3.6.1.4.1.20632.8.1.3	Temperature of any one of CPU1, CPU1 VRM, CPU2, CPU2 VRM, board or RAM exceeded its threshold value.
TempHigh	1.3.6.1.4.1.20632.8.1.4	Temperature of any one of CPU1, CPU1 VRM, CPU2, CPU2 VRM, board or RAM is higher than 80C.
SystemFailOver	1.3.6.1.4.1.20632.8.1.5	System failed over to redundant system.
SwitchingToMaintMode	1.3.6.1.4.1.20632.8.1.6	System is switching to Maintenance mode.
FanDead	1.3.6.1.4.1.20632.8.1.7	One of the system fan is dead.
DataPortLinkDown	1.3.6.1.4.1.20632.8.1.8	Data port link (interface1 or interface2) is down.
ServerDown	1.3.6.1.4.1.20632.8.1.9	Server is down.
PeerDown	1.3.6.1.4.1.20632.8.1.10	Peer is down in redundant environment.
DataPortLinkUp	1.3.6.1.4.1.20632.8.1.11	Data port link (interface1 and interface2) is up.
ServerUp	1.3.6.1.4.1.20632.8.1.12	Server is up.
PeerUp	1.3.6.1.4.1.20632.8.1.13	Peer is up in redundant environment.
CookieEncryptionKeyAboutToExpire	1.3.6.1.4.1.20632.8.1.16	Shared secret key is about to expire.
CookieEncryptionKeyExpired	1.3.6.1.4.1.20632.8.1.17	Shared secret key has expired.
FirmwareStorageHigh	1.3.6.1.4.1.20632.8.1.18	Firmware storage exceeds 75%.
LogStorageHigh	1.3.6.1.4.1.20632.8.1.19	Log storage exceeds 85%.
RaidDegrading	1.3.6.1.4.1.20632.8.1.20	One of the RAID arrays is degrading.

In addition to the above trap messages the Barracuda Web Application Firewall sends out emails for the following three system alerts. System alerts notify you when:

- Your Energize Update subscription is about to expire

- New firmware updates are available
- Your system is low on disk space

Apart from that, you can also use the SNMP GET commands to view important statistics of the Barracuda Web Application Firewall. For more information on the SNMP GET commands, refer *Table D.2: SNMP GET Command* on page 220.



# Maintaining the Barracuda Web Application Firewall

---

This section describes how to manage and maintain your Barracuda Web Application Firewall using the Web administration interface. This section covers the following topics:

<i>Backing up and Restoring your System Configuration.....</i>	<i>170</i>
<i>Updating the Firmware of your Barracuda Web Application Firewall.....</i>	<i>170</i>
<i>Updating the Attack, Virus and Security Definitions.....</i>	<i>171</i>
<i>Replacing a Failed System .....</i>	<i>171</i>
<i>Reloading, Restarting, and Shutting Down the System .....</i>	<i>172</i>
<i>Using the Built-in Troubleshooting Tools .....</i>	<i>172</i>
<i>Using the Task Manager .....</i>	<i>172</i>
<i>Setting the System Configuration.....</i>	<i>173</i>
<i>Rebooting the System in Recovery Mode.....</i>	<i>173</i>

## Backing up and Restoring your System Configuration

---

The **ADVANCED > Backup** page lets you backup and restore the configuration of your Barracuda Web Application Firewall. You should backup your system on a regular basis in case you need to restore this information on a replacement Barracuda Web Application Firewall or in the event your current system data becomes corrupt.

If you are restoring a backup file on a new Barracuda Web Application Firewall that is not configured, you need to assign your new system an IP address and DNS information on the **BASIC > IP Configuration** page.

Note the following about the backup file:

- Do not edit backup files. Any configuration changes you want to make need to be done through the Web interface. The configuration backup file contains a checksum that prevents the file from being uploaded to the system if any changes are made.
- The following information is not included in the backup file:
  - System password
  - System IP information
  - DNS information

## Updating the Firmware of your Barracuda Web Application Firewall

---

The **ADVANCED > Firmware Update** page allows you to manually update the firmware version of the system or revert to a previous version. The only time you should revert back to an old firmware version is if you recently downloaded a new version that is causing unexpected problems. In this case, call Barracuda Networks Technical Support before reverting back to a previous firmware version.

If you have the latest firmware version already installed, the **Download Now** button will be disabled.

### Note



Applying a new firmware version results in a temporary loss of service. For this reason, you should plan a down-time period.

## Updating the Attack, Virus and Security Definitions

The **ADVANCED > Energize Updates** page allows you to manually update the attack, virus and security definition, as well as change the interval at which the Barracuda Web Application Firewall checks for updates. Energize Updates provide the Barracuda Web Application Firewall with the latest definitions.

We recommend that the **Automatically Update** setting be set to **Hourly** so your Barracuda Web Application Firewall receives the latest definitions as soon as new threats are identified by Barracuda Central.

The following table describes the common fields for Attack, Virus and Security Definition Updates. Click **Save Changes** after making any changes.

*Table 13.3: Definition Updates*

Field	Description
Current Installed Version	Starts the Barracuda Web Application Firewall in the normal (default) mode. This option is automatically selected if no other option is specified within the first three (3) seconds of the splash screen appearing.
Latest General Release	Displays the latest version that is available. If the current version running on the Barracuda Web Application Firewall is not the latest, click <b>Update</b> to download the latest version. The Update button is disabled if the system already has the latest version.
Previously Installed Version	Displays the previously installed version that was running on the system. To go back to this version of the definitions, click <b>Revert</b> .
Automatically Update	Determines the frequency at which the Barracuda Web Application Firewall checks for updates. To disable automatic updates, select Off. Hourly updates occur at the beginning of each hour. Daily updates occur at 12:20am (twenty after midnight) based on the system time zone. The recommended setting is Hourly.
Subscription Status	Informs you if your Energize Updates are current and when your subscription expires.

## Replacing a Failed System

Before you replace your Barracuda Web Application Firewall, use the tools provided on the **ADVANCED > Troubleshooting** page to try to resolve the problem.

In the event that a Barracuda Web Application Firewall fails and you cannot resolve the issue, customers that have purchased the Instant Replacement service can call Technical Support and arrange for a new unit to be shipped out within 24 hours.

After receiving the new system, ship the old Barracuda Web Application Firewall back to Barracuda Networks at the address below with an RMA number marked clearly on the package. Barracuda Networks Technical Support can provide details on the best way to return the unit.

Barracuda Networks  
3175 S. Winchester Blvd  
Campbell, CA 95008

**Note**



To set up the new Barracuda Web Application Firewall so it has the same configuration as your old failed system, restore the backup file from the old system onto the new system, and then manually configure the new system's IP information on the **BASIC > IP Configuration** page. For information on restoring data, refer to *Backing up and Restoring your System Configuration* on page 170.

## Reloading, Restarting, and Shutting Down the System

---

The **System Reload/Shutdown** section on the **BASIC > Administration** page allows you to shutdown, restart, and reload system configuration on the Barracuda Web Application Firewall.

Shutting down the system powers off the unit. Restarting the system reboots the unit. Reloading the system re-applies the system configuration.

You can also reset the Barracuda Web Application Firewall by pressing **RESET** on the front panel of the system. The following actions occur:

- Reboots the system
- Resets the firmware version to the factory setting

Do not press and hold the **RESET** button for longer than a few seconds. Doing so changes the IP address of the system. Pushing and holding the **RESET** button for eight seconds changes the default IP address to 192.168.1.200. Holding the button for 12 seconds changes the IP address to 10.1.1.200.

## Using the Built-in Troubleshooting Tools

---

The **ADVANCED > Troubleshooting** page provides various tools that help troubleshoot network connectivity issues that may be impacting the performance of your Barracuda Web Application Firewall.

For example, you can test your Barracuda system's connection to Barracuda Central to make sure it can successfully download the **latest attack, virus and security definitions**. You can also ping other devices from the Barracuda Web Application Firewall, perform a traceroute from the Barracuda Web Application Firewall to any another system, and other tasks.

Clicking **Clear Configuration** clears the entire configuration and restores the Barracuda Web Application Firewall to its initial configuration.

## Using the Task Manager

---

The **ADVANCED > Task Manager** page provides the list of all tasks that are in the process of being performed, and also displays any errors encountered when performing these tasks.

## Setting the System Configuration

---

The **ADVANCED > System Configuration** page allows the share a common encryption key to encrypt and decrypt the user session data. Cookies or hidden parameters are used to store the encrypted state information on client's navigation platforms such as browsers and other user agents. This is useful to avoid any traffic interruption during failover or when there are multiple Barracuda Web Application Firewalls deployed to scale the performance capacity.

## Rebooting the System in Recovery Mode

---

If your Barracuda Web Application Firewall experiences a serious issue that impacts its core functionality, you can use diagnostic and recovery tools that are available at the reboot menu to return your system to an operational state.

Before you use the diagnostic and recovery tools, do the following:

- Use the built-in troubleshooting tools on the **ADVANCED > Troubleshooting** page to help diagnose the problem.
- Perform a system restore from the last known good backup file.
- Contact Barracuda Networks Technical Support for additional troubleshooting tips.

As a last resort, you can reboot your Barracuda Web Application Firewall and run a memory test or perform a complete system recovery, as described in this section.

### To perform a system recovery or hardware test:

1. Connect a monitor and keyboard directly to your Barracuda Web Application Firewall.
2. Reboot the system by doing one of the following:
  - Click **Restart** on the **BASIC > Administration** page.
  - Press the Power button on the front panel to turn off the system, and then press the Power button again to turn back on the system.

The Barracuda splash screen displays with the following three boot options:

```
Barracuda  
Recovery  
Hardware_Test
```

3. Use your keyboard to select the desired boot option, and press **Enter**.

You must select the boot option within three seconds of the splash screen appearing. If you do not select an option within three seconds, the Barracuda Web Application Firewall defaults to starting up in the normal mode (first option).

For a description of each boot option, refer to *Reboot Options* on page 174.

### Note



To stop a hardware test, reboot your Barracuda Web Application Firewall by pressing Ctrl-Alt-Del.

## Reboot Options

Table 13.4 describes the options available at the reboot menu.

Table 13.4: Reboot Options

Reboot Options	Description
Barracuda	Starts the Barracuda Web Application Firewall in the normal (default) mode. This option is automatically selected if no other option is specified within the first three (3) seconds of the splash screen appearing.
Recovery	<p>Displays the Recovery Console where you can select the following options:</p> <ul style="list-style-type: none"><li>• <b>Perform filesystem repair</b>—Repairs the file system on the Barracuda Web Application Firewall.</li><li>• <b>Perform full system re-image</b>—Restores the factory settings on your Barracuda Web Application Firewall and clears out all configuration information.</li><li>• <b>Enable remote administration</b>—Initiates a connection to Barracuda Central that allows Barracuda Networks Technical Support to access the system. Another method for enabling this troubleshooting connection is to click <a href="#">Establish Connection to Barracuda Central</a> on the <a href="#">ADVANCED &gt; Troubleshooting</a> page.</li><li>• <b>Run diagnostic memory test</b>—Runs a diagnostic memory test from the operating system. If problems are reported when running this option, we recommend running the Hardware_Test option next.</li></ul>
Hardware_Test	<p>Performs a thorough memory test that shows most memory related errors within a two-hour time period. The memory test is performed outside of the operating system and can take a long time to complete.</p> <p>Reboot your Barracuda Web Application Firewall to stop the hardware test.</p>



# Chapter 14

## XML Firewall

---

This chapter explains how the Barracuda Web Application Firewall can protect a Web service against XML and SOAP based attacks. This chapter covers the following topics:

- *Web Services* on page 177
- *XML Firewall* on page 182
- *XML Validations* on page 183
- *XML Protections* on page 185

### Note



For more detailed information about a specific page in the Web interface, view the online help by clicking the question mark icon on the right side of the interface.

# Web Services

---

A Web application is designed to take input from a human user and display output to a human user. In contrast, a Web service is an application that is accessible on the Web but is intended to be used by another application. Web services share business logic, data, and processes through a programmatic interface. Web services are emerging as the new standard for organizations to streamline business processes with increased efficiency and reduced application integration costs. Web services are used primarily as a means for businesses to communicate with each other and clients, and they allow organizations to communicate without intimate knowledge of each other's infrastructure and security configurations.

## Web Services Implementation

---

Web services use a universal language to send data and instructions to one another over the Internet with no translation required. The term Web service describes a standardized way of integrating Web-based applications using the Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), and Universal Description, Discovery, and Integration (UDDI) open standards over an Internet protocol (usually HTTP). XML is used to tag the data, SOAP is used to exchange the data, WSDL is used for describing the services available, and UDDI is used for listing what services are available.

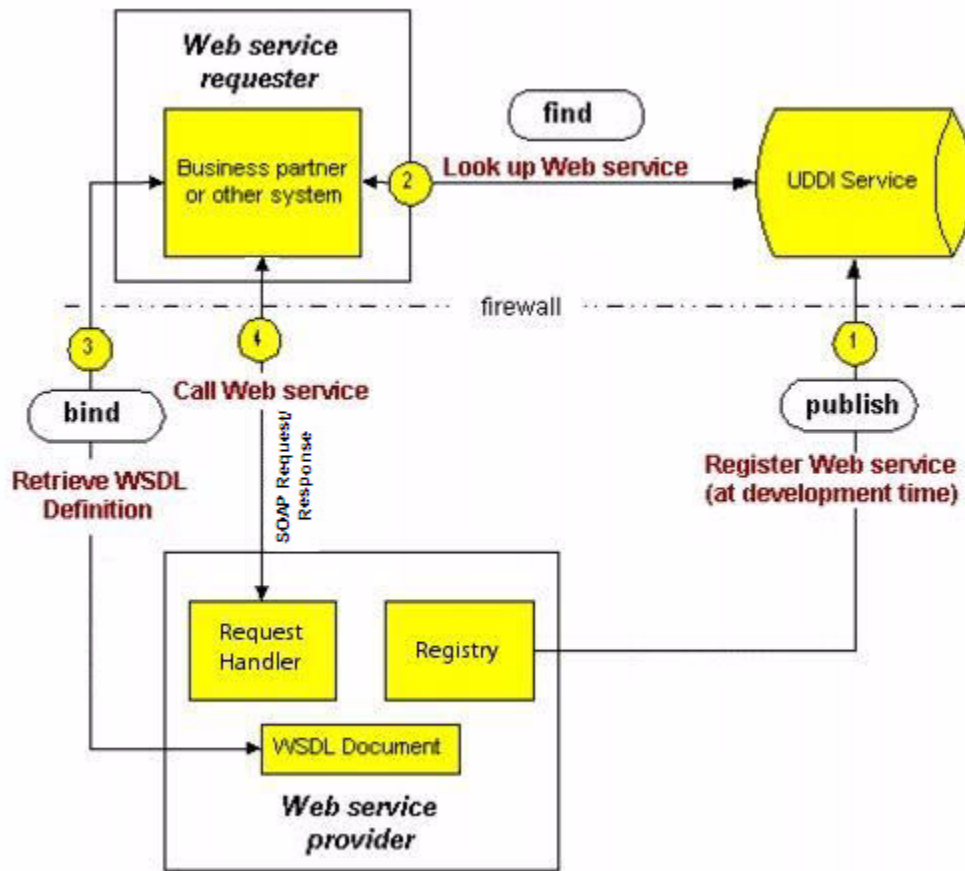
For example, consider two banks that want to set up a Web service to share account balance information. *Figure 14.1* illustrates the steps:

1. Bank A creates a Web service description (in the XML-based WSDL format) that describes what the Web service requires as input and what information should be returned, such as the customer's account number and password, and sends a SOAP request to register it with a UDDI service.
2. Bank B sends a SOAP request to the UDDI service to look up information about Bank A's Web service. (While using a UDDI service is common practice, it is not a requirement for a Web service.)
3. Bank B sends a SOAP request to Bank A's Web service to retrieve the WSDL definition and bind to that Web service.
4. Bank B sends a SOAP request to Bank A's Web service that conforms to the WSDL definition.

In this example, access to account information must be limited to a select list of approved intermediaries, which requires authentication through passwords, public keys, or other mechanisms. In addition, the bank might want to prioritize requests (such as by how much customers are paying for the service), confirm that payment for the service is received, and send a receipt. At each step, information must be secured from unauthorized access, attack, and data theft.



Figure 14.1: Web Service example



A business can combine multiple Web services to accomplish a task. For example, a travel service might define one Web service for interacting with its client application, another Web service for communicating with a credit card service (with the travel service acting as the client of the credit card service), another Web service for communicating with one or more hotel services, and another Web service for communicating with one or more airline services.

## WSDL

A Web service description (WSDL document) is a human-readable document that describes the expectations and functionality of a particular Web service, so a potential client can read the description and understand how to correctly interact with the service. Even though it is written solely from the point of view of the Web service, it is inherently intended for use by both the client and the service. It specifies the rules for how they should interact.

WSDL defines an XML grammar for describing network services as collections of communication end points capable of exchanging messages. WSDL service definitions provide documentation for distributed systems and serve as a recipe for automating the details involved in application communications.

See *WSDL 1.2 Syntax* on page 206 for the complete syntax defined in WSDL version 1.2.

## Web Service Vulnerabilities

---

Web services are vulnerable to many risks, such as cross site scripting, SQL injection, and denial of service. In addition, Web services come with their own specific vulnerabilities. Some of these include the following:

- Each Web service has an associated WSDL document that is basically a blueprint for the service. The document details the messaging request and response for the service in XML, what parameters (including data type) the service expects, and what operations are available through the service. By analyzing a service's WSDL document, a hacker knows exactly what the service is supposed to do and which parts are open to attack through techniques such as malformed SOAP messages and other XML parser attacks. A WSDL document might also reveal what tools generated the Web service, providing attackers with additional information on the environment.
- SOAP and XML are standards used to wrap data for easy consumption. SOAP provides enveloping information to deliver messages in a seamless fashion between heterogeneous applications. XML includes metadata to describe the structure of the information. Malicious code can be embedded into the elements or CDATA of the information. CDATA is used to delineate information in the message that should not be parsed. Embedded characters or malicious code can be sent. The receiving application may display or execute the data in unintended ways. Cross-site scripting referred to as XML encapsulation can be used to embed commands that can tie up system resources or gain unauthorized access.
- XML-based attacks take advantage of the XML parsers that process the SOAP message. Web services and existing infrastructure do not provide protection for XML-based attacks. Putting in recursive relationships to create entity expansions, bogus parameters, and significant amounts of white space can cause XML parsers to be overloaded or to perform unexpected problems.
- Any type of application can be behind the Web service interface, including packaged applications, internally developed applications, desktop applications, and legacy mainframe applications. These applications carry their own security vulnerabilities, which are likely to be even more exposed through a Web services interface. In addition, they all approach security in different ways. This presents a significant security challenge to protect these services consistently.

## Web Service Protections

---

The following table describes the possible attack techniques and how a Barracuda Web Application Firewall can protect against each technique.

*Table 14.1: Protection against XML and SOAP based attacks*

Technique	Description	Protection
Schema Poisoning	Manipulating the WS schema to alter the data processed by the application.	The Barracuda Web Application Firewall protects against schema poisoning by validating that content adheres to the defined WSDL and schema.
XML Parameter Tampering	Injection of malicious scripts or content into XML parameters.	The Barracuda Web Application Firewall protects against parameter tampering by validating that parameter values are consistent with the WSDL and schema specifications.

*Table 14.1: Protection against XML and SOAP based attacks*

Technique	Description	Protection
Inadvertent XDoS	Sending poorly encoded SOAP messages that cause the application to fail.	The Barracuda Web Application Firewall inspects SOAP at the header, envelope, and message level to ensure proper structure and content.
WSDL Scanning	Scanning the WSDL (business API) to reveal sensitive information about the data format of the application.	The Barracuda Web Application Firewall uses Web services cloaking to hide the true internal URI of sensitive Web services.
Coercive Parsing	Injection of malicious content into the XML.	The Barracuda Web Application Firewall utilizes real-time WS-I checking and content inspection to block malicious payloads.
Oversized Payload	Sending oversized files to create an XDoS attack (similar to a buffer overflow attack).	The Barracuda Web Application Firewall inspects the payload and enforces element, document, and other maximum payload sizes.
Recursive Payload	Sending mass amounts of nested data to create an XDoS attack against the XML parser.	The Barracuda Web Application Firewall validates WSDL and schema formats, inspects SOAP headers, envelopes, and messages, and ensures that WS-I standards are met.
SQL Injection	Disguising a malicious SQL command inside a SOAP wrapper in an attempt to disclose or modify back-end data.	The Barracuda Web Application Firewall utilizes real-time WS-I checking and content inspection by checking against schema.
Replay Attacks	The use of repetitive SOAP messages to force an XDoS attack.	The Barracuda Web Application Firewall includes request-level throttling technology to ensure resources cannot reach a fail state.
External Entity Attack	An attack on an application that parses XML input from untrusted sources using an incorrectly configured XML parser.	The Barracuda Web Application Firewall can suppress external URI references to protect against external manipulation of data.
Information Disclosure	Unencrypted data in a Web service message is exposed to anyone watching application traffic.	The Barracuda Web Application Firewall has extensive SSL security capabilities at the ASIC level and can ensure that no unencrypted XML traffic traverses the network under any circumstance.
Malicious Code Injection	Scripts embedded deep within a SOAP message can be delivered directly to applications and databases.	The Barracuda Web Application Firewall has extensive SSL security capabilities at the ASIC level and can ensure that no unencrypted XML traffic traverses the network under any circumstance.
Identity Centric Attack	Credentials are forged or impersonated in an attempt to access sensitive data.	The Barracuda Web Application Firewall enforces basic or strong authentication at the SOAP message level.
Processing Instructions (PI)	A PI is a text data section that is ignored by the XML parser and is passed on as instructions to applications.	The Barracuda Web Application Firewall can block requests containing Processing Instructions (PI).

*Table 14.1: Protection against XML and SOAP based attacks*

Technique	Description	Protection
Inline or external DTDs	DTDs describe the elements and their nesting structure in an XML document.	The Barracuda Web Application Firewall can block requests containing both inline or external DTDs.
External References	Requests containing external entities including external URI references or external DTDs.	The Barracuda Web Application Firewall can block requests containing external entities including external URI references or external DTDs.

# XML Firewall

---

The XML Firewall feature on the Barracuda Web Application Firewall comprises of **XML Validations** and **XML Protection** UI screens, available only with the **Barracuda Web Application Firewall Model 660** and higher.

Barracuda Web Application Firewall protects the Web services by applying the following XML firewall features:

- **WSDL** - Sets rules against requests and responses.
- **XML Validation Settings** - Sets default limit values with the assumption that any requests or responses with lengths greater than the defaults are possible attacks.
- **WS-I Basic Profile Tests** - Sets what WSI 1.0 basic profile tests to apply.
- **SOAP Validations** - Sets what SOAP conformance policies to apply.

XML Firewall is an optional feature and is disabled by default. Enable XML Firewall checks for XML and SOAP based validations and also activate the default XML protections on the **ADVANCED > XML Protection** page.

## To enable XML Firewall

- From the **ADVANCED > XML Validations** page, set the parameter **Enable XML Firewall** to 'Yes'.

## Tasks to enforce XML Firewall

---

The Barracuda Web Application Firewall allows you to protect a Web service by configuring XML Firewall in the following steps:

1. Create a service on the **BASIC > Services** page. (for more information refer *Creating Services* on page 42).
2. Locate the associated WSDL and Schema files for your Web application and import them on to the Barracuda Web Application Firewall (for more information refer *Import Schema/WSDL* on page 183). Associate the imported WSDL and Schema files with the target Web service. The Barracuda Web Application Firewall can then validate SOAP messages that pass through the Barracuda Web Application Firewall against the Web service policies defined in the WSDL file.
3. Configure an XML firewall (refer *XML Firewall* on page 182) by setting the following firewall features:
  - 3a. Set SOAP validation parameters to ensure requests conform to SOAP standards.
  - 3b. Enable WSI basic profile test assertions applied during run-time validation of XML and SOAP messages.
  - 3c. XML Validation rules to ensure SOAP messages conform to customized policies such as size limits, and XML filters.

# XML Validations

---

The [WEBSITE > XML Validations](#) page protects the Web services from XML based attacks by importing a WSDL file into the Barracuda Web Application Firewall and binding it with the target Web service. By default XML Validations is disabled. To enable XML Validations, first you need to enable XML Firewall.

## To enable XML Firewall

- From the [ADVANCED > XML Validations](#) page, set the parameter **Enable XML Firewall** to 'Yes'.

### Note



Only HTTP and HTTPS web services can be protected using the WSDL policy. The **Protected URLs** section displays the list of HTTP and HTTPS Web services configured on the Barracuda Web Application Firewall.

## Import Schema/WSDL

---

To apply the WSDL policy to a Web service, you need locate the associated WSDL and Schema files for your Web application and then import the Schema and WSDL files on to the Barracuda Web Application Firewall. WSDL (Web Services Description Language) is an XML based document that describes a Web service, its network location and the operations it supports. A Schema file is an XML based alternative to Document Type Definitions (DTDs), which describes the structure of an XML document, in this case the WSDL document.

For more information about a WSDL file, refer *WSDL* on page 178 and *WSDL 1.2 Syntax* on page 206.

### Note



A WSDL file can itself reference one or more Schema files for validation or to reuse Web Services interfaces. All the referenced files must be imported before the corresponding WSDL file can be imported. This allows the system to enforce validation checks and properly create reference associations between files at the time of import.

## To Import Schema/WSDL files on to the Barracuda Web Application Firewall.

1. Import the Schema and the WSDL files from the Import Schema/WSDL section by doing the following:
  - **Filetype** - Select the Filetype: WSDL or Schema. The Schema files referred to in the WSDL should be imported before the WSDL itself is imported.
  - **Name** - Enter a name for the WSDL/Schema file.
  - **Namespace** - Enter the target namespace defined in the WSDL/Schema file or any valid URI which can be used further to refer this namespace.
  - **File Path** - Browse the WSDL file on the local disk and click Open.
2. Click **Import**. The imported WSDL/Schema files are listed under **Imported Schema/WSDL** section.

You can perform the following tasks on the imported Schema/WSDL files:

- Bind the imported WSDL to the URLs in your Web services, as described in *Protected URLs* on page 184.

- Click **Export** to view the contents of the imported Schema/WSDL file.
- Click **Details** to view the available services and its port information of the imported WSDL file.

## Protected URLs

---

This section displays the list of HTTP and HTTPS Web services configured on the Barracuda Web Application Firewall. Click **Add** to bind a WSDL file with a Web service.

### Note



It is recommended that you enable XML Firewall selectively on URLs that require it, since XML Firewall requires validating the request contents which introduces additional latency in serving the request.

### To bind a WSDL file to the appropriate URLs in your Web service

1. Once the WSDL file is imported, you need to bind it to the appropriate URLs in your Web service. From the Protected URLs section, click **Add** for the specific Web service. The **Add WSDL** dialog box appears. Specify values for the following:
  - **Service Name** - Specifies the name of the service.
  - **Data Format** - Specify the format of messages to be validated through this service. Select **SOAP** to intercept SOAP based messages or **XML** to intercept general XML data.
  - **Enforce WSDL** - Specifies the name of the WSDL. All the imported WSDL files are listed in the drop-down list. Select a WSDL from the list. Note: This is enabled only when the parameter **Data Format** is set to **SOAP**.
  - **URL** - Specify the URL pattern for which XML Validations are to be enforced. Example: `"/MathService.asmx"`.
  - **Direction** - Specifies the direction to apply WSDL rules. Select whether you want the WSDL rules to be applied to the requests or responses or both.
  - **Enforce XML Validations** - Specifies whether to enforce XML validations for this service. Select 'Yes' to enforce XML validation settings defined in the **ADVANCED > XML Protection > XML Validation Settings** section.
  - **Enforce WS-I Validations** - Specifies whether to enforce WS-I validations for this service. Select 'Yes' to enforce WS-I validation settings defined in **ADVANCED > XML Protection > WS-I Basic Profile Assertions** section. Note: This is enabled only when the parameter **Data Format** is set to **SOAP**.
  - **Enforce SOAP Validations** - Specifies whether to enforce SOAP validations for this service. Select 'Yes' to enforce SOAP validation settings defined in **ADVANCED > XML Protection > SOAP Validations** section. Note: This is enabled only when the parameter **Data Format** is set to **SOAP**.
  - **Status** - Specifies the status of the bound WSDL. If set to 'On', the WSDL rules are applied. If the status is set to 'Off', the parameters **Enforce XML Validations**, **Enforce WS-I Validations** and **Enforce SOAP Validations** are disabled.
2. Click **Add**. The WSDL file is now bound with the Web service.

# XML Protections

---

The [ADVANCED > XML Protection](#) page provides default XML and SOAP based validation checks when enabled. By default XML Protection is disabled. To enable XML Protection, first you need to enable XML Firewall.

## To enable XML Firewall

- From the [ADVANCED > XML Validations](#) page, set the parameter **Enable XML Firewall** to 'Yes'.

## XML Validation Settings

---

The XML Validation Settings allows you to set custom validation rules for XML requests or responses. For example, a rule that aborts request processing if there are more than 50 total elements in the XML or limit the message size or total number of bytes, minimizing the chance of an unknown attacker flooding the service with too much data.

While the **SOAP Validations** and the **WS-I Basic Profile tests** (described in the next sections) determine whether a SOAP message is valid, they only mark an invalid message as an intrusion. Blocking invalid messages is enabled through **XML Validation Settings**.

The XML validation parameters are set to a default value. You can edit the existing values.

The XML requests which violates the XML validation rules are listed under the attack group **xmlfw-dos-violations** on the [SECURITY > Action Policy](#) page. Action policy specifies the action to be taken when a violation occurs. You can edit the default attack action settings for a policy.

## WS-I Basic Profile Assertions

---

The Web Services Interoperability Organization (WS-I) published the Basic Profile Version 1.0 to help customers validate their Web services. This profile contains implementation guidelines for the core Web services specifications: XML 1.0, XML Schema 1.0, SOAP 1.1, and WSDL 1.1. These guidelines are a set of requirements that define how these specifications should be used to develop inter-operable Web services. The WS-I test tools Basic Profile Test Assertions can be used to verify that a Web service conforms to these requirements. The Barracuda Web Application Firewall performs these tests during run time validation for SOAP messages.

There are forty two test case parameters, which are all set to **Yes** by default. 'Yes' means apply this test; 'No' means ignore this test. You can edit the existing values.

The XML requests which violates the WS-I Basic Profile Assertions are listed under the attack group **xmlfw-wsi-assertion-failures** on the [SECURITY > Action Policy](#) page. Action policy specifies the action to be taken when a violation occurs. You can edit the default attack action settings for a policy.

## SOAP Validations

---

SOAP is the transfer mechanism protocol for sending Web service descriptions in an HTTP message, and the SOAP validation parameters set the SOAP validation checks to apply. (These checks ensure the message adheres to SOAP standards.)



SOAP is a lightweight communication protocol for exchanging data using XML over HTTP. SOAP is not a defined Web application; however, it is a mechanism that provides communication between Web applications. SOAP is both platform independent and language independent. SOAP was developed as a W3C standard protocol. SOAP is a call-response mechanism that operates in a client-server paradigm. The client application makes a call to the server, passing in parameters, and the server provides a response. Both call and response are transported in the form of XML documents.

SOAP messages are susceptible to a number of potential attacks. Unintentionally exposing SOAP services could make the back-end server or application vulnerable to attacks. These attacks include the same attacks as in HTTP, such as input validation, SQL injection, and buffer overflow attacks. SOAP makes the back-end server more vulnerable because it allows actions to be invoked remotely on the back-end server.

There are four SOAP validation parameters, which are all set to **No** by default. You can edit the existing values and set it to **Yes** to validate these SOAP standards.

The XML requests which violates the SOAP Validations are listed under the attack group **xmlfw-soap-violations** on the [SECURITY > Action Policy](#) page. Action policy specifies the action to be taken when a violation occurs. You can edit the default attack action settings for a policy.



# Role Based Administration

---

This chapter describes Role Based Administration (RBA) feature of the Barracuda Web Application Firewall. The following are the sections included in this chapter:

- *Overview* on page 189
- *Roles* on page 189
- *Users* on page 191
- *Privileges* on page 191
- *Creating New Role* on page 192
- *Creating External Authentication Service* on page 193
- *Creating New Local Administrator Account* on page 193

# Overview

The Barracuda Web Application Firewall has the capability to provide role based administration (RBA). Role based administration is a mechanism of restricting access to system resources based on the roles assigned to users within an organization. The Barracuda Web Application Firewall is shipped with predefined roles. Each predefined role has distinct operational and configuration privileges that are listed under *Predefined Roles* section. In addition to predefined roles, the Barracuda Web Application Firewall enables you to create custom roles and define access privileges for those roles. These roles can be assigned to the users to perform specific job functions. The 'admin' role which is by default assigned to the 'admin' user has permission for role management.

The RBA feature in the Barracuda Web Application Firewall introduces the following components:

- Roles
- Users
- Privileges

## Roles

A role is a set of privileges or permissions on the available system resources. Roles are created for various job functions. The 'admin' role is allowed to create, modify and delete roles. A role can be assigned to multiple users within an organization. Assigning a role to a user confers the set of privileges on the system resources included in the role definition. All users who assume the same role, operate in the same environment and access the same resources. For example, if an administrator is assigned 'audit-admin' role, he can only view logs on the system and is exempted from accessing any other object.

### Note



Only the 'admin' role can create, modify and delete roles.

### Predefined Roles

The Barracuda Web Application Firewall provides a set of predefined (Factory Shipped) roles. Each predefined role is comprised of distinct access privileges on similar system resources. These roles cannot be modified or deleted. The following table briefly describes the predefined roles:

*Table 15.1: Predefined Roles*

Role	Description
admin	This is the super administrator role. The default 'admin' user is assigned this role. This role has privilege to perform all the system operations. An admin is responsible for creating and assigning roles.
audit-manager	This role defines the auditing capabilities. The role's responsibility is: <ul style="list-style-type: none"><li>• View logs</li></ul>

Role	Description
certificate-manager	<p>This role defines certificate management capabilities. The role's responsibility include:</p> <ul style="list-style-type: none"> <li>• Uploading certificates</li> <li>• Creating certificates</li> <li>• Uploading Trusted certificates</li> </ul>
service-manager	<p>This role defines the service management capabilities. The role's responsibility include:</p> <ul style="list-style-type: none"> <li>• Adding Server</li> <li>• Creating URL ACLs</li> <li>• Configuring Website Translation rules</li> <li>• Adding URL and Parameter profiles</li> <li>• Configuring Traffic Management rules</li> </ul> <p><b>Note:</b> This role is exempted from creating and deleting services.</p>
policy-manager	<p>This role defines the security policy management capabilities. The role's responsibility include:</p> <ul style="list-style-type: none"> <li>• Managing default and customized security policies</li> <li>• Modifying security policies</li> </ul> <p><b>Note:</b> This role is exempted from creating or deleting security policies.</p>
network-manager	<p>This role defines the networking capabilities. The role's responsibility include:</p> <ul style="list-style-type: none"> <li>• Advanced IP configuration</li> <li>• Configuring SNAT and ACL's</li> <li>• Network Troubleshooting</li> </ul>
monitoring-manager	<p>This role defines the system monitoring capabilities. The role's responsibility include:</p> <ul style="list-style-type: none"> <li>• View logs</li> <li>• Configuring email notifications</li> <li>• Exporting System logs, Application Logs and FTP Access Logs</li> <li>• Generating and scheduling reports</li> </ul>
guest	<p>User assigned to this role can view all the configuration, but is exempted from modifying the configuration.</p>

## New Roles

In addition to the factory shipped roles, the Barracuda Web Application Firewall enables you to create new roles. You can specify the privileges for these roles, and then assign these roles to the users. For more information on how to create a new role, refer *Creating New Role*.

## Users

---

A user is an individual who can use the Barracuda Web Application Firewall. The set of operations that can be performed by the user is defined by the role associated with it. Users are not assigned permissions directly, but only acquire them through the associated role. A 'user' can be categorized as 'local' or 'external'. Users need to be associated with a role during the creation of user accounts. Once the user account is created for a user, the user can access the system. When a user attempts to login, the Barracuda Web Application Firewall first tries to authenticate the user credentials against the configured local administrators. If a user cannot be authenticated locally, it queries the configured external authentication service. Once authenticated, the user inherits the privileges from the associated role.

### Local Users

Local administrators or users are local to the Barracuda Web Application Firewall. Local users are authenticated internally in the Barracuda Web Application Firewall. The admin user can create local users on the **ADVANCED > Admin Access Control** page. If you delete a local administrator account, the user is denied access to the system. For more information on how to add a new user, refer *Creating New Local Administrator Account*.

### External Users

External administrators or users are part of an external authentication service like the Lightweight Directory Access Protocol (LDAP). The Barracuda Web Application Firewall enables you to configure external authentication service, and allow the external users to access the system. An External user cannot be created but is synced internally from the LDAP server when a user is successfully authenticated with the configured directory services. You can override the default role association for an external user by editing the user. When an external user is not a part of LDAP database anymore, then the user needs to be manually deleted from the Barracuda Web Application Firewall. External authentication will however fail when a user is not part of LDAP anymore. For more information on how to create an external authentication service, refer *Creating External Authentication Service*.

## Privileges

---

A privilege means an access right or permission on a system resource. Privileges are used to control access to the system. You can grant privileges to a role, and then grant the role to one or more users. There are two distinct categories of privileges:

- Object Privileges
- Screen and Operation Privileges

### Object Privileges

The following are the key configuration objects that are classified in the role based administration:

Table 15.2: Object Privileges

Object	Description	Permission
Services	Exhibits all the services that are configured on the Barracuda Web Application Firewall.	<b>Read:</b> Enables the user to view the configuration of an object, but exempts from modifying the object.
Security Policies	Exhibits all the default and customized security policies.	<b>Write:</b> Enables the user to view and modify the configuration of an object, but exempts from deleting the object.
Authentication Services	Exhibits all the authentication services such as Lightweight Directory Access Protocol (LDAP) and Remote Authentication Dial In User Service (RADIUS).	<p><b>Read All:</b> Enables the user to view and modify the configuration of all objects, but exempts from modifying the objects.</p> <p><b>Write All:</b> Enables the user to view and modify the configuration of all objects, but exempts from deleting the objects.</p>

## Screen and Operation Privileges

The Barracuda Web Application Firewall provides several distinct operations. These operations include tasks such as shutting down the system, changing the system time and date, backup the system configuration etc. You can grant these operations to a role. A role granted with a specific operation can execute only that operation, and is prevented from executing any other operation in the system. For example, when a user is granted ‘appearance’ operation, he gains access to change the system name and reset the image used on the Web interface.

If you want to select an operation, ensure that the corresponding **Primary Tab** and **Secondary Tab** are selected. If you deny the primary tab, the corresponding secondary tab and operations become inaccessible. The admin user should determine the screens viewable to a user by selecting the Primary and Secondary Tabs.

## Creating New Role

The procedure below describes how to create a new administrator role that can be later delegated to your user account.

### To create new role:

1. Go to **ADVANCED > Admin Access Control** page.
2. In **Administrator Roles** section, click **Add Administrator Role**. The **Add Administrator Role** pop-up window appears.
3. Specify the new role name, and choose privileges for that role. Following are the sections displayed under **Add Administrator Role** window:
  - **Role Name:** Specify a name for the new administrator role.
  - **Services:** Specify read or write permission on the services. Click the Read or/and Write check box(es) against the service for which you want to give permission.
  - **Security Policies:** Specify read or write permission on the security policies. Click the Read or/and Write check box(es) against the security policies for which you want to give permission.

- **Authentication Services:** Specify read or write permission on the authentication service. Click the **Read** or/and **Write** check box(es) against the authentication database for which you want to give permission.
  - **Screen and Operation Privileges:** Specify permission on screens and operations for a role. Select the **Operations** listed under **Secondary Tab** that you want the user to perform.
4. Click **Create Role** to add the new role with the above privileges.

## Creating External Authentication Service

---

The Barracuda Web Application Firewall authenticates external administrators to this LDAP server. The procedure below describes how to create an external authentication service:

### To create external authentication service:

1. Go to **ADVANCED > Admin Access Control** page
2. In **External Authentication Service** section, click **Add Directory Service**. The **Add Directory Service** pop-up window appears.
3. Specify the values for the following fields:
  - **Realm Name:** Specify the name of the realm under which the Barracuda Web Application Firewall admins are stored (A realm identifies a collection of users and groups. It specifies information, in a flat directory structure, such as where users are located and where groups are located).
  - **IP Address:** Specify the IP address of an external LDAP server used for authenticating users.
  - **Port:** Specify the port number of the external LDAP server used for authenticating users. The standard port for LDAP is port 389 for non-SSL connections and 636 for SSL connections.
  - **Encryption:** Specify the type of encryption protocol to be used by the Barracuda Web Application Firewall when querying the LDAP database for user authentication and role retrieval.
  - **Bind DN:** Specify a Distinguished Name (DN) that can be used to query the LDAP server to search for the users/roles.
  - **Bind Password:** Specify the password used for querying the LDAP server using the bind DN.
  - **LDAP Search Base:** Specify the Distinguished Name (DN) at which to start the search, specified as a sequence of relative distinguished names (RDN), connected with commas and without any blank spaces.
4. Click **Add** to add this service.

## Creating New Local Administrator Account

---

The procedure below describes how to create a new local administrator account and delegate role to the administrator.

### To create new local administrator account:

1. Go to **ADVANCED > Admin Access Control** page.



2. In **Administrator Accounts** section, click **Add Local Administrator**. The **Local Administrator Account** pop-up window appears.
3. Specify the values for the following fields:
  - **User Name**: Specify the name of the user.
  - **Password**: Specify the password for the user.
  - **Role**: Select a role from the drop-down list that you want to assign the user. The drop-down lists predefined and customized roles.
  - **Email Address**: Specify the email address of the user.
4. Click **Add** to add the new local administrator account.



# Chapter 16

## Network Firewall

---

This chapter gives you an overview of Network Firewall features of the Barracuda Web Application Firewall and explains how to configure Source Network Address Translation (SNAT) and Access Control List (ACL) on the Barracuda Web Application Firewall. The following are the topics covered in this chapter:

- *Network Firewall Overview* on page 197
- *Configuring Source Network Address Translations (SNATs)* on page 199
- *Configuring Access Control Lists (ACLs)* on page 201

### Note



For more detailed information about a specific page in the Web interface, view the online help by clicking the question mark icon on the right side of the interface.

# Network Firewall Overview

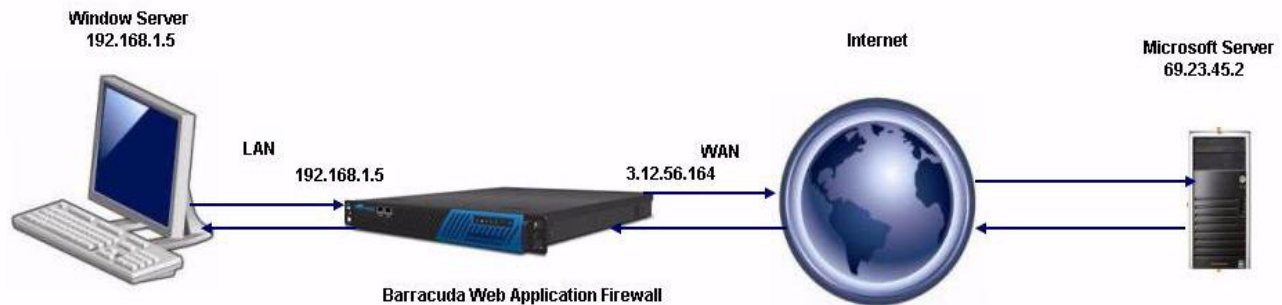
A Network Firewall is a system that inspects network traffic passing through it, and controls access between any two networks (LAN, WAN or MGMT) based on the pre-configured rules or filters.

Network firewalls create access control where unauthorized network-level requests are blocked at Network layer. At the network level, the Barracuda Web Application Firewall enforces network layer Access Control Lists (ACLs) and Source Network Address Translations (SNATs). ACLs make access control decisions in Network Firewall. SNAT rule maps multiple internal IP addresses to a single external IP address. This IP address translation is done to prevent exposing an internal address during routing. ACLs are created to set up the IP firewall access rules for the Barracuda Web Application Firewall. A network firewall rule specifies matching criteria for packets and corresponding action on each packet. If a packet matches, the configured action takes place. Several ACLs can be configured for network firewall support.

In Reverse Proxy mode, by default the Barracuda Web Application Firewall denies all the traffic originating from LAN to go out on the WAN. Only the configured traffic from LAN is allowed to go out on the WAN. For example, consider a Windows Server with internal IP address 192.168.1.5 on the LAN needs Microsoft updates from the Microsoft Server (69.23.45.2). The Windows Server will not be able to send the request as by default, the Barracuda Web Application Firewall denies all the requests unless explicitly configured. This can be accomplished with the combination of SNAT and ACLs. SNAT and ACLs are interdependent; SNAT is configured to map the internal IP address to an external IP address and ACLs to distinguish whether the request has to be sent out to the Microsoft Server or to be denied.

Use the **ADVANCED > Network Firewall** page to configure Source Network Address Translation (SNAT) rule and Access Control List (ACL) rule.

*Figure 16.1: Example for Reverse Proxy mode*



## Note



The Barracuda Web Application Firewall's network firewall module employs a positive security policy when active, that is, by default it denies all requests unless it is explicitly configured in the allowed list. Therefore, most firewall configuration tasks involve masking addresses, identifying which requests to allow or to drop.

By configuring SNAT and ACL rules in Network Firewall you can achieve the following:

- Non-routable internal IP addresses are translated to a single unique routable external IP address.
- Allows your internal routers to download files from the outside server.

For example:

- Allow the internal server to initiate DNS lookups with any external DNS server.
- Allow the internal mail client to initiate mail requests with a specific external SMTP server.
- Allow the internal FTP client to initiate an FTP session with a specific external FTP server for file uploads and downloads.

To accomplish the above goals, the following tasks are required:

1. Create a Source NAT that translates the internal server IP address to the external IP address. This in effect masks the internal server address from the outside exposure.
2. Create an ACL that allows the internal server to send a lookup request to an external DNS server.
3. Create an ACL that allows the internal server to send a mail request to a specified external SMTP server.
4. Create an ACL that allows the internal FTP client to establish a FTP session to a specified external FTP server.

#### Note



In Bridge mode, the Barracuda Web Application Firewall will only inspect traffic for the configured VIP and bridge all other traffic. In this mode, the traffic is bridged through the firewall and thus does not reach the NATs and ACLs in the layer 3 (L3). For example, if a Virtual IP address 198.156.132.122 on port 80 is created, the Barracuda Web Application Firewall will only inspect port 80 traffic for that Virtual IP address (198.156.132.122) and allow any other traffic, such as SSH or remote desktop to pass through.

## Configuring NAT for LAN Servers

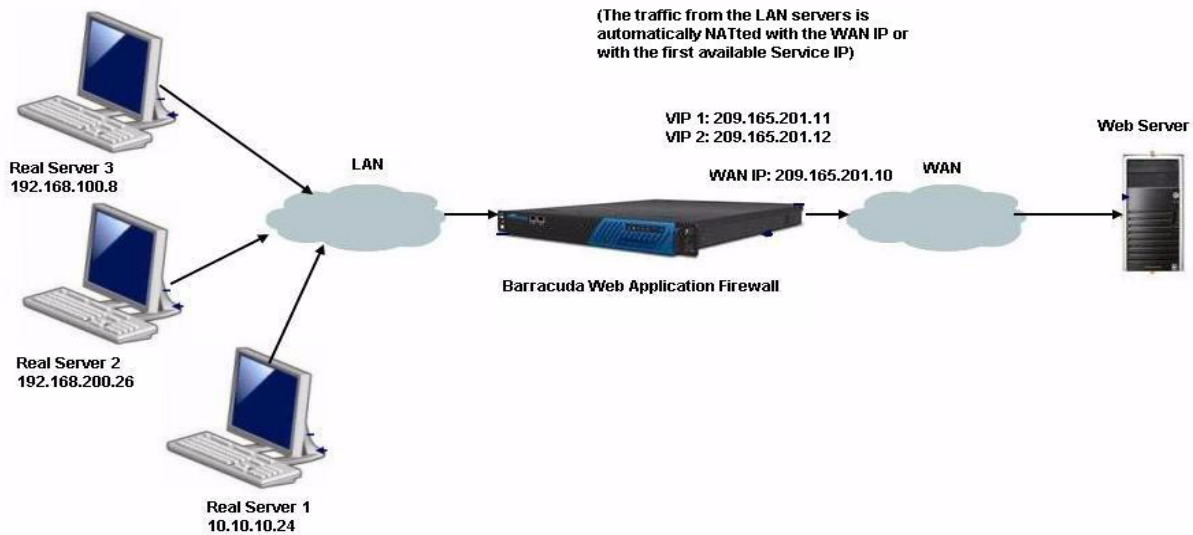
---

Network Address Translation (NAT) for LAN Servers is an enhanced feature, that automatically NATs all the servers on LAN with a single check option. When this option is enabled, all traffic originating from LAN to go out on the WAN is automatically NATted with the WAN interface IP address or with the first available Service IP address.

The user is not required to configure SNAT and ACL rule for the LAN servers, as the Barracuda Web Application Firewall automatically NATs and allows the LAN traffic to go out on the WAN.

For example, consider the LAN servers with the IP addresses 10.10.10.24, 192.168.32.10 and 192.168.30.15 wants to go out on WAN through the Barracuda Web Application Firewall, the traffic from the LAN servers is automatically NATted with the WAN interface IP address (209.165.201.10) or with the first available Service IP address 209.165.201.11.

Figure 16.2: NAT for LAN Servers



## Configuring Source Network Address Translations (SNATs)

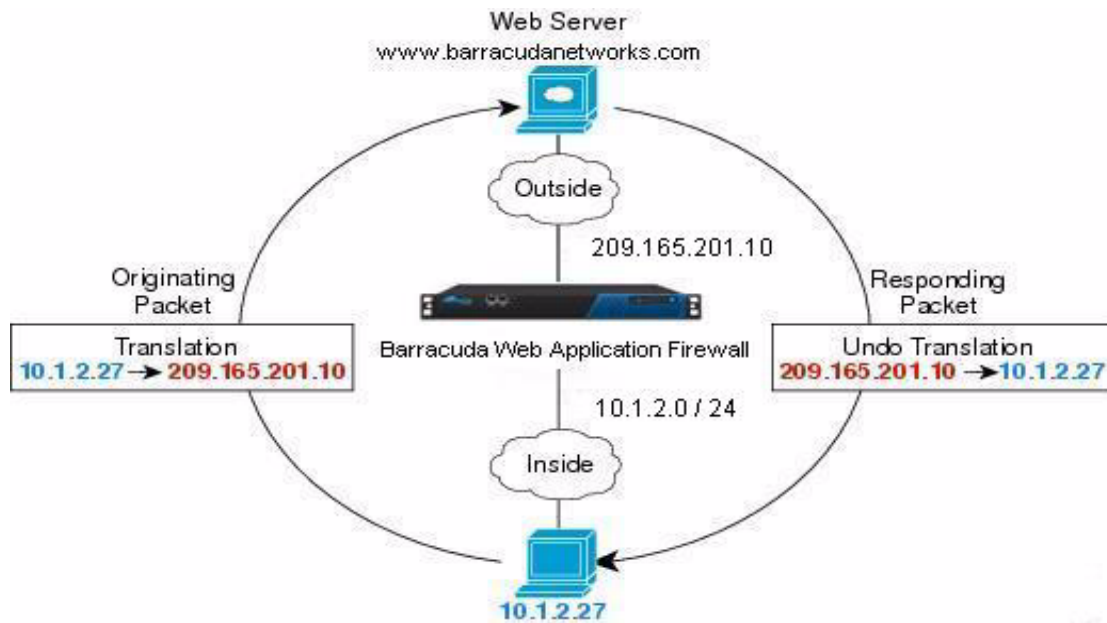
Source Network Address Translation (SNAT) is a technique that maps internal IP (private IP) addresses to an external IP (public IP) address. This IP address translation is done to prevent exposing an internal IP address during routing. The Source Network Address Translation (SNAT) re-writes the IP address of the computer that originated the packet.

SNAT is composed of two steps:

- The process of translating an internal IP address into an external IP address.
- The process to undo translation for returning traffic i.e. it re-writes the IP address of the computer that originated the packet.

For example, consider an internal IP 10.1.2.27 sends a packet to a Web server, the Barracuda Web Application Firewall translates the internal IP 10.1.2.27 to an external IP 209.165.201.10. When the Web server responds, the external IP 209.165.201.10 receives the packet and sends it to the internal IP 10.1.2.27. See Figure 14.2.

Figure 16.3: Example for SNAT



Following are the SNAT functionalities performed by the Barracuda Web Application Firewall:

- **Dynamic NAT:** Sets up a sequential translation between internal IP addresses and external IP addresses. You can specify a range of external IP addresses, and the Barracuda Web Application Firewall dynamically maps the internal IP address with the available external IP address. For example, enter the internal IP address 10.1.2.0 in **Pre SNAT Source** with subnet mask 255.255.255.0 in **Pre SNAT Source Mask** and enter a range of external IP addresses (209.165.201.11 - 209.065.201.16) in **Post SNAT Source**. The Barracuda Web Application Firewall will start to translate internal source IP address with the available external IP address.
- **Static NAT:** Sets up a one to one translation between a single internal IP address and a single external IP address. For example, an internal IP address of 10.1.2.27 will always translate to 209.165.201.10.
- **PAT:** The source Port Address Translation (PAT) is automatically done when encountered with the same source port numbers from different internal IP addresses. PAT is useful when you have few registered addresses and want those addresses to serve numerous internal addresses or when you want to funnel all traffic through a single address.

#### To configure Source Network Address Translations (SNATs):

1. From the **ADVANCED** tab, select **Network Firewall** menu.
2. Under **Source NAT**, Specify the values for the following fields:
  - **Pre SNAT Source** - Enter the Source IP address (the internal IP address) before translation.
  - **Pre SNAT Source Mask** - Enter the associated address space mask for the Source IP address before translation.
  - **Protocol (TCP/UDP)** - Select the protocol from the drop-down list.
  - **Destination Port** - Enter the destination port number of the network connection that has to be translated. By default the Barracuda Web Application Firewall SNATs any port in the range of 1-65535. You can configure the Destination Port field to restrict this to a narrower range or to a single port as required.
  - **Outgoing Interface** - Select the outgoing network port from which the traffic passes through (WAN, LAN or MGMT).

- **Post SNAT Source** - Enter the IP address (external address or the public IP address assigned by an ISP).
3. Click **Add** to add the configuration settings.

## Configuring Access Control Lists (ACLs)

---

Access Control List (ACL) is a list of permissions/access rules specified to a packet. ACLs are created to set up the IP firewall access rules. The list specifies the access rule for each packet. If a packet matches the specified rule, the configured action is performed. The action can be either **ALLOW** that accepts the packet and allows access or **DENY** that drops the packet and access is denied.

- ACLs are capable of confining the flow of traffic of individual IP address or range of IP addresses.
- ACLs can be bound to any of the interfaces (LAN, WAN or MGMT) of the Barracuda Web Application Firewall. This allows you to specify distinct restrictions for front-end and back-end traffic.

### To create an ACL:

1. From the **ADVANCED** tab, select **Network Firewall** menu.
2. Under **Network ACLs**, Specify the values for the following fields:
  - **From Address**: Enter the IP address from where the traffic is generated.
  - **From Netmask**: Enter the associated address space mask from where the traffic is generated.
  - **Interface**: Select the network port from which the traffic passes through.
  - **Protocol**: Select the protocol for the network.
  - **Service Ports**: Enter the associated service port or range of ports (using the format starting port -ending port).

### Note



If the source IP is set to 0.0.0.0, the source mask should also be set to 0.0.0.0.

- **To Address**: Enter the destination network or IP address.
  - **To Netmask**: Enter the associated address space mask of destination network or IP address.
  - **Action**: Select the action to be performed for the packet that matches the specified criteria (ALLOW or DROP).
  - **ALLOW**: This action states that a packet is allowed if it matches the specified criteria.
  - **DENY**: This action states that a packet is dropped if it does not match the specified criteria.
3. Click **Add** to add the configuration settings.



## **Barracuda Web Application Firewall Hardware**

---

This appendix provides hardware information for the Barracuda Web Application Firewall. The following topics are covered:

- *Barracuda Web Application Firewall Front Panel* on page 203
- *Barracuda Web Application Firewall Back Panel* on page 204
- *Hardware Compliance* on page 205

# Barracuda Web Application Firewall Front Panel

The following figure shows the front panel components as described in Table A.1.

Figure A.1: Front Panel of Barracuda Web Application Firewall



The following table describes the front panel components on the Barracuda Web Application Firewall.

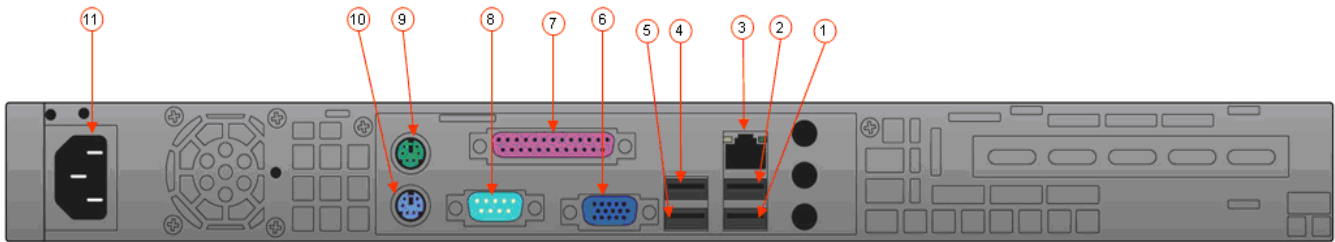
Table A.1: Barracuda Web Application Firewall Front Panel Component Descriptions.

Diagram Location	Description
1	On/Off button
2	Reset button
3	Power Indicator
4	Disk Activity
5	Management Network Activity
6	Attack Activity
7	Attack or Virus Downloads
8	LAN Port
9	WAN Port

# Barracuda Web Application Firewall Back Panel

The following figure shows the back panel components as described in Table A.2.

Figure A.2: Back Panel of Barracuda Web Application Firewall



The following table describes the back panel components on the Barracuda Web Application Firewall.

Table A.2: Barracuda Web Application Firewall Back Panel Component Descriptions.

Diagram Location	Description
1	Unused USB Port
2	Unused USB Port
3	Unused Network Port
4	Unused USB Port
5	Unused USB Port
6	VGA Display (console)
7	Unused Printer port
8	SerialPort
9	Mouse
10	Keyboard
11	Redundant Power Supply

# Hardware Compliance

---

This section contains compliance information for the Barracuda Web Application Firewall hardware.



## Notice for the USA

---

Compliance Information Statement (Declaration of Conformity Procedure) DoC FCC Part 15: This device complies with part 15 of the FCC Rules.

Operation is subject to the following conditions:

1. This device may not cause harmful interference, and
2. This device must accept any interference received including interference that may cause undesired operation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try one or more of the following measures:
  - Reorient or relocate the receiving antenna.
  - Increase the separation between the equipment and the receiver.
  - Plug the equipment into an outlet on a circuit different from that of the receiver.
  - Consult the dealer or an experienced radio/ television technician for help.

## Notice for Canada

---

This apparatus complies with the Class B limits for radio interference as specified in the Canadian Department of Communication Radio Interference Regulations.



## Notice for Europe (CE Mark)

---

This product is in conformity with the Council Directive 89/336/EEC, 92/31/EEC (EMC).



## Appendix B

# WSDL Files

---

This appendix describes the standard WSDL syntax and provides sample code. This appendix covers the following topics:

- *Elements of a WSDL file* on page 206
- *WSDL 1.2 Syntax* on page 206
- *Sample WSDL Code* on page 208

### Elements of a WSDL file

---

A WSDL document defines services as collections of network end points (ports). In WSDL, the abstract definitions of end points and messages are separated from their concrete network deployment or data format bindings. This allows the reuse of abstract definitions. The concrete protocol and data format specifications for a particular port type constitutes a reusable binding. A port is defined by associating a network address with a reusable binding, and a collection of ports defines a service. Hence, a WSDL document uses the following elements in the definition of network services:

- **Types** - a container for machine- and language-independent data type definitions using a system such as XSD that provides information about complex data types used in the WSDL document
- **Message** - an abstract definition of the data being communicated
- **Operation** - an abstract description of an action supported by the service
- **Port Type** - an abstract set of operations supported by one or more end points that describes the interfaces (legal operations) exposed by a Web service
- **Binding** - a concrete protocol and data format specification for a particular port type that describes how the operation is invoked for a Web service
- **Port** - a single end point defined as a combination of a binding and a network address that specifies a single communication end-point (binding) address
- **Service** - a collection of related end points (ports) that specifies the address(es) of the binding

### WSDL 1.2 Syntax

---

The following code is the syntax defined for WSDL version 1.2 by the W3C:

```
<wsdl:definitions name="nmtoken"? targetNamespace="uri">
  <import namespace="uri" location="uri"/> *
  <wsdl:documentation .... /> ? <wsdl:types> ?
  <wsdl:documentation .... /> ?
```

```

    <xsd:schema .... /> *
</wsdl:types>
<wsdl:message name="ncname"> *
    <wsdl:documentation .... /> ?
    <part name="ncname" element="qname"? type="qname"?/> *
</wsdl:message>
<wsdl:portType name="ncname"> *
    <wsdl:documentation .... /> ?
    <wsdl:operation name="ncname"> *
        <wsdl:documentation .... /> ?
        <wsdl:input message="qname"> ?
            <wsdl:documentation .... /> ?
        </wsdl:input>
        <wsdl:output message="qname"> ?
            <wsdl:documentation .... /> ?
        </wsdl:output>
        <wsdl:fault name="ncname" message="qname"> *
            <wsdl:documentation .... /> ?
        </wsdl:fault>
    </wsdl:operation>
</wsdl:portType>
<wsdl:serviceType name="ncname"> *
    <wsdl:portType name="qname"/> +
</wsdl:serviceType>
<wsdl:binding name="ncname" type="qname"> *
    <wsdl:documentation .... /> ?
    <!-- binding details --> *
    <wsdl:operation name="ncname"> *
        <wsdl:documentation .... /> ?
        <!-- binding details --> *
        <wsdl:input> ?
            <wsdl:documentation .... /> ?
            <!-- binding details -->
        </wsdl:input>
        <wsdl:output> ?
            <wsdl:documentation .... /> ?
            <!-- binding details --> *
        </wsdl:output>
        <wsdl:fault name="ncname"> *

```

```

        <wsdl:documentation .... /> ?
        <!-- binding details --> *
    </wsdl:fault>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="ncname" serviceType="qname"> *
    <wsdl:documentation .... /> ?
    <wsdl:port name="ncname" binding="qname"> *
        <wsdl:documentation .... /> ?
        <!-- address details -->
    </wsdl:port>
</wsdl:service></wsdl:definitions>

```

## Sample WSDL Code

---

The following code is a sample WSDL file for a simple math service that adds, subtracts, multiplies, and divides.

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions targetNamespace="http://example.com/webservices"
    xmlns:tm="http://example.com/wsdl/mime/textMatching/"
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
    xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:s0="http://example.com/webservices"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:s="http://www.w3.org/2001/XMLSchema"
    xmlns="http://schemas.xmlsoap.org/wsdl/">
    <types>
        <s:schema elementFormDefault="qualified"
            targetNamespace="http://example.com/webservices">
            <s:element name="Divide">
                <s:complexType>
                    <s:sequence>
                        <s:element maxOccurs="1" minOccurs="1"
                            name="a" type="s:int"/>
                        <s:element maxOccurs="1" minOccurs="1"
                            name="b" type="s:int"/>
                    </s:sequence>
                </s:complexType>
            </s:element>
        </s:schema>
    </types>

```

```

</s:element>
<s:element name="DivideResponse">
  <s:complexType>
    <s:sequence>
      <s:element maxOccurs="1" minOccurs="1"
        name="DivideResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="Multiply">
  <s:complexType>
    <s:sequence>
      <s:element maxOccurs="1" minOccurs="1"
        name="a" type="s:int"/>
      <s:element maxOccurs="1" minOccurs="1"
        name="b" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="MultiplyResponse">
  <s:complexType>
    <s:sequence>
      <s:element maxOccurs="1" minOccurs="1"
        name="MultiplyResult" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="Subtract">
  <s:complexType>
    <s:sequence>
      <s:element maxOccurs="1" minOccurs="1"
        name="a" type="s:int"/>
      <s:element maxOccurs="1" minOccurs="1"
        name="b" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="SubtractResponse">
  <s:complexType>

```



```

        <s:sequence>
            <s:element maxOccurs="1" minOccurs="1"
                name="SubtractResult" type="s:int"/>
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="Add">
    <s:complexType>
        <s:sequence>
            <s:element maxOccurs="1" minOccurs="1"
                name="a" type="s:int"/>
            <s:element maxOccurs="1" minOccurs="1"
                name="b" type="s:int"/>
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="AddResponse">
    <s:complexType>
        <s:sequence>
            <s:element maxOccurs="1" minOccurs="1"
                name="AddResult" type="s:int"/>
        </s:sequence>
    </s:complexType>
</s:element>
</s:schema>
</types>
<message name="DivideSoapOut">
    <part name="parameters" element="s0:DivideResponse"/>
</message>
<message name="SubtractSoapOut">
    <part name="parameters" element="s0:SubtractResponse"/>
</message>
<message name="MultiplySoapOut">
    <part name="parameters" element="s0:MultiplyResponse"/>
</message>
<message name="MultiplySoapIn">
    <part name="parameters" element="s0:Multiply"/>
</message>
<message name="SubtractSoapIn">

```

```

    <part name="parameters" element="s0:Subtract"/>
</message>
<message name="DivideSoapIn">
    <part name="parameters" element="s0:Divide"/>
</message>
<message name="AddSoapOut">
    <part name="parameters" element="s0:AddResponse"/>
</message>
<message name="AddSoapIn">
    <part name="parameters" element="s0:Add"/>
</message>
<portType name="TestServiceSoap">
    <operation name="Divide">
        <input message="s0:DivideSoapIn"/>
        <output message="s0:DivideSoapOut"/>
    </operation>
    <operation name="Multiply">
        <input message="s0:MultiplySoapIn"/>
        <output message="s0:MultiplySoapOut"/>
    </operation>
    <operation name="Subtract">
        <input message="s0:SubtractSoapIn"/>
        <output message="s0:SubtractSoapOut"/>
    </operation>
    <operation name="Add">
        <input message="s0:AddSoapIn"/>
        <output message="s0:AddSoapOut"/>
    </operation>
</portType>
<binding name="TestServiceSoap" type="s0:TestServiceSoap">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="Divide">
        <soap:operation
soapAction="http://example.com/webservices/Divide"
style="document"/>
        <input>
            <soap:body use="literal"/>
        </input>
    </operation>

```

```

        <output>
            <soap:body use="literal"/>
        </output>
    </operation>
    <operation name="Multiply">
        <soap:operation
            soapAction="http://example.com/webservices/Multiply"
            style="document"/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
    </operation>
    <operation name="Subtract">
        <soap:operation
            soapAction="http://example.com/webservices/Subtract"
            style="document"/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
    </operation>
    <operation name="Add">
        <soap:operation
            soapAction="http://example.com/webservices/Add"
            style="document"/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
    </operation>
</binding>
<service name="TestService">

```

```
<port name="TestServiceSoap" binding="s0:TestServiceSoap">
  <soap:address
    location="http://192.168.0.229:8090/MathService.asmx"/>
</port>
</service>
</definitions>
```

# Extended Match and Condition Expressions

---

An Extended Match expression is used in various configuration screens to specify a rule that matches a request or response. Different elements of the request can be matched and can be combined in a very flexible manner to identify a request and apply security measures on those requests alone.

A few examples:

- Header Host co example.com - match a request whose Host header contains example.com
- Parameter userid ex - match any request in which the parameter 'userid' is present
- (Header Host eq www.example.com) && (Client-IP eq 10.0.0.0/24) - match a request whose host header is www.example.com and the request client's IP address is in the 10.0.0.\* subnet.

## Quick reference

---

- Extended Match Expression:
  - Element Match
  - (Expression) [Join (Expression) ...]
- Join:
  - &&, ||
- Element Match:
  - Element [Element Name] Operator [Value]
- Element:
  - Request Elements: Method, HTTP-Version, Client-IP, URI, URI-Path, Header
  - Request Parameters: Parameter, Pathinfo
  - Response Elements: Status-code, Response-Header
- Operator:
  - Matching: eq, neq, req, nreq
  - Containing: co, nco, rco, nrco
  - Existence: ex, nex

## Structure of an Extended Match Expression

---

The following explains the components of an Extended match expression.

An expression consists of one or more **Element Matches**, combined using **Join** operators to indicate AND and OR operations to combine the Element Matches. Parentheses must be used to delimit individual Element Matches when using join operators. Parentheses can be nested.

An Element Match consists of an **Element**, an optional **Element Name**, an **Operator** followed by an optional **Value**. Some elements like "Header" require an Element Name like "User-Agent", whereas some elements like "HTTP-Version" require no further qualification. Also, some operators like "eq" (stands for "equals") require a value, whereas some operators like "ex" (stands for "exists") require no value.

Tokens are delimited by space and the parenthesis characters. Double quotes (") can be used to enclose single tokens which contain parenthesis characters or spaces. The back-slash character can also be used to escape, that is, remove the special meaning of the special characters (space and parentheses).

## Operators

---

The following are the possible operators in an Element Match. The operators are case insensitive, for example "eq", "Eq" and "EQ" are all treated the same.

- **eq** - true if the operand is equal to the given value. A case insensitive string comparison is performed. Thus, a value of "01" is not the same as a value of "1", whereas values "one" and "ONE" are treated the same.
- **neq** - true if the operand is not equal to the given value. A case insensitive string comparison is performed.
- **co** - true if the operand contains the given value.
- **nco** - true if the operand does not contain the given value.
- **rco** - true if the operand contains the given value, which is treated as a regular expression.
- **nrco** - true if the operand does not contain the given value, which is treated as a regular expression.
- **req** - true if the operand matches the given value, which is treated as a regular expression.
- **nreq** - true if the operand does not match the given value, which is treated as a regular expression.
- **ex** - true if the operand exists. A value is not required
- **nex** - true if the operand does not exist. A value is not required

## Elements

---

The following are the different Elements allowed in the expression. Elements and Element Names are case insensitive, so "Method" and "METHOD" are treated the same.

- **Method** - The HTTP Method that was received in the request. Example: (Method eq GET)
- **HTTP-Version** - This refers to the version of the HTTP protocol of the request. Example: (HTTP-Version eq HTTP/1.1)
- **Header** - An HTTP header in the request. An Element Name to identify which header is required to follow the word "Header". Example: (Header Accept co gzip). This will check if the "Accept:" header contains the string "gzip".
- **Client-IP** - This refers to the IP address of the client sending the request. The IP address can be either host IP address or subnet IP address specified by a mask. Only "eq" and "neq" operations

are possible for this element. Examples: (client-ip eq 192.168.1.0/24), (Client-IP eq 192.168.1.10)

- **URI** - The URI is the Uniform Resource Identifier in the request. This includes any query parameters in the request. Example: (URI rco /abc.\*html?userid=b)
- **URI-path** - This refers to the path portion of the URI, which excludes any query parameters. Example: (URI-path req \..\*copy%20[^/]\*)
- **Pathinfo** - This refers to the portion of URL which is interpreted as PATH\_INFO on the server. The Web Application Firewall uses a set of known extensions to determine whether a portion of the URL is a Pathinfo or not. For example, if the request URL is /twiki/view.cgi/Engineering, then, "/Engineering" is considered to be the pathinfo rather than part of the URL. Example: (PathInfo rco abc\*)
- **Parameter** - This refers to a parameter in the query string part of the URL. the servers as a name-value pair. The special parameter "\$NONAME\_PARAM" is used to refer to the case where the parameter name is absent. Examples: (Parameter sid eq 1234), (Parameter \$NONAME\_PARAM co abcd)
- **Status-code** - This refers to the status code of the response returned by the servers. Example: (status-code eq 302)
- **Response-header** - This refers to the HTTP response header in the response. The term "Response-header" should be followed by the name of the header on which the action is to be applied. Example: (Response-Header Set-Cookie co sessionid)

Not all elements are allowed in different kinds of expressions. The following restrictions apply:

- Request rules (ACLs, URL Policy, URL Profiles) allow the elements Method, HTTP-Version, Header, Client-IP, URI, URI-Path, PathInfo, and Parameter.
- Request Rewrite Condition allows the elements Method, HTTP-Version, Header, Client-IP, and URI.
- Response Rewrite Condition allows the elements Method, HTTP-Version, Header, Client-IP, URI, Status-code and Response-Header.

## Joins

---

Each expression can be joined with another expression by one of the following:

- **||** - This checks if either of the expressions are true.
- **&&** - This checks if both the expressions are true.

## Combining

---

More than one Element Match can be combined together by using the join operators || and && provided the Element Matches are enclosed in parentheses. Combining Element Matches without parentheses is not allowed. Example: (Header cookie ex) && (URI rco .\*\.html) && (Method eq GET)

Nested sub-expressions can be created by enclosing parentheses within expressions. This makes the expression more readable as well as unambiguous. Example: (HTTP-Version eq HTTP/1.1) && ((Header Host eq www.example.com) || (Header Host eq website.example.com))

## Escaping

---

The space character and the parentheses characters are special characters since they cause the parser to split the string into tokens at these separators. In some cases, it is required to specify these characters as part of the value itself. For example, the User-Agent header typically contains both spaces and parentheses, as in:

User-Agent: Mozilla/5.0 (Linux i686; en-US; rv:1.8.1.3) Firefox/2.0.0.3

The spaces and parenthesis characters in such cases must be escaped by prefixing these characters with a back-slash (\), or the entire value can be enclosed in double-quotes ("). Examples:

- Header User-Agent eq "Mozilla/5.0 (Linux i686; en-US; rv:1.8.1.3) Firefox/2.0.0.3"
- Header User-Agent eq Mozilla/5.0 \ (Linux\ i686;\ en-US;\ rv:1.8.1.3)\ Firefox/2.0.0.3

To specify the double-quote character itself, it must be escaped with a back-slash. This is true inside a quoted string, or a non-quoted string. Note that the single quote character has no special meaning, and is treated as any other character.

To specify the back-slash character itself, it must be escaped as "\\". This is true within quoted strings or non-quoted strings.

The back-slash character escapes all characters, not just the special characters. Thus, "\c" stands for the character "c" etc. In other words, back-slash followed by any character stands for the character, whether or not that character has a special meaning in the extended match syntax.



## Appendix D

# Usage Guidelines

Determining the policies to apply and understanding their ramifications can be confusing. This appendix provides guidelines in various areas to help your decision process.

### Macro Definitions

The Barracuda Web Application Firewall supports several macros to assist in configuring policies. The following table describes these macros arranged as per the areas where they can be used. The URI in these cases does not include the host.

*Table D.1: Macro Definitions*

Name	Description
<b>Request Rewrites</b>	
\$SRC_ADDR	Inserts the source (client) IP address. You can use it for the new value (Rewrite Value parameter) when inserting or rewriting a header (see <i>Configuring Request Rewrite</i> on page 68 for an example).
\$URI	Should be specified in the new value, if you are rewriting or redirecting the URI. \$URI specifies the complete request URI including the query string.
\$X509_VERSION	The client certificate's X509 version string.
\$X509_SERIAL_NUMBER	The serial number of the client certificate.
\$X509_SIGNATURE_ALGORITHM	The Signature Algorithm used in the client certificate.
\$X509_ISSUER	The client certificate's issuer string.
\$X509_NOT_VALID_BEFORE	Time from which the client certificate is valid.
\$X509_NOT_VALID_AFTER	Time after which the client certificate is invalid.
\$X509_SUBJECT	The client certificate's Subject string.
\$X509_SUBJECT_PUBLIC_KEY_TYPE	The X509 Certificate Subject Key Identifier String of the client certificate.
\$X509_SUBJECT_PUBLIC_KEY	Public Key modulus of the client certificate.

Name	Description
\$X509_SUBJECT_PUBLIC_KEY_RSA_BITS	Size of the client certificate's public key, in bits.
\$X509_EXTENSIONS	The client certificate's X509 Extensions String.
\$X509_HASH	The X509 Hash string of the client certificate.
\$X509_WHOLE	The X509 client certificate represented as a string in PEM format.
\$AUTH_USER	Adds the username.*
\$AUTH_PASSWD	Adds the password.*
\$AUTH_GROUPS	Adds the user roles.*  <b>*Note:</b> (1) The URL is not protected, i.e. access-control or authentication is off. The value substituted for the above three macros will be the special string "NCURLNotProtected". (2) The client has not logged in. The value substituted for the above three macros will be the special string "NCNoUserSession". (3) The user does not belong to any groups. The value substituted for \$AUTH_GROUPS will be the special string "NCNOUserRoles".
<b>URL ACLs</b>	
\$NONAME_PARAM	Inserts a parameter with no name (see <i>No Name Parameters</i> on page 220)
<b>Redirect Policy</b>	
%s Load Balancing	Represents the complete request URI.
%10sLoad Balancing	Represents the first 10 characters of the request URI.
<b>Response Page</b>	
%action-id	The attack id of the violation which resulted in this response page to be displayed.
%host	The host which sent this request.
%s	The URL of the request which caused this violation.
%client-ip	The Client IP of the request which caused the violation.
%attack-time	The time at which the violation occurred.
%attack-name	The attack name of the violation which resulted in the response page to be displayed.

## No Name Parameters

There might be times when you want to configure a parameter without a name. For example, consider a site that pops up an advertising window when a user lands there. A Javascript adds a query string that results in the following GET request:

```
GET /ad?xxx
```

### Note



The Barracuda Web Application Firewall does not learn “no name” parameters such as query strings like "GET /ad?0" added by a Javascript. Workaround: Add a null value URL ACL.

The Barracuda Web Application Firewall treats `xxx` as the value of a parameter. In this case, you cannot create an exception rule based on the `xxx` value because there is no way to associate it with a named parameter.

To address such situations (that is, requests with parameter name-value pairs of the type `?xxx` or `?=xxx` where `xxx` is the value), you can use a special token: `$NONAME_PARAM` (case insensitive). This token allows you to create an expression for a parameter without a name as in the following examples:

```
set extended match = parameter $NONAME_PARAM ex
set extended match = parameter $NONAME_PARAM eq 0
set extended match = parameter $noname_param co xxx
```

POST\_PARAM\_META\_VIOLATION log entries which contain parameters with no name to be logged as parameter " ".

Workaround: create a parameter with header = "Parameter \$NONAME\_PARAM ex" and remove the corresponding metacharacter from the list.

POST\_PARAM\_DIRECTORY\_TRAVERSAL\_VIOLATION entries which contain parameters with no name to be logged as parameter name "-".

Workaround: create a parameter with extended-match = "Parameter \$NONAME\_PARAM ex"

## Available Statistics from SNMP GET Command

The following table describes in detail the SNMP GET commands to view important statistics of Barracuda Web Application Firewall.

Table D.2: SNMP GET Command

Name	Object ID	Description
TotalApplications	1.3.6.1.4.1.20632.8.2	Total applications configured.
TotalServers	1.3.6.1.4.1.20632.8.3	Total servers configured.
TotalAttacks	1.3.6.1.4.1.20632.8.4	Count of attacks in last one hour.
ActiveApplications	1.3.6.1.4.1.20632.8.5	Total applications configured whose status is ON.

Name	Object ID	Description
ActiveServers	1.3.6.1.4.1.20632.8.6	Total servers whose operational status is in-service.
bwsMessage	1.3.6.1.4.1.20632.8.7	System log message.
SystemLoad	1.3.6.1.4.1.20632.8.8	System load in percentage.
CPUFanSpeed	1.3.6.1.4.1.20632.8.9	CPU fan speed in rotations per min.
SystemFanSpeed	1.3.6.1.4.1.20632.8.10	System fan speed in rotations per min.
CPUTemperature	1.3.6.1.4.1.20632.8.11	CPU temperature in degree celsius.
FirmwareStorage	1.3.6.1.4.1.20632.8.12	Firmware storage in percentage.
LogStorage	1.3.6.1.4.1.20632.8.13	Log Storage in percentage.
HighAvailabilityStatus	1.3.6.1.4.1.20632.8.14	High Availability Status.
OperationalMode	1.3.6.1.4.1.20632.8.15	Operation mode.
DataPathStatus	1.3.6.1.4.1.20632.8.16	Data Path Status.
LinkStatus	1.3.6.1.4.1.20632.8.17	Link Status.

# **Limited Warranty and License**

---

## **Limited Warranty**

---

Barracuda Networks, Inc., or the Barracuda Networks, Inc. subsidiary or authorized Distributor selling the Barracuda Networks product, if sale is not directly by Barracuda Networks, Inc., ("Barracuda Networks") warrants that commencing from the date of delivery to Customer (but in case of resale by a Barracuda Networks reseller, commencing not more than sixty (60) days after original shipment by Barracuda Networks, Inc.), and continuing for a period of one (1) year: (a) its products (excluding any software) will be free from material defects in materials and workmanship under normal use; and (b) the software provided in connection with its products, including any software contained or embedded in such products will substantially conform to Barracuda Networks published specifications in effect as of the date of manufacture. Except for the foregoing, the software is provided as is. In no event does Barracuda Networks warrant that the software is error free or that Customer will be able to operate the software without problems or interruptions. In addition, due to the continual development of new techniques for intruding upon and attacking networks, Barracuda Networks does not warrant that the software or any equipment, system or network on which the software is used will be free of vulnerability to intrusion or attack. The limited warranty extends only to you the original buyer of the Barracuda Networks product and is non-transferable.

## **Exclusive Remedy**

---

Your sole and exclusive remedy and the entire liability of Barracuda Networks under this limited warranty shall be, at Barracuda Networks or its service centers option and expense, the repair, replacement or refund of the purchase price of any products sold which do not comply with this warranty. Hardware replaced under the terms of this limited warranty may be refurbished or new equipment substituted at Barracuda Networks option. Barracuda Networks obligations hereunder are conditioned upon the return of affected articles in accordance with Barracuda Networks then-current Return Material Authorization ("RMA") procedures. All parts will be new or refurbished, at Barracuda Networks discretion, and shall be furnished on an exchange basis. All parts removed for replacement will become the property of the Barracuda Networks. In connection with warranty services hereunder, Barracuda Networks may at its discretion modify the hardware of the product at no cost to you to improve its reliability or performance. The warranty period is not extended if Barracuda Networks repairs or replaces a warranted product or any parts. Barracuda Networks may change the availability of limited warranties, at its discretion, but any changes will not be retroactive. IN NO EVENT SHALL BARRACUDA NETWORKS LIABILITY EXCEED THE PRICE PAID FOR THE PRODUCT FROM DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM THE USE OF THE PRODUCT, ITS ACCOMPANYING SOFTWARE, OR ITS DOCUMENTATION.

## Exclusions and Restrictions

---

This limited warranty does not apply to Barracuda Networks products that are or have been (a) marked or identified as "sample" or "beta," (b) loaned or provided to you at no cost, (c) sold "as is," (d) repaired, altered or modified except by Barracuda Networks, (e) not installed, operated or maintained in accordance with instructions supplied by Barracuda Networks, or (f) subjected to abnormal physical or electrical stress, misuse, negligence or to an accident.

EXCEPT FOR THE ABOVE WARRANTY, BARRACUDA NETWORKS MAKES NO OTHER WARRANTY, EXPRESS, IMPLIED OR STATUTORY, WITH RESPECT TO BARRACUDA NETWORKS PRODUCTS, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTY OF TITLE, AVAILABILITY, RELIABILITY, USEFULNESS, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR ARISING FROM COURSE OF PERFORMANCE, DEALING, USAGE OR TRADE. EXCEPT FOR THE ABOVE WARRANTY, BARRACUDA NETWORKS PRODUCTS AND THE SOFTWARE IS PROVIDED "AS IS" AND BARRACUDA NETWORKS DOES NOT WARRANT THAT ITS PRODUCTS WILL MEET YOUR REQUIREMENTS OR BE UNINTERRUPTED, TIMELY, AVAILABLE, SECURE OR ERROR-FREE, OR THAT ANY ERRORS IN ITS PRODUCTS OR THE SOFTWARE WILL BE CORRECTED. FURTHERMORE, BARRACUDA NETWORKS DOES NOT WARRANT THAT BARRACUDA NETWORKS PRODUCTS, THE SOFTWARE OR ANY EQUIPMENT, SYSTEM OR NETWORK ON WHICH BARRACUDA NETWORKS PRODUCTS WILL BE USED WILL BE FREE OF VULNERABILITY TO INTRUSION OR ATTACK.

## Software License

---

PLEASE READ THIS SOFTWARE LICENSE AGREEMENT ("AGREEMENT") CAREFULLY BEFORE USING THE BARRACUDA SOFTWARE. BY USING THE BARRACUDA SOFTWARE YOU ARE AGREEING TO BE BOUND BY THE TERMS OF THIS LICENSE. IF YOU DO NOT AGREE TO THE TERMS OF THIS LICENSE DO NOT USE THE SOFTWARE. IF YOU DO NOT AGREE TO THE TERMS OF THIS LICENSE YOU MAY RETURN THE SOFTWARE OR HARDWARE CONTAINING THE SOFTWARE FOR A FULL REFUND TO YOUR PLACE OF PURCHASE.

1. The software, documentation, whether on disk, in read only memory, or on any other media or in any other form (collectively "Barracuda Software") is licensed, not sold, to you by Barracuda Networks, Inc. ("Barracuda") for use only under the terms of this License and Barracuda reserves all rights not expressly granted to you. The rights granted are limited to Barracuda's intellectual property rights in the Barracuda Software and do not include any other patent or intellectual property rights. You own the media on which the Barracuda Software is recorded but Barracuda retains ownership of the Barracuda Software itself.

2. Permitted License Uses and Restrictions. This License allows you to use the Software only on the single Barracuda labeled hardware device on which the software was delivered. You may not make copies of the Software and you may not make the Software available over a network where it could be utilized by multiple devices or copied. You may not make a backup copy of the Software. You may not modify or create derivative works of the Software except as provided by the Open Source Licenses included below. The BARRACUDA SOFTWARE IS NOT INTENDED FOR USE IN THE OPERATION OF NUCLEAR FACILITIES, AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, LIFE SUPPORT MACHINES, OR OTHER EQUIPEMENT IN WHICH FAILURE COULD LEAD TO DEATH, PERSONAL INJURY, OR ENVIRONMENTAL DAMAGE.

3. You may not transfer, rent, lease, lend, or sublicense the Barracuda Software.
4. This License is effective until terminated. This License is automatically terminated without notice if you fail to comply with any term of the License. Upon termination you must destroy or return all copies of the Barracuda Software.
5. YOU EXPRESSLY ACKNOWLEDGE AND AGREE THAT THE USE OF THE BARRACUDA SOFTWARE IS AT YOUR OWN RISK AND THAT THE ENTIRE RISK AS TO SATISFACTION, QUALITY, PERFORMANCE, AND ACCURACY IS WITH YOU. THE BARRACUDA SOFTWARE IS PROVIDED "AS IS" WITH ALL FAULTS AND WITHOUT WARRANTY OF ANY KIND, AND BARRACUDA HEREBY DISCLAIMS ALL WARRANTIES AND CONDITIONS WITH RESPECT TO THE BARRACUDA SOFTWARE, EITHER EXPRESSED OR IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES AND/OR CONDITIONS OF MERCHANTABILITY, OF SATISFACTORY QUALITY, OF FITNESS FOR ANY APPLICATION, OF ACCURACY, AND OF NON-INFRINGEMENT OF THIRD PARTY RIGHTS. BARRACUDA DOES NOT WARRANT THE CONTINUED OPERATION OF THE SOFTWARE, THAT THE PERFORMANCE WILL MEET YOUR EXPECTATIONS, THAT THE FUNCTIONS WILL MEET YOUR REQUIREMENTS, THAT THE OPERATION WILL BE ERROR FREE OR CONTINUOUS, OR THAT DEFECTS WILL BE CORRECTED. NO ORAL OR WRITTEN INFORMATION GIVEN BY BARRACUDA OR AUTHORIZED BARRACUDA REPRESENTATIVE SHALL CREATE A WARRANTY. SHOULD THE BARRACUDA SOFTWARE PROVE DEFECTIVE, YOU ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION.
6. License. YOU EXPRESSLY ACKNOWLEDGE AND AGREE THAT YOU WILL PROVIDE AN UNLIMITED ZERO COST LICENSE TO BARRACUDA FOR ANY PATENTS OR OTHER INTELLECTUAL PROPERTY RIGHTS UTILIZED IN THE BARRACUDA SOFTWARE WHICH YOU EITHER OWN OR CONTROL.
7. Limitation of Liability. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT SHALL BARRACUDA BE LIABLE FOR PERSONAL INJURY OR ANY INCIDENTAL SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER, INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, LOSS OF DATA, BUSINESS INTERRUPTION, OR ANY OTHER COMMERCIAL DAMAGES OR LOSSES, ARISING OUT OF OR RELATED TO YOUR ABILITY TO USE OR INABILITY TO USE THE BARRACUDA SOFTWARE HOWEVER CAUSED, REGARDLESS OF THE THEORY OF LIABILITY AND EVEN IF BARRACUDA HAS BEEN ADVISED OF THE POSSIBILITY OF DAMAGES. In no event shall Barracuda's total liability to you for all damages exceed the amount of one hundred dollars.
8. Export Control. You may not use or otherwise export or re-export Barracuda Software except as authorized by the United States law and the laws of the jurisdiction where the Barracuda Software was obtained.

## **Energize Update Software License**

---

PLEASE READ THIS ENERGIZE UPDATE SOFTWARE LICENSE CAREFULLY BEFORE DOWNLOADING, INSTALLING OR USING BARRACUDA NETWORKS OR BARRACUDA NETWORKS-SUPPLIED ENERGIZE UPDATE SOFTWARE.

BY DOWNLOADING OR INSTALLING THE ENERGIZE UPDATE SOFTWARE, OR USING THE EQUIPMENT THAT CONTAINS THIS SOFTWARE, YOU ARE CONSENTING TO BE BOUND BY THIS LICENSE. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS LICENSE, THEN (A) DO NOT DOWNLOAD, INSTALL OR USE THE SOFTWARE, AND (B) YOU MAY RETURN THE SOFTWARE FOR A FULL REFUND, OR, IF THE SOFTWARE IS

SUPPLIED AS PART OF ANOTHER PRODUCT, YOU MAY RETURN THE ENTIRE PRODUCT FOR A FULL REFUND. YOUR RIGHT TO RETURN AND REFUND EXPIRES 30 DAYS AFTER PURCHASE FROM BARRACUDA NETWORKS OR AN AUTHORIZED BARRACUDA NETWORKS RESELLER, AND APPLIES ONLY IF YOU ARE THE ORIGINAL PURCHASER.

The following terms govern your use of the Energize Update Software except to the extent a particular program (a) is the subject of a separate written agreement with Barracuda Networks or (b) includes a separate "click-on" license agreement as part of the installation and/or download process. To the extent of a conflict between the provisions of the foregoing documents, the order of precedence shall be (1) the written agreement, (2) the click-on agreement, and (3) this Energize Update Software License.

License. Subject to the terms and conditions of and except as otherwise provided in this Agreement, Barracuda Networks, Inc., or a Barracuda Networks, Inc. subsidiary (collectively "Barracuda Networks"), grants to the end-user ("Customer") a nonexclusive and nontransferable license to use the Barracuda Networks Energize Update program modules and data files for which Customer has paid the required license fees (the "Energize Update Software"). In addition, the foregoing license shall also be subject to the following limitations, as applicable:

Unless otherwise expressly provided in the documentation, Customer shall use the Energize Update Software solely as embedded in, for execution on, or (where the applicable documentation permits installation on non-Barracuda Networks equipment) for communication with Barracuda Networks equipment owned or leased by Customer; Customer's use of the Energize Update Software shall be limited to use on a single hardware chassis, on a single central processing unit, as applicable, or use on such greater number of chassis or central processing units as Customer may have paid Barracuda Networks the required license fee; and Customer's use of the Energize Update Software shall also be limited, as applicable and set forth in Customer's purchase order or in Barracuda Networks' product catalog, user documentation, or Web site, to a maximum number of (a) seats (i.e. users with access to the installed Energize Update Software), (b) concurrent users, sessions, ports, and/or issued and outstanding IP addresses, and/or (c) central processing unit cycles or instructions per second. Customer's use of the Energize Update Software shall also be limited by any other restrictions set forth in Customer's purchase order or in Barracuda Networks' product catalog, user documentation or Web site for the Energize Update Software.

General Limitations. Except as otherwise expressly provided under this Agreement, Customer shall have no right, and Customer specifically agrees not to:

- i. transfer, assign or sublicense its license rights to any other person, or use the Energize Update Software on unauthorized or secondhand Barracuda Networks equipment, and any such attempted transfer, assignment or sublicense shall be void;
- ii. make error corrections to or otherwise modify or adapt the Energize Update Software or create derivative works based upon the Energize Update Software, or to permit third parties to do the same; or
- iii. decompile, decrypt, reverse engineer, disassemble or otherwise reduce the Energize Update Software to human-readable form to gain access to trade secrets or confidential information in the Energize Update Software.

Upgrades and Additional Copies. For purposes of this Agreement, "Energize Update Software" shall include (and the terms and conditions of this Agreement shall apply to) any Energize Update upgrades, updates, bug fixes or modified versions (collectively, "Upgrades") or backup copies of the Energize Update Software licensed or provided to Customer by Barracuda Networks or an authorized distributor/reseller for which Customer has paid the applicable license fees. NOTWITHSTANDING ANY OTHER PROVISION OF THIS AGREEMENT: (1) CUSTOMER HAS NO LICENSE OR RIGHT TO USE ANY SUCH ADDITIONAL COPIES OR UPGRADES UNLESS CUSTOMER, AT THE TIME OF ACQUIRING SUCH COPY OR UPGRADE, ALREADY HOLDS A VALID



LICENSE TO THE ORIGINAL ENERGIZE UPDATE SOFTWARE AND HAS PAID THE APPLICABLE FEE FOR THE UPGRADE; (2) USE OF UPGRADES IS LIMITED TO BARRACUDA NETWORKS EQUIPMENT FOR WHICH CUSTOMER IS THE ORIGINAL END USER PURCHASER OR LESSEE OR WHO OTHERWISE HOLDS A VALID LICENSE TO USE THE ENERGIZE UPDATE SOFTWARE WHICH IS BEING UPGRADED; AND (3) USE OF ADDITIONAL COPIES IS LIMITED TO BACKUP PURPOSES ONLY.

**Energize Update Changes.** Barracuda Networks reserves the right at any time not to release or to discontinue release of any Energize Update Software and to alter prices, features, specifications, capabilities, functions, licensing terms, release dates, general availability or other characteristics of any future releases of the Energize Update Software.

**Proprietary Notices.** Customer agrees to maintain and reproduce all copyright and other proprietary notices on all copies, in any form, of the Energize Update Software in the same form and manner that such copyright and other proprietary notices are included on the Energize Update Software. Except as expressly authorized in this Agreement, Customer shall not make any copies or duplicates of any Energize Update Software without the prior written permission of Barracuda Networks. Customer may make such backup copies of the Energize Update Software as may be necessary for Customer's lawful use, provided Customer affixes to such copies all copyright, confidentiality, and proprietary notices that appear on the original.

**Protection of Information.** Customer agrees that aspects of the Energize Update Software and associated documentation, including the specific design and structure of individual programs, constitute trade secrets and/or copyrighted material of Barracuda Networks. Customer shall not disclose, provide, or otherwise make available such trade secrets or copyrighted material in any form to any third party without the prior written consent of Barracuda Networks. Customer shall implement reasonable security measures to protect and maintain the confidentiality of such trade secrets and copyrighted material. Title to Energize Update Software and documentation shall remain solely with Barracuda Networks.

**Indemnity.** Customer agrees to indemnify, hold harmless and defend Barracuda Networks and its affiliates, subsidiaries, officers, directors, employees and agents at Customer's expense, against any and all third-party claims, actions, proceedings, and suits and all related liabilities, damages, settlements, penalties, fines, costs and expenses (including, without limitation, reasonable attorneys fees and other dispute resolution expenses) incurred by Barracuda Networks arising out of or relating to Customer's (a) violation or breach of any term of this Agreement or any policy or guidelines referenced herein, or (b) use or misuse of the Barracuda Networks Energize Update Software.

**Term and Termination.** This License is effective upon date of delivery to Customer of the initial Energize Update Software (but in case of resale by a Barracuda Networks distributor or reseller, commencing not more than sixty (60) days after original Energize Update Software purchase from Barracuda Networks) and continues for the period for which Customer has paid the required license fees. Customer may terminate this License at any time by notifying Barracuda Networks and ceasing all use of the Energize Update Software. By terminating this License, Customer forfeits any refund of license fees paid and is responsible for paying any and all outstanding invoices. Customer's rights under this License will terminate immediately without notice from Barracuda Networks if Customer fails to comply with any provision of this License. Upon termination, Customer must cease use of all copies of Energize Update Software in its possession or control.

**Export.** Software, including technical data, may be subject to U.S. export control laws, including the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations in other countries. Customer agrees to comply strictly with all such regulations and acknowledges that it has the responsibility to obtain licenses to export, re-export, or import Energize Update Software.

**Restricted Rights.** Barracuda Networks' commercial software and commercial computer software documentation is provided to United States Government agencies in accordance with the terms of this

Agreement, and per subparagraph "(c)" of the "Commercial Computer Software - Restricted Rights" clause at FAR 52.227-19 (June 1987). For DOD agencies, the restrictions set forth in the "Technical Data-Commercial Items" clause at DFARS 252.227-7015 (Nov 1995) shall also apply.

**No Warranty.** The Energize Update Software is provided AS IS. Customer's sole and exclusive remedy and the entire liability of Barracuda Networks under this Energize Update Software License Agreement will be, at Barracuda Networks option, repair, replacement, or refund of the Energize Update Software.

**Renewal.** At the end of the Energize Update Service Period, Customer may have the option to renew the Energize Update Service at the current list price, provided such Energize Update Service is available. All initial subscriptions commence at the time of sale of the unit and all renewals commence at the expiration of the previous valid subscription.

In no event does Barracuda Networks warrant that the Energize Update Software is error free or that Customer will be able to operate the Energize Update Software without problems or interruptions. In addition, due to the continual development of new techniques for intruding upon and attacking networks, Barracuda Networks does not warrant that the Energize Update Software or any equipment, system or network on which the Energize Update Software is used will be free of vulnerability to intrusion or attack.

**DISCLAIMER OF WARRANTY.** ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS, AND WARRANTIES INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTY OR CONDITION OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, SATISFACTORY QUALITY OR ARISING FROM A COURSE OF DEALING, LAW, USAGE, OR TRADE PRACTICE, ARE HEREBY EXCLUDED TO THE EXTENT ALLOWED BY APPLICABLE LAW. TO THE EXTENT AN IMPLIED WARRANTY CANNOT BE EXCLUDED, SUCH WARRANTY IS LIMITED IN DURATION TO THE WARRANTY PERIOD. BECAUSE SOME STATES OR JURISDICTIONS DO NOT ALLOW LIMITATIONS ON HOW LONG AN IMPLIED WARRANTY LASTS, THE ABOVE LIMITATION MAY NOT APPLY TO YOU. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS, AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM JURISDICTION TO JURISDICTION.

**General Terms Applicable to the Energize Update Software License Disclaimer of Liabilities.** IN NO EVENT WILL BARRACUDA NETWORKS BE LIABLE FOR ANY LOST REVENUE, PROFIT, OR DATA, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL, OR PUNITIVE DAMAGES HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY ARISING OUT OF THE USE OF OR INABILITY TO USE THE ENERGIZE UPDATE SOFTWARE EVEN IF BARRACUDA NETWORKS OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. In no event shall Barracuda Networks' liability to Customer, whether in contract, tort (including negligence), or otherwise, exceed the price paid by Customer. BECAUSE SOME STATES OR JURISDICTIONS DO NOT ALLOW LIMITATION OR EXCLUSION OF CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

This Energize Update Software License shall be governed by and construed in accordance with the laws of the State of California, without reference to principles of conflict of laws, provided that for Customers located in a member state of the European Union, Norway or Switzerland, English law shall apply. The United Nations Convention on the International Sale of Goods shall not apply. If any portion hereof is found to be void or unenforceable, the remaining provisions of the Energize Update Software License shall remain in full force and effect. Except as expressly provided herein, the Energize Update Software License constitutes the entire agreement between the parties with respect to the license of the Energize Update Software and supersedes any conflicting or additional terms contained in the purchase order.

## Open Source Licensing

---

Barracuda products may include programs that are covered by the GNU General Public License (GPL) or other "open source" license agreements. The GNU license is re-printed below for you reference. These programs are copyrighted by their authors or other parties, and the authors and copyright holders disclaim any warranty for such programs. Other programs are copyright by Barracuda Networks.

### GNU GENERAL PUBLIC LICENSE, (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public

License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

#### GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide

if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

*one line to give the program's name and an idea of what it does.*

Copyright (C) yyyy name of author

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19yy name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Barracuda Products may contain programs that are copyright (c)1995-2005 International Business Machines Corporation and others. All rights reserved. These programs are covered by the following License:

"Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation."

Barracuda Products may include programs that are covered by the BSD License: "Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE."

Barracuda Products may include the libspf library which is Copyright (c) 2004 James Couzens & Sean Comeau All rights reserved. It is covered by the following agreement: Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS MAKING USE OF THIS LICENSE OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Barracuda Products may contain programs that are Copyright (c) 1998-2003 Carnegie Mellon University. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. The name "Carnegie Mellon University" must not be used to endorse or promote products derived from this software without prior written permission. For permission or any other legal details, please contact Office of Technology Transfer Carnegie Mellon University 5000 Forbes Avenue Pittsburgh, PA 15213-3890 (412) 268-4387, fax: (412) 268-7395 tech-transfer@andrew.cmu.edu .Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by Computing Services at Carnegie Mellon University (<http://www.cmu.edu/computing/>)." CARNEGIE MELLON UNIVERSITY DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL CARNEGIE MELLON UNIVERSITY BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Barracuda products may include programs that are covered by the Apache License or other Open Source license agreements. The Apache license is reprinted below for your reference. These programs are copyrighted by their authors or other parties, and the authors and copyright holders disclaim any warranty for such programs. Other programs are copyright by Barracuda Networks.

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend

that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

#### **Source Code Availability**

Per the GPL and other "open source" license agreements the complete machine readable source code for programs covered by the GPL or other "open source" license agreements is available from Barracuda Networks at no charge. If you would like a copy of the source code or the changes to a particular program we will gladly provide them, on a CD, for a fee of \$100.00. This fee is to pay for the time for a Barracuda Networks engineer to assemble the changes and source code, create the media, package the media, and mail the media. Please send a check payable in USA funds and include the program name. We will mail the packaged source code for any program covered under the GPL or other "open source" license.



# Index

---

## A

- ACLs 201
- Active Mode 50
- Administration 165
- Appearance 165
- Attack Types 143
- Authentication 107
  - SSO 108
- Authentication Services 113
  - LDAP 114
  - RADIUS 115
- Authorization 112

## B

- Backup 170
- Backup Server 49
- Basic Security for a Service 47
- Bridge-Path 33

## C

- Caching 92
- Certificate Components 99
  - CA Certificate 100
  - Distinguished Name 100
  - Key Pair 99
  - Token 100
- Client Certificates 116
- Compression 94
- Creating a High Availability (HA) Environment 157
- Custom Parameter Class 145
- Custom Patterns 143
- Custom Service 45
- Custom SSL Service 45

## E

- Energize Updates 171

## F

- Failback 159
- Failover 159
- Firmware Update 170

- FTP Security 152
- FTP Service 46

## H

- Header ACL 52
- HTTP service 42
- HTTPS service 43

## I

- Identity Theft Patterns 142
- Initial Setup 36
- Input Types 144
- Instant SSL service 43
- Interface Routes 140
- Intermediary Certificates 104
- Internal Patterns 150

## L

- Local Users/Groups 115
- Logs 132
  - Access Logs 133
  - Audit Logs 133
  - Export Logs 134
  - FTP Web Logs 135
  - Search Logs 134
  - Web Firewall Logs 132

## M

- Multiple IP Address Configuration 139

## N

- Network Firewall 75, 189, 197

## O

- One-armed 33

## P

- Passive Mode 50
- Policy Tuner 153

## R

- Rate Control Pool 146
- Recovery Mode 173
- Redirect service 44
- Reload/Shutdown 172
- Removing the units for RMA 161
- Removing units from a cluster 160
- Reports 136
- Request Limits 54
- Response Page 145
- Reverse Proxy 32

## S

- Session Identifiers 146
- Session Tracking 153
- Signed Certificate 104
- SNATs 199
- SSL 49
- Static Routes 139
- System Configuration 173

## T

- Time Zone 165
- Traffic Management 90
- Trap Messages 168
- Troubleshooting 171

## U

- URL ACLs 51
- URL Policy 150
  - Bruteforce Prevention 151

## V

- VLAN 140

## X

- X.509 Certificate 99
- XML Firewall 182