Prepare for the Installation ...................................................... 40
Connect Barracuda Web Application Firewall to Network. .......... 41
Configure IP Address and Network Settings .......................... 41
Configure the Barracuda Web Application Firewall .................. 42
Activate Subscription Status. .................................................. 43
Update the Barracuda Web Application Firewall Firmware .......... 44
Update Attack, Virus, and Security Definitions ....................... 44
Configuring Your First Service .............................................. 45
Creating and Editing a Service .............................................. 45
Deploying an HTTP service .................................................. 45
Deploying an HTTPS service ............................................... 46
Configuring Load Balance in Proxy Mode ............................... 46
Configuring Security for Services ......................................... 47
Configuring Basic Security for a Service ............................... 47
Enabling Additional Security for a Service ......................... 48
Tuning .......................................................... 50
Introduction to Working with Logs ..................................... 50
Tuning using Web Firewall Logs and the Policy Tuner Tool ........ 50
Enforcing Security ......................................................... 51

Chapter 4 – Services ......................................................... 53
Overview .......................................................... 54
Securing an HTTP Web Site with HTTPS ............................ 55
Creating a Redirect Service .............................................. 56
Creating an FTP Service ................................................ 56
Creating an FTP SSL Service ......................................... 56
Creating a Custom Service ............................................. 56
Creating a Custom SSL Service ..................................... 57
Configuring Servers for a Service ..................................... 57

Chapter 5 – Customized Security for Web sites ..................... 59
Tuning Security Policies ................................................. 60
Request Limits ......................................................... 61
Cookie Security ....................................................... 61
URL Protection ......................................................... 62
Parameter Protection ................................................... 63
Cloaking .............................................................. 63
Data Theft Protection ................................................... 64
URL Normalization .................................................... 64
Global ACLs ........................................................... 65
Action Policy .......................................................... 65
Creating a New Security Policy ...................................... 66

Chapter 6 – Granular Security ............................................. 67
Security Profiles ......................................................... 68
Web Site Profiles ....................................................... 68
URL Profiles ........................................................ 68
Trusted Hosts .......................................................... 69
Chapter 7 – Adaptive Security

Overview
Layout of Adaptive Security
Exception Heuristics
Working with Exception Profiling
  Configuring Exception Profiling
  Learning from Trusted Hosts
  Learning Concurrently from Trusted and Non-Trusted Traffic
  Pending Recommendations
Working with Adaptive Profiling
  Configuring Adaptive Profiling
  Working with Navigation parameters
  Configuring URLs to be Excluded from Adaptive Profiling
Understanding Request and Response Learning
Viewing Newly Generated Profiles
Enforcing Learned Profiles
Using Strict Profile Checks to Enforce Positive Security
Recommended way to use the Adaptive Security feature

Chapter 8 – Advanced Security

Advanced Attack Protection
URL Policy
Session Tracking to Prevent Session DoS
Masking Sensitive Data in Logs

Chapter 9 – Advanced Tuning

User Defined Data Patterns
Internal Patterns
Creating and Using Custom Identity Theft Patterns
Creating and Using Custom Attack Types
Creating and Using Custom Input Types
Creating and Using Custom Parameter Class
Creating and Using Custom Response Pages
Creating Session Identifiers
Rate Control

Chapter 10 – Traffic Management

Traffic Management
Load Balancing
Monitoring the Health of the Server
Configuring a Backup Server
Content Rule
Configuring Caching
Configuring Compression
Chapter 11 – Web Site Translation ........................................ 107
Configuring Web Site Translation ........................................ 108

Chapter 12 – User Access Control ...................................... 111
Overview ........................................................................... 112
Steps to Configure Access Control to your Web site ................. 113
Configuring an Internal Authentication Database ....................... 113
Configuring an External Authentication Database ....................... 113
Creating an Authentication Policy ......................................... 114
Creating an Authorization Policy .......................................... 114
Single Sign-On (SSO) ....................................................... 115
Allowing/Denying Client Certificates ...................................... 117
Client Certificate Validation using OCSP ............................... 118

Chapter 13 – Keys and Certificates .................................... 121
Overview ........................................................................... 122
SSL Implementation and Configuration .................................. 123
Certificates ......................................................................... 124
Certificate Components ...................................................... 124
Creating a Test Certificate .................................................. 124
Saved Certificates .............................................................. 125
CSR .................................................................................. 125
Uploading a Certificate ........................................................ 126
Uploading a Trusted Certificate ............................................. 127

Chapter 14 – XML Firewall ............................................... 129
Web Services ....................................................................... 130
Web Services Implementation ................................................ 130
Web Service Vulnerabilities .................................................. 131
Web Service Protections ...................................................... 132
XML Firewall ..................................................................... 134
Tasks to enforce XML Firewall ............................................. 134
XML Validations .................................................................. 135
Import Schema/WSDL .......................................................... 135
Protected URLs ................................................................... 136
Advanced Web Service Validation .......................................... 137
XML Validation Settings ...................................................... 137
WS-I Basic Profile Assertions ............................................... 137
SOAP Validations ............................................................... 137

Chapter 15 – Monitoring, Logging, and Reporting ............... 139
Monitoring Barracuda Web Application Firewall ....................... 140
Viewing Performance Statistics ............................................ 140
Health Indicators for Services and Servers .............................. 141
Viewing System Tasks ........................................................ 141
Chapter 16 – Advanced Deployment .............................................. 151

Advanced Deployment .......................................................... 152
  Service Virtual Interfaces .................................................. 152
  Custom Virtual Interfaces ................................................. 152
  Static Routes ................................................................... 152
  Interface Routes .................................................................. 153
  VLAN (Virtual Local Area Network) .................................... 153

Chapter 17 – Network Firewall ..................................................... 155

Network Firewall Overview ...................................................... 156
  Configuring NAT for LAN Servers ...................................... 157
  Configuring Source Network Address Translations (SNATs) ... 158
  Configuring Access Control Lists (ACLs) ............................ 160

Chapter 18 – High Availability .................................................... 161

Creating a High Availability (HA) Environment ....................... 162
  Failover ........................................................................... 164
  Failback ........................................................................... 165

Chapter 19 – Administering and Maintaining your System .......... 169

Administrative Settings .......................................................... 170
  Controlling Access to the Administration Interface ................ 170
  Customizing the Appearance of the Web Interface ................. 170
  Setting the Time Zone of the System ................................... 170
  Configuring NTP Server for Time Synchronization ................ 170
  Enabling SSL for Administration ....................................... 171
  Maintaining the Barracuda Web Application Firewall ............. 173
  Backing up and Restoring your System Configuration ............. 173
  Updating the Barracuda Web Application Firewall Firmware .... 173
  Updating the Attack, Virus, and Security Definitions .............. 174
  Replacing a Failed System ............................................... 174
  Reloading, Restarting, and Shutting Down the System .............. 175
  Using the Built-in Troubleshooting Tools .............................. 175
  Using Session Recording .................................................... 176
  Using the Task Manager ....................................................... 176
Chapter 1

Introduction

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

- **Overview** ................................................................. 10
- **Core Functionality of the Barracuda Web Application Firewall** .... 12
- **Technical Support** ..................................................... 15
- **The Barracuda Web Application Firewall Models** ................. 16
- **How to Use This Guide** ............................................... 18
Overview

The Barracuda Web Application Firewall is an integrated hardware and software solution that offers comprehensive Web application and Web services security, access control, 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 Web interface.

The Barracuda Web Application Firewall 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 security vulnerabilities. It is designed to easily fit into any existing data center environment, rapidly securing and accelerating new and existing Web applications out-of-the-box. Deployment options include inline or offline modes.

Figure 1.1: Standard Barracuda Web Application Firewall Deployment

Incoming application layer protocols are terminated and validated. Secure Socket Layer (SSL) traffic is decrypted, and the data is normalized to detect attacks in spite of multiple encoding formats. Web site user access control checks are applied, then the application layer traffic is deep inspected for any vulnerabilities. Malicious traffic is denied, while safe traffic can be load balanced among configured Web servers.

Web Server responses are inspected and masked to prevent the release of sensitive data. Responses are cloaked to hide server specific information which could be used to refine future attacks. Data can then be cached and compressed to accelerate application delivery. Finally, the data is encrypted and sent to the Web application clients.

Figure 1.2: Barracuda Web Application Firewall Architecture
All data and actions taken are logged, and a rich set of real-time reporting and alerting features based on the logs, actions, and system state are provided. Security features and policies are updated by automatically downloading the latest definitions from the Barracuda Energize Update Servers.
Core Functionality of the Barracuda Web Application Firewall

The architecture of the Barracuda Web Application Firewall, depicted in Figure 1.2: Barracuda Web Application Firewall Architecture on page 10, shows the core functions which are introduced below.

Security

The Barracuda Web Application Firewall performs deep inspection of all requests and responses to and from the Web servers for any malicious content or security vulnerability. It includes both protection from known attacks and from unknown, or Zero Day attacks depending on configuration.

Advanced Web Application features provide instant and comprehensive security to HTTP based Web Applications, XML based Web Services, and SSL-enabled applications.

Bi-directional Content Inspection and Security

- **HTTP, HTTPS and FTP protocol compliance**: Terminates and validates all inbound requests enforcing compliance with the HTTP, HTTPS and FTP specifications respectively, and maximizing visibility, security, and control.
- **Integrated Anti Virus**: Scans for embedded viruses and malware in all file uploads to the Web application using the integrated anti-virus engine of the Barracuda Web Application Firewall.
- **Protection against common, high-visibility attacks**: Protects your Web applications from vulnerabilities in online Web forms, preventing injection attacks including SQL injections, OS command injections, and cross-site scripting.
- **Protection against attacks based on session state**: Protects your Web applications from any attacks based on session state, such as forms tampering or cookie tampering.
- **Brute Force Attack Prevention**: Prevents hackers from guessing passwords using readily available password dictionaries.
- **Application denial of service (DoS) protection**: Implements strict content length checks in individual requests and sets maximum thresholds for sessions and request rates to prevent malicious users from subverting your Web applications.
- **Fine-grained control**: Features fine-grain rules creation based on both HTTP requests and responses down to the level of individual HTML elements.
- **Full Regex Support**: Allows fine tuning of security policies by matching custom patterns, defined using powerful PCRE regular-expression support, to the HTTP requests and responses.
- **Outbound data theft protection**: Deep inspects all server responses to prevent leakage of sensitive information using provided default patterns (credit card data, social security numbers, etc.) or custom user configured patterns.
- **Web site cloaking**: Strips identifying banners of Web server software and version numbers and provides customizable HTTP error handling to defeat server fingerprinting attacks.
- **Active/Passive Mode**: Maintains throughput while tuning security policy. Passive mode of enforcement allows perceived threats to be passed to servers, but logged to assist in refining the security policy. Once refinements are sufficient, the Active mode enforces policy according to the action policy.
- **Action Policies**: Customizes the action taken for a defined attack group. Attacks can be handled by denial of response, redirection, or a response page with follow up actions that can block future attacks.
Automated Policy Tuning and Configuration

- **Exception Profiling**: Helps prevent strict security policies from denying genuine requests by automatically refining, or recommending refinements, to policies based on logged request and response traffic.
- **Adaptive Profiling**: Creates narrowly defined allowable profiles for requests based on trusted traffic, and denies all else to provide zero-day protection from never before seen attacks.
- **Policy Tuner**: Recommends policy refinements based on log entries to prevent a logged event from being considered an attack in the future.

Web Service Security

- **XML Firewall**: Provides 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.

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**: Customizes access policies 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.

Load Balancing

- **TCP, HTTP Load Balancing**
- **Application Content Routing**: Partitions and routes safe request traffic to configured Web servers based on the content types.
- **Persistence**: Routes user session traffic to the same Web server in a server farm.
- **Server Monitoring**
- **Choice of Scheduling Algorithms**

Application Acceleration and Assurance

- **Caching**: Reduces load on back-end Web servers and increases performance by caching Web content, avoiding repetitious requests to back-end Web servers.
- **Compression**: Reduces network traffic requirements by automatically applying GZIP compression to HTML content.
- **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.
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.

Administration Capabilities

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, Barracuda Web Application Firewall activity, Web services activity, network firewall activity and traditional Web logs.

Barracuda Energize Updates

To minimize ongoing administration associated with security and to provide your organization with maximum protection against the latest Internet threats, engineers at Barracuda Central monitor the Internet for trends in Web based attacks, collecting data from worldwide collection points. Barracuda Central, the 24/7 operations center operated by Barracuda Networks to monitor and block the latest Internet threats, automatically delivers Energize Updates to your Barracuda Web Application Firewall around the clock. These updates include firmware updates, including attack and security updates, content categories, and virus definitions for the most current protection against security threats. By identifying attack trends at an early stage, the team at Barracuda Central can quickly develop new and improved blocking techniques that are automatically made available to your Barracuda Web Application Firewall.

Role Based Administration

Access to system components can be limited by assigning administrative users to configured designated roles with the minimum required administrative privileges needed to complete assigned tasks.
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
- 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

<table>
<thead>
<tr>
<th>Feature</th>
<th>Model 360</th>
<th>Model 460</th>
<th>Model 660</th>
<th>Model 860</th>
<th>Model 960</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Web Application Firewall</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTTP/HTTPS/FTP Protocol Validation</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Protection Against Common Attacks</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Form Field Meta Data Validation</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Web Site Cloaking</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Response Control</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Outbound Data Theft Protection</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Granular Policies to HTML Elements</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Protocol Limit Checks</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>File Upload Control</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Logging, Monitoring and Reporting</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Brute Force Attack Protection</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Session Tracking</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Web Address Translation</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Rate Control</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>XML Firewall</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Adaptive Profiling</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Anti Virus Checks</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>Application Delivery and Acceleration</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High Availability</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SSL Offloading</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Application Content Routing</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Response Caching</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Response Compression</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Connection Pooling</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Load Balancing</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Feature</td>
<td>Model 360</td>
<td>Model 460</td>
<td>Model 660</td>
<td>Model 860</td>
<td>Model 960</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
</tr>
<tr>
<td><strong>User Access Control</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authentication and Authorization</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Single Sign-On</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>LDAP/RADIUS Integration</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td><strong>Management</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Web Based User Interface</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>XML RPC API</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Role Based Access</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
How to Use This Guide

1. Read Chapter 2, Web Application Firewall Concepts to understand the concepts related to the Web Application Firewall technology. You need to understand what the device can do, and decide which features of the product you want to use.

2. Read Choosing Your Deployment Mode on page 35 to understand which deployment option will provide the best overall security for your networks, while also enabling you to use the features that are most important to your installation.

3. Once you have chosen the deployment method, refer to Initial Setup on page 40 to install, connect, and activate the appliance on your network.

4. Configure one or more services using instructions in Configuring Your First Service on page 45.


Chapter 2

Web Application Firewall Concepts

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

- Deployment Mode Introduction ........................................... 20
- Service Types Mapped to Deployment Modes ..................... 21
- Extended Match and Condition Expressions Introduction ...... 23
- Security Policy .................................................................... 25
- Bidirectional Content Inspection and Security ...................... 26
- Authentication, Authorization, and Access Control .............. 29
- Application Acceleration .................................................... 30
- Load Balancing .................................................................. 30
- Network Firewall ............................................................... 28
Deployment Mode Introduction

An introductory explanation of the deployment modes is included here because the features available on the Barracuda Web Application Firewall depend on the deployment mode you choose. For further deployment mode information, see Choosing Your Deployment Mode on page 35.

The Barracuda Web Application Firewall can be deployed as a proxy, in full reverse proxy or in one-armed proxy configurations, or it can be deployed as a bridge path in bridge mode. When deployed as a proxy, the Barracuda Web Application Firewall is in-line with the Web Servers, and intercepts and inspects incoming and outgoing traffic, preventing attacks from reaching the Web Servers and preventing the leak of sensitive data to the requesting clients. This mode allows all security features to be employed, and prevents potential security threats from reaching the Web application. When deployed in a bridge mode, all data is passed simultaneously to the Web application and to the Web Application Firewall. Bridge mode deployment can be achieved with no changes to the network configuration of the Web Servers.

Deployment Terminology

The following table introduces deployment terms used by the Barracuda Web Application Firewall.

Table 2.1: Deployment terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real Server</td>
<td>Identifies the server (IP address, port) that hosts the Web application that will be protected by the Barracuda Web Application Firewall.</td>
</tr>
<tr>
<td>Virtual IP address (VIP)</td>
<td>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.</td>
</tr>
<tr>
<td>Services</td>
<td>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.</td>
</tr>
</tbody>
</table>

PROXY MODE

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Reverse Proxy</td>
<td>In full reverse proxy mode, the Barracuda Web Application Firewall is deployed in-line, using both the physical ports (WAN and LAN) of the device. <strong>This is the recommended configuration as it provides the best security.</strong></td>
</tr>
<tr>
<td>One-arm Proxy</td>
<td>Deployed in One-armed proxy mode, incoming and outgoing network traffic to the Application Firewall passes through the WAN port. A high level of security is achieved with this configuration, but because traffic is limited to the WAN port, network throughput is decreased.</td>
</tr>
</tbody>
</table>

BRIDGE MODE

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bridge Path</td>
<td>Deployed as an in-line Bridge Path, the Barracuda Web Application Firewall uses the same address for the VIP and back-end server, so data is passed through to the Web Application, including potential attacks, even as the security checks are performed. This configuration does not require changes to the existing network infrastructure</td>
</tr>
</tbody>
</table>
SSL Implementation

SSL establishes an encrypted link between clients and Web applications. This standard security technology also employs certificates for server and/or client authentication. The data between client and server, both incoming and outgoing, is encrypted. The Barracuda Web Application Firewall can be configured to provide encrypted transactions end to end. It can provide a configured server certificate when interacting with clients on the front-end, and it can validate a client certificate, and process encrypted requests and responses so that all security features are maintained on incoming data. For each configured server, the appliance can maintain an encrypted back-end data link providing a configured client certificate for back-end transactions, as well as requiring and validating server certificates. For a discussion of SSL technology and the Barracuda Web Application Firewall implementation of it, and for certificate authentication information, see Keys and Certificates on page 121.

Service Types Mapped to Deployment Modes

Depending on incoming data protocol and whether SSL is employed, you must choose an appropriate service, or Web application entry point, to configure Web application security. FTP protocol validation is only available in proxy mode, while HTTP protocol validation is available in proxy or bridge mode. Custom data can be passed through to the servers without processing in either mode.

SSL enabled services handle the encryption and decryption of SSL traffic properly so all other security features are maintained. These include the following services: HTTPS, FTPS, Custom SSL, and Instant SSL.

The Barracuda Web Application Firewall can be configured with the following service types:

Services Available In Proxy Mode (Full Reverse Proxy / One-armed Proxy):

HTTP
HTTPS
FTP
FTPS
Custom
Custom SSL
Instant SSL
Redirect

Services Available in Bridge mode (Bridge Path):

HTTP
HTTPS
Security for a Web site

The Barracuda Web Application Firewall implements Web site protection by associating a security policy and error logging level with a service. Default security provides an adequate amount of protection for the majority of Web sites. In addition, several other parameters can be set to expand Web site attack prevention. The following concepts clarify important considerations when deciding what security policy and logging levels to associate with your Web application.

Negative and Positive Security Models

The Barracuda Web Application Firewall employs both negative and positive security models. The negative model rejects requests matching known malicious signatures, preventing them from reaching the servers while allowing all others. The positive security model allows the creation, based on trusted traffic, of highly refined explicitly allowable data ranges or signatures, and denies any request failing to match them.

Negative Security Model

The negative security model relies on regularly updated security policies and patterns to identify malicious requests, preventing them from reaching the servers. Anything not matching the known attack patterns is allowed by default. Legacy network security solutions like IPS and IDS exclusively employed a negative security model. Negative security models require maintaining and regularly updating attack signatures, and may experience decreased performance when the list gets long. Signatures must be experienced as attacks before they are identified and added to the malicious request list. This implies a never before seen or “zero day attack” could defeat negative security.

Policy Tuner

With negative security in place, innocent requests can be wrongly interpreted as attacks, and the policy must be adjusted, or tuned, to allow valid requests to be distinguished from actual attacks. Policy Tuner assists in the fine tuning of security policies by integrating with the Barracuda Web Application Firewall log entries. For each log entry it provides a recommendation for adjusting the policy so the logged entry will not be considered an attack in the future. For more information refer to Tuning using Web Firewall Logs and the Policy Tuner Tool on page 50.

Positive Security Model

A positive security model (also known as "whitelist") is one that explicitly defines what is allowed, and rejects everything else. The creation of highly refined explicitly allowable security definitions, based on trusted traffic, prevents “zero day attacks,” because even never before seen attacks fail to match and are denied. An example is a FORM field containing an age, which, if restricted to the range 1-120 only, will always block new injection attacks without requiring updated signatures.

Exception Profiling

A tight positive security policy may sometimes deny genuine requests. Exception Profiling assists in refining the allowed settings, automatically reducing the number of false positives. You can enable exception profiling for desired Web site sections, identifying false positives and automatically refining the security policy rules for these site sections. The profiler can issue recommendations rather than automatically modifying the policy. A heuristics screen allows you to configure the exception profiling criteria for different violations.
**Adaptive Profiling**

Adaptive Profiling generates a positive security profile for your applications, 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 prevent anomalous traffic from generating the profile.

**Extended Match and Condition Expressions Introduction**

You can define the matching criteria for policies so security, data acceleration, and access control policies are applied only to appropriate data. The criteria can be as simple as specifying the Host and URL of matching data. To define more complex criteria you can use the extended match field, which allows you to combine (&& or ||) condition expressions, comparing expected data elements (for example Host, URL, Header Host, or userid) to values (for example userid eq 123456). Extended matching is only applied to requests or responses with a matching host and URL. For more information, refer to Appendix B on page 203, Extended Match and Condition Expressions.
The security architecture of the Barracuda Web Application Firewall associates security policies with Web services. A security policy has pre-set configured security settings which apply to any associated Web service. Security policies are shareable, so once a policy is created, it can be assigned to more than one Web service. A security policy is a set of rules which specify inspection criteria for input or output data, identifying malicious or vulnerable data. Security policies include mostly negative and some positive elements. For most Web sites, security policies sufficiently implement good Web application security.

Policies can be created or refined manually, or can be refined using automated tools. They include general URL and Parameter protection applied to all Web service requests. When required, more customized URL and Parameter protection can be implemented using Web Site Profiles, URL profiles, and parameter profiles.

Figure 2.1: Security Architecture: Security Policy applied to multiple services
**Security Policy**

When a service is created, default security, including Default security policy and log levels, are automatically in force for the service. Barracuda Web Application Firewall includes the following pre-configured security policies:

- Default
- Sharepoint
- OWA
- Oracle

When a Web service requires it, you can change the security policy assigned to the Web service, or refine a policy. Refined or newly created policies can be associated with any Web service.

Security policies specify limits to requests using negative and positive security. They include settings for cookie security, URL and parameter protection, cloaking, data theft protection, and URL Normalization. All of these elements are discussed in more detail below. In addition, security policies have associated Global Access Control Lists (ACLs) specifying the matching criteria which are compared to Web service requests. Requests that match are subject to corresponding actions (Deny, Allow, etc.). Every Security Policy also includes an Action Policy which specifies the Barracuda Web application Firewall response to various attack types.

**Security Profiles**

Using a Security Policy to determine whether a request should be allowed or denied is not always sufficient. Sometimes defending against potential attacks requires security customized for the Web site to further analyze a request before determining whether it should be allowed or denied. Web Site Profiles contain customized URL and Parameter profiles enforced for individual URL spaces of a Web application. Conceptually, the profiles are structured:

```
Web Site Profiles > URL Profiles > Parameter Profiles
```

Web Site Profiles are comprised of URL Profiles which have associated Parameter Profiles.

Security Profiles can implement positive security. That is, they can specifically define allowable request characteristics, denying requests which do not match. Alternatively, they can be used in conjunction with Security Policy settings combining positive and negative security.

Custom profiles, for specified URLs of a Web site, can be created and refined manually, or automatically using Exception Profiling or Adaptive Profiling. Refer to Adaptive Security on page 73 for details about creating, refining, and enforcing profiles.

**Web Customization**

Because Security Profiles are customized for a Web site, they are not applied to multiple Services or applications. In addition, Web site security can be customized using URL Policies and Web site Allow/Deny rules. URL Policies allow advanced security features to be enabled only for relevant URL spaces. For more information refer to URL Policy on page 88. Web site Allow/Deny rules are explained below.

Because increasingly complex security configurations require increased processing time, the best security solution uses the simplest combination of policies and profiles which can correctly distinguish destructive traffic from legitimate requests.
Bidirectional Content Inspection and Security

By deep inspecting incoming requests from clients, and responses from Web servers, the Barracuda Web Application Firewall can identify security threats. Inbound security requires termination and protocol validation, then data decryption and normalization, before identifying threats. Authentication and authorization are also performed before inspection.

For a detailed list of known attack types and the corresponding protection provided by the Barracuda Web Application Firewall, refer to Table F.1: Attack Types and Corresponding Barracuda Web Application Firewall Defense on page 233.

Inbound Security

The purpose of Inbound Security is to protect the Web application from potential attacks launched through client requests over the Internet.

Allow/Deny Rules

Allow/Deny Rules restrict access to specified URL spaces. These rules are applied before other security policies and override them. If a request’s URL matches an allow rule, then the request is passed through without applying other configured rules. On matching a deny rule, the request is immediately dropped. For more information refer to Allowing/Denying Specific URLs on page 70 and to Allowing/Denying Specific Headers on page 71.

Input Validation

The Barracuda Web Application Firewall allows validation of inputs 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. For more information refer to Customized Security for Web sites on page 59.

File Upload Validation

The Barracuda Web Application Firewall integrates with anti-virus software to scan all incoming requests and all uploaded files at the network periphery. Requests containing viruses are prevented from reaching the Web servers. The Barracuda Energize Updates maintains current virus signatures automatically, relieving administrators of worries about out of date definitions. For more information refer to Anti-Virus Protection for File Uploads on page 48.

Brute Force Protection

Brute Force Protection limits the number of requests allowed from a single client, or from all clients, to a URL space within a defined time interval. Offending clients are blocked from making further requests. This prevents attempts to guess usernames and passwords. It also blocks distributed Denial of Service (DoS) attacks. For more information refer to Preventing Brute Force Attacks on page 88.

Session Security

A Session refers to all server requests from a specific user on a single client. Session Tracking can limit the number of sessions originating from a particular client in a given interval of time. Limiting
the session generation rate by client IP address helps prevent session-based DoS attacks. For more information refer to *Session Tracking to Prevent Session DoS* on page 89.

**Rate Control**

Rate Control allows the configuration of a maximum number of allowed connections from a single IP address. If the threshold is exceeded, further connections are queued, and eventually blocked. Rate Control thus throttles requests to the Web Application preventing DoS attacks. For more information refer to *Rate Control* on page 95.

**Web Service Security**

XML data is used over HTTP to integrate systems using the SOAP messaging framework and interactive Web 2.0 XML applications. While XML-based applications are susceptible to the same attacks as other Web applications, they require additional defense from attacks specific to XML vulnerabilities. The XML Firewall module validates data to defend against attacks embedded in SOAP messages and XML data. Attacks are blocked before they can reach the servers. The Barracuda Web Application Firewall can validate SOAP messages, and enforce the WS-I Basic Profile Assertions for validating Web Services. For more information refer to *XML Firewall* on page 129.

**Outbound Security**

Outbound Security protects Web applications from exposing network, server, or user specific information which could assist hostile clients in launching future successful attacks against the Web application, Web servers, or their clients.

**Outbound Cloaking**

The Barracuda Web Application Firewall hides URL return codes, HTTP headers and content in error responses, as well as back-end IP addresses, making it less likely that hackers or worms will be able to launch a successful attack in the future using that information. For more information refer to *Cloaking* on page 63.

**Data Theft Protection**

Sensitive data patterns such as social security numbers or credit card information, once defined, can be protected from unauthorized exposure. For more information refer to *Data Theft Protection* on page 64.

**Web Site Translation**

Web site translation changes the internal codes, headers, and cookies in requests and responses so they are concealed or protected. For more information refer to *Configuring Web Site Translation* on page 108.
Security Violation Handling

Action Policies

Action Policies are associated with Security Policies and specify the action taken when a request violates a Security Policy in Active mode (see below). Action policies are set for defined attack groups, each with set attack actions. When a violation occurs, the action policy determines which attack action to take depending on the violation attack group. For more information, refer to Action Policy on page 65.

Active and Passive Modes

The Mode of security enforcement determines how the Barracuda Web Application Firewall handles the violations it matches to a Service, Policy, or Profile.

Passive Mode does not block or deny matching requests or responses, but only logs them for analysis and tuning purposes. Active Mode enforces the corresponding Action Policy defined for the violation attack group.

A Service in Passive mode only logs perceived attacks. This setting overrides all other more granular settings of enforcement Mode. A Service in Active mode responds to attacks according to more granular mode settings. Modes are properties of Services, Profiles, and Advanced Security (or URL Policy). Active vs Passive Mode can be applied to every service request, or more granularity to requests matching Allow /Deny rules (URL ACLs with corresponding actions), to profiles (Web site or URL), or to Advanced Security for Web sites. Consider using the passive mode (default setting at service creation) at initial deployment to reduce the impact of false positives. Then, change the policy to active mode as the incidence of false positives reduces to an acceptable level. As custom security is implemented, customized URL or Web site security is initiated in passive mode, even though the service remains in active mode.

Network Firewall

Network Firewall inspects network traffic, controlling access between any two networks according to configured rules or filters. Unauthorized network-level requests can be blocked at the network layer by configuring Access Control Lists (ACLs) enforced by the Barracuda Web Application Firewall. Source Network Address Translations (SNATs) map multiple internal IP addresses to a single external IP address. This IP address translation prevents exposing an internal address during routing. Network firewall rules specify matching criteria for packets and corresponding actions for matching packets. For more information refer to Network Firewall Overview on page 156.
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 integrated with corporate authentication databases such as LDAP or Radius, or with an internal user database created on the Barracuda Web Application Firewall itself. This lets you consolidate the authentication configuration and maintenance in a single centralized device for multiple back-end applications.

Authorization

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

Two Factor Authentication

Certificate validation on the Barracuda Web Application Firewall allows two factors to be verified before authenticating, for example, hardware generated token and username/password.

Client Certificate Validation

Client Certificates can be validated and an OCSP request can verify that the certificate holder remains a valid certificate holder. For more information refer to Client Certificate Validation using OCSP on page 118.

Single Sign-On (SSO)

SSO provides single sign-on functionality across single or multiple cookie domains. 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 to Single Sign-On (SSO) on page 115.
Application Acceleration and Assurance

Application Acceleration

The following features accelerate the application delivery and reduce the page load times improving the Web site browsing experience.

Content Routing

The Barracuda Web Application Firewall allows routing of application layer (Layer 7) content, based on the content type. Servers can be partitioned according to content type and process the requests more efficiently by directing them to the relevant server. For example, all image content can be routed to a designated server optimized for image delivery. A server optimized for CGI scripts execution can be set up to handle all script requests (e.g., *.cgi). For more information, refer to Content Rule on page 102.

Caching

Static outgoing content can be cached in local memory and served from there for subsequent requests, reducing the processing load on the Web servers. For more information, refer to Configuring Caching on page 104.

Compression

The system can dynamically compress outgoing content which can significantly reduce the transmission times, especially over slow network connections. Compression is also desirable when using verbose protocols over HTTP, like XML. For more information, refer to Configuring Compression on page 105.

Connection Pooling

Connection Pooling reuses the back-end server connections from a pool of established connections to deliver requests faster, and decrease connection set up and tear down overhead. It also reduces the server load, freeing resources to handle other important tasks.

Load Balancing

The load balancing module allows configuration of designated servers for a protected Web site, and allows adding or removing servers without interrupting existing traffic. This can improve Web site performance and add redundancy to handle server failures. For detailed information refer to Load Balancing on page 100.

Session Persistence

Session Persistence directs requests from a particular client to the same server, to allow the server to maintain session information (for example, shopping cart contents). Layer 7 session persistence is based on either the client IP address 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 server’s health can be monitored. If a server stops responding, its persistence requests may be directed to a healthy server, since requests are not sent to an unhealthy server. Monitoring may be performed by examining the server response codes in-band, or by generating independent requests out-of-band.

Load Balancing Algorithm

Round robin, weighted round robin, or least requests algorithms can be used to make 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 to Creating a High Availability (HA) Environment on page 162.
This chapter provides general instructions for deploying and installing the Barracuda Web Application Firewall and covers the following topics:

Four Steps to Getting Up and Running ............................................. 34
Choosing Your Deployment Mode ............................................... 35
Initial Setup ............................................................................. 40
Configuring Your First Service ............................................... 45
Configuring Security for Services .......................................... 47
Tuning ................................................................................. 50
Enforcing Security ................................................................. 51
Four Steps to Getting Up and Running

Choose and Implement a Deployment Mode

The Barracuda Web Application Firewall supports three different deployment modes. To select the preferred mode which meets both the varied needs of your network and the operational needs of your organization requires consideration of security, throughput, and ease of implementation. For a discussion of these factors, refer to Choosing Your Deployment Mode on page 35.

Configure a Service

Once the Barracuda Web Application Firewall has been deployed on the network, configure services to establish incoming and outgoing data paths to the protected Web servers. For more information refer to Configuring Your First Service on page 45.

Configure Security

The Barracuda Web Application Firewall provides robust default security to protect your Web applications. Once you have configured a service for the Web application, the service protects the Web application using default security settings in passive mode. Enforced security policies for the Web service can be selected to provide security specific to the Web application. For more information refer to Configuring Security for Services on page 47.

Tune and Enforce Policy

Security settings can be refined manually, or with exception profiling, before enforcing them. In addition, the Barracuda Web Application Firewall provides a Policy Tuner which integrates with the Web firewall logs to adjust or tune existing security. Once policies have been fine tuned, the administrator can switch the Web application running in passive mode into active protection mode. This blocks all traffic that violates configured security policies. For more information refer to Tuning on page 50.

This chapter explains each of the above steps using an HTTP / HTTPS service as an example.
Choosing Your Deployment Mode

Interfaces of the Barracuda Web Application Firewall

Before deploying the Barracuda Web Application Firewall, get acquainted with its physical interfaces:

- **WAN** - The WAN interface connects to the client side or Internet facing network.
- **LAN** - The LAN interface connects to the server side or Application facing network.
- **MGMT (Management)** - The MGMT interface connects to a separate out-of-band network for securely managing the Barracuda Web Application Firewall. Configuration can be done through the WAN port, but configuring your Barracuda Web Application Firewall through the MGMT port is highly recommended. **The WAN IP address and the Management IP address should be in different subnets to ensure that the Management interface is secure.** Figure 3.1 shows a secure configuration of the MGMT interface.
- **Virtual IP Address (VIP)** - An IP address that is configured for each service on the Barracuda Web Application Firewall. Depending on the mode of deployment, the VIP can be the same as or different from the Real Server IP addresses. In Bridge Path mode they are one and the same, while in Proxy mode they are different.

Services on the Barracuda Web Application Firewall can be used in the following three deployments:

- Reverse Proxy (Recommended) .......................................................... 36
- One-Armed Proxy .............................................................................. 37
- Bridge Path ........................................................................................ 38

Proxy deployments accept traffic on the virtual IP address and proxy the traffic to a private network behind the Barracuda Web Application Firewall. The Reverse proxy is a full proxy, through which all incoming and outgoing data to the application flow. The One-Armed proxy allows selected data to route through the Barracuda Web Application Firewall, while other data can be sent directly to the back-end servers. In a Bridge path deployment the Barracuda Web Application Firewall sits inline between the network firewall and the Web servers, inspecting configured traffic and bridging any other traffic to the servers.

The deployment you choose depends on the current network configuration and on the features of the Barracuda Web Application Firewall you want to use. Reverse Proxy is recommended for initial deployment, and for this reason the Barracuda Web Application Firewall is shipped configured in the Proxy mode. Depending on your needs, you can choose the mode of operation. To change the mode of operation, go to the BASIC > IP Configuration page, select the appropriate radio button under Operation Mode and click Save Changes.

**Note**

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

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

**Table 3.1: Deployment Options**

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Reverse Proxy</th>
<th>One-Armed</th>
<th>Bridge Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create secure path to Web servers.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Load Balance multiple Web servers while securing the Web application.</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Maximize network bandwidth (use both ports).</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>High availability for deployed applications.</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Integrate with existing enterprise load balancers.</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Establish multiple paths to servers for testing.</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Minimize change to existing network infrastructure.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Retain existing IP addresses.</td>
<td>X**</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

**Note:**

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

** - Server host can retain IP address, but DNS must be updated with the VIP for the Web site on the Barracuda Web Application Firewall. Direct access to the server IP address bypasses the Barracuda Web Application Firewall leaving the server vulnerable.

**Proxy Deployment**

**Reverse Proxy (Recommended)**

In a Reverse Proxy configuration, the Barracuda Web Application Firewall is deployed in-line using both physical ports (WAN and LAN) of the device. This configuration is recommended because it provides the best security and can utilize all traffic acceleration features. Deploying in Reverse Proxy requires changes to the existing network infrastructure.

Reverse Proxy deployment requires the WAN and LAN interfaces of the Barracuda Web Application Firewall to be on separate logical networks.

- The servers must be on a private network connected through a switch on the LAN port.
- The WAN port connects to the Internet through a switch where the publicly accessible IP addresses will be routed.
Each server in the private network is assigned a virtual IP address on the Barracuda Web Application Firewall (for example, 192.168.9.110, 192.168.9.120, and 192.168.9.130 in Figure 3.1). The virtual IP addresses should be accessible from the Internet, 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 and redirects it to the real server on the private network via the LAN port (for example, the VIPs map to real servers 10.10.10.10, 10.10.10.20, and 10.10.10.30 in Figure 3.1).

The following table describes the advantages and considerations of deploying your Barracuda Web Application Firewall in Reverse Proxy mode.

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full feature availability including Load Balancing and Instant SSL.</td>
<td>Requires network changes to Server IP addresses and DNS mappings.</td>
</tr>
<tr>
<td>Most secure deployment choice since back-end servers are completely isolated.</td>
<td>Deployment requires cut-over of live services.</td>
</tr>
<tr>
<td>Fast High Availability failover.</td>
<td>Network reconfiguration required to restore network to original state.</td>
</tr>
</tbody>
</table>

*Figure 3.1: Sample Reverse Proxy network layout*

**One-Armed Proxy**

One-Armed proxy deployment requires minimal changes to the existing infrastructure. Because a single port (the WAN port) is shared by both external and internal traffic passing through the Barracuda Web Application Firewall, the network throughput is reduced. One-Armed Proxy deployment allows the retention of alternate paths to access the servers. This deployment configuration can be used to load balance HTTP/HTTPS traffic at the application layer (for more information, see *Content Rule* on page 102), while letting SMTP and other traffic go directly to the
The following table describes the advantages and considerations of deploying your Barracuda Web Application Firewall in One-Armed Proxy mode.

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requires fewer network configuration changes than Reverse Proxy. Network infrastructure and partitioning unchanged.</td>
<td>Requires DNS, IP address changes.</td>
</tr>
<tr>
<td>Allows multiple access paths to servers for testing.</td>
<td>Decreased throughput with only one port (WAN) used.</td>
</tr>
<tr>
<td>Integrates easily with existing enterprise load balancers.</td>
<td>Potentially compromises server security by providing direct server access, unlike Reverse Proxy configuration.</td>
</tr>
</tbody>
</table>

**Figure 3.2: Sample One-armed network layout**

- **Management**
  - 172.10.10.5

- **One-Armed Proxy**
  - VIP1: 192.168.9.110
  - VIP2: 192.168.9.120
  - VIP3: 192.168.9.130

- **WAN**
  - 192.168.9.1

- **Servers**
  - 10.10.10.1
  - 10.10.10.2
  - 10.10.10.3

**Migration from One-Armed to Full Reverse Proxy**

Changing IP addressing schemes presents a challenge for some network environments. Deploying a One-Armed Proxy initially requires fewer changes to network settings, and allows administrators to validate operation while retaining access to the servers. Then migrating to Reverse Proxy requires just a few more network and configuration changes, to take advantage of the best security and other features.

**Bridge Path**

Bridge Path provides the easiest configuration scenario because it uses the same IP address for the VIP and back-end server, so it does not require you to change the server IP addresses or DNS mappings. You can place the Barracuda Web Application Firewall inline with the existing IP address infrastructure and add servers without changing IP addresses. In a 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. See **Figure 3.3** for a sample Bridge Path deployment configuration.
The following table describes the advantages and considerations of deploying your Barracuda Web Application Firewall in Bridge Path mode.

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimal network changes since the existing IP address infrastructure is reused.</td>
<td>Sensitive to broadcast storms and address resolution looping errors.</td>
</tr>
<tr>
<td>Real Servers keep existing IP addresses.</td>
<td>Less resilient to network misconfigurations.</td>
</tr>
<tr>
<td></td>
<td>Features like Load balancing, Instant SSL, and TCP pooling are not available.</td>
</tr>
</tbody>
</table>

Figure 3.3: Sample Bridge Path network layout

Practical Deployment Configuration Considerations

Practical points from the deployment discussion above:

- Using one interface, One-Armed proxy 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 change of IP addresses for the servers or for the client facing IP address for the Service is required. The bridge is transparent, so no existing services are disrupted. Disconnecting 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.
Initial Setup

Once you’ve chosen the right deployment for your network by considering the tradeoffs explained in *Choosing Your Deployment Mode* on page 35, you’re ready to install the Barracuda Web Application Firewall and configure the initial settings. The following steps are explained in this section:

- Prepare for the Installation ............................................................... 40
- Connect Barracuda Web Application Firewall to Network ............... 41
- Configure IP Address and Network Settings ................................. 41
- Configure the Barracuda Web Application Firewall ......................... 42
- Activate Subscription Status .............................................................. 43
- Update the Barracuda Web Application Firewall Firmware............. 44
- Update Attack, Virus, and Security Definitions ............................ 44

Prepare for the Installation

Before installing your Barracuda Web Application Firewall:

- Installing the Barracuda Web Application Firewall may require certain changes to the existing network depending upon the network configuration and the deployment mode of the Barracuda Web Application Firewall you choose. 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 in a 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* on page 42.
- Note the server IP address and TCP port of Web applications you want to protect.
- Verify that 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 2a. and 2b. 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

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

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

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 Front Panel Indicator Lights on page 141.

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 address configuration of the Barracuda Web Application Firewall.

3. Using your Tab key, select Change and press Enter to change the IP address 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 may be entered at this step or from the Web interface in **3b**). 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. The Web interface of the Barracuda Web Application Firewall is accessed from a Web browser on any machine that can communicate with the appliance, so it must be on the same network, or have routing set up accordingly.

**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.
   
   
   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 Address 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 address 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 to set the time zone correctly because it is used to coordinate traffic
distribution and timestamps appear 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 a Web administration session expiration length (in minutes), after which an
administrator will be required to log back in.

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 Subscription Status**

After installation, Barracuda Energize Updates and other applicable subscriptions must be activated
for the Barracuda Web Application Firewall to be fully enabled, and to continue to receive the latest
updates to all virus, attack, and security definitions from Barracuda Central. The Barracuda Energize
Updates service downloads these updates to your Barracuda Web Application Firewall on an hourly
basis.

**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 Web Application Firewall to enable functionality. (Click here to activate)

2. Click on the link and the **Product Activation** page opens in a new browser window.

3. 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, note the **Activation Code** displayed, which will need to be
manually entered 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 required, enter the **Activation Code** from step 3a by first entering the code, then
clicking **Activate**. This completes activation of the 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 subscription status does not change to **Current**, or help is needed to fill out the **Product Activation**
page, call a Barracuda Networks sales representative.
Update the Barracuda Web Application Firewall Firmware

Update the firmware on the Barracuda Web Application Firewall using ADVANCED > Firmware Update:

1. Read the release notes to learn about the latest features and updates provided in the new firmware version.

2. 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 your appliance 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.

3. Click Apply Now when the download completes.

4. 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, Virus, and Security Definitions

To apply the latest attack, virus, and security definitions:

In ADVANCED > Energize Updates, turning Automatic Update ON sets up repeated definition updates as needed. Any definition update can also be started manually when the Current Installed Version is not the Latest Version. The latest attack, virus, and security definitions will be downloaded and installed by initiating the Update. For more detailed instructions about setting intervals for update or manually updating definitions, see the online help.
Configuring Your First Service

A Service is a combination of a Virtual IP (VIP) address and a TCP port. Traffic arriving at the designated VIP and port is subjected to security checks configured for the service, and then passed to one of the Real Servers associated with that Service. The services supported by the Barracuda Web Application Firewall depend on the deployment mode you choose. See Service Types Mapped to Deployment Modes on page 21.

When you successfully create a service, it appears in the Services list with a green, orange, or red health indicator next to it. The service settings can be edited, including the service definition, the security settings for the service, and the SSL settings.

Using any service, the Barracuda Web Application Firewall acts as a server to which the client connects on the front end. On the back end, the Web service acts as a client to the real servers. The Barracuda Web Application Firewall fulfills each of these roles using the service and its associated configuration settings.

A Web application with HTTP or HTTPS input and output can be protected with an HTTP or HTTPS Service on the Barracuda Web Application Firewall, as described below.

Creating and Editing a Service

From the BASIC > Services page you can create and edit services. All services require a Service Name you choose, and the Virtual IP Address and Port of the incoming and outgoing data to the Web application. In addition, you must choose the Type of service you need to best protect your Web application, and the Real Servers where allowed requests will be sent to the Web application.

You can associate services into groups you define. This could help you identify ripple effects when later changes to the configuration are made. All services remain in a default group unless you specify otherwise.

Depending on the service type, other parameters may need to be specified which are explained below, or in Services on page 53 where deployment of various service types is discussed in detail. After entering the required parameters, click Add.

Only after successfully creating a service, so it is visible in the Services list on the BASIC > Services page, can the Basic Security and SSL settings (when applicable) be configured. Initially, the service is associated with the default security policy, and has a passive enforcement mode. These settings can be modified by clicking Edit next to the Service on the Services list. The Service page allows you to choose a different Web Firewall Policy and Log Level, and change the enforcement mode to Active, in addition to editing other security settings which are discussed in detail in online help.

Deploying an HTTP service

An HTTP service is the controlled entry point for an unencrypted HTTP Web application on the server. Create an HTTP Service by choosing HTTP as the service type, in addition to the parameters discussed in Creating and Editing a Service above.

To provide an encrypted entry point to an unencrypted HTTP Web application, please refer to Securing an HTTP Web Site with HTTPS on page 55.
Deploying an HTTPS service

An HTTPS service is a controlled entry point for an encrypted HTTPS Web application on the server. This service handles encrypted transactions between clients and the Barracuda Web Application Firewall, authenticating itself with certificates while acting as a server to requesting clients. Create an HTTPS service by choosing HTTPS as the type of service. In addition to the parameters discussed in Creating and Editing a Service on this page above, you must select a Certificate the service will present to clients. The available certificates are created or uploaded on the BASIC > Certificates page, and can be selected from the Certificate list.

For further information refer to Configuring SSL for SSL Enabled Services on page 55.

Note

To encrypt transactions between the Barracuda Web Application Firewall and the back-end servers, refer to Back-end SSL Server Configuration on page 57.

Configuring Load Balance in Proxy Mode

To configure Load Balancing on the Barracuda Web Application Firewall use Edit of the service on the BASIC > Services page. Load Balancing uses a configured load balancing Algorithm, Persistence Method, and Failover Method. The Algorithm logically determines the server to which a client’s first request should be routed. Persistence Method specifies how client connections which require maintained state information will continue to be routed to the same back-end server for each subsequent request. Failover Method specifies how to handle persistent client requests when the server that handled the original request is out of service. Other required settings for implementing the chosen Persistence Method are configured as a property of the service as well. See online help for detailed information about these settings.

Once the Service has been configured for load balancing, any server associated with the service must be configured for load balancing. Add servers for this service using Add on the BASIC > Services page in the Services section. Load balancing requires monitoring of server health so that persistence is maintained on the same server when possible, but in the event a server is out of service, the request is handled appropriately. Edit a server from the list of Servers associated with the service in the BASIC > Services list to configure load balancing server health checks. You can configure various thresholds, setting limits which determine when a server is considered out of service. In-band server health checks are done by default, checking the health of the server by monitoring user traffic. Additionally you can configure out-of-band checks, which initiate data transmissions independent of user traffic. Servers are marked as out-of-service when they fail to respond correctly. No further user traffic is sent to an out-of-service server. If other servers are available for load balanced requests, further requests will be routed to them. Otherwise, an error response is returned to the client.

Out-of-band server health checks allow for reinstatement of a server to in-service if the server subsequently responds to requests. In addition to out-of-band checks, you can configure application layer server health checks using HTTP requests to the server. Server responses are checked for the configured content to verify server health. See the online help on the BASIC > Services page, Services section, for details on configuring the server load balancing advanced settings.
Configuring Security for Services

Configuring Basic Security for a Service

At creation, Services default to a Passive mode of enforcement using the default Security Policy provided and updated by Barracuda Central. When refinements to default security are required for a Web application, a variety of options provide increasingly refined settings. Edit a Service from the BASIC > Services page to configure the following security settings.

Associating a Policy with Service

To refine the security provided for a Web site, the enforced Security Policy can be changed from the default policy to any existing security policy including any of the Barracuda Web Application Firewall provided and maintained security policies, or any previously saved customized policy. A list of available policies can be viewed in SECURITY POLICIES > Policy Manager under Policy Overview. By editing a service, from BASIC > Services, the Web Firewall Policy can be selected from a drop down list. For information on refining a security policy or creating a custom policy, refer to Tuning Security Policies on page 60.

Setting Log Levels for Service

A Web Firewall Log Level is a threshold indicating which error messages need to be logged for a service. This log level determines whether only the most urgent attack information, or less serious attack information, including even warnings or debug information, are written to the logs for this service. When the action policy associated with an attack requires logging, the log level determines whether the current attack meets the error message logging threshold. For more information on configuring Web Firewall Log Levels, see the online help.

Security Mode defaults to Passive

Services can provide security enforcement in Passive or Active mode. Services default to Passive mode, but the enforcement mode can be changed at any time. To do so, Edit the service from the BASIC > Services list, and change Mode to Active.

- Passive mode responds to offending traffic only by logging the event.
- Active mode performs the action configured in association with the perceived threat.

Normally, a service should remain in Passive mode while settings and throughput are verified, at which time you would switch the service enforcement mode to Active.

Enforcement mode settings are associated with services and more fine grained partitions of your Web application. A Passive mode for a Service overrides the mode settings of the more fine grained partitions of a Web Application. An Active enforcement mode for a Service does not prevent passive enforcement for a Web site partition, for example using the Web site Profile Mode or URL Profile Mode.

Trusted Hosts

A Group of Trusted Hosts can be listed with a Trusted Hosts Action. If a policy violation comes from one of the Trusted Hosts, the Trusted Hosts Action overrides the Action configured for other hosts. You can set the Trusted Hosts Action to:
Allow: Ignore attack,

Passive: Treat as if in Passive Mode; Log but allow request through, or

Default: Trusted hosts are treated the same as all other clients.

For more information, refer to Trusted Hosts on page 69.

Ignore case determines how, for this service, the URLs are matched to rules like URL ACLs and URL Profiles. When Yes, the case of the text in the URL is ignored for matching with any Barracuda Web Application Firewall rule.

With the Barracuda Web Application Firewall in proxy mode, the IP address of the actual client needs to be retained and passed on to back-end servers. The actual requesting client IP address is needed to properly log data and events concerning its request, but because the Web service acts as the client when forwarding requests to the back-end server, the Web service’s IP address appears to be the client IP address. Header Name For Actual Client IP specifies the name of the header in which the actual client IP address is stored and passed to the server.

**Rate Control**

To enforce Rate Control for this service, Edit the service from the BASIC > Services list, and turn Rate Control Status On. The Rate Control Pool must be configured from the available list. Use the default, or create a custom rate control pool using ADVANCED > Libraries. For more information on configuring Rate Control, refer to Rate Control on page 95.

**Enabling Additional Security for a Service**

**Brute Force Protection**

To enable Brute Force protection, edit the default URL policy from WEBSITES > Advanced Security. Brute Force attacks attempt unauthorized access by repeatedly bombarding the system with guessed parameters. For more information, refer to Preventing Brute Force Attacks on page 88. Detailed instructions for implementing Brute Force protection can be found in online help.

**Anti-Virus Protection for File Uploads**

The Barracuda Web Application Firewall integrates with anti-virus software to scan incoming requests for virus signatures. Virus scanning is enabled for a service using URL Policies. WEBSITES > Advanced Security shows existing services and the URL Policies applied to each. To enable virus scanning on a service, Add or Edit the applicable URL Policy. Before enabling virus scanning, create a customized URL policy which matches only to Web application spaces which upload information. Then enable virus scanning for the custom URL policy, so it applies only for those spaces. Requests containing viruses are blocked from reaching the Web servers. The Barracuda Web Application Firewall updates the virus signature database automatically using Barracuda Energize Updates so administrators need not manually update definitions. For more information on configuring Virus Scan protection refer to Enabling and Disabling Virus Protection on page 88.

**Data Theft Protection**

The Barracuda Web Application Firewall can detect and protect configured patterns of data from being revealed inadvertently by server responses. SECURITY POLICIES > Data Theft Protection allows configuration of Identity Theft data types for a Security Policy. You can enable protection for
specific URLs using WEBSITES > Advanced Security. Security Policy Data Theft settings are then enforced only for configured URLs.

While Barracuda Energize Updates provides a set of default protected patterns such as credit card and social security numbers, these can be expanded or customized, using ADVANCED > Libraries, to include other Web application specific data patterns needing protection from disclosure. Any configured pattern can be masked, or the response blocked altogether, if a protected pattern occurs in the server response. For more information about configuration of data theft protection, see Data Theft Protection on page 64.
Tuning

Tuning configuration on the Barracuda Web Application Firewall is the process of adjusting settings to achieve the desired balance between throughput and performance of the Web application, and protection of the application and hosting network from attacks or leaks.

The analysis of incoming and outgoing data and the Barracuda Web Application Firewall corresponding actions enable you to tune configuration and security settings so future data is handled appropriately.

Introduction to Working with Logs

The Barracuda Web Application Firewall records significant events in comprehensive logs. All actions and events are recorded in the Web firewall log, so reviewing this log reveals how the current settings are working, suggesting refinements for improved future processing. Use this log to verify and refine settings and policies.

Every entry in the Barracuda Web Application Firewall log has an associated error level indicating the severity of the event. The error level threshold for logging is configured by the administrator to control the volume of logs being stored and persisted. The logs are maintained in internal storage, and with the configuration of external syslog servers, are also saved in external storage.

For more detailed information, refer to Logs on page 145.

Tuning using Web Firewall Logs and the Policy Tuner Tool

Use the BASIC > Web Firewall Logs page to view the actions and events logged for the Barracuda Web Application Firewall. Built-in filters help locate log entries of interest. The display can be adjusted to your preference.

Each Web Firewall Log entry includes a Fix which can be applied to the rule violated by this entry using the Policy Tuner tool. The fix ensures the logged entry will no longer be considered an attack. Note the Service IP, Client IP, Mask, and Date of any Web Firewall Log entry which should be allowed in the future. Use the ADVANCED > Policy Tuner filter criteria to identify the entry. Selecting Apply Fix on a logged entry or entries adjusts the rule.

After initial tuning, the administrator should continue the process of tuning throughout the life of the service, according to observed traffic patterns. For further information about tuning the service, see Tuning Security Policies on page 60.
Enforcing Security

Once you have inspected and validated the logs of the Barracuda Web Application Firewall, tuning and verifying the settings and policies to apply to traffic, and are satisfied with your protection and throughput, switch the enforcement mode to Active. Traffic which violates the security settings or policies is now treated according to the action policy associated with the corresponding violation type.
This chapter describes various controlled entry points, or services, that can be created on the Barracuda Web Application Firewall to secure a Web application. To configure an HTTP or HTTPS service, refer to Configuring Your First Service on page 45. Detailed configuration of other service types are covered on the following pages:

- Securing an HTTP Web Site with HTTPS ........................................ 55
- Creating a Redirect Service .......................................................... 56
- Creating an FTP Service ............................................................... 56
- Creating an FTP SSL Service ....................................................... 56
- Creating a Custom Service ......................................................... 56
- Creating a Custom SSL Service .................................................. 57
- Back-end SSL Server Configuration ........................................... 57
Overview

A Service is an entry point to a Web application specified by the combination of a Virtual IP (VIP) address and a TCP port. Traffic arriving at the designated VIP and port is subjected to validation and security checks configured for the service, and then passed to one of the Real Servers associated with that Service. The type of service to configure depends on the protocol of the transferred data and desired validations. For example, HTTP and FTP protocols can be terminated and validated by services of those types. Furthermore, encrypted data must be handled by a service which can decrypt, do security checks, and re-encrypt transferred data, in order to fully utilize the validation features of the Barracuda Web Application Firewall. Custom services forward data without protocol validation allowing non-http non-ftp data to be passed through the Barracuda Web Application Firewall to the Web application.

Services act as servers to which the client connects on the front-end. On the back-end, the service acts as a client to the real servers. The Barracuda Web Application Firewall fulfills each of these roles using the service and its associated configuration settings.

For detailed instructions to create and edit a service, refer to Creating and Editing a Service on page 45, which explains configuration of features common to all services. Various service types also require additional fields specific to the data they process, which are discussed in this chapter, or on the referenced pages.

The following services handle incoming and outgoing unencrypted or encrypted HTTP data to the Web application:

- HTTP service
- HTTPS service

For more information about creating an HTTP or HTTPS service, refer to Deploying an HTTP service or Deploying an HTTPS service on page 46.

The following services implement off-loaded front-end encryption for an HTTP Web Application:

- Instant SSL service
- Redirect service

The following services handle unencrypted and encrypted (SSL) non-HTTP data:

- FTP service
- FTP SSL service
- Custom service
- Custom SSL service

Note: In Bridge All Traffic mode, only HTTP and HTTPS services can be created. Because the same IP address is used for both the VIP (service) and the server, the Real Server field cannot be configured in this mode. To change the operation mode, go to the BASIC > IP Configuration page, and set the Mode of Operation to Proxy under Operation Mode section.

Creating SSL Enabled Services

To use SSL you need to select a Certificate which the service presents to authenticate itself to a client. Certificates are created or uploaded using the BASIC > Certificates page, where you can add a certificate to the available Certificate list. You choose your Service certificate from this list before using Add to create the service. You can change the certificate to any available certificate by clicking Edit for the service in the Services list.
Configuring SSL for SSL Enabled Services

By clicking Edit next to the service on the Services section of the BASIC > Services page, you can configure the SSL supported protocols. SSL status defaults to On for a newly created SSL enabled service. If you set Enforce Client Certificate to Yes, any request from a client without a certificate immediately terminates. If you set Enable Client Authentication to Yes, the Barracuda Web Application Firewall authenticates the client with the presented certificate, or authenticates the client using an authorization policy configured through ACCESS CONTROL > Authorization. Authentication of certificates uses selected trusted certificates.

SSL enabled services allow configuration of encryption between the requesting client and the Barracuda Web Application Firewall. To encrypt transactions between the appliance and the back-end servers, refer to Back-end SSL Server Configuration on page 57.

Securing an HTTP Web Site with HTTPS

Creating a service of type Instant SSL provides an encrypted entry point to an unencrypted HTTP Web application, offloading SSL encryption and authentication for the application to the Barracuda Web Application Firewall. Using Add on BASIC > Services to create an Instant SSL service 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 HTTP requests to the HTTPS service on port 443. See Creating SSL Enabled Services above for instructions to configure the SSL Certificate. Domain is the main domain identifying links to be rewritten from 'http' to 'https'. At least one Secure Site Domain should be specified before adding the service. More domains can be added using the Instant SSL section on the Services page, as described below. See the example in Figure 4.1 where http://www.barracuda.com is specified, so those links found in the outgoing response will be rewritten to https://www.barracuda.com.

Figure 4.1: Instant SSL Service

Clicking Add creates both services, one Redirect and one HTTPS.

Edit the HTTPS service Instant SSL section to verify Instant SSL Status is On, and to specify additional Secure Site Domains requiring links to be converted from http: to https:.

Sharepoint Rewrite Support is relevant only if an Instant SSL service is created to protect a Microsoft SharePoint application. Normally, an Instant SSL Service rewrites the HTTP links in the responses to HTTPS using HTML tags, like href. But Sharepoint applications also insert hyperlinks outside the
basic HTML tags which require extra processing. Turn SharePoint Rewrite Support On to ensure HTTP links outside HTML tags are also properly rewritten to HTTPS.

To Configure SSL for the HTTPS service, refer to Configuring SSL for SSL Enabled Services on page 55. For more information on SSL implementation between clients and the Barracuda Web Application Firewall, refer to SSL for Client to Barracuda Web Application Firewall Transmissions on page 123.

Creating a Redirect Service

A redirect service is a non-SSL service that redirects all HTTP requests to another service. Since the only purpose of this service is to redirect to an existing service, the Real Server IP address cannot be specified. To create a Redirect service, select Redirect as the type of service. If you need additional instructions to create the redirect service, see Creating and Editing a Service on page 45.

Creating an FTP Service

An FTP service is the controlled entry point for an unencrypted FTP Web application on the server. To create an FTP service, select FTP as the type of service. For additional instructions to create an FTP service, see Creating and Editing a Service on page 45. To configure FTP attack protection, see Configuring FTP Attack Prevention below.

Creating an FTP SSL Service

An FTP SSL Service is a controlled entry point for an encrypted FTP Web application on the server. This handles encrypted transactions between clients and the Barracuda Web Application Firewall and authentication with certificates. To create an FTP SSL service, select FTP SSL as the type of service. If you need additional instructions to create an FTP SSL service, see Creating and Editing a Service on page 45. This is an SSL enabled service, so see Creating SSL Enabled Services on page 54 for instructions on selecting a certificate, and Configuring SSL for SSL Enabled Services on page 55 for configuring the SSL settings on the newly created service.

For more information on SSL implementation of secured transmission between clients and the Barracuda Web Application Firewall, refer to SSL for Client to Barracuda Web Application Firewall Transmissions on page 123.

Configuring FTP Attack Prevention

The Barracuda Web Application Firewall allows configuration of FTP verbs which are explicitly allowed using WEBSITES > FTP Security. Disallow all other verbs by enabling FTP Attack Prevention for an FTP Service or FTP SSL Service, using Edit on the BASIC > Services page.

Creating a Custom Service

A Custom Service allows the Barracuda Web Application Firewall to process any application layer protocol over TCP. 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 create a Custom service, select **Custom** as the type of service. If you need additional instructions to create a Custom service, see *Creating and Editing a Service* on page 45.

**Creating a Custom SSL Service**

A **Custom SSL Service** is a controlled entry point for an encrypted non-HTTP non-FTP Web application on the server. This handles encrypted transactions between clients and the Barracuda Web Application Firewall and authentication with certificates. If you need additional instructions to create a custom SSL service, see *Creating and Editing a Service* on page 45. This is an SSL enabled service, so see *Creating SSL Enabled Services* on page 54 for instructions on selecting a certificate, and *Configuring SSL for SSL Enabled Services* on page 55 for configuring the SSL settings of the Custom SSL service.

For more information on SSL implementation of secured transmission between clients and the Barracuda Web Application Firewall, refer to *SSL for Client to Barracuda Web Application Firewall Transmissions* on page 123.

**Configuring Servers for a Service**

Once you create a Service you can add specific servers to it using **Add** next to the Service in the **BASIC > Services** list.

**Back-end SSL Server Configuration**

To use SSL encryption between the Barracuda Web Application Firewall and a server, **Edit** the server from the list of Servers in the **BASIC > Services** list and set **Server uses SSL** to **Yes** in the **SSL (Server)** section. The Barracuda Web Application Firewall can validate the server certificate using trusted certificates. If the server provides a self signed certificate, **Validate Server Certificate** should be **No**. You can also configure a client certificate for the service to present to the server. This certificate configuration is needed if the server requires client authentication, because the service acts as a client to the back-end server when it forwards requests. Online help provides more detailed instructions for configuring these settings. For more information on SSL implementation between the Barracuda Web Application Firewall and back-end servers, refer to *SSL for Barracuda Web Application Firewall to Server Transmissions* on page 123.
This chapter describes how to tune the default security policies for a service, or to create a custom Security Policy for the service to use. For information on configuring Basic Security (including Policy and Log Levels) for a Service, refer to *Configuring Basic Security for a Service* on page 47. The following topics are covered:

- Tuning Security Policies ................................................................. 60
- Creating a New Security Policy..................................................... 66
Tuning Security Policies

The Barracuda Web Application Firewall provides a variety of security policies to protect Web sites and Web services. For information about the security policies provided and maintained on the Barracuda Web Application Firewall by Barracuda Energize Updates, refer to Security Policy on page 25. To configure basic security for a new service, or to change the service Security Policy applied for this service, refer to Configuring Basic Security for a Service on page 47.

Security Policies define matching criteria for requests, and specify what actions to take when a request matches. All policies are global and they can be shared among multiple services configured on the Barracuda Web Application Firewall, as depicted in Figure 2.1: Security Architecture: Security Policy applied to multiple services on page 24.

When a Service requires customized settings, the provided security policies can be tuned, or customized policies can be created. Each policy is a collection of nine sub-policies. Modify a policy by editing the value of the parameter(s) on the sub-policy page. Details of each of the nine sub-policies are on the referenced page:

- Request Limits.................................................................................... 61
- Cookie Security.................................................................................. 61
- URL Protection................................................................................... 62
- Parameter Protection........................................................................ 63
- Cloaking.............................................................................................. 63
- Data Theft Protection........................................................................ 64
- URL Normalization.......................................................................... 64
- Global ACLs...................................................................................... 65
- Action Policy...................................................................................... 65
Request Limits

The SECURITY POLICIES > Request Limits page displays the configured maximums for request fields.

When Enable Request Limits is Yes, Request Limits are used to enforce size limits on HTTP request header fields. The requests that have fields larger than the specified maximums are dropped. Properly configured limits mitigate buffer overflow exploits, preventing Denial of Service (DoS) attacks. Request Limits are enabled by default, and the default limits are set to prevent potential buffer overflow attacks. Edit the fields to change one or more of the default values if necessary. Save Changes after modifying any field.

For more information on modifying Request Limits, refer to SECURITY POLICIES > Request Limits online help.

Cookie Security

Cookies provide a mechanism to maintain service state information between otherwise stateless transactions. Cookies store user preferences, shopping cart items, and can include sensitive information like registration or login credentials. If a cookie can be viewed or changed, the system is vulnerable to attack and any sensitive information can be stolen.

To set and apply security features to HTTP cookies sent from the Web servers to clients, use SECURITY POLICIES > Cookie Security. The Barracuda Web Application Firewall’s Cookie Security policy provides two cryptographic techniques. Tamper Proof Mode can be set to:

- **Encrypted**: Encrypts the cookie data
- **Signed**: Attaches a digital signature to the server generated cookie.

When the Tamper Proof Mode is None, cookies pass through the Barracuda Web Application Firewall without cryptographic checking.

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 and encrypts or signs the cookie before delivering it to the client. When a subsequent request from the client returns this cookie, the Barracuda Web Application Firewall intercepts the request and decrypts it or verifies its signature. If the cookie is intact, the Barracuda Web Application Firewall forwards the original cookie to the server. If the cookie has been altered, decryption or signature verification fails, and the Barracuda Web Application Firewall removes the cookie and forwards the request to the server without the cookie.

Encryption prevents the viewing of or tampering with cookies, but also prevents the client from access to cookie values. When the client needs access to the cookie values, signing is the preferred option. When signing cookies, the Barracuda Web Application Firewall actually forwards two cookies to the client browser; one plain text cookie and one signed cookie. When a subsequent request from the client returns the cookies, if either cookie is altered, signature verification fails, and the Barracuda Web Application Firewall removes the cookies before forwarding the request to the server.

Cookie Security Interaction with other Security Features

Both Encryption and Signing may change the cookie length, but only signing changes the number of headers in the message, appending one or more headers to the forwarded message. If the SECURITY
**POLICIES > Request Limits** constrains the number or length of HTTP headers, signing or encrypting a cookie may violate the request limits. Messages thus rejected are logged as ‘Cloak’ under Action on the **BASIC > Web Firewall Logs** page.

### Configuring Cookie Security

Cookie Security provides the following configurable settings:

- **Cookie Max Age**: Configured maximum cookie age after which cookies are automatically expired. The unique cookie owner can be configured to be determined in various ways, including using custom headers. Cookies can be refused from clients unless the client uses SSL. The client can be refused any access to the cookies. Cookie security can be configured to allow unrecognized cookies for a configured number of days to allow cached old cookies in clients whose access to the Web application preceded the installation of the Barracuda Web Application Firewall. Cookies known to be appropriately modified by client side scripts can be explicitly exempted from the cookie security checks.

After altering Cookie Security, save changes, then **Edit** the affected service from **BASIC > Services**. Select the relevant policy for **Web Firewall Policy**, and verify **Active** as the **Mode** before saving changes to the service **Basic Security**. Cookie security using the new settings is now enforced.

For more detailed configuration instructions, refer to online help from **SECURITY POLICIES > Cookie Security**.

### URL Protection

URL requests and embedded parameters in them can contain malicious script. Attacks embedded in URL requests or their parameters are executed with the permissions of the executing component. Injection of operating system or database commands into the parameters of a URL request, cross site scripting, remote file inclusion attacks, and buffer overflow attacks can all be perpetrated through unchecked URL requests or their parameters.

Here is an example of malicious script within a URL Request:

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

Defense from these attacks is achieved by restricting the allowed methods in headers and content for invoked URL requests, restricting the number of request parameters and their lengths, limiting file uploads, and specifying attack types to explicitly detect and block. (Attack types are configured on **ADVANCED > Libraries** or **ADVANCED > View Internal Patterns**.) URL Protection uses a combination of these techniques to protect against various URL attack types. URL Protection defends the service from URL request attacks when no URL Profile is configured to do it. To view and configure URL Protection settings, use **SECURITY POLICIES > URL Protection**.

### Limiting Allowed Methods in HTTP Headers and Content

While GET and POST are the predominant methods used by Web servers for information access, HTTP allows several less known methods*:

- HEAD
- GET
- POST
- PUT
- DELETE
*RFC 2616 describes the above HTTP methods in detail.*

The OPTIONS command allows clients to determine which methods the Web server supports. Some methods allow modification of stored files, stealing of user credentials, or bypassing environment level access control checks. URL protection allows an explicit way to specify allowed or disallowed methods in URL calls. Disallowing PUT, DELETE, and TRACE is recommended. The allowed request content-types also need to be carefully restricted to prevent similar security threats.

For more information about configuring URL Protection, see SECURITY POLICIES > URL Protection online help.

### Parameter Protection

To protect a service from attacks using the parameters of a URL query string or form POST parameters, use SECURITY POLICIES > Parameter Protection. Parameter Protection defends the service from Parameter based attacks when no Parameter Profile is configured to do it.

Parameters that contain special characters may have SQL or html tagging expressions embedded in them. Embedded SQL keywords like "OR," "SELECT," or "UNION" in a parameter, or system commands such as "xp_cmdshell" can exploit vulnerabilities. These attack patterns can be configured in Parameter Protection, and compared to requests and responses. If a parameter matches, the corresponding request or response is not processed.

Parameter protection is configured as part of a security policy. Enable Parameter Protection should be Yes to enforce parameter protection. List the Denied Metacharacters for the policy. Set the maximum length of a parameter using Max Param Value Length. Enter the file extensions allowed for uploaded files using File Upload Extensions. Set the maximum size for individual files that can be uploaded in one request in Max Upload File Size. Select Blocked attack types, or configure and select custom blocked attack types (configured on ADVANCED > Libraries). Explicit exceptions to the blocked patterns can be specified in Exception Patterns using pattern names configured on ADVANCED > Libraries or ADVANCED > View Internal Patterns. Parameters may be exempted from the Parameter Protection checks using Ignore Parameter. Saving the changes adds the configured parameter protection to the Security Policy.

For more information, refer to the online help.

### Cloaking

Cloaking prevents hackers from obtaining information that could be used to launch a successful subsequent attack. HTTP headers and return codes are masked before sending a response to a client. Enable a security policy to cloak a Web site or Web service using SECURITY POLICIES > Cloaking.  

Enabling Filter Response Header removes listed Headers to Filter (for example, Server) from responses. Enabling Suppress Return Code cloaks client error (status code 4xx) and server error (status code 5xx) messages from responses.

All Web site 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. Saving the changes adds the configured cloaking settings to the Security Policy.

For more information, refer to the online help.
Data Theft Protection

Data theft protection prevents unauthorized disclosure of confidential information. Configuring data theft protection requires two steps:

- Specify any at risk data elements handled by the Web application using Security Policy.
- Enable protection of these elements where needed, using URL Policy.

Sensitive data elements may require masking to prevent their unauthorized disclosure, or requests containing sensitive data may be blocked altogether. Using Security Policy, you can configure any sensitive data elements which may need protection, along with the desired way to handle them. These settings can then be used by any service associated with the security policy. URL policies applied to narrowly defined URL spaces requiring this protection can individually enable it as needed. Other URL spaces operate without unnecessarily incurring the processing hit. To optimize performance, enable data theft protection only for parts of the site known to carry sensitive information.

Use **SECURITY POLICIES > Data Theft Protection** to edit the security policy and specify at risk data. Set the **Data Theft Element Name**, **Identity Theft Type** (using standard or custom data types) and **Action** (Cloak based on configuration, or Block). Then use **WEBSITES > Advanced Security** to **Enable Data Theft Protection** by using **Edit** on the applicable URL policy of the Service. More information is available using online help.

Note

Data Theft Protection will only be enforced for URLs matching a URL policy for which **Enable Data Theft Protection** on **WEBSITES > Advanced Security** is **Yes**. The settings enforced for the URL policy are configured using **SECURITY POLICIES > Data Theft Protection**.

URL Normalization

The Barracuda Web Application Firewall normalizes all traffic before applying any security policy string matches. For HTTP data, this requires decoding Unicode, UTF, or Hex to base text, to prevent disguised attacks using encoding formats for which string matches are not effective.

Normalization 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. ASCII is the default. Additional checks to prevent path traversal and path disclosure attacks can be set using **SECURITY POLICIES > URL Normalization**.

In some cases multiple character set encoding is needed, as for a Japanese language site which might need both Shift-JIS and EUC-JP encoding. To add character set encoding, set the **Detect Response Charset** parameter to **Yes**. All response headers will be searched for a META tag specifying the character set encoding type and any supported types will be added dynamically.

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.

<table>
<thead>
<tr>
<th>Escape</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%5C</td>
<td>%5C Normal UTF-8 escape of the backslash character.</td>
</tr>
<tr>
<td>%255C</td>
<td>%25, the escape for % followed by 5C.</td>
</tr>
<tr>
<td>%35%63</td>
<td>The % character followed by %35, the escape for 5, and %63, the escape for C.</td>
</tr>
<tr>
<td>%25%35%63</td>
<td>The individual escapes for %, 5, and C.</td>
</tr>
</tbody>
</table>

### Global ACLs

Global ACLs (URL ACLs) are strict allow/deny rules shareable among multiple services configured on the Barracuda Web Application Firewall. They are associated with configured Security Policies. Use **SECURITY POLICIES > Global ACLs** to add or modify Global ACLs.

Each global ACL is created with **URL ACL Name**, **URL Match**, and **Extended Match** criteria to precisely define which URLs are subject to the rule. **Extended Match Sequence** clarifies which match takes precedence when multiple rules match. The **Action** specifies what action to take on matching data, including Allow, Deny, or Process. When the **Action** or **Deny Response** requires redirection, the **Redirect URL** specifies the redirection location. **Response Page** specifies the response to send when **Deny Response** requires one.

**BASIC > Policy Manager** provides a **Policy Overview** of existing Security Policies and the number of Allow and Deny URL Policies associated with each.

For more information, refer to online help.

### Action Policy

Specify what action to take when a violation occurs using **SECURITY POLICIES > Action Policy**. A set of attack groups and associated attack actions can be viewed. 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

Modify the attack action by selecting an attack action name and using **Edit** to enter the desired **Action**, **Deny Response**, and other **Options**. Use online help for more information about configuring attack actions.
Creating a New Security Policy

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

Use SECURITY POLICIES > Policy Manager Create New Policy to enter the new policy name and Add. The new policy with default values appears in the Policy Overview list. To modify a policy, go to the desired sub-policy page and select the policy from the Policy Name drop-down list. Change the value of the parameter(s) and Save Changes to save and activate the new settings. Delete removes a policy. All policies can be removed except the default policy.

All services use the default policy at creation. If desired, change the policy for a service by selecting the service and using Edit under Actions from the BASIC > Services page. Once the service page opens, select the desired policy from the Web Firewall Policy drop-down list under Basic Security to edit values. Save Changes to save and activate the new settings.
This chapter describes how to further refine security for your Web site using Security Profiles, implementing allow/deny rules, and specifying trusted hosts. To better understand how Profiles fit within the architecture of the Web Application Firewall, refer to Security Architecture on page 24. This chapter covers the following topics:

- Security Profiles ................................................................. 68
- Trusted Hosts ................................................................. 69
- Allowing/Denying Specific URLs ........................................... 70
Security Profiles

Security profiles are customized settings for a Web site partition. Because they are narrowly defined for the Web application, they allow the configuration of positive security, disallowing any request not fitting within the profile. Profiles can also be used in conjunction with negative security if desired. Web site Profiles contain URL profiles tailored to more granular URL partitions of the Web site. URL profiles in turn are associated with Parameter profiles for the parameters of the individual URL.

See Security Architecture on page 24 for more information about how profiles fit into the Barracuda Web Application Firewall security enforcement.

Web Site Profiles

Configure Web Site Profiles using WEBSITES > Web Site Profiles. When a new service is added, a default Web Site Profile is created and enabled for that service. To view and modify the default settings, use Service > Web Site on the WEBSITES > Web Site Profiles page. Select the service and use Edit to change default values.

When configuring a Web Site Profile, setting Use Profile to Yes allows the use of URL Profiles and Parameter Profiles to validate requests to that service. To implement strict positive security, set Strict Profile Check to Yes, which denies any request for which there is no matching URL Profile or Parameter Profile. Setting Strict Profile Check to No subjects the incoming request to negative security checks and also to profiles if they exist. Explicit Exclude URL Patterns can be listed (Example: *.html,*.htm,*.jpg, *.gif,*.css,*.js) and exceptions to the excluded patterns can be defined in Include URL Patterns (excluding, for example, all html requests but the exception). Allowed Domains defines the domain attribute of the session cookie to allow a broader set of domains valid cookie access without the cookie being rejected by the Barracuda Web Application Firewall.

The enforcement Mode of a Web Site Profile can be set to Passive, Active, or Learning. If the Mode of the Service is set to Passive, all attacks on that service are logged but allowed to pass through the Barracuda Web Application Firewall, overriding the Mode setting in the Web Site Profile for that service. If the Mode of the Service is set to Active, the Web Site Profile can override the setting to Passive for the Web site. When the Web Site Profile Mode is set to Active, an attack is allowed or denied based on the Mode setting of the matching URL Profile.

Learning mode uses the settings of WEBSITES > Adaptive Profiling to learn from allowed requests and responses. Learning mode is used to develop profiles to implement positive security. For more information on filling out fields of the Web Site Profile, see online help.

URL Profiles

URL Profiles are used to validate incoming requests to the service. Matching a request to a URL profile depends on the URL and Extended Match settings. The URL Profile enforcement Mode is overridden by a Passive mode setting of either the Service or Web Site Profile. If both Service and Web Site Profile modes are set to Active, the Mode of the URL Profile determines passive or active enforcement of associated actions for matches to the profile. Services can have multiple URL profiles. URL Profiles can be created automatically using WEBSITES > Adaptive Profiling with the URL Profile Mode set to Learning. For more information on using Adaptive Profiling to create URL Profiles, refer to Working with Adaptive Profiling on page 79.

To manually create a URL Profile, use Add URL from URL Profiles in WEBSITES > Web Site Profiles. Provide the URL Profile Name and set the Status to On to enable matching to it. Explicitly allowed
query strings, content types and methods can be specified for this URL. Maximum Parameter Name and Content lengths can be set. **Maximum Upload Files** threshold can be set for the URL. List **Allowed Methods** in requests to the URL. List **Allow Content Types** in POST body for the URL. Choose whether to **Allow Query String** in the URL. **CSRF Prevention** (only applicable if Parameter Profile exists) and **Hidden Parameter Protection** for forms, or forms and URLs can be enabled. **Blocked Attack Types**, both provided and customized can be selectively enforced. Exception Patterns list explicit exceptions to blocked patterns using configured names from ADVANCED > Libraries or ADVANCED > View Internal Patterns. For more information on manually creating a URL Profile, see online help.

**Parameter profile**

Parameter Profiles validate the parameters of requests coming into the service. Multiple parameter profiles can be created for a URL. Select the URL Profile from WEBSITES > Web Site Profiles Directories list, then click **Add Param** and fill in the Create Parameter Profile dialog box. Enter the Parameter Profile Name and set Status to **On** to validate requests using this profile. Specify the parameter name in Parameter. Select the Parameter Type. Values allows the entry of fixed strings to match against the parameter's value, if the parameter Type is set to **Global Choice**. Select Parameter Class, to be enforced on the parameter. Select the Custom Parameter Class to be used if the Parameter Class is set to **Custom**. Refer to Creating and Using Custom Parameter Class on page 94 for more information. Set the Max Value Length and specify whether the Parameter is Required in all requests or if the Barracuda Web Application Firewall should Ignore the parameter, doing no validation on it at all. The **Max Instances** threshold puts a limit on the number of times a parameter is allowed. Specify allowed **File Upload Extensions** and **Allowed Metacharacters** before using Add to save the configuration. For more information on adding a new Parameter Profile, see online help.

**Trusted Hosts**

WEBSITES > Trusted Hosts allows designation of trusted hosts, by IP address and Mask, for which security checks are not necessary. Traffic coming from trusted hosts is assumed safe. Using **Add New Trusted Host**, enter a group name in Trusted Host Group Name field and Add. To add a trusted host to an existing Trusted Host Group, choose Add Host from the Options column for the existing group listed in Trusted Hosts.
Allowing/Denying Specific URLs

**WEBSITES > Allow/Deny** defines strict allow/deny rules for a Web site. URL ACLs allow customization of access by partitioning a Web site into security zones using host and URL to specify the zone and then configuring different allow or deny rules on each zone. Explicit deny ACLs can also be configured for known or observed attacks.

Configure URL ACLs using **WEBSITES > Allow/Deny**. Select **Add** for the desired Service and fill in the criteria on **Create ACL**. Security zones are partitions of a site specified using a URL ACL key. The key includes **URL Match**, **Host Match**, and an optional **Extended Match** rule (a value or expression for the header parameter). Matching URL ACL uses a best match algorithm including host, URL, and extended match fields in configured sequential order. In most cases, only host and URL specify an ACL. The Barracuda Web Application Firewall optimizes the search for matches 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 other fields in the request (eg client IP address or HTTP header) use **Extended Match** rules. Only **Extended Match** rules specified for the matching host and URL are considered for evaluation. If more than one rule is specified, they are evaluated according to the **Extended Match Sequence**. For more information on building an Extended Match expression, refer to **Extended Match and Condition Expressions** on page 203.

Select the Action taken in the event of a match to this Parameter Profile. **Deny Response** defines the response if the **Action** indicates **Deny**. **Response Page** specifies the page to send to the client if the **Action** is set to **Custom Response**. If **Deny Response** is **Redirect**, the **Redirect URL** specifies where to redirect the client.

Consider that there are two ways to redirect a request with two different implications to logs and processing:

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

The first case is not considered an attack, therefore:

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

The second case is a suspected attack, therefore:

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

**Note**

If the Mode of a service is passive, that enforcement setting overrides the actions set in the URL ACLs under **WEBSITES > Allow/Deny > URL Allow/Deny Rules** for that service. Matches will be logged but not enforced.
Allowing/Denying Specific Headers

Use **WEBSITES > Allow/Deny Header: Allow/Deny Rules** to define strict limitations on incoming headers intended for a Web site or Web service. A header ACL can be configured to protect against attack types and potentially malicious metacharacters and keywords placed in a header.

Select **Add** under **Options** for the service for which you want a Header Allow/Deny Rule. The **Create Header ACL** dialog requires entry of the **Header ACL Name**, the **Header Name** to validate. Enable by setting **Status** to **On**, and enter the **Mode**, Active (blocking) or Passive (monitoring and logging only), overridden to passive by a Passive Service Mode. Set the **Max Header Value Length** threshold and the **Denied Metacharacters**. Use **Add** to save the newly created Header ACL.
This chapter explains how to effectively use Adaptive Security, the automated profiling and tuning tools provided by the Barracuda Web Application Firewall. The tools covered in this chapter are Exception Profiling, including Exception Heuristics, and Adaptive Profiling. They are covered on the following pages:

- Overview ................................................................. 74
- Exception Heuristics ............................................... 76
- Working with Exception Profiling ............................. 78
- Working with Adaptive Profiling ............................... 79
- Recommended way to use the Adaptive Security feature........ 85
Overview

Adaptive security is comprised of:

- Exception Profiling
- Adaptive Profiling

Both assist in development of fine grained security settings. Exception profiling works with generated log files to refine security settings, customizing them to the Web application. Adaptive profiling analyzes the request and response traffic to generate customized security profiles for the Web application.

**Exception Profiling** uses a heuristics based strategy to adjust Web application security settings in response to logged traffic. Web applications are dynamic and vary widely, so a one size fits all security strategy might not be adequate across a Web site. Processing data through a service and making appropriate adjustments to security settings produces more mature security tailored to the application. Mature Web site security more accurately distinguishes between safe and unsafe traffic.

Exception Heuristics allow configuration of an exception strategy for a service, based on how tolerant the Web application is of violations detected in non-trusted traffic. A Low tolerance of violations means recommended or actual updates to security settings are triggered by fewer repetitions of the same violation. A High tolerance of violations means more repetitions of a violation are required before a recommendation or actual modification to security settings is made. Each Service has an associated Exception Profiling Level which can be set to Low, Medium, or High for untrusted traffic. A separate trusted setting applies to violations encountered in trusted traffic.

Exception Profiling modifies URL and Parameter profiles to relax the configured security policy so logged violations will be allowed when seen in the future. Exception Profiling either creates a new profile, or fine tunes an existing profile to reflect the exception. Exception profiling automates the otherwise manual process of evaluating and responding to logged violations. With exception profiling, the system learns exceptions and creates appropriate URL and parameter profiles automatically to allow them, by observing logged user traffic.

**Adaptive Profiling** learns the intricate structure of an application and enforces conformance to it. Detailed security profiles are created by Learning from requests and responses served by a particular Web application. Learning creates a positive security model, generating a whitelist of valid URLs and parameters, which can be configured to be allowed to the exclusion of all others. This positive security enforcement can provide protection against attacks like Forceful Browsing and Parameter Tampering.

The learned structure of the application is called a profile of the Web site. A Web Site 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 safe incoming traffic for the Web application. Strict profile checking can be enforced denying any request which falls outside of the Web Site Profile.

Hackers commonly probe Web applications for weak files, scripts, and directories vulnerable to attack or which may contain sensitive content. These attacks are denied when Web Site Profiles are developed and enforced, because they fall outside of the Web Site Profile.

To create URL profiles the adaptive profiler observes the valid HTTP methods for a URL: the allowed content types, maximum content, parameter lengths, maximum file uploads and referrers.

To create parameter profiles, the profiler observes the valid parameters of the URL, parameter length restrictions and HTML markups in the response body, and predefined values in FORM input elements such as radio buttons, check boxes, and drop-down lists. Protection from tampering of hidden parameters and against range and type violations in input parameters are also implemented.
Settings for Adaptive Security through the Web user interface appear on the following related pages:

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

For more information on Web Site Profiles, refer to Security Profiles on page 68.

Implementing the learning process requires configuration settings on all four pages. Configuration of Exception Profiling requires settings on 1, 2, and 4, while configuring Adaptive Profiling requires settings on 3 and 4.
Exception Heuristics

Automated exception creation assists the administrator in adjusting profiles to properly distinguish between allowable requests and violations. As traffic is analyzed and violations of security settings are logged, the Exception profiler suggests adjustments to security profile settings.

The Barracuda Web Application Firewall has four policies governing Exception Profiling. Each service has a configurable exception profiling level for untrusted traffic, and can apply a trusted profiling level for trusted traffic. For untrusted traffic Low, Medium, and High levels are provided. These levels correspond to increasing tolerance of violations in untrusted traffic. The heuristics settings automate the exception profiler’s response to logged errors. Creation of an actual or recommended exception happens after repeated observed violations. Heuristics settings are tailored to defined violation types, so responses to different violation types can be set appropriately. Exceptions can modify profiles automatically, with administrator approval, or not at all.

A Low setting indicates a low tolerance to violations, so logged traffic violations are reviewed frequently to properly adjust security settings. On the other hand, a High exception heuristics level indicates a greater tolerance of violations, and a higher confidence in the security settings, so review or adjustment of the profile only happens if a violation is seen more frequently. A separate setting for trusted hosts allows exception profiling to treat trusted traffic violations differently, by default automatically updating the respective profiles to reflect changes in trusted traffic. The Trusted settings are applied to all trusted traffic for all services.

There are only three levels of Exception Profiling Heuristics for untrusted traffic which apply to all services, so a change in settings of any level applies to any service using that level (Low, Medium, or High). Exception profiling provides default settings for each violation type. The settings indicate how exceptions update profiles (Automatically, Manually, or not at all), how the new setting in the profile is generated (increasing the current value, or accepting the observed value, for example) and how many times the logged error needs to be seen before generating an exception (Trigger Count). These default settings for an Exception Profiling Level can be edited and saved. Edits will apply to all services using that level.

Selecting an Exception Profiling Level of Low will increase the number of exceptions or recommendations for profile adjustment, causing a more rapid adjustment of the profile to reflect observed traffic. On the other hand, an Exception Profiling Level of High results in fewer exceptions and pending recommendations, indicating increased confidence in the profile, and higher tolerance for traffic violations.

Use ADVANCED > Exception Heuristics to adjust settings for exception heuristics for various Violation Types including:

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

For each violation type, set the following parameters:

- **Setting**: How exception will be created, automatically, manually through approval of Pending Recommendations, or no exceptions should be created.
- **New Value**: How to modify the parameter after learning. **New Value** can be a function of the old value (increase 100% for example). Or the new value can be based on the default option provided. **New Value** is selected from provided options.
- **Trigger Count**: This threshold sets the number of times a violation must be received from unique sources before triggering exception learning either automatically or manually. Only
unique requests from a client are counted. Multiple violations from the same client generate a single violation in the trigger count, preventing repeated hacker attacks from inspiring an exception.

**Working with Exception Profiling Levels**

**Exception Profiling Level** determines the exception creation heuristics for the Service to which it is bound. Four policies, or levels, are provided: Low, Medium, High (for untrusted traffic) and Trusted. To view the settings for a profiling level, select the level and **Show Definition**. The **Request Violation Handling** module gets populated with the settings for that level and can be modified. The levels are shareable across multiple Services. Any change made to an exception heuristics level setting applies to any Service bound to this level. Services may have an untrusted traffic exception profiling level (Low, Medium, or High) and also have designated trusted hosts using the Trusted Hosts exception profiling settings (not depicted in **Figure 7.1**).

*Figure 7.1: Shared Exception Heuristics Levels for Services*
Working with Exception Profiling

Configuring Exception Profiling

Exception profiling for a service can be configured using WEBSITES > Exception Profiling. By default Exception Profiling Level is None and Learn From Trusted Host Group is No. To learn from trusted hosts enable Learn From Trusted Host Group. To learn from untrusted traffic, select an Exception Profiling Level indicating a low, medium, or high tolerance of violations in untrusted traffic. For more information about Exception Profiling Levels, see Exception Heuristics on page 76. Detailed heuristics for each of the Exception Profiling policies are defined on ADVANCED > Exception Heuristics.

Learning from Trusted Hosts

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 from 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.

1. If Learn From Trusted Hosts Group is Yes and Exception Profiling Level is either Low, Medium, or High, then exceptions from trusted hosts are directly learned using trusted hosts heuristics. Concurrently, exceptions from non-trusted hosts are learned, according to the Exception Profiling Level.
2. If Learn From Trusted Hosts Group is Yes and Exception Profiling Level is None, then only exceptions from trusted hosts are learned.

Pending Recommendations

Exceptions for attacks when Setting is Manual on ADVANCED > Exception Heuristics are not applied automatically, but show up as Pending Recommendations on WEBSITES > Exception Profiling. For details on Exception Heuristics, refer to Exception Heuristics on page 76.

Pending Recommendations displays a table of violations considered false positives with the recommended fixes. Single or multiple entries can be selected; then Apply Fix refines the policy settings to prevents the logged data from being considered an attack in the future.

Decide whether to Apply Fix or Ignore the exception after looking at Pending Recommendations. Ignore removes the selected log entry from Pending Recommendations. If the same attack is encountered again, the Barracuda Web Application will continue to log it under Pending Recommendations. To prevent repeated log entries for this case, turn Off exception profiling for that Violation Type using ADVANCED > Exception Heuristics.

For more information, refer to WEBSITES > Exception Profiling online help.
Working with Adaptive Profiling

Configuring Adaptive Profiling

Learning using adaptive profiling selectively profiles URL spaces defined on WEBSITES > Web Site Profiles, including critical application components. Configure Adaptive Profiling to learn using WEBSITES > Web Site Profiles by setting the Mode of the URL Profile to Learn. Adaptive Profiling creates URL and Parameter profiles from requests, responses, or both.

Configure learning settings using WEBSITES > Adaptive Profiling. For example, turn on adaptive profiling for /cgi-bin/*/scripts using Content Types. The application profile developed by Adaptive Profiling is displayed on WEBSITES > Web Site Profiles. The configuration settings are explained below.

Edit Service Adaptive Profiling

Each Service is initially configured for adaptive profiling with predefined settings. Edit the predefined settings for adaptive profiling by clicking on the Edit link under Actions.

Add Adaptive Profiling Rules

Create one or more learn rules for a URL space using Add under Actions. The profiling rule defines the rules for learning the specified URL space.

To Start Adaptive Profiling

Use Start Learning under the WEBSITES > Web Site Profiles > Service to initiate learning.

To Stop Adaptive Profiling

Use Stop Learning under WEBSITES > Web Site Profiles > Service to stop learning.

Working with Navigation parameters

To distinguish between two requests with the same URL but different query parameters, specify those parameters as Navigation Parameters. Doing this uniquely defines configuration settings for the combination of the request URL and the navigation parameters.

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 values of query parameters primary_tab and secondary_tab together determine which page displays; different value combinations of navigation parameters display completely different pages, containing different FORM elements and content.

To protect this application, primary_tab and secondary_tab would be defined as Navigation Parameters forcing the profiler to generate separate profiles for each possibility. For example, the above case would produce the following profiles:

/cgi-bin/index.cgi?primary_tab=basic&secondary_tab=ip_config
By default all parameters are considered non-navigation parameters.

**WEBsITES > Web Site Profiles** URL Profile > **Nav Params** indicates values for all the navigation parameters used by the URL profile.

### Configuring URLs to be Excluded from Adaptive Profiling

Certain static content with no URL or FORM parameters should be excluded from profiling. This includes images, style sheets, and flash files among others. Configure these using **WEBsITES > Web Site Profiles**. The above examples are included in the default settings. Customize the list for a Web site so unnecessary profiles are not generated. Smaller configuration sizes improve performance.

Targeted learning takes advantage of the powerful automated profiling tools, without the performance impact of fully profiling every space in a URL. Targeted learning means excluding URL spaces from profiling when doing so does not reduce security effectiveness.

### Understanding Request and Response Learning

The following elements of profiles are learned through request and response learning:

**Table 7.1: Elements learned during learning phase**

<table>
<thead>
<tr>
<th>Profile Type</th>
<th>Elements learned during learning phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL Profile</td>
<td>Allowed Query string, Allowed Methods, Allowed Content Types, Max Content Length, Max Parameters, Max Upload Files, Max Parameter Name Length, Referrers.</td>
</tr>
<tr>
<td>Parameter Profile</td>
<td>Type, Allowed Meta-characters, Max Parameter Value Length, Required, Ignore, Max Instances, File Upload Extensions, Max Upload File Size.</td>
</tr>
</tbody>
</table>

The initial values of these elements are taken from the applicable security policy.

As an example, when a new URL profile is generated the parameter **Max Content Length** is set to 32k if the service uses default security policy. If the profiler receives a successful request with content length 35k, the new value of **Max Content Length** is set to 35k for the request URL.

Other security policy elements for the service specified for the Web site (e.g. Advanced Security elements, Allow/Deny rules, etc.) and those inherited from the associated Security Policy (e.g. Cookie Security, Normalization, etc.) continue to apply during the learning phase. This maintains protection from application attacks even during the profile development phase. Whether attacks are blocked or just logged during this phase depends on the Service’s **Mode** setting.
The following URL and Parameter profile configuration elements are not learned by the Adaptive Profiler. They continue to be applied even during the learning phase:

### Table 7.2: Elements not learned during learning phase

<table>
<thead>
<tr>
<th>Profile Type</th>
<th>Elements not learned during learning phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL Profile</td>
<td>Blocked Attack Types, Custom Blocked Attack Types</td>
</tr>
<tr>
<td>Parameter Profile</td>
<td>Blocked Attack Types, Custom Blocked Attack Types</td>
</tr>
</tbody>
</table>

The following table describes the different Types for a parameter profile. Read Only, File Upload, Global Choice, and Input parameter types are automatically learned by the profiler. Session Choice and Session Invariant need to be manually specified.

### Table 7.3: Different Types for a Parameter Profile

<table>
<thead>
<tr>
<th>Param Type</th>
<th>FORM Attribute Used for Learning</th>
<th>Description</th>
<th>Allowed Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read Only</td>
<td>type=hidden</td>
<td>All hidden FORM parameters are learned as Read Only. Known by the &lt;type=hidden&gt; attribute in the HTML response content, the value of the parameter is learned on a per session basis.</td>
<td>When profile mode is Active: Allowed value for the parameter in a request is exactly equal to that learned from the response. When profile mode is Learning: If the value varies in requests during the learning stage, the type updates to Input.</td>
</tr>
<tr>
<td>File Upload</td>
<td>type=file</td>
<td>The parameter of type file upload in FORM is treated as File Upload type.</td>
<td></td>
</tr>
<tr>
<td>Global Choice</td>
<td>type=checkbox</td>
<td>radio</td>
<td>submit</td>
</tr>
<tr>
<td>Session Choice</td>
<td>NA</td>
<td>Has to be specified manually. Similar to Global Choice, but values are learned on a per session basis.</td>
<td>The system constructs an allowed list of values learned by observing the values on a per session basis.</td>
</tr>
<tr>
<td>Session Invariant</td>
<td>NA</td>
<td>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: session-id.</td>
<td>The unique value learned for a parameter per session.</td>
</tr>
<tr>
<td>Input</td>
<td></td>
<td>Any parameter not of the above types is treated as Input type.</td>
<td>All values allowed by the regex for Comments Data Type element on ADVANCED &gt; View Internal Patterns.</td>
</tr>
</tbody>
</table>

---
Response Learning

If Response Learning is 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 WEBSITES > Adaptive Profiling.

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 differ in a subsequent request while the URL space is still being learned, their type changes to Input.

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 relevant URL space).

For a POST request, the profiler may have already learned the FORM and query parameters from a prior response. For example, 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 an earlier response, the profiler will update the parameter type to Input when Request Learning is On for this URL and the profile is learning. When Request Learning is Off, and the profile is learning, the request is allowed, but the parameter type is not changed. When the profile is Active (learning is turned Off), 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

<table>
<thead>
<tr>
<th>URL Profile</th>
<th>Parameter Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.html</td>
<td>Query Params (q1, q2)</td>
</tr>
</tbody>
</table>

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

Response: a.html?q1=abc&q2=defsdfrom

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

Request: b.html?q3=userinfo

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

<table>
<thead>
<tr>
<th>URL Profile</th>
<th>Parameter Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.html</td>
<td>Query Params (q1, q2)</td>
</tr>
<tr>
<td>b.html</td>
<td>Form params: {firstname, lastname, married}</td>
</tr>
<tr>
<td></td>
<td>Query params: (q3)</td>
</tr>
</tbody>
</table>

**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 Web Site and Directory. The URL/Parameter profiles are displayed in red as shown in the figure below.

*Figure 7.2: Newly generated profile*

One measure of the maturity of a security profile is the number of hits, or matching requests, which it has encountered. If Hits is small, the profile will reflect only that small number of requests or responses it has encountered and not reflect the spectrum of potential requests and responses.

Filter the list of profiles by selecting Profiles not reviewed or URLs with Params not reviewed. Keep track of viewed profiles with the Mark Read option from More Actions. The profile will be listed in black font next time it is viewed. This way the administrator can track which learned profiles still need review.
Enforcing Learned Profiles

Once satisfied with generated profiles, select them, and Lock Profiles using More Actions to begin their enforcement. The system now considers violations to these profiles as attacks, no longer learning from them. The Mode setting for these URLs determines how attacks are handled.

To assist in this transition, the system displays as Hits on WEBSITES > Web Site Profiles the number of successful requests matching generated URL profiles after the last update. The number of hits to a profile is a good indicator of the profile maturity. The higher the number of hits, the more mature the profile is.

Using Strict Profile Checks to Enforce Positive Security

Enforcing Strict Profile Checks denies requests which do not match any profile. If set to No, the service's default security settings will be applied to those requests which do not match a profile. Strict Profile Check cannot be modified when Adaptive Profiling is On. The enforcement of profile security depends on Service > Web Site > Use Profile (found on WEBSITES > Web Site Profiles) according to the following table.

Table 7.4: Strict Profile Check parameter behavior

<table>
<thead>
<tr>
<th>Use Profile</th>
<th>Strict Profile Check</th>
<th>Behaviour</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Yes, No</td>
<td>Profiles not used.</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Requests not matching any profile use assigned security policy protections, enforcing a negative security model. Exceptions can be added to any security policy violations by adding the relevant profiles.</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>A &quot;deny unless allow&quot; strict rule is enforced dropping requests with no matching profiles, and enforcing a positive security model.</td>
</tr>
</tbody>
</table>
Recommended way to use the Adaptive Security feature

1. From WEBSITE > Web Site Profiles, select the Web Site and Start Learning.
2. Either manually browse through the application (recommended) or crawl the application.
3. Let Adaptive Profiling populate the URL and parameter profiles automatically.
4. View the created profiles to review them. If satisfactory, Stop Learning to stop the profiling for the Service. Select a subset of the profiles and enforce them by selecting Lock Profiles when desired.
5. The profile mode is initially Passive. Look for any false positives using BASIC > Web Firewall Logs. Examine the Hits statistic on WEBSITES > Web Site Profiles, under URL Profiles. If satisfactory, select Lock all Profiles from the More Actions drop-down list to turn all profiles to Active.
7. If possible, manually combine the learned profiles to optimize the configuration.
8. If the back-end application has changed, revise the profiles accordingly using Resume Learning from More Actions for the URL.

For more information on Adaptive Security, refer to online help.
This chapter describes advanced security configuration on the Barracuda Web Application Firewall. It includes the following:

- **URL Policy** .................................................................................................................. 88
- **Session Tracking to Prevent Session DoS** ................................................................. 89
- **Masking Sensitive Data in Logs** .................................................................................. 89
**Advanced Attack Protection**

The Barracuda Web Application Firewall provides advanced capabilities enforceable for URL partitions of the Web site. The following briefly explains various Barracuda Web Application Firewall protections that make up a URL Policy. These include Rate Based protection (including Rate Control and Brute Force Prevention), Data Theft, and Virus Scan protection.

In addition to URL Policy settings, advanced security settings include Session Tracking and Mask Sensitive Data settings, configured using [WEBSITES > Advanced Security](#), and described below.

**URL Policy**

A URL Policy is made up of advanced protection settings configured using Advanced Security on [WEBSITES > Advanced Security](#). One or more URL Policies are associated with a service, specifying which advanced protections are enabled or disabled for URL spaces of the Web application. A default URL Policy is provided which applies to all data that does not correspond to a more refined URL space and URL Policy. The default URL policy leaves all advanced protections disabled. The default URL policy can be deleted, disabled, or refined. Add a URL Policy to a service or Edit an existing URL Policy after selecting it from listed URL policies associated with the service. Host Match, URL Match, Extended Match, and Extended Match Sequence fields determine which URL Policy applies to a request or response of a service. Parse URLs in Scripts enables the Barracuda Web Application Firewall to parse scripts identifying URLs which need URL translation or instant SSL domain translation. This should only be enabled when required, as it is processing intensive.

Data Theft Protection can be enabled for URLs matching the URL Policy according to the Data Theft Protection settings configured in the Security Policy for the service. For more information on configuring Data Theft Protection refer to [Data Theft Protection](#) on page 64.

**Enabling and Disabling Virus Protection**

Use the [WEBSITES > Advanced Security](#) page to turn on Virus Scan. Under Advanced Security section, click Edit for a service to enable or disable Virus Scan. By default, Virus Scan is set to No, and the virus definitions are updated on a regular basis (hourly by default) using Barracuda Energize Updates.

When Virus Scan is Yes for a URL Policy, all requests matching that URL Policy are scanned for viruses and any traffic containing a virus is blocked. The Barracuda Web Application Firewall logs requests, viewable using [BASIC > Web Firewall Logs](#), so the system administrator can take appropriate action.

**Preventing Brute Force Attacks**

Brute Force protection sets a maximum number of requests (all requests or only invalid requests) to a URL space from a single client, or from all sources, within a configured time interval. It blocks offending clients from making further requests. You can specify exception clients for which no maximum is enforced. Brute force protection prevents the following types of rate based attacks:

- **Brute force attempts to gain access** – Repetitive login failures in quick succession may be an attempt to gain unauthorized access using guessed credentials.
- **Brute force attempts to steal session tokens** – Session tokens, authentication mechanisms for requests by already authenticated users, can be guessed and stolen through repeated requests.
- **Distributed Denial of Service attacks (DDoS)** – Repeated requests for the same resource can impair critical functionality by exhausting server resources.
Vulnerability scanning tools – High rates of requests can probe Web applications for weaknesses. Typically these tools execute a database of commonly known and unknown (blind) attacks which are executed in quick succession.

Note
1. To detect brute force attacks against session management (too many sessions given out to a single IP address or range), use session tracking.
2. To control the rate of requests to specific resources (URL spaces), and to provide different levels of service to different sets of clients, use rate control pool.

Preventing Rate Based Attacks using Rate Control Pool

URL Policies allow the selection of a rate control pool to throttle requests to a Web application. Rate Control Pools are configured using ADVANCED > Libraries. Once created they can be applied using the URL Policy Rate Control Pool setting. A Rate Control Pool throttles the incoming requests to the URL to prevent the application from being overwhelmed and falling victim to Denial of Service attacks.

For more information on configuring Rate Control, refer to Rate Control on page 95.

Session Tracking to Prevent Session DoS

A Session refers to all requests 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 requests from that user. Every user has a unique session identified by a unique session identifier.

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 address helps prevent session-based Denial of Service (DoS) attacks.

To configure Session Tracking use WEBSITES > Advanced Security and choose Edit from Options. Enable Session Tracking by setting Status to On, then specify the desired session protection fields: New Session Count: maximum number of new sessions allowed per IP address; Interval: the time in seconds to track requests; Session Identifiers: the token type used to recognize sessions; and Exception Clients can be listed which are exempted from this protection.

To configure the Barracuda Web Application Firewall to recognize the session identifier, refer to Creating Session Identifiers on page 95. For more information on configuring Session Tracking, refer to online help.

Masking Sensitive Data in Logs

Data masking security of the Barracuda Web Application Firewall obscures sensitive data elements before logging them. Configured parameters like social security numbers, credit card information, or other proprietary data in the URL parameters of a request can be protected from unauthorized exposure in the logs. Data masking is configured for an application using parameter names to specify...
sensitive data. Logged data appears in **BASIC > Access Logs**, with the sensitive data overwritten by ‘X’es.

**Note**

1. Masking cannot be applied to sensitive data in custom parameters or custom headers.
2. Once masked, the original data cannot be retrieved, recovered, or restored.

To configure Data Masking, use **Mask Sensitive Data** on **WEBSITES > Advanced Security**. Edit the service for which masking is necessary. The **Mask Sensitive Data** window allows the entry of **Sensitive Parameter Names**. Provide multiple parameters separated by commas with no spaces between parameter names (e.g., cardId, securityNumber, password). **Save Changes**.
This chapter describes advanced user defined security configurations you can create and enforce using the Barracuda Web Application Firewall. The following topics are covered:

- **User Defined Data Patterns** .............................................................. 92
- **Rate Control**....................................................................................... 95
User Defined Data Patterns

The Barracuda Web Application Firewall allows you to create customized data patterns which can be detected and handled according to configured security settings.

The Barracuda Web Application Firewall uses regular expressions (regex) to define data type patterns. Custom data types can be defined using regex patterns to implement advanced data type enforcement on input parameters. (See Extended Match and Condition Expressions on page 203 for guidelines on how to write regular expressions.) The pattern-match engine recognizes the lexical patterns in text and compares inputs to defined data type patterns. For example, the following is the default regex pattern for a Visa credit card:

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

In addition, a pattern can also be associated with an algorithm. For example, an algorithm which validates a credit card number can be associated with a credit card pattern. The algorithm is then run on all strings matching the regular expression before deciding whether they actually conform to this pattern.

Internal Patterns

The ADVANCED > View Internal Patterns page includes Identity Theft Patterns, Attack Types, Input Types, and Parameter Class. These patterns can be bound to a policy or to profiles of an application to validate the incoming requests.

The patterns specified under each pattern group cannot be modified directly, but can be copied using the Copy function, then the copied pattern (found in ADVANCED > Libraries) can be modified. You can modify or delete the patterns as required, and then apply the group to a Web application's security policy. For more information on how to copy a pattern group, see Copying a Pattern Group below.

The following provides a brief description of internal patterns.

Identity Theft Patterns

Identity theft is the loss of personal data resulting in fraud. Disclosure of sensitive information such as credit card numbers, banking information, passwords, or usernames in service communication might enable identity theft. The Barracuda Web Application Firewall prevents unauthorized exposure of at risk data.

The Identify Theft container includes Credit Cards, Social Security Number, and Directory Indexing data types. In addition, customized identity theft patterns can be created and used. For more information, see Creating and Using Custom Identity Theft Patterns below.

Attack Types

An attack is a technique used to exploit vulnerabilities in Web applications. Attacks can insert or modify code in requests. If a request contains an attack pattern, it is dropped. The attack data type container includes Cross-site Scripting, Remote-file Inclusion, SQL Injection, Directory Traversal, and OS Command Injection. In addition customized attack data types can be created and used. For more information, see Creating and Using Custom Attack Types below.
Input Types

Input data types are used to validate the HTTP request parameters. Inputs come from Web forms, Applications and Services, Custom client applications, or File based records. This validation ensures that the data conforms to the correct syntax, is within length boundaries, and contains only permitted characters or numbers. Requests failing validation are identified as intrusions and blocked.

Input types are defined using reg-ex patterns. Default Input Types including credit cards, numeric, hex-number, alpha, alphanumeric, string, name, and date are provided. In addition, customized Input Types can be defined and used. For more information, see Creating and Using Custom Input Types below.

Parameter Class

Parameter class defines acceptable values for parameters. Parameter classes are bound to Parameter Profiles using WEBSITES > Web Site Profiles Parameter Profiles and specify validation criteria for parameters in a request. In addition to the internal parameter classes, customized parameter classes can be created and used. For more information, see Creating and Using Custom Parameter Class below.

Copying a Pattern Group

To modify a pattern group, first make a Copy, providing a name for the New Group. After clicking Paste, the new group will appear in the list on ADVANCED > Libraries. From ADVANCED > Libraries, Edit Pattern or Delete can be used to modify or delete the copied pattern.

Creating and Using Custom Identity Theft Patterns

Create custom Identity Theft Patterns using ADVANCED > Libraries Identity Theft Patterns. Identity Theft Patterns are compared to responses to identify potentially sensitive information. One or more patterns defining a custom data type format can be associated to each group. To create a new custom Identity theft group, simply provide a name for the New Group and click Add. The new Identity Theft group is listed under Identity Theft Patterns. Add Pattern allows the addition of one or more patterns you specify for that group. See online help for more information on filling in fields and values. To use the newly created patterns, they must be configured for a security policy bound to a service.

Using a Custom Identity Theft Pattern

To configure a security policy to use the newly created Identity Theft Pattern, use SECURITY POLICIES > Data Theft Protection for the applicable Security Policy. For more information refer to Data Theft Protection on page 64 or to online help.

Creating and Using Custom Attack Types

ADVANCED > Libraries Attack Types allows creation of custom attack data types which, when detected in a request, identify the request as an attack. One or more patterns which define the format of the attack type can be added to each group. To create a custom attack type, create a New Group, then add one or more Patterns to that Attack Group. To create a new custom attack type group, simply provide a name for the New Group and click Add. The new group is listed under Attack Types. Add Pattern allows the addition of one or more patterns you specify for that group. See online help for
more information on filling in fields and values. To use the newly created patterns, they must be configured for a security policy bound to a service.

**Using a Custom Attack Type**

The added attack type pattern is listed under Custom Blocked Attack Types on the following pages and sections:

- ADVANCED > Libraries > Custom Parameter Class
- WEBSITES > Web Site Profiles > URL Profiles
- SECURITY POLICIES > URL Protection
- SECURITY POLICIES > Parameter Protection

The Custom Blocked Attack Types default to enabled for ADVANCED > Libraries > Custom Parameter Class and on WEBSITES > Web Site Profiles URL Profiles. Manual selection is required to enable it for both SECURITY POLICIES > URL Protection and SECURITY POLICIES > Parameter Protection.

The following sections describe the use of Custom Patterns in more detail.

---

**Creating and Using Custom Input Types**

The Barracuda Web Application Firewall includes a collection of pre-defined and custom input data types, which can be used to validate HTTP Request parameters. Input data types are used to validate that request parameters conform to expected formats. Most attacks can be prevented by properly validating input parameter values against expected input data types. Input Type validation enforces the expected formats rather than trying to identify malicious values. Requests failing validation are identified as intrusions and blocked. Default Input Types including alpha-numeric strings, credit card, date and positive-long-integer are provided. Custom Input Data Types can also be added.

ADVANCED > Libraries Input Types allows creation of customized input data types. One or more patterns which define the format of the custom input type can be added to a new input type group. To create a new custom input type group, simply provide a name for the New Group and click Add. The new group is listed under Input Types. Add Pattern allows the addition of one or more patterns you specify for that group. See online help for more information on filling in fields and values. The custom Input Type can now be used to validate input parameters using Custom Parameter Class. Newly created Custom Input Data Types are used by configuring Custom Parameter Class to enforce conformance of input parameters with the custom input data type. See Creating and Using Custom Parameter Class below or online help for more detailed instructions.

---

**Creating and Using Custom Parameter Class**

ADVANCED > Libraries Custom Parameter Class allows creation of custom parameter classes which enforce expected input formats and block attack formats for request parameters. One or more patterns which define the format of the data type can be added to each group. Bind the custom parameter class to a parameter profile by adding a new parameter profile or editing an existing parameter profile using WEBSITES > Web Site Profiles.

To create a Custom Parameter Class, use Add Custom Parameter Class. Name the Custom Parameter Class, and select the Input Type Validation to enforce. To use a Custom Input Data Type, select Custom for the Input Type Validation, and fill in the Custom Input Type Validation indicating which custom input data type to enforce. Parameters are also checked for Denied Metacharacters.
Blocked Attack Types, and Custom Blocked Attack Types configured for the class. See online help for more information about configuring fields and values.

Using a Custom Parameter Class

Custom Parameter Class is bound to a Web Site Profile using WEBSITES > Web Site Profiles.

Select the service which requires a custom parameter class.

Use Add Param in Parameter Profiles to create a parameter profile for the URL profile selected. Name and enable the created Parameter Profile. To use a custom parameter class, select CUSTOM for Parameter Class and specify the Custom Parameter Class to enforce. The parameter profile is now used to validate the requests for the configured service. The response to requests failing validation depends on Mode in the URL Profile.

Creating and Using Custom Response Pages

The ADVANCED > Libraries > Response Pages section allows creation of customized HTML response pages for HTTP requests that violate security policies on the Barracuda Web Application Firewall. Either Edit an existing default response page or use Add Response Page to add customized response pages that can be shared among multiple services. Refer to online help for more details.

Added response pages are listed as response page options when configuring:

- SECURITY POLICIES > Global ACLs > Existing Global ACLs
- SECURITY POLICIES > Action Policy > Action Policy
- WEBSITES > Allow/Deny > URL : Allow/Deny Rules

In each case, to use the custom response page requires updating the Deny Response to Response Page and selecting the desired Response Page from the list of added customized response pages.

Creating Session Identifiers

Configuration of Session Identifiers allows the Barracuda Web Application Firewall to recognize session information in requests and responses. Recognizing Session Identifiers requires the identifier name and the beginning and end delimiters. For example, “JSESSIONID=12345;” would be configured with Name JSESSIONID, Starting Delimiter = and ending delimiter; to allow WAF to successfully extract the Session Id 12345. To configure customized Session Identifiers, use ADVANCED > Libraries Session Identifiers. Edit or Add Session Identifiers, which then become available to configure for the service using WEBSITES > Advanced Security > Session Tracking.

Rate Control

Rate Control allows configuration of the maximum number of connections allowed from any specific IP address. When the Rate Control threshold is reached, the Barracuda Web Application Firewall queues further requests up to a configured threshold, then blocks further connections.

The Rate Control policy is a shareable policy for controlling the rate of request to a Web application. A Rate Control Pool specifies the maximum number of Active Requests and Maximum Client Backlogs allowed when Active requests exceed the threshold. A set of Preferred Client IP addresses or IP address ranges can be configured with associated weights. The Barracuda Web Application
Firewall uses the 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 is the lowest priority and 100 is the highest priority.

**Rate Control**

A default rate control pool is provided by Barracuda Energize Updates which allows the easy set up of Rate control. To set up a customized Rate Control Pool, use ADVANCED > Libraries. WEBSITES > Advanced Security allows rate control to be enabled by editing the default URL Policy, enabling the protection, and setting the rate control pool to the default. For more information on rate control, refer to Preventing Rate Based Attacks using Rate Control Pool on page 89. Detailed configuration instructions are available through online help for setting up Rate Control Pool and enabling rate control for the URL Policy.

**Before you set up a Rate Control Pool**

Answer the following questions:

1. What are the maximum simultaneous requests that can be served by the resource being protected? This determines the Maximum Active Requests setting.
2. What, if any, are the bonafide gateways and mega-proxies that will be accessing the protected resources? These are Preferred Clients. If they proxy client requests, assign a suitable weight to the proxy IP address 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 to allow for IP addresses not defined in Step 2? This defines the Maximum per client backlog setting.

**Benefits of a Rate Control Pool**

A Rate Control Pool helps defend against rate control attacks by:

- Throttling attackers attempting to flood the application with DoS attacks. The requests get queued for weighted round robin scheduling, slowing down the request rate seen by the server.
- Protecting “Load Sensitive” applications, such as search or DBMS intensive applications, from application DoS attacks.
- Allowing bonafide gateways and mega-proxies access while preventing attacks.

**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 from each unconfigured client queue followed by 5 requests from the Preferred client queue.

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.

**Creating Rate Control Pool**

1. From the ADVANCED > Libraries Rate Control Pool use Add Rate Control Pool to set up a new rate control pool.

   - **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.

- **Maximum per client backlog** - Enter the number of requests per client IP address that will be queued when the Barracuda Web Application Firewall has reached the Maximum Active Requests limit. For example, if **Maximum 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 up to 32 requests. Any requests after that will be dropped until a request is deleted from the queue.

- **Maximum 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 **Maximum Unconfigured Clients** is set to 100 and **Maximum 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.

2. Click **Add** to add the above configuration.

Use **Add Preferred Clients** to add a range of IP addresses to that pool which will be treated in a preferential manner. If the preferred client queue represents a range of IP addresses, the queue for preferred clients will contain the requests from all the clients falling within that range.

**Creating Preferred Clients**

3. Click **Add Preferred Clients**, under **Options**. The **Add Preferred Client** window appears.

4. Specify values for the following fields:
   - **Name** - Enter the name for the client weight.
   - **Status** - Sets the status of the preference. Enabling this makes the client IP address range a 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 **Maximum Per Client Backlog**, times the weight. For example, if **Maximum 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 lowest priority and 100 the highest priority).

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

6. Click **Edit** against any Rate Control Pool to modify the configuration.

7. Click **Delete** to delete the created Rate Control Pool from the list.
This chapter describes major traffic management features: Load Balancing, Content Rule, Caching, and Compression. The following topics are covered:

- Load Balancing ................................................................. 100
- Content Rule ................................................................. 102
- Configuring Caching ..................................................... 104
- Configuring Compression ............................................. 105
Traffic Management

The Barracuda Web Application Firewall provides traffic management features including load balancing, caching, and compression of Web site traffic to improve the performance of Web sites. It can load balance all TCP traffic in proxy deployments, and using Content Rules can implement Load Balancing, Caching, and Compression of HTTP application layer data for any deployment.

When the Barracuda Web Application Firewall is in proxy mode, all incoming TCP traffic can be distributed to balance the processing load across multiple back-end servers. Persistence is maintained for requests from the same client, and server health is monitored, so requests are served as efficiently as possible.

In addition, for HTTP data, content rules can be used to route requests to servers optimized to handle specific content types. This layer 7 load balancing is available only for HTTP data, and can be used with any deployment configuration of the Barracuda Web Application Firewall. Content rules also provide caching or compression options for matching HTTP data, which can decrease data processing demands on the Web site.

Load Balancing

A load balancer is a networking device that distributes traffic across multiple back-end servers in order to improve Web site response times. The Barracuda Web Application Firewall can act as a stand-alone load balancer or work in conjunction with other load balancers. Situated in front of back-end servers, it distributes incoming traffic across the servers using the configured algorithm. The Barracuda Web Application Firewall supports load balancing of all types of applications.

Load balancing ensures that subsequent requests from the same IP address will be routed to the same back-end server as the initial request. This guarantee of persistence requires an awareness of server health so subsequent requests are not routed to a server which is no longer responding. The Barracuda Web Application Firewall can monitor server health by tracking server responses to actual requests and marking the server as out-of-service when errors exceed a user configured threshold. In addition, the Barracuda Web Application Firewall can perform out-of-band health checks, requests created and sent to a server at configured time intervals to verify its health.

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

- Distributes traffic requests among back-end servers according to a user-configured algorithm.
- Automatically identifies server status to ensure appropriate traffic routing.
- Adds and removes servers without interrupting network traffic.
- Provides persistence support allowing a user to maintain connection integrity between client and Web service.
- Provides for configuration of a backup server, used only when all other servers being load balanced are out-of-service.

Load balancing can be configured at two levels:

- General (all TCP Traffic--layer 4)
- Content Rule (HTTP traffic only--layer 7)

The general policy is configured for a service and applies to all TCP requests to the service, while the content rule policy applies to HTTP requests matching the configured content rule only. The general and content rule configuration procedures are identical. There are three steps to configure load balancing on a Barracuda Web Application Firewall:

- Configure the load balancing algorithm and other general parameters.
• Configure a persistence method to maintain the integrity of stored state information.
• Configure a failover method to handle requests for a server which is down.

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

General load balancing, routing requests to back-end servers based on a user configured algorithm, is configured for a service. From BASIC > Services, choose Edit under Options for the service. Choose the Algorithm, Persistence Method, and Failover Method. The algorithm determines where the first request from a source IP address is routed. Future requests from the same client will be routed to the same server according to the configured Persistence method. Failover Method only applies when the persisted server is “out-of-service”. For detailed instructions to configure load balancing, see online help.

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 load balanced servers are out-of-service, requests are sent to the Backup server. For more information, refer to Configuring a Backup Server on page 102.

Monitoring the Health of the Server
Load balancing distributes requests to servers, sending subsequent requests from the same client to the same back-end server. To prevent requests from being sent to an unresponsive server, the health of all back-end servers must be monitored. The Barracuda Web Application Firewall monitors server health three ways: using In-Band, Out-of-Band, and Application layer health checks. In-Band and Application Layer Health Checks can only change a server status to out-of-service from an online state; but Out-Of-Band Health Checks, which perform periodic tests of all servers, allow a server state to change from out-of-service to online when the health checks succeed.

Note
In-Band and Application Layer Health Checks are performed only if the parameter Enable OOB Health Checks is set to Yes for Out-Of-Band Health Checks. This prevents Servers from being marked out-of-service indefinitely. Disabling Out-of-Band Health Checks disables monitoring server health.

For detailed configuration instructions, see the online help by clicking Edit for the server on the BASIC > Services page.

In-Band Health Checks
In-Band Health checks monitor a server’s connections and response to user traffic. The In-Band Health Check policy specifies layer 4 and layer 7 error thresholds. The server connections and responses are monitored for errors. When error counts exceed configured thresholds, the server is marked out-of-service.

Servers marked out-of-service no longer receive requests. Traffic is routed to other load balanced servers if possible. When no healthy server is available to serve a request, an error response is sent to the client.
In-band monitoring is enabled by default, and default parameters are provided. The settings can be modified if desired. In-band monitoring is disabled if Out-of-Band Health Checks are disabled.

**Out-Of-Band Health Checks**

The Barracuda Web Application Firewall also monitors server health by sending requests at configured intervals which are independent of incoming traffic. Out-of-Band health checks performed in addition to user traffic connections. The Out-of-Band Health Check parameters specify layer 4 and layer 7 server monitoring.

If a server health check fails, the server is marked as out-of-service. Out-of-service servers continue to be sent data based on Out-Of-Band Health Check configuration, so when a health check succeeds, the server's status reverts to in-service. An out-of-service server can only be restored to service using out-of-band health checks because in-band checks require user traffic to be sent to the server, and user traffic isn’t sent to an out-of-service server.

**Application Layer Health Check**

Application Layer Health Check sends an HTTP request to verify the server is responding correctly. A correct response verifies the server is healthy. Otherwise, the server is marked as out-of-service. The Application Layer Health Check settings define the HTTP request type (URL, Method, Headers), and healthy response (Status Code, Match Content String).

**Configuring a Backup Server**

An optional backup Web server can be configured to be used only when all other load balanced servers fail. For detailed instructions refer to online help.

**Content Rule**

A Web site 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. For requests matching the rule, the configured content rule policies are applied. Content rules allow management of HTTP traffic flow for a Web application.

Configuring a content rule requires the following steps:

- Create the content rule for the target Web service.
- Add one or more rules to define match criteria for this content rule.
Configure the policies to apply to matching requests.

You can configure settings for a content rule which apply to three traffic management techniques:

- **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 configuration specifies which server group to use. This allows distribution of requests based on the content type (see Example1: Content Rule for Images on page 103).

- **Caching** - Sets caching policy for the content rule (refer to Configuring Caching on page 104). This allows selective caching 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 the specific content type.

Rules are evaluated based on a key comprised of the URL, host, and an optional extended match rule in specified sequential order. In most cases, only host and URL are 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 matching is determined by a best match algorithm where the best match is the rule with the longest matching host and URL keys. For more information refer to Extended Match and Condition Expressions on page 203.

**Example1: Content Rule for Images**

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

To direct all requests for image content from the images directory in the Web server to a different set of servers, say S4 and S5, do the following:

- Use Rule for the Web service on BASIC > SERVICES > Services section to create a content rule for requests matching the URL /images/* as shown below: (For detailed instructions on creating a content rule, see online help).

![Figure 10.1: Configuring Content Rule](image)

- Add one or more servers for this content rule (S4 and S5 in this case) using the Server option under Add. 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 customize the load balancing policy used by the servers for this content rule, Edit the content Rule.

To configure caching for a content rule, create caching rules (see Figure 10.2) specifying file extensions and size restrictions for objects that should be cached on the Barracuda Web Application Firewall. These objects will be retrieved from the cache directly to serve future requests, rather than fetching the object content from the back-end servers.
Create compression rules for a content rule, specifying what response content should be compressed by the Barracuda Web Application Firewall to improve available network bandwidth. For more information on configuring compression, see Configuring Compression on page 105.

**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.

For detailed instructions on configuring Caching for Content Rules, see the online help in WEBSITES > Traffic Management Caching.

**Content Rules and Dynamic Pages**

Requests are compared to Web service content rules and if caching is enabled for the matching rule, the response is cached and served from the cache for subsequent requests. When disabled, each subsequent request is forwarded to the back-end server for the reply.

Caching works well for responses containing static pages which remain unmodified over multiple requests. When response content is likely to change for each request due to context or conditions (for example, when server side scripting is used to generate context-sensitive responses based on URL/form params in requests) dynamic response is required.

A content rule with a URL of /reports/* matches all pages under /reports. If cache is enabled on this content rule, all pages under /reports are cached. If the /reports folder also contains dynamic pages that shouldn’t be cached, the user has the following options:

- Move all dynamic pages into another folder, for example /reports/cgi_bin, and create a separate content rule for the URL of /reports/cgi_bin* disabling caching 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 their life span, and indicates when to retrieve a newer version from the originating server. Objects are designated with a freshness algorithm, and when an object is stale, this algorithm directs requests to the originating server. Otherwise, the request is served with the locally stored cached copy. If the cached object expires, it is still served from the cache, but the response includes a Warning 110 (response is stale) header.

The following algorithm is used for calculating object freshness. For this algorithm, age is calculated as follows:
age = (current_time - time_retrieved) + 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 10.1: Object Freshness Calculations**

<table>
<thead>
<tr>
<th>If ...</th>
<th>Then object freshness is calculated as ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ignore Response Headers is enabled.</td>
<td>freshness = age - expiry_age</td>
</tr>
<tr>
<td>Cached response had an expiration time.</td>
<td>freshness = current_time - object_expiration_time</td>
</tr>
<tr>
<td>Age of an object is greater than Expiry Age.</td>
<td>freshness = age - expiry_age</td>
</tr>
<tr>
<td>Cached response has a time last modified header.</td>
<td>stale_age = time_object_retrieved + object_age - time_object_last_modified&lt;br&gt;stale_age = stale_age * &lt;br&gt;age_from_last_modified_percentage / 100&lt;br&gt;if age &gt; stale_age, freshness = age - stale_age</td>
</tr>
<tr>
<td>Cached response does not have a time last modified header:</td>
<td>freshness = age - expiry_age</td>
</tr>
<tr>
<td>Staleness &lt; 0, a min-fresh header request is present, and it is set to be greater than the staleness value (positive value of it).</td>
<td>The object is considered stale.</td>
</tr>
<tr>
<td>Staleness &lt; 0, and a min-fresh header request is not present.</td>
<td>The object is considered fresh.</td>
</tr>
<tr>
<td>A max-stale header request is present, and it is set to be greater than the staleness.</td>
<td>The object will expire.</td>
</tr>
<tr>
<td>A max-stale header request is not present.</td>
<td>The object is considered stale.</td>
</tr>
</tbody>
</table>

**Configuring Compression**

Compression improves response time for clients reducing the quantity of data transferred. Web pages that use HTML, JavaScript, Java, and other text-based languages, can be compressed to improve traffic management and significantly reduce download time. Compression can be applied for all client requests, and to specific client requests that match Content Rules. Enabling compression for a service
applies compression to all requests. For detailed instructions on configuring Compression for Content Rules, see the online help in **WEBSITES > Traffic Management Compression**.

**Note**
Compression should be turned off in back-end Web servers. The Barracuda Web Application Firewall will not uncompress, inspect, and recompress compressed responses originating from the back-end servers. Instead, back-end servers should send uncompressed content, letting the Barracuda Web Application Firewall examine it for security violations and then compress it and send it out to clients.

**Note**
The cache should be cleared when enabling compression to avoid retrieving already present uncompressed cached objects.
This chapter describes the configuration and monitoring tasks you can perform from the Web interface. The following topics are covered:

Configuring Web Site Translation.................................................... 108
Configuring Web Site Translation

Web site translation allows Web site cloaking and translation of URLs and headers in requests and responses. It translates the internal codes, headers, and cookies so they are concealed from external users. Web site translation allows you to set a variety of address translation rules for application-specific packets sent through the Barracuda Web Application Firewall.

Note

Web Site Translation is only available with model 460 and above.

Configuring URL Translation

When a Web server returns a URL, sensitive information about the Web server may be revealed, which could be used to launch a variety of Web attacks against the server. URL translation modifies the prefix, domain, and response body of an internal URL to an externally viewable URL, thus preventing potential attacks.

URL translation can externalize internal applications, which link to internal servers (not defined in the external DNS name space). For example, Company ABC has an internal application registered in the internal DNS as finance.abc. URL translation can 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 so the internal application is available on the public Internet as www.companyabc.com/finance.abc.

To configure URL Translation, use WEBSITES > Web Site Translations URL Translations. For detailed configuration instructions, see online help.

Configuring HTTP Request Rewrite

HTTP Request Rewrite allows incoming requests to be rewritten or redirected. Headers can be added, removed, or edited on the Barracuda Web Application Firewall before the request is forwarded to the back-end server. The URL can be rewritten to map to a different resource. A redirect response can also be issued to the clients to point them to an updated location or resource.

For example, Request Rewrite could relay the client IP address to the back-end server (in Proxy mode), by inserting the header X-Forwarded-For with the value of the client IP. The back-end server could then extract and use this value, to perform client analytics, for example. Similarly authentication parameters (like certificate details or user name) could be forwarded by inserting request headers and using macros.

HTTP Request Rewrite is available only with Barracuda Web Application Firewall model 460 and higher.

To configure HTTP Request Rewrite, use WEBSITES > Web Site Translations HTTP Request Rewrite. For detailed configuration instructions, see online help. To format a Request Rewrite Condition refer to Format of Request Rewrite and Response Rewrite Conditions on page 109.

Configuring HTTP Response Rewrite

This policy sets rewrite rules for outbound responses. It allows you to add, delete, or rewrite headers. Response Rewrites are used for many purposes. For example, if a response included a header listing
the source IP address, response rewrite could delete that header preventing external users from seeing
the actual IP address of the server.

To configure HTTP Response Rewrite, use **WEBSITES > Web Site Translations HTTP Response
Rewrite**. For detailed configuration instructions, see online help. To format a Response Rewrite
Condition refer to **Format of Request Rewrite and Response Rewrite Conditions** on page 109

**Format of Request Rewrite and Response Rewrite Conditions**

Request rewrite condition and Response rewrite condition are expressions that specify the condition
for which a rewrite will occur. They include a combination of tokens and operations on those tokens.
Tokens differ for Requests and Responses, but operations acting on the tokens are common to both
Request Rewrite Conditions and Response Rewrite Conditions. An asterisk indicates there are no
conditions for rewrite, so the rewrite applies to all.

**Operations for Request Rewrite and Response Rewrite Conditions**

The following are operations that can be used in a Rewrite Condition 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:

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

More than one expression can be grouped together using parenthesis '()' and '}'.

**Response Rewrite Tokens**

A response rewrite condition expression consists of an operation(s) being carried out on one of the
following tokens. Each of the following tokens are case insensitive.

- **Header**: 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**: 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**: The status code of the response returned by the servers. Example: Status-Code eq
  200

**Request Rewrite Tokens**

The Request Rewrite condition expression consists of an operation being carried out on one of the
following tokens. Each of the following tokens are case insensitive.
- **Header**: 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**: 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". No other operation is 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 Uniform Resource Identifier which identifies the resource upon which to apply the request. Example: URI rco /abc*html

- **Method**: The HTTP method in the request. Example: Method eq GET

- **Http-Version**: The version of the HTTP protocol of the request. Example: HTTP-Version eq HTTP/1.1

- **Parameter**: The 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**: The portion of URL which contains extra information about the path of the resource on the server. Example: pathinfo rco abc*

### Configuring Response Body Rewrite

This policy sets the rule for searching and replacing any text string in the response body. Only responses whose content-type begins with text/ can be searched, including text/html, text/plain, text/javascript, text/css, text/xml. Neither flash nor applet content can be searched. The search and replace strings should be text rather than regex expressions. Metacharacters cannot be used, such as \r or \n in either search or replace, which means you cannot search and replace any multi-byte charset strings.

To configure Response Body Rewrite, use **WEBSITES > Web Site Translations > Response Body Rewrite**. For detailed configuration instructions, see online help.
This chapter describes how to configure user authentication and access control through the Barracuda Web Application Firewall. The following topics are covered:

- **Overview** .......................................................... 112
- **Steps to Configure Access Control to your Web site** ........ 113
- **Configuring an Internal Authentication Database** ............. 113
- **Configuring an External Authentication Database** ............ 113
- **Creating an Authentication Policy** ................................ 114
- **Creating an Authorization Policy** ................................. 114
- **Allowing/Denying Client Certificates** ............................ 117
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 can access your Web sites and what level of access each user has. By combining these with SSL encryption, you can create a secure VPN tunnel to your Web sites.

Authentication can be implemented only for HTTP or HTTPS Web services. The authentication process requires users to provide a valid name and password to gain access. A validated user has qualified access to the Web site; that is, the data and services this user can access depend on the authorization level. The following figure illustrates the authentication process:

![Authentication Process Diagram](image)

**Figure 12.1: Authentication Process Diagram**

The user accesses a login page (a GET request), a form for entering a username and password. The login form must be accessible to all users, but need not reside on a back-end server. The Barracuda Web Application Firewall includes a default login form which can be used instead of creating your own login page. The user submits the form (a POST request) and the Barracuda Web Application Firewall compares the submitted information against an internally or externally located authentication database. If successfully authenticated, the user receives a cookie back, and is redirected to a success page. On subsequent requests, after verifying proper authorization, the Barracuda Web Application Firewall forwards the request to the desired location.

If a user fails authentication, he is redirected to a failed authorization page (not illustrated in the figure). When an authenticated user attempts to access an unauthorized page, for which he does not have permission, he is redirected to a denied authorization page.
Steps to Configure Access Control to your Web site

1. Configure an internal or external authentication database. To configure an internal database, set up local users and groups on the ACCESS CONTROL > Local User/Groups page; to configure an external database, use the ACCESS CONTROL > Authentication Services page.

2. Associate the authentication database with your Web site, using the ACCESS CONTROL > Authentication page.

3. After setting up authentication, configure the authorization policy for your Web site using the ACCESS CONTROL > Authorization page. You can configure the authorization policy for the whole Web site (for example, /*) or for only certain parts of the Web site (for example, /abc/* only).

Configuring an Internal Authentication Database

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.

Detailed instruction for configuring an internal Authentication Service are available using ACCESS CONTROL > Local Users/Groups online help.

Configuring an External Authentication Database

External authentication databases are configured on the ACCESS CONTROL > Authentication Services page.

**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.

Detailed instruction for configuring an LDAP Authentication Service are available using ACCESS CONTROL > Authentication Services online help.

**RADIUS**

The RADIUS protocol is based on a client/server model. The Barracuda Web Application Firewall can 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.

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 to authenticate a user.

Detailed instructions for configuring a Radius Authentication Service are available using **ACCESS CONTROL > Authentication Services** online help.

**Configuring a Secondary RADIUS Server**

The Barracuda Web Application Firewall allows you to add a secondary RADIUS server for authenticating users, to which a request is forwarded if the primary RADIUS server fails to respond. When configuring the secondary server, note all parameter values including shared secret of the secondary RADIUS server must be identical to the primary RADIUS server, except the server IP address and port number. To configure a secondary RADIUS server, click **Add** next to the RADIUS authentication service you want to add the secondary server to. Then specify values for **Secondary Server IP** and **Secondary Server Port**. Use **ACCESS CONTROL > Authentication Services** online help for more detailed instructions.

**SiteMinder**

SiteMinder Authentication service identifies a database server supporting the SiteMinder protocol, which 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 SiteMinder directory server for authenticating a user.

Detailed instruction for configuring a Siteminder Authentication Service are available using **ACCESS CONTROL > Authentication Services** online help.

**RSA SECURID**

RSA SecurID authentication service uses the RSA Authentication Manager database to authenticate the identity of users based on two factors: the current code generated on the user's assigned RSA SecurID authenticator, and a secret memorized Personal Identification Number (PIN) before granting access to protected resources.

Detailed instruction for configuring an RSA SecurID Authentication Service are available using **ACCESS CONTROL > Authentication Services** online help.

**Creating an Authentication Policy**

The **ACCESS CONTROL > Authentication** page allows you to specify the parameters and resources to bind a configured authentication database with your service and configure authentication of users. For detailed service authentication configuration instructions, refer to online help.

**Creating an Authorization Policy**

The **ACCESS CONTROL > Authorization** page allows you to configure custom access across your Web site allowing or denying users or groups access to specific services. Access control for a service is configured according to URL and Host Match. Configure access control for a URL key of a service to restrict which users/groups can access that service. Customized access is allowed based on the category of the user/group.
Detailed instructions for configuring an Authorization policy are available using **ACCESS CONTROL > Authorization** online help.

### Single Sign-On (SSO)

Single Sign-On (SSO) allows a user to access multiple applications across different Web servers and platforms, without having to re-authenticate for each. A single set of login credentials authenticates and authorizes a user across configured domains of the Web site. With a successful login, an SSO User Session Cookie is generated providing the user authentication to all configured parts of the Web site for some duration. If login is unsuccessful, the user’s authentication request is rejected. The Barracuda Web Application Firewall supports both single domain and multi-domain SSO.

#### Single domain SSO

You can set up single sign-on to a domain, allowing authenticated users to access all or a subset of restricted resources by authenticating just once. To set up Single Sign-on for a single domain, use the **ACCESS CONTROL > Authentication** page, and identify the service for which you want to set up single sign-on. Edit that service, selecting an **Authentication Service** from the drop-down list to be bound with the service. Make sure to set the **Session-Cookie Domain** to the same domain name as the other services for which you want to configure single domain SSO. (For example, service1 and service2 would both have ‘barracuda.com’ as the session cookie domain). More detailed instructions are available in online help for configuring SSO.

**Note**

In a Single domain SSO set up, ensure that you configure same **Session Cookie Domain** name for each service 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 a user logged into ‘host1.bc.com’, ‘host2.bc.com’ and ‘host3.bc.com’ using ‘bc.com’ as the cookie domain. A logout in host1.bc.com removes the user session cookie from the browser. This automatically logs the user 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 authentication to be honored by 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 the 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, /ncsso.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 '/ncsso.process', then the master service URL for the slave domain is 'http://www.abc.com/ncsso.process'.

**Multi-domain Single Sign-On Functionality**

Attempts to access the master domain first are handled differently than attempts to access the slave domain first. For example, consider master domain 'www.abc.com’ and slave domain ‘www.xyz.com’. Attempts to access the master domain first (www.abc.com) result in a challenge to provide login credentials. An SSO User Session Cookie is generated on successful login. The user gains access to the master domain and can navigate to the slave domains using the generated session cookie without having to re-authenticate. Attempts to access the slave domain first (www.xyz.com) redirect the user to the master service URL for authentication using login credentials. If successful, SSO User Session Cookies are generated for both domains (master and slave) and access to the slave domain is allowed.

To set up Multi-domain Single Sign-On, use the ACCESS CONTROL > Authentication page. First identify the service that you want to configure as a master domain, clicking Edit next to that service. You must select an Authentication Service to bind with the service. To identify this service as the master service that provides cookies for the subsequent slave domains, set Master Service to ‘Yes’. Also, specify the URL path in Master Service URL. To configure a slave domain, identify a service that you want to configure as a slave domain clicking Edit next to that service. Again select the Authentication Service to bind with the service. In this case, the parameter Master Service should be ‘No’ so this service is identified as a slave domain. Specify the Master Service protocol, host, master domain and URL path in Master Service URL. More detailed instructions for configuring Single Sign-On are available in Online help.
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 browser by expiring it. For a logout performed from the slave domain, the slave domain’s cookie is removed by expiring it and the user is redirected to the master domain, which removes the master ‘s cookie.

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, with master domain www.xyz.com. When a user performs a logout in the master domain (www.xyz.com), the user session cookie is removed by expiring it, automatically logging 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, and redirects to the master domain (www.xyz.com) requesting the master domain to expire its cookie to logout the user. The master domain www.abc.com then redirects the user to logout from www.def.com. To achieve this, configure Auth Logout Success URL as ‘http://www.def.com/nclogin.submit?f_method=LOGOUT’ under authentication of www.abc.com. This assumes that ‘nclogin.submit’ is configured as the login-processor-path in www.def.com.

For multiple slave domains, you need to configure the Auth Logout Success URL in www.def.com for the corresponding next domain (e.g. ‘http://www.ghi.com/nclogin.submit?f_method=LOGOUT’ for next domain www.ghi.com) and so on.

Steps involved in chained logout:
2. www.xyz.com expires it's 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 it's 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 step2.
6. The SSO User Session cookie of all the three domains have 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.

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.

Allowing/Denying Client Certificates

The ACCESS CONTROL > Client Certificates page allows you to define allow/deny rules based on Client Certificates. These settings aren’t used unless Enable Client Authentication is Yes for the service, configurable using Edit on the service from the BASIC > Services page.

When Client Authentication is turned on for a service, all clients are required to present a certificate to access the Web Site. The certificate is first checked for validity. A valid certificate cannot have expired, and must be signed by a certificate authority (CA) listed under Trusted Certificates for the
service. Even a valid certificate signed by a trusted CA can be rejected based on the certificate attributes. This allows revocation of a once valid certificate.

Each Allow/Deny rule has the following important attributes:

- A sequence number specifying the order in which to evaluate the rule.
- A set of attribute matches (like 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 compared to all Allow/Deny rules. The rules are compared in sequence number order, starting from the lowest sequence number. Each attribute in the rule is compared, and if all attributes match a rule, the corresponding action (Allow or Deny) is taken and no further rules are compared.

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

To require only matching Client Certificates to be allowed, create a Deny rule with a high sequence number (10000, for example) which matches all rules (has * for all attributes) and the action Deny. Every certificate which doesn’t match any other rule will be denied. This requires that every allowed certificate must have a corresponding Allow rule with a lower sequence number.

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

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

While complex rules can be built if needed, in general, allowing all certificates signed by a trusted CA is recommended, using the Allow/Deny list only to revoke access for certificates that have been issued but are no longer allowed. Since the serial number is unique among all certificates issued by a single CA, it can be reliably used to uniquely identify a certificate. The Common Name (CN) may also uniquely identify a certificate that must be revoked.

**Configuring Allow/Deny Certificate Rules**

Detailed instructions for configuring Allow/Deny Certificate rules are available through ACCESS CONTROL > Client certificates online help.

**Client Certificate Validation using OCSP**

The Barracuda Web Application Firewall supports Online Certificate Status Protocol (OCSP) to determine updated status of a digital certificate. While Certificate Revocation Lists (CRLs) provide certificate status which is updated periodically, OCSP provides more recently updated revocation status information for certificates. A central OCSP server is configured to collect and update CRLs from the different Certificate Authority (CA) servers. OCSP server (also know as OCSP responder) is a trusted Certificate Authority (CA) server. When OCSP is enabled, the Barracuda Web Application Firewall communicates with the OCSP server to validate the revocation status of client...
certificates. SSL connections from the clients are allowed or denied based on the status of the client certificate presented to the Barracuda Web Application Firewall.

**Functioning of OCSP Validation**

When a user attempts to access a server, an OCSP status request for the user certificate is sent to an OCSP responder (central OCSP server). The OCSP responder receives the request and validates whether the request contains information required to identify the certificate. The OCSP responder returns a signed response message indicating the status of the certificate as follows:

- "GOOD" indicates a positive response that the certificate is not revoked.
- "REVOKED" indicates that the certificate has been revoked.
- "UNKNOWN" indicates that the responder has no information about the requested certificate.

In case of any error or failure, the responder may return an unsigned message indicating the failed communication, logged under **System Logs**. Errors could occur because of a malformed request, an internal error, or an unauthorized request.

**Note**

**Enforce Client Certificate** should be **Yes** for a service on the **BASIC > Services** page for Barracuda Web Application Firewall to perform client certificate authentication using OCSP.

**Configuring OCSP Validation**

Detailed instructions for configuring a service to enforce client certificate validation using OCSP are available in **ACCESS CONTROL > Client Certificates** online help.
This chapter provides an introduction to the Public Key Infrastructure (PKI) technology, including a system overview of how the Barracuda Web Application Firewall uses PKI encryption to protect traffic:

Overview ........................................................................................................ 122
SSL Implementation and Configuration ....................................................... 123
Certificates ..................................................................................................... 124
Overview

The Barracuda Web Application Firewall implements Secure Socket Layer (SSL) encryption using PKI objects. Besides encrypting transmitted data and this technology allows authentication of sender and receiver and is the most effective way to securely send confidential information over the Internet. The Barracuda Web Application Firewall allows you to use SSL encryption between a client and the Barracuda Web Application Firewall, and/or between the Barracuda Web Application Firewall and Web servers. To implement SSL, the Barracuda Web Application Firewall allows the creation or upload of Public Key Infrastructure (PKI) objects like keys and certificates.

PKI technology allows secure exchange of data over the Internet using key pairs for authentication and encryption. This type of cryptography starts with the creation of two keys: a public key known by everyone, and a private key known only by its owner. This key pair is used to encrypt and decrypt messages sent by an owner. The public key allows initiation of a secure communication. The private portion of the key pair confirms the owner’s identity.

In an SSL transmission between a client and a server, the client requests a secure connection, and the server responds with a certificate, identifying the certificate authority (CA) and the server’s public encryption key. This allows the client to verify the server identity. If satisfied with the authenticity of the server, the client sends a test transmission which can only be decrypted with the private key of the server. This transmission allows both parties to generate encryption and decryption for the impending transaction. In addition, a server may require the client to authenticate itself by providing a certificate, and refuse to communicate with clients who fail to do so.

The Barracuda Web Application Firewall acts as a server on the front-end (Internet facing), receiving client requests. On the back end, the Barracuda Web Application Firewall acts as a client to the Web servers, forwarding safe requests to the servers. In each case, data can be secured using SSL, providing end-to-end secure data for requests and responses.

The Barracuda Web Application Firewall allows Certificates obtained from a trusted CA to be uploaded, or can create of a self-signed certificate to implement SSL.

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 an entity and then allows access to the Web site.
- **Privacy** - A certificate 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, authorization required users to give an ID and password. Certificates guarantee the authenticity of each user, thus providing a level of authorization.

Digital certificates created using the Barracuda Web Application Firewall are of the standard X.509 format and are considered self-signed.
SSL Implementation and Configuration

SSL for Client to Barracuda Web Application Firewall Transmissions

The Barracuda Web Application Firewall receives requests from clients on behalf of the back-end server. When a request is received from a client, the Barracuda Web Application Firewall acts as a server to the requesting client. It can be configured to provide a certificate (a self signed certificate created on the appliance, or a certificate issued by a trusted CA uploaded to the appliance) which allows the client to authenticate the request transactions and send them in encrypted form.

To generate self-signed certificates, or to upload trusted certificates, saving them on the Barracuda Web Application Firewall, use BASIC > Certificates, and follow the instructions provided in online help.

To configure the Barracuda Web Application Firewall to use SSL in client communication, create an SSL enabled service and refer to Configuring SSL for SSL Enabled Services on page 55 for specific instructions on configuring SSL. Additionally, the Barracuda Web Application Firewall can be configured to require the client to provide a certificate for authentication, denying communication with clients who fail to do so. For more information on Client Certificates, refer to Allowing/Denying Client Certificates on page 117.

SSL for Barracuda Web Application Firewall to Server Transmissions

The Barracuda Web Application Firewall also provides server-side encryption, and can provide a certificate to the servers for client authentication (the Barracuda Web Application Firewall acting as the client to the back-end servers). This protects services configured on the Barracuda Web Application Firewall. The client-server negotiations include the following:

- The Barracuda Web Application Firewall receives and verifies the Real server’s certificate.
- The Barracuda Web Application Firewall may provide a certificate in return if client authentication is required by the back-end server.

The SSL handshake allows the server and the Barracuda Web Application Firewall to authenticate each other. Once mutually authenticated, both use keys for encryption, decryption, and tamper detection during the SSL sessions.

To configure the Barracuda Web Application Firewall to use SSL in Server communication, Add a server for the respective service on BASIC > Services, and configure the Barracuda Web Application Firewall to validate the server certificate and optionally to present a client certificate using the instructions in Back-end SSL Server Configuration on page 57. For more information on Client Certificates, refer to Allowing/Denying Client Certificates on page 117.
Certificates

A signed certificate is a digital identity document that enables both server and client to authenticate each other. It can be generated or procured from a third party CA. Generated certificates 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.

Certificate Components

Key Pair

The Barracuda Web Application Firewall implements an asymmetric methodology for encryption, where two related keys are used in combination. A key pair consists of a public key and a private key which work together, with one of the key pair encrypting messages, and the other decrypting encrypted messages.

Exposure of the public key does not endanger the secure transactions because the private key cannot be derived from it.

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 trusted certificate is a third-party certificate issued by a Certificate Authority (CA) which can be uploaded and saved on the Barracuda Web Application Firewall. This certificate can be added to a certificate chain, where it is used for encryption and authentication. Browsers requiring certificates from a CA will require the procurement and upload of the certificate before communication between a client and a server can be established.

Creating a Test Certificate

A self-signed X.509 digital certificate can be created by the Barracuda Web Application Firewall. This certificate is one of the most commonly used types of certificates 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 accepted by clients or Web servers. This certificate is a self-signed certificate and is also called as a user certificate.
To create a test certificate use BASIC > Certificates Certificate Generation and follow the detailed instructions in online help.

**Saved Certificates**

All the created (self-signed) certificates and uploaded (generated then signed by a third party, or trusted) certificates are listed under BASIC > Certificates Saved Certificates. The certificates created using the Barracuda Web Application Firewall are secure and authentic, though 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. A generated certificate can also be sent to a third party for signing, by downloading the CSR file and sending it to a trusted CA such as VeriSign or Thwate for signing. The following are the two options for uploading a signed certificate:

**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 file can be 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 signed 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.

The key can be extracted from the certificate, and then the certificate can be uploaded to the Barracuda Web Application Firewall using BASIC > Certificates Upload Certificate.

**Extracting the key from the Certificate**

Once the CSR is signed and returned, the certificate file is replaced by the new certificate. 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* below for more information about uploading a signed certificate.

**Note**

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

---

**Uploading a Certificate**

**BASIC > Certificates Upload Certificate** allows signed certificates to be uploaded on the Barracuda Web Application Firewall and used for SSL encryption. A signed-certificate can be obtained by generating and sending out a certificate to a third party Certificate Authority (CA) organization such as VeriSign or Thawte. You can upload the returned signed certificate in PKCS12 or PEM format.

**To upload a certificate in PKCS12 Token Format**

1. In the **Upload Certificate** section, specify values for the following fields:
   - **Certificate Name** - Specify a name to identify this signed certificate.
   - **Certificate Type** - Select **PKCS12 Token** as the certificate type.
   - **Signed Certificate** - Click **Browse** to select the signed certificate file in PKCS12 format. When uploading a signed certificate as a PKCS12 token, ensure that the file is in .pfx extension otherwise it is treated as a pem file. Also note that when uploading the certificate in pfx format, the intermediary certificates, if any, should also be bundled in the pfx file.
   - **Certificate Password** - Specify the password used to generate the PKCS12 token for the signed certificate being uploaded.
   - **Disallow private key export** - Selecting this option will lock the private key corresponding to this certificate. Normally certificates are downloaded in PKCS12 format which includes the private key and certificate. When the key is locked, you can only download the certificate in PEM format. Also you cannot take a backup when the private key is locked.

2. Click **Upload Now** to upload the certificate.

**To upload a certificate in PEM Format**

1. In the **Upload Certificate** section, specify values for the following fields:
   - **Certificate Name** - Specify a name to identify this signed certificate.
   - **Certificate Type** - Select **PEM Certificate** as the certificate type.
   - **Signed Certificate** - Click **Browse** to select the signed certificate file in PEM format.
   - **Assign the associated key** - Select 'Yes' if the private key corresponding to this certificate is already uploaded on the Barracuda Web Application Firewall. If not, select 'No' and upload the private key in the **Certificate Key** field.
   - **Certificate Key** - Click **Browse** to select the corresponding private key for the signed certificate being uploaded. This key must be unencrypted and it should be in PEM format.
   - **Intermediary Certificates** - Click **Browse** to select the intermediary CA certificate. You can add a single or a set of intermediary CA certificates by clicking the '+' button. Note
that uploading the signed certificate in pfx format and specifying the corresponding intermediary certificates in pem format here would not work.

- **Disallow private key export** - Selecting this option will lock the private key corresponding to this certificate. You cannot take a backup when the private key is locked.

2. Click **Upload Now** to upload the certificate.

**Uploading a Trusted Certificate**

A trusted-certificate is a certificate generated by and 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.
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** ................................................................. 130
- **XML Firewall** .............................................................. 134
- **XML Validations** ......................................................... 135
- **Advanced Web Service Validation** .............................. 137
Web Services

Web applications are designed to serve human users, taking input from and sending output to human users. In contrast, a Web service is designed to serve other Web applications. Web services share business logic, data, and processes through a programmatic interface. Web services allow businesses to communicate with each other and with clients without requiring inside knowledge of each other's infrastructure and security configurations. They are used to assist organizations in streamlining business processes, providing increased efficiency and reduced application integration costs.

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 available services.

For example, consider two banks needing to share account balance information. Figure 14.1 illustrates the steps to share the data using a Web service:

1. Bank A creates a Web service description (in the XML-based WSDL format) that describes Web service required inputs and outputs (for example, 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 restricted to approved intermediaries, requiring 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, or send a receipt. Each function requires that information be secure from unauthorized access, attack, and data theft.
A business can combine multiple Web services to accomplish a task. For example, a travel service might define one Web service for interacting with a 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 for communicating with one or more hotel services, and another for communicating with one or more airline services.

**WSDL**

A Web service description (WSDL document) is a human-readable document, written from the Web service perspective, that describes the expectations and functionality of a particular Web service, and clarifies how client and service should interact. A potential client reads the Web service description, to learn how to correctly interact with the service.

WSDL is an XML grammar for describing network services as collections of communication endpoints capable of exchanging messages. WSDL service definitions provide documentation for distributed systems and serve as a recipe for automating the details of application communications. See *WSDL 1.2 Syntax* on page 214 for the complete syntax defined in WSDL version 1.2.

**Web Service Vulnerabilities**

Web services are vulnerable to many of the same attack risks as other Web applications, but also face additional vulnerabilities including:
• Publicly available WSDL documents provide a blueprint for the service. The document details messaging request and response, expected parameters (including data type), and available operations for the service. By analyzing its WSDL document, a hacker not only knows exactly what the service is supposed to do, but also which parts are open to attack through techniques such as malformed SOAP messages and other XML parser attacks. A WSDL document may reveal what tools generated the Web service, providing attackers with more insight into potential vulnerabilities.

• SOAP and XML are standards used to wrap data for easy consumption. SOAP envelops information to deliver messages seamlessly between applications. XML includes metadata to describe the structure of the information. CDATA is used to delineate information in the message that should not be parsed. Malicious code or characters can be embedded in the elements or CDATA, allowing unintentional display or execution by the receiving application or service. XML encapsulation (a form of cross site scripting) can embed commands that tie up system resources or gain unauthorized access.

• XML-based attacks can overload XML parsers which process SOAP messages. Attackers that put in recursive relationships to create entity expansions, bogus parameters, or even significant amounts of white space, can cause XML parsers to be overloaded or to behave unpredictably.

• Any type of application behind a Web service interface, including a packaged application, an internally developed application, a desktop application, or a legacy mainframe application carries its own security vulnerabilities. These inherent security risks are even more exposed through a Web service interface. Because each application type approaches security its own way, it’s a significant security challenge to protect these services.

### Web Service Protections

The following table describes possible Web service attack techniques and the corresponding protection provided by Barracuda Web Application Firewall.

*Table 14.1: Protection against XML and SOAP based attacks*

<table>
<thead>
<tr>
<th>Technique</th>
<th>Description of Attack</th>
<th>Barracuda Web Application Firewall Protection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema Poisoning</td>
<td>Manipulating the WS schema to alter the data processed by the application.</td>
<td>Protects against schema poisoning by validating that content adheres to the defined WSDL and schema.</td>
</tr>
<tr>
<td>XML Parameter Tampering</td>
<td>Injection of malicious scripts or content into XML parameters.</td>
<td>Protects against parameter tampering by validating that parameter values are consistent with the WSDL and schema specifications.</td>
</tr>
<tr>
<td>Inadvertent XDoS</td>
<td>Sending poorly encoded SOAP messages that cause the application to fail.</td>
<td>Inspects SOAP at the header, envelope, and message level to ensure proper structure and content.</td>
</tr>
<tr>
<td>WSDL Scanning</td>
<td>Scanning the WSDL (business API) to uncover sensitive information about the application data format.</td>
<td>Uses Web services cloaking to hide the true internal URI of sensitive Web services.</td>
</tr>
<tr>
<td>Coercive Parsing</td>
<td>Injection of malicious content into the XML.</td>
<td>Utilizes real-time WS-I checking and content inspection to block malicious payloads.</td>
</tr>
<tr>
<td>Oversized Payload</td>
<td>Sending oversized files to create an XDoS attack (similar to a buffer overflow attack).</td>
<td>Inspects transmitted data and enforces element, document, and other maximum sizes.</td>
</tr>
</tbody>
</table>
### Table 14.1: Protection against XML and SOAP based attacks

<table>
<thead>
<tr>
<th>Technique</th>
<th>Description of Attack</th>
<th>Barracuda Web Application Firewall Protection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recursive Payload</td>
<td>Sending mass amounts of nested data to create an XDoS attack on the XML parser.</td>
<td>Validates WSDL and schema formats, inspects SOAP headers, envelopes, and messages, and ensures that WS-I standards are met.</td>
</tr>
<tr>
<td>SQL Injection</td>
<td>Hiding a malicious SQL command inside a SOAP wrapper attempting to uncover or modify back-end data.</td>
<td>Utilizes real-time WS-I checking and content inspection to compare to schema.</td>
</tr>
<tr>
<td>Replay Attacks</td>
<td>Using repetitive SOAP messages to force an XDoS attack.</td>
<td>Includes request-level throttling technology to ensure resources cannot reach a fail state.</td>
</tr>
<tr>
<td>External Entity Attack</td>
<td>Parses XML input from untrusted sources using an incorrectly configured XML parser.</td>
<td>Can suppress external URI references to protect against external manipulation of data.</td>
</tr>
<tr>
<td>Information Disclosure</td>
<td>Exposes unencrypted Web Service message data to anyone watching application traffic.</td>
<td>Has extensive SSL security capabilities at the ASIC level to ensure end to end encryption of XML traffic.</td>
</tr>
<tr>
<td>Malicious Code Injection</td>
<td>Delivers scripts embedded deep within SOAP messages directly to applications and databases.</td>
<td>Ensures SOAP messages conform to customized policies.</td>
</tr>
<tr>
<td>Identity Centric Attack</td>
<td>Forges credentials in an attempt to access sensitive data.</td>
<td>Enforces authentication (basic or strong) at the SOAP message level.</td>
</tr>
<tr>
<td>Processing Instructions (PI)</td>
<td>Uses PI (a text data section that is ignored by the XML parser) to pass instructions to applications.</td>
<td>Can block requests containing Processing Instructions (PI).</td>
</tr>
<tr>
<td>Inline or external DTDs</td>
<td>Uses DTDs (a text data section that is ignored by the XML parser) to pass instructions to applications.</td>
<td>Can block requests containing both inline or external DTDs.</td>
</tr>
<tr>
<td>External References</td>
<td>Uses requests containing external entities including external URI references or external DTDs (text data sections that are ignored by the XML parser) to pass instructions to applications.</td>
<td>Can block requests containing external entities including external URI references or external DTDs.</td>
</tr>
</tbody>
</table>
The Barracuda Web Application Firewall XML Firewall includes XML Validations and XML Protection. It is available only on Barracuda Web Application Firewall Model 660 and higher.

Barracuda Web Application Firewall protects Web services using the following XML firewall features:

- **WSDL** - Allows import of WSDL file and binding of the WSDL to configured Web Sites. Then enforces rules for requests, responses, or both for SOAP data.
- **XML Validation Settings** - Sets default length limits; assumes requests or responses with lengths greater than the defaults are possible attacks.
- **WS-I Basic Profile Tests** - Applies WS-I 1.0 basic profile tests in run-time according to configuration.
- **SOAP Validations** - Enforces SOAP conformance according to configuration.

XML Firewall is disabled by default. When enabled, XML firewall enforces XML and SOAP based validations configured on ADVANCED > XML Protection.

**To enable XML Firewall**

- From WEBSITES > XML Validations, set the parameter Enable XML Firewall to 'Yes'.

**Tasks to enforce XML Firewall**

Configure XML Firewall to protect a Web service using the following steps:

1. Create a service using BASIC > Services. (for more information refer to Services on page 53).
2. Locate and import the WSDL and Schema files for your Web application to the Barracuda Web Application Firewall (for more information refer to Import Schema/WSDL on page 135). Associate the imported WSDL and Schema files with the target Web service (for more information refer to Protected URLs on page 136). The Barracuda Web Application Firewall can then compare and validate SOAP messages using Web service policies created from the WSDL file.
3. Configure an XML firewall by setting the following firewall features:
   - **3a.** Set SOAP validation parameters to ensure requests conform to SOAP standards.
   - **3b.** Enable WS-I basic profile test assertions to be applied during run-time validation of XML and SOAP messages.
   - **3c.** Set XML Validation rules to ensure SOAP messages conform to customized policies such as size limits, and XML filters.
**XML Validations**

**WEBSITE > XML Validations** allows the import of a WSDL file into the Barracuda Web Application Firewall and binds it with the target Web service. By default XML Validations is disabled. To enable XML Validations, first enable XML Firewall.

**To enable XML Firewall**

- From the **WEBSITES > 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**

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. To enforce WSDL specifications for a Web service, locate and import the associated WSDL and Schema files for your Web application to the Barracuda Web Application Firewall.

For more information about a WSDL file, refer to **WSDL** on page 131 and **WSDL 1.2 Syntax** on page 214.

**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 136.
   - 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 only on URLs that require it, since XML Firewall validates request content, introducing additional latency in serving requests.

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** - Specify 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". Note: Ensure you specify a valid URL that matches with any of the Port URLs of the selected WSDL file. If the URL contains regular expression such as "/*", the following validations are exempted for the Web service:
     - WSI1011: Request content matches WSDL - If the enforcing direction is Request or Both.
     - WSI1013: Response content matches WSDL - If the enforcing direction is Response or Both.
   - **SOAP Validations** - Validate SOAP headers defined in WSDL, Allow additional SOAP Headers and Validate SOAP body from WSDL schema.
   - **Direction** - Specify 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** - Specify 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.
Advanced Web Service Validation

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 Enable XML Firewall on WEBSITES > XML Validations.

To enable XML Firewall
- From the WEBSITES > XML Validations page, set the parameter Enable XML Firewall to 'Yes'.

XML Validation Settings

The XML Validation Settings allow configuration of custom validation rules for XML requests or responses. Rules can enforce a maximum number of total elements in the XML, or limit message size or total number of bytes per message. These help prevent attacks from flooding the service with too much data.

SOAP Validations and the WS-I Basic Profile tests (described in the next sections) determine whether a SOAP message is valid, identifying invalid messages as intrusions. Blocking the invalid messages is enabled through XML Validation Settings.

The XML validation parameters are set to default values which can be modified.

XML requests violating the XML validation rules are associated with 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 to implement the desired attack response.

WS-I Basic Profile Assertions

The Web Services Interoperability Organization (WS-I) Basic Profile Version 1.0 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 define how the 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 to validate SOAP messages.

There are forty two test case parameters, all set to Yes by default, meaning the test is applied; a No setting would cause that test to be ignored. You can modify existing settings.

XML requests violating 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 to implement the desired attack response.

SOAP Validations

SOAP is the transfer mechanism protocol for sending Web service descriptions in an HTTP message. The SOAP validation parameters set the SOAP validation checks to apply. (These checks verify the message adheres to SOAP standards.)
SOAP is a lightweight communication protocol for exchanging data using XML over HTTP. SOAP 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 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 setting them to Yes to validate these SOAP standards.

The XML requests which violate 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 to implement the desired attack response.
This chapter describes and explains monitoring features, logs, and reports available for the Barracuda Web Application Firewall. The following topics are covered:

- Monitoring Barracuda Web Application Firewall .......... 140
- Logs ................................................................................. 145
- Reports ............................................................................... 150
The Barracuda Web Application Firewall incorporates hardware and software fail-safe mechanisms and provides system visibility to the administrator through alerts and logs. The powerful reporting engine provides a broad spectrum of Web traffic statistics and user-level activity reports which can be created ad-hoc, emailed to administrators, or sent to an FTP server. Monitoring multiple Barracuda Web Application Firewalls can be achieved using the Barracuda Control Center, a centralized management Web interface for managing, configuring, and reporting on multiple devices from one central Web console.

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

- Viewing Performance Statistics ....................................................... 140
- Health Indicators for Services and Servers..................................... 141
- Viewing System Tasks....................................................................... 141
- Front Panel Indicator Lights ........................................................... 141
- Managing Multiple Systems With Barracuda Control Center............. 143
- Receiving Trap Messages and System Alerts................................. 143

### Viewing Performance Statistics

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

- Traffic statistics, showing the number of requests for various types of traffic since the last system reset.
- Subscription status of Barracuda Energize Updates.
- Performance statistics, such as CPU temperature and system load, with values color coded to indicate when they are outside normal ranges.

<table>
<thead>
<tr>
<th>Indicator Color</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green</td>
<td>The component is working as expected.</td>
</tr>
<tr>
<td>Orange</td>
<td>Indicates warning. The component is operating outside of standard limits.</td>
</tr>
<tr>
<td>Red</td>
<td>Indicates danger. Needs immediate attention.</td>
</tr>
</tbody>
</table>

- Hourly and daily traffic statistics.
Health Indicators for Services and Servers

**BASIC > Services** displays the health of Services and Servers.

The following table describes the health indicators displayed for each service and server:

<table>
<thead>
<tr>
<th>Service and Server Health Indicators</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green dot</td>
<td>Service is up; Server is responding.</td>
</tr>
<tr>
<td>Orange dot</td>
<td>Degraded service is available; for example, 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.</td>
</tr>
<tr>
<td>Red dot</td>
<td>Service is down; Server is not responding.</td>
</tr>
</tbody>
</table>

**Viewing System Tasks**

**ADVANCED > Task Manager** displays a list of currently processing tasks. Any errors encountered performing these tasks are also displayed.

Tasks tracked by the Barracuda Web Application Firewall include:

- Cluster setup
- Configuration restoration

Use **Cancel** next to a task name to stop a task. A task taking longer than expected can be cancelled and run at a later time when the system is less busy.

The **Task Errors** section continues to display an error until it is manually removed from the list.

**Front Panel Indicator Lights**

**Indicator Lights on Model 360 / 460 / 660**

The front panel of the Barracuda Web Application Firewall Model 360, 460 and 660 has five indicator lights. These lights blink when the system processes any traffic. *Figure 15.1* displays the location of each of the lights. *Table 15.1* describes each indicator light.

*Figure 15.1: Indicator Lights (Model 360 / 460 / 660)*
Table 15.1: Description of the Indicator Lights

<table>
<thead>
<tr>
<th>Light</th>
<th>Color</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Failed</td>
<td>Red</td>
<td>Blinks when the Barracuda Web Application Firewall is in failed state, which includes STM Down, System Reboot or Kernel Panic.</td>
</tr>
<tr>
<td>Bypass Mode</td>
<td>Orange</td>
<td>Blinks when the Barracuda Web Application Firewall is in bypass mode.</td>
</tr>
<tr>
<td>Network Traffic</td>
<td>Green</td>
<td>Blinks when the Barracuda Web Application Firewall processes traffic.</td>
</tr>
<tr>
<td>Data I/O</td>
<td>Green</td>
<td>Blinks during data transfer.</td>
</tr>
<tr>
<td>Power</td>
<td>Green</td>
<td>Displays a solid green light when the system is powered on.</td>
</tr>
</tbody>
</table>

Indicator Lights on Model 860 and 960

The front panel of the Barracuda Web Application Firewall Model 860 and 960 has six indicator lights. These lights blink when the system processes any traffic. Figure 15.2 displays the location of each of the lights. Table 15.2 describes each indicator light.

Figure 15.2: Indicator Lights (Model 860 and 960)

Table 15.2: Description of the Indicator Lights from left to right.

<table>
<thead>
<tr>
<th>Indicator</th>
<th>LED Color</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Failed</td>
<td>Red</td>
<td>Blinks when the Barracuda Web Application Firewall is in failed state, which includes STM Down, System Reboot or Kernel Panic.</td>
</tr>
<tr>
<td>Bypass Mode</td>
<td>Red</td>
<td>Blinks when the Barracuda Web Application Firewall is in bypass mode.</td>
</tr>
<tr>
<td>Network Traffic</td>
<td>Green</td>
<td>Not used.</td>
</tr>
<tr>
<td>Network Traffic</td>
<td>Green</td>
<td>Blinks when the Barracuda Web Application Firewall processes traffic.</td>
</tr>
<tr>
<td>Data I/O</td>
<td>Orange</td>
<td>Blinks during data transfer.</td>
</tr>
<tr>
<td>Power</td>
<td>Green</td>
<td>Displays a solid green light when the system is powered on.</td>
</tr>
</tbody>
</table>
Managing Multiple Systems With Barracuda Control Center

The Barracuda Control Center enables administrators to manage, monitor, and configure multiple Barracuda Web Application Firewalls at one time from one console. The same tabbed pages are available on the Barracuda Control Center for managing all aspects of Barracuda Web Application Firewall configuration that you see in each individual Web interface, and you can create aggregated reports for multiple devices from the Barracuda Control Center console. You can connect one or more Barracuda Web Application Firewalls to the Barracuda Control Center by doing the following:

1. If you don't already have an account with Barracuda Networks, visit http://login.barracudanetworks.com to create one.
2. Make a note of your username (email address) and password.
3. Log into your Barracuda Web Application Firewall as the administrator. From the ADVANCED > Firmware Update page, check to make sure you have the latest firmware installed. If not, download and install it now.
4. From the ADVANCED > Control Center page, enter the Barracuda Networks username and password you created and click Yes to connect to the Barracuda Control Center. Note that your Barracuda Web Application Firewall can connect with only one Barracuda Control Center account at a time.
5. Log into the Barracuda Control Center with your username and password and you will see your Barracuda Web Application Firewall statistics displayed on the BASIC > Status page. To access the Web interface of your Barracuda Web Application Firewall, click on the link in the Products column in the Control Center pane on the left side of the page. Or you can click on the product name in the Product column of the Unit Health pane on the right side of the page.
6. Follow steps 3 and 4 to connect every subsequent Barracuda Web Application Firewall to the Barracuda Control Center.

To disconnect your Barracuda Web Application Firewall from the Barracuda Control Center, from ADVANCED > Control Center of the Web interface, enter the Barracuda Control Center username and password and click No for Connect to Barracuda Control Center. Do this when you know that there will be a loss of connectivity between the appliance and the Barracuda Control Center due to the appliance being physically moved or other network connectivity issues.

Receiving Trap Messages and System Alerts

From the BASIC > Administration page under Trap Receivers, specify the client IP address and port number to receive trap messages. An alert email is sent to the recipient’s email address, if the email is configured under BASIC > Administration Email Notifications.

Trap defines the SNMP trap for generating customized alerts for an event. The default alerts and their descriptions are given in the following table:

Table 15.3: Trap Messages

<table>
<thead>
<tr>
<th>Trap Name</th>
<th>Object ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TempCritical</td>
<td>1.3.6.1.4.1.20632.8.1.3</td>
<td>Temperature of any one of CPU1, CPU1 VRM, CPU2, CPU2 VRM, board or RAM exceeded its threshold value.</td>
</tr>
</tbody>
</table>
In addition to the above trap messages, the Barracuda Web Application Firewall sends out emails for the following three system alerts:

- Your Energize Update subscription is about to expire.
- New firmware updates are available.
- Your system is low on disk space.

Apart from these, you can use the SNMP GET commands to view important statistics of the Barracuda Web Application Firewall. For more information on the SNMP GET commands, refer to Table C.2: SNMP GET Command on page 211.

<table>
<thead>
<tr>
<th>Trap Name</th>
<th>Object ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TempHigh</td>
<td>1.3.6.1.4.1.20632.8.1.4</td>
<td>Temperature of any one of CPU1, CPU1 VRM, CPU2, CPU2 VRM, board or RAM is higher than 80C.</td>
</tr>
<tr>
<td>SystemFailOver</td>
<td>1.3.6.1.4.1.20632.8.1.5</td>
<td>System failed over to redundant system.</td>
</tr>
<tr>
<td>SwitchingToMaintMode</td>
<td>1.3.6.1.4.1.20632.8.1.6</td>
<td>System is switching to Maintenance mode.</td>
</tr>
<tr>
<td>FanDead</td>
<td>1.3.6.1.4.1.20632.8.1.7</td>
<td>One of the system fan is dead.</td>
</tr>
<tr>
<td>DataPortLinkDown</td>
<td>1.3.6.1.4.1.20632.8.1.8</td>
<td>Data port link (interface1 or interface2) is down.</td>
</tr>
<tr>
<td>ServerDown</td>
<td>1.3.6.1.4.1.20632.8.1.9</td>
<td>Server is down.</td>
</tr>
<tr>
<td>PeerDown</td>
<td>1.3.6.1.4.1.20632.8.1.10</td>
<td>Peer is down in redundant environment.</td>
</tr>
<tr>
<td>DataPortLinkUp</td>
<td>1.3.6.1.4.1.20632.8.1.11</td>
<td>Data port link (interface1 and interface2) is up.</td>
</tr>
<tr>
<td>ServerUp</td>
<td>1.3.6.1.4.1.20632.8.1.12</td>
<td>Server is up.</td>
</tr>
<tr>
<td>PeerUp</td>
<td>1.3.6.1.4.1.20632.8.1.13</td>
<td>Peer is up in redundant environment.</td>
</tr>
<tr>
<td>CookieEncryptionKeyAboutToExpire</td>
<td>1.3.6.1.4.1.20632.8.1.16</td>
<td>Shared secret key is about to expire.</td>
</tr>
<tr>
<td>CookieEncryptionKeyExpired</td>
<td>1.3.6.1.4.1.20632.8.1.17</td>
<td>Shared secret key has expired.</td>
</tr>
<tr>
<td>FirmwareStorageHigh</td>
<td>1.3.6.1.4.1.20632.8.1.18</td>
<td>Firmware storage exceeds 75%.</td>
</tr>
<tr>
<td>LogStorageHigh</td>
<td>1.3.6.1.4.1.20632.8.1.19</td>
<td>Log storage exceeds 85%.</td>
</tr>
<tr>
<td>RaidDegrading</td>
<td>1.3.6.1.4.1.20632.8.1.20</td>
<td>One of the RAID arrays is degrading.</td>
</tr>
</tbody>
</table>
Logs

The Barracuda Web Application Firewall has a comprehensive logging feature to record significant events. Events related to http traffic, actions of the Barracuda Web Application Firewall, and user actions are captured in logs the administrator can view using the Web interface. These log messages enable a system administrator to:

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

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

- **Web Firewall Logs** - Logs all actions/events on the Barracuda Web Application Firewall. These logs help the administrator analyze traffic for suspicious activity and fine tune the security settings.
- **Access logs** - Logs all Web traffic activities. These logs provide information about the Web site traffic and performance.
- **Audit logs** - Logs all administration and configuration activities. This information assists in audits.

Severity Levels are associated with each log, indicating the severity threshold an event or action must surpass to be logged. An administrator configures the severity level according to what errors need to be addressed, and what volume of logs can be persisted.

Besides persisting the logs in internal storage, the Barracuda Web Application Firewall also allows configuration of external syslog servers for persistent external storage.

For more detailed information about the logs, refer to the document *Syslog and the Barracuda Web Application Firewall*.

Web Firewall Logs

**BASIC > Web Firewall Logs** displays all actions/events logged by the Barracuda Web Application Firewall. Use the built-in filters to quickly locate specific types of log entries. Use Preferences to set the number of messages to be displayed per page.

**Note**
Log request processing lengthens when:
1. An invalid filter criteria is entered.
2. The log matching the filter criteria, lies at the end of the log database.

Use Back or More to toggle between pages of log entries. Web Firewall Logs are enabled by default.

**Export to CSV** saves logs in .csv format to your computer.

To configure the log level for Web Firewall Logs use **BASIC > Services**, to Edit the service. Under Basic Security, select the **Web Firewall Log Level** you want. The log level is a threshold, so all errors meeting or surpassing the severity threshold are logged. A lower log level (higher severity threshold) logs less information.

Web Firewall Logs can be used with Policy Tuner to refine security settings. For more information, see *Tuning using Web Firewall Logs and the Policy Tuner Tool* on page 50.
Access Logs

**BASIC > Access Logs** displays all logged Web traffic activities. Use the built-in filters to quickly locate specific types of log entries. Use **Preferences** to set the number of messages to be displayed per page.

**Note**
Log request processing lengthens when:
1. An invalid filter criteria is entered.
2. The log matching the filter criteria, lies at the end of the log database.

Use **Back** or **More** to toggle between different pages of log entries. Access Logs are enabled by default. To disable Access Logs, use **BASIC > Services** to **Edit** the service and set the **Enable Access Logs** parameter to **Off**. **Export to CSV** saves logs in .csv format to your computer.

Audit Logs

**BASIC > Audit Logs** displays recorded activities of the users logged in to the Web Interface of the Barracuda Web Application Firewall. Use the built-in filters to quickly locate specific types of log entries. Use **Preferences** to set the number of messages to be displayed per page.

**Note**
Log request processing lengthens when:
1. An invalid filter criteria is entered.
2. The log matching the filter criteria, lies at the end of the log database.

Use **Back** or **More** to toggle between different pages of log entries. **Export to CSV** saves logs in .csv format to your computer.

**In the following cases, “Logout” is not logged in the Audit Logs:**
- When Barracuda Web Application Firewall is restarted because critical processes have crashed, the current existing sessions won't be logged out.
- When maintenance command is executed by a user or by Barracuda Web Application Firewall, the current existing sessions won't be logged out.

**In the following situation, “Login” is not logged in Audit Logs:**
- When maintenance command is executed by a 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 with the search option to quickly locate specific types of log entries.

**To use the search criteria:**
1. Select the filter column, an appropriate operator from the drop-down list, and enter a search value. Click **Search**, to display the log entries matching the filter criteria.
2. Click ‘+’ to add search fields and ‘-’ to remove them. Multiple search criteria can be specified, with either an AND or OR combination of the filters.

3. Specify the complete timestamp when searching for log messages generated in a specified time. While using multiple search criteria, you cannot duplicate the same fields in the criteria. Back or More toggle between different pages of log entries. Export to CSV saves the logs in .csv format to your computer. You can specify regular expressions (Regexp) for selected fields. See online help for more detailed instructions.

**Export Logs**

ADVANCED > Export Logs displays all parameters and resources for configuring the log policy on a service. The Barracuda Web Application Firewall uses this policy to generate the logs in standard and custom formats, and then exports them to the configured servers.

To export Web log messages to an FTP server, you need to configure ADVANCED > Export Logs FTP Access Logs.

**Note**

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

**Syslog**

The Syslog section on ADVANCED > Export Logs is a standard UNIX/Linux tool for exporting logs to remote syslog servers. Enter the name and IP addresses of up to 3 syslog servers to which you will export System Events, Web Firewall logs, Access logs, and Audit logs. If you are running syslog on a UNIX machine, be sure to start the syslog daemon process with the “-r” option so that it can receive messages from sources other than itself. Windows users have to install a separate program to utilize the syslog since the Windows OS does not include the syslog capability. Kiwi Syslog is a popular solution, but there are many others to choose from, both free and commercial.

The syslog messages are sent over UDP to the standard syslog port of 514. If there are any firewalls between the Barracuda Web Application Firewall and the servers receiving the syslog messages, then be sure that port 514 is open on the firewalls.

To configure System Logs, use the ADVANCED > Export Logs page, Syslog section. You need to enter the name and IP address of the syslog server, and choose whether to time stamp log entries or log the unit name of the Barracuda Web Application Firewall which generated the log entry. For more detailed instructions for configuring system logs, see the online help.

To monitor the system logs, use ADVANCED > Export Logs, the Syslog section, and click Monitor Syslog.

**Syslog Facility**

The syslog receives different types of log messages from various hosts. Each log message contains an actual message and IP address, as well as a logging priority and logging facility. To differentiate and store logged messages coming to the same syslog server in unique log files according to their log type, use the logging facility.
All log messages can be marked with one of the following facilities: local0, local1, local2, local3, local4, local5, local6, or local7. Setting a different facility (default = local0) for each log type allows the syslog server to segregate the logs into different files. Otherwise, all log messages will be in one file.

To configure facilities for different log types use ADVANCED > Export Logs Syslog and click Syslog Settings. Here you can select the appropriate facility (Local0 to Local7) from the drop-down list for each log type.

Note
You can set the same facility for all log types. Then System Logs, Web Firewall Logs, Access Logs, and Audit Logs would be in the same file.

To configure log levels for different modules, use ADVANCED > Export Logs Module Log Levels. Specify a Name, and select a Module and Log Level. For more detailed instructions, see the online help.

**Custom Logs Formats**

The format of the Web Firewall Logs, Access Logs, and Audit Logs to be sent to the syslog sever can be customized. You can choose between the Common Log Format, NCSA Extended Format, W3C Extended Format, Default, or Custom Format. The Common Log Format, NCSA Extended Format, W3C Extended Format, and Default formats are already defined and cannot be edited. Given below are the steps to specify the Custom Format.

To customize the log format for any Log Type (except System Logs) use ADVANCED > Export Logs. On the Logs Format section, select Custom Format for any of the log types. Online help explains the ways Custom Format can be defined.

For information on how to manage these logs please see the documentation available for your syslog server.

**FTP Access Logs**

FTP Access Logs allow configuration of the FTP server that will host the Access logs and the format of log events being transported.

To configure FTP Access Logs, specify the IP address, port and login credentials of the FTP server. Indicate the destination directory for the logs, and the logged data format. For detailed configuration instructions, see online help.

*Table 15.4: Table of Logs*

<table>
<thead>
<tr>
<th>System Logs</th>
<th>Web Firewall Logs</th>
<th>Access Logs</th>
<th>Audit Logs</th>
</tr>
</thead>
<tbody>
<tr>
<td>%t - Time Stamp</td>
<td>%t - Time Stamp</td>
<td>%t - Time Stamp</td>
<td>%t - Time Stamp</td>
</tr>
<tr>
<td>%md - Module Name</td>
<td>%un - Unit Name</td>
<td>%un - Unit Name</td>
<td>%un - Unit Name</td>
</tr>
<tr>
<td>%ll - Log Level</td>
<td>%lt - Log Type</td>
<td>%lt - Log Type</td>
<td>%lt - Log Type</td>
</tr>
<tr>
<td>%ei - Event ID</td>
<td>%sl - Severity Level</td>
<td>%ai - Application IP</td>
<td>%an - Admin Name</td>
</tr>
<tr>
<td>%ms - Message</td>
<td>%ad - Attack Description</td>
<td>%ap - Application Port</td>
<td>%ct - Client Type</td>
</tr>
<tr>
<td>System Logs</td>
<td>Web Firewall Logs</td>
<td>Access Logs</td>
<td>Audit Logs</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------</td>
<td>-------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>%ci - Client IP</td>
<td>%ci - Client IP</td>
<td>%li - Login IP</td>
<td></td>
</tr>
<tr>
<td>%cp - Client Port</td>
<td>%cp - Client Port</td>
<td>%lp - Login Port</td>
<td></td>
</tr>
<tr>
<td>%ai - Application IP</td>
<td>%id - Login ID</td>
<td>%trt - Transaction Type</td>
<td></td>
</tr>
<tr>
<td>%ap - Application Port</td>
<td>%cu - Certificate User</td>
<td>%tri - Transaction ID</td>
<td></td>
</tr>
<tr>
<td>%ri - Rule ID</td>
<td>%m - Method</td>
<td>%cn - Command Name</td>
<td></td>
</tr>
<tr>
<td>%rt - Rule Type</td>
<td>%p - Protocol</td>
<td>%cht - Change Type</td>
<td></td>
</tr>
<tr>
<td>%at - Action Taken</td>
<td>%h - Host</td>
<td>%ot - Object Type</td>
<td></td>
</tr>
<tr>
<td>%fa - Follow-up Action</td>
<td>%v - Version</td>
<td></td>
<td></td>
</tr>
<tr>
<td>%adl - Attack Details</td>
<td>%s - HTTP Status</td>
<td></td>
<td></td>
</tr>
<tr>
<td>%m - Method</td>
<td>%bs - Bytes Sent</td>
<td>%ov - Old Value</td>
<td></td>
</tr>
<tr>
<td>%u - URL</td>
<td>%br - Bytes Received</td>
<td>%nv - New Value</td>
<td></td>
</tr>
<tr>
<td>%p - Protocol</td>
<td>%ch - Cache Hit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>%sid - Session ID</td>
<td>%tt - Time Taken</td>
<td></td>
<td></td>
</tr>
<tr>
<td>%ua - User Agent</td>
<td>%si - Server IP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>%px - Proxy IP</td>
<td>%sp - Server Port</td>
<td></td>
<td></td>
</tr>
<tr>
<td>%pp - Proxy Port</td>
<td>%st - Server Time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>%au - Authenticated User</td>
<td>%sid - Session ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>%r - Referrer</td>
<td>%rtf - Response Type Field</td>
<td></td>
<td></td>
</tr>
<tr>
<td>%aid - Attack ID</td>
<td>%pmf - Profile Matched Field</td>
<td></td>
<td></td>
</tr>
<tr>
<td>%ag - Attack Group</td>
<td>%pf - Protected Field</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>%wmf - WF Matched Field</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>%u - URL</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>%q - Query</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>%r - Referrer</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>%c - Cookie</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>%ua - User Agent</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>%px - Proxy IP</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>%pp - Proxy Port</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>%au - Authenticated User</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>%cs1 - Custom Header 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>%cs2 - Custom Header 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>%cs3 - Custom Header 3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Reports

You can configure and generate reports of various types, based on all logged information, which help manage day-to-day operation. Barracuda Web Application Firewall reports are broadly classified into four functional groups, each containing a predefined set of report types. Select a Report Group, then a corresponding Report Type from the drop-down list. The four report groups are:

Security and Traffic Reports
Security and Traffic reports contain the Web attack prevention activity performed by the Barracuda Web Application Firewall. Note: Some 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
Audit reports contain server details and the login/logout activities performed by different user roles.

Config Summary Reports
Config Summary reports contain:
- Performance of the Barracuda Web Application Firewall features such as Load Balancing, Rate Control, Learning, etc.
- Details of the digital certificates like issuing date, expiry date, and associated services.
- Details of accounts, their users, privileges assigned to them, permitted operations, etc.

PCI Reports
PCI reports detail compliance with PCI (Payment Card Industry) standards and display:
- Combined details of the PCI attacks such as top attacking Clients, and top attacked Services, Domains, and URLs.
- Details of the PCI directives and the Barracuda Web Application Firewall compliance with those directives.

Generating Reports
Use BASIC > Reports to choose different reports that can help you keep track of the activity of the Barracuda Web Application Firewall. Generate a report on demand or configure the Barracuda Web Application Firewall to automatically generate reports on a daily, weekly, or monthly basis, emailing the reports to configured email addresses. For information on aggregating reports across multiple Barracuda Web Application Firewall appliances, see Managing Multiple Systems With Barracuda Control Center on page 143. Reports can be based on user-activity, content, or actions. For detailed descriptions, see online help on the BASIC > Reports page.

Note
Only one-time reports can be displayed when report processing is done. Scheduled reports are emailed when done.
This chapter describes the advanced security configurations you can deploy using the Barracuda Web Application Firewall. The following topics are covered:

*Advanced Deployment* ................................................................. 152
Advanced Deployment

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

- Service Virtual Interfaces ................................................................. 152
- Custom Virtual Interfaces ................................................................. 152
- Static Routes .................................................................................... 152
- Interface Routes ............................................................................... 153
- VLAN (Virtual Local Area Network) ................................................. 153

Service Virtual Interfaces

The Service Virtual Interfaces 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 the traffic to flow 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.

Custom Virtual Interfaces

The Custom Virtual Interfaces section allows you to add new virtual interface(s) to the physical port (WAN or LAN or MGMT) used to communicate with the servers.

To configure Custom Virtual Interfaces

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 port over which communication will be transmitted. To do this, select either WAN or LAN. (Back-end traffic is normally over LAN.)

2. Click Add to add the new virtual interface.

Static Routes

The Static Routes section allows you to create a static route by specifying 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 are translated into CSV (Comma-Separated Values) format and displayed in the Bulk Edit window. Then manually add, modify, or remove as many rows as you desire and **Save Changes**.

Interface Routes

The Interface Routes section allows you to create an interface route by specifying 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. Also, the ports on a switch with VLAN capabilities can 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. Alternatively, **VLAN ports** are those ports on the VLAN switch which belong to a single VLAN and therefore only see the broadcast traffic of that VLAN.
VLAN Configuration

To be able to route to a VLAN through any one of the interfaces (WAN, LAN, or MGMT) a VLAN interface must be added to it. This interface receives the broadcast traffic from the VLAN. It is also used to route 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 have 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 **VLAN Configuration**.

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.

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 VLANs with which the interface communicates. 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.
This chapter gives 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 it. The following topics are covered in this chapter:

- **Network Firewall Overview** ............................................................. 156
- **Configuring Source Network Address Translations (SNATs)** ............. 158
- **Configuring Access Control Lists (ACLs)** ........................................ 160
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 implement access control so unauthorized network-level requests are blocked at the 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 rules map 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 address firewall access rules for the Barracuda Web Application Firewall. A network firewall rule specifies matching criteria for packets and corresponding actions for matching packets. 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 prevents all traffic originating from LAN from going out on the WAN. Only traffic specifically configured to go out on the WAN from the LAN is allowed. For example, consider a Windows Server with internal IP address 192.168.1.5 on the LAN that needs Microsoft updates from a Microsoft Server (69.23.45.2). The Windows Server will not be able to send a request until explicitly configured to do so, since the Barracuda Web Application Firewall denies all requests by default. To accomplish the update, a combination of SNAT and ACLs must be configured. SNAT and ACLs are interdependent; SNAT is configured to map the internal IP address to an external IP address, while ACLs distinguish whether the request will be sent out or be denied.

Use the ADVANCED > Network Firewall page to configure Source Network Address Translation (SNAT) rule and Network Access Control Lists (ACLs) rule.

Figure 17.1: Example for Reverse Proxy mode

Note
The Barracuda Web Application Firewall’s network firewall employs a positive security policy when active; that is, by default it denies all requests unless 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.
- Internal routers can allow downloading files from outside servers.

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 prevents outside exposure of the internal server address.
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 an FTP session to a specified external FTP server.

In Bridge mode, the Barracuda Web Application Firewall only inspects traffic for the configured VIP and bridges all other traffic. In this mode, traffic is bridged through the firewall and thus does not reach the NATs and ACLs in 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 servers on the LAN with a single check box option. Selecting this option enables all traffic originating from LAN to go out on the WAN, 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 as in Figure 17.2. 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.
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 of undoing translation for returning traffic i.e. re-writing the IP address of the computer that originated the packet.

For example, consider an internal IP address 10.1.2.27 sends a packet to an external Web server. The Barracuda Web Application Firewall translates the internal IP address 10.1.2.27 to an external IP address 209.165.201.10. When the external Web server responds, the external IP address 209.165.201.10 receives the packet and sends it to the internal IP address 10.1.2.27. See Figure 17.3.
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** page.
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 applied to a packet. ACLs are the IP address firewall access rules. The list specifies access rules for each packet. If a packet matches the specified rule, the configured action is performed. The action can be either **ALLOW** which accepts the packet and allows access or **DENY** which drops the packet and access is denied.

- ACLs are capable of constraining the flow of traffic to an 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** page.
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 address 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 DENY).
   - **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.
This chapter describes the configuration, function, and administration of a High Availability environment using the Barracuda Web Application Firewall. It includes the following topic:

*Creating a High Availability (HA) Environment* ........................................... 162
Creating a High Availability (HA) Environment

The ADVANCED > High Availability page allows you to link a second Barracuda Web Application Firewall to your network to act as a backup to the primary, as in Figure 18.1. Both systems must be 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, providing continuous network availability. The Barracuda Web Application Firewall uses ports 8001 and 8002 to synchronize configuration between linked systems.

Figure 18.1: High Availability Pair of Barracuda Web Application Firewall

Each linked Barracuda Web Application Firewall sends a custom “heartbeat” to the other using UDP, providing continual status updates. The backup unit automatically becomes active, taking over the services of the primary system, if the primary system fails to send a heartbeat for nine (9) seconds, or sends a status indicating its state as “Failed”.

Each Barracuda Web Application Firewall to be added to a cluster must meet the following requirements:

• Have a unique WAN IP address. The Barracuda Web Application Firewalls use the WAN IP address (UDP port) to communicate for HA.
• Have connectivity to (can ping successfully) the other appliance on the WAN interface.
• Be co-located (WAN Interface) 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. Do the same for Barracuda Web Application Firewall 2. No processes should be running on either appliance 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 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. Save Changes.
   4b. In the Clustered Systems section, enter the WAN IP address of Barracuda Web Application Firewall 1, and 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.
6. The **High Availability Status** can be viewed from **BASIC > Status Performance Statistics** page. This shows the role and state of that unit.

*Figure 18.2 and Figure 18.3 show how the performance statistics appear before and after the second Barracuda Web Application Firewall has been clustered with the primary unit.*

**Figure 18.2: An unclustered Barracuda Web Application Firewall**

Image of unclustered firewall showing one entry with status green.

**Figure 18.3: Two clustered Barracuda Web Application Firewalls**

Image of clustered firewalls with two entries showing status green.

**To ensure proper routing from the back-end servers in case of failover**

- Add a virtual IP address on the LAN interface from the **ADVANCED > Advanced IP Config** page.
- Use this virtual IP address as the routing address for WAN traffic on the real server routing tables (or to the intermediate router's routing tables if the server is in a different subnet).

**Note**

Do not use the LAN IP address for routing in a HA setup as it is not synchronized or failed over in a cluster.

**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 High Availability Status** on the **BASIC > Status** page. When a single unit is deployed, **High Availability Status** is displayed as **Stand-alone**.
The following table describes the possible system states.

Table 18.1: System Status

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>• 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).</td>
</tr>
<tr>
<td>Standby</td>
<td>• 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).</td>
</tr>
<tr>
<td>Failed</td>
<td>• 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.</td>
</tr>
</tbody>
</table>

**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 be in Standby 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, when the link is down for any one of these interfaces, the system goes into a Failed state.

2. **Data Path Process Crash** - If the system had an outage, particularly if it has been crashing frequently, the system will be placed in the Failed state. Frequently is determined 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 has not received a heartbeat from 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 if the back-up unit is also down.

**Failover in Automatic Mode**

When the primary unit that is active and handling traffic fails, the primary unit automatically fails over to the backup unit. The backup unit assumes the services from the failed (primary) unit and continues to process the traffic until the failed unit is restored.

**Failover in Manual Mode**

Failover in the manual mode occurs in two ways. They are:
Two ways of Failover in Manual Mode:
1. When the primary unit fails.
2. Clicking the Failover button on the primary unit.

The first case is similar to the automatic failover. When the primary unit that is active and handling traffic fails, the primary unit automatically fails over to the backup unit. The backup unit assumes the services of the failed (primary) unit and continues to process the traffic.

In the second case, you can force a failover on the primary unit by clicking the Failover button in the Clustered Systems section. The primary unit goes to the Standby state and the backup unit assumes the services from the primary unit and continues to process the traffic.

Failback

Failback restores functioning services that have failed over from the primary unit to the backup unit. When the primary unit returns from the Failed state to the Active state (that is, it can now actively serve application requests), the services are automatically failed back (released) from the backup unit. The backup unit now goes into Standby state.

Failback in Automatic Mode

When the primary unit restores from the Failed state and is ready to resume operation, the backup unit automatically fails back to the primary unit. The primary unit resumes operation and the backup unit is restored to Standby state.

Failback in Manual Mode

Failback in the manual mode occurs in two ways. They are:

Two ways to Failback in Manual Mode:
1. Clicking the Failback button on the backup unit.
2. If the backup unit fails and the primary unit is in Standby state

In Manual mode, the backup unit remains active even if the primary unit restores from the Failed state and is ready to resume operation. You need to click on the Failback button in the Clustered Systems section of the backup unit to resume operation on the primary unit.

In the second case, the primary unit automatically resumes operation if the backup unit fails. This can occur only when the primary unit is not in Failed state and is ready to resume operation.

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 18.2 identifies the data that is propagated when two systems are linked.

Table 18.2: Data Propagated Between Linked Systems

<table>
<thead>
<tr>
<th>Propagated Data</th>
<th>Data Not Propagated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any configuration changes through the Administration interface.</td>
<td>• System IP configuration (IP address, netmask, gateway, and DNS server) and Monitor Link information configured on the BASIC &gt; IP Configuration page.</td>
</tr>
<tr>
<td></td>
<td>• System password and time zone as configured on the BASIC &gt; Administration page.</td>
</tr>
</tbody>
</table>

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 address conflicts and services to go down.

Do the following to remove the units:

1. Take backups of both the units. Refer to Backing up and Restoring your System Configuration on page 173.
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, remove the unit from the cluster. 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 units:

* Replace a failed primary unit one of two ways:
  * backup unit becomes the new primary
  * a new unit assumes the role of the primary
* Replace a failed backup unit

To remove the cluster when the primary is dead:

1. To replace the primary unit with backup unit.
   1a. Take a backup of the backup unit.
   1b. Ensure that the backup unit 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 a failed primary unit with a new unit as primary unit, the pre-requisites are:
   * A backup of the failed primary unit was made before it went down
   * A down-time period is planned
   2a. Take a backup of 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 new unit is the primary unit and 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 primary unit, and click Join Cluster.
This chapter provides general instructions for administering and maintaining the Barracuda Web Application Firewall. This chapter covers the following topics:

Administrative Settings ................................................................. 170
Maintaining the Barracuda Web Application Firewall ............... 173
Administrative Settings

This section covers the basic administrative settings for your Barracuda Web Application Firewall.

Controlling Access to the Administration Interface............................170
Customizing the Appearance of the Web Interface.......................... 170
Setting the Time Zone of the System................................................ 170
Configuring NTP Server for Time Synchronization......................... 170
Enabling SSL for Administration..................................................... 171

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). You can specify the NTP server(s) to use on the ADVANCED > System Configuration page.

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.

Configuring NTP Server for Time Synchronization

The ADVANCED > System Configuration NTP Server Settings section allows you to define one or more NTP server(s) to act as a master clock for the Barracuda Web Application Firewall. When multiple servers are defined, NTP uses the server whose time is most accurate based on various factors like the time variation and distance to the server.

To configure a NTP server
1. Go to the ADVANCED > System Configuration NTP Server Settings section.
2. Specify values for the following:
Administering and Maintaining your System 171

- **Name**: Defines a name to identify the NTP server that will be used to synchronize time.
- **Server IP**: Specifies the IP address or host name of the NTP server.

3. Click **Add** to add the NTP server settings.

**Note**
Please ensure that the DNS is configured properly, otherwise, ntpdate takes a long time to report the error, slowing down the boot process.

A **REBOOT** of the system is recommended to help the Barracuda Web Application Firewall synchronize the time and also to ensure all services run smoothly with the synchronized time.

### 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 allow only secured connections, set up SSL.

**Note**
The SSL configuration referred to here is related only to 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>
<thead>
<tr>
<th><strong>Table 19.1: SSL Fields</strong></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Web Interface HTTPS/SSL Configuration</strong></td>
<td>Select <strong>Yes</strong> to enable SSL and only allow access to the Web administration interface via SSL. Select <strong>No</strong> to use standard HTTP access.</td>
</tr>
<tr>
<td><strong>HTTPS/SSL access only</strong></td>
<td>The SSL port used by the Barracuda Web Application Firewall. Default port for SSL is 443.</td>
</tr>
</tbody>
</table>
### SSL Certificate Configuration

<table>
<thead>
<tr>
<th>Certificate Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default (Barracuda Networks)</td>
<td>Certificates are free but generate browser alerts. The default certificate is signed by Barracuda Networks and provided free as the default type of certificate.</td>
</tr>
<tr>
<td>Private (self-signed)</td>
<td>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.</td>
</tr>
<tr>
<td>Trusted</td>
<td>Certificates are issued by trusted Certificate Authorities (CA), which are usually recognized by your Web browser so no additional configuration is required.</td>
</tr>
</tbody>
</table>

### Certificate Generation

<table>
<thead>
<tr>
<th>Organization Info</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Name</td>
<td>is the fully qualified domain name used to access the administration interface. For example: “barracuda.yourdomain.com”</td>
</tr>
<tr>
<td>Country</td>
<td>is the two-letter country code where your organization is located.</td>
</tr>
<tr>
<td>State or Province Name</td>
<td>is the full name of the state or province where your organization is located.</td>
</tr>
<tr>
<td>Locality Name</td>
<td>is the city where your organization is located.</td>
</tr>
<tr>
<td>Organization Name</td>
<td>is the legal name of your company or organization.</td>
</tr>
<tr>
<td>Organization Unit Name</td>
<td>is an optional field in which to specify a department or section within your organization.</td>
</tr>
</tbody>
</table>

### Trusted Certificate

<table>
<thead>
<tr>
<th>Upload Signed Certificate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>After purchasing a certificate, browse to the location of the certificate and click <strong>Upload Now</strong>. Once you upload the certificate, your Barracuda Web Application Firewall automatically begins using it.</td>
<td></td>
</tr>
<tr>
<td>Once you have uploaded your signed certificate, make sure <strong>Trusted</strong> is selected for the Certificate Type (described above).</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Upload Backup SSL Private key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>After downloading the corresponding private key, browse to the location of the key and click <strong>Upload Now</strong>.</td>
<td></td>
</tr>
</tbody>
</table>
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:

- Backing up and Restoring your System Configuration ................................................................. 173
- Updating the Barracuda Web Application Firewall Firmware ............................................................... 173
- Updating the Attack, Virus, and Security Definitions.................................................................................. 174
- Replacing a Failed System ........................................................................................................................... 174
- Reloading, Restarting, and Shutting Down the System .................................................................................. 175
- Using the Built-in Troubleshooting Tools ..................................................................................................... 175
- Using Session Recording .............................................................................................................................. 176
- Using the Task Manager ............................................................................................................................... 176
- Setting the System Configuration................................................................................................................ 177
- Rebooting the System in Recovery Mode ...................................................................................................... 177

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 address information
  - DNS information

Updating the Barracuda Web Application Firewall Firmware

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 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 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 definitions, as well as change the interval at which the Barracuda Web Application Firewall checks for updates. Barracuda 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 19.2: Definition Updates

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Installed Version</td>
<td>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.</td>
</tr>
<tr>
<td>Latest General Release</td>
<td>Displays the latest version that is available. If the current version running on the Barracuda Web Application Firewall is not the latest, click Update to download the latest version. The Update button is disabled if the system already has the latest version.</td>
</tr>
<tr>
<td>Previously Installed Version</td>
<td>Displays the previously installed version that was running on the system. To go back to this version of the definitions, click Revert.</td>
</tr>
<tr>
<td>Automatically Update</td>
<td>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.</td>
</tr>
<tr>
<td>Subscription Status</td>
<td>Informs you if your Barracuda Energize Updates are current and when your subscription expires.</td>
</tr>
</tbody>
</table>

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 address information on the BASIC > IP Configuration page. For information on restoring data, refer to Backing up and Restoring your System Configuration on page 173.

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 impact 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.

Click Clear Configuration to clear the entire configuration and restore the Barracuda Web Application Firewall to its initial configuration.
Using Session Recording

The Session Recording section on the ADVANCED > Troubleshooting page enables you to capture the requests and responses in the Barracuda Web Application Firewall from a specified Client IP address or User ID. The captured session is stored in an XML file.

Note

When a new session is captured, the previous session is overwritten by the new session.

To enable session recording

1. Enter the following information:
   - **Service IP** - Click the drop-down list and select the service for which you want to enable session recording.
   - **Client Filter** - Specify the client IP address or user ID for which you want to enable the requests to be captured. The session recording captures the requests coming from the specified client ID or user ID. Note: The AAA should be enabled on the Barracuda Web Application Firewall when User ID is selected as the Client Filter.
   - **Response Content Types** - Specify the content types that have to be captured in the requests and/or responses during a session. The specified content types are matched with the responses. If the content types match the response content, session recording captures both the requests and responses of that session. If not, session recording ignores the requests and responses of that session.
   - **Capture Request** - Specifies whether to capture the request or not. Select Yes to capture the requests.
   - **Max Request Body Limit** - Specify the maximum request size limit to be captured during a session.
   - **Capture Response** - Specifies whether to capture the response or not. Select Yes to capture the responses.
   - **Max Response Body Limit** - Specify the maximum response size limit to be captured during a session.
   - **Max Number of Requests** - Specify the maximum number of requests to be captured during a session.

2. Perform the following tasks one after the other in the given order:
   - Click Start Capture to start the session recording.
   - Click Stop Capture to stop the session recording.
   - Click Download to download the captured session file (XML file).
   - Click Remove Capture File to discard the existing captured session file.

Using the Task Manager

The ADVANCED > Task Manager page provides the list of all currently processing tasks, and also displays any errors encountered when performing these tasks.
Setting the System Configuration

The **ADVANCED > System Configuration** page allows sharing of a common encryption key to encrypt and decrypt user session data. Cookies or hidden parameters are used to store the encrypted state information on a client's navigation platforms such as browsers or 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 178.

   **Note**

   To stop a hardware test, reboot your Barracuda Web Application Firewall by pressing Ctrl-Alt-Del.
Reboot Options

Table 19.3 describes the options available at the reboot menu.

Table 19.3: Reboot Options

<table>
<thead>
<tr>
<th>Reboot Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barracuda</td>
<td>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.</td>
</tr>
<tr>
<td>Recovery</td>
<td>Displays the Recovery Console where you can select the following options:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Barracuda Recovery</strong>—Repairs the file system on the Barracuda Web Application Firewall.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Full Barracuda Recovery</strong>—Restores the factory settings on your Barracuda Web Application Firewall and clears out all configuration information.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Enable remote administration</strong>—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 Establish Connection to Barracuda Central on the <strong>ADVANCED &gt;Troubleshooting</strong> page.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Diagnostic memory test</strong>—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.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Exit</strong>—Exits from the Recovery Console.</td>
</tr>
<tr>
<td>Hardware_Test</td>
<td>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. Reboot your Barracuda Web Application Firewall to stop the hardware test.</td>
</tr>
</tbody>
</table>
Chapter 20
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 .......................................................................................... 180
Roles ................................................................................................ 180
Users ................................................................................................ 181
Privileges ......................................................................................... 182
Creating A New Role ........................................................................ 183
Creating An External Authentication Service .................................. 183
Creating A New Local Administrator Account .............................. 185
The Barracuda Web Application Firewall provides role based administration (RBA). RBA restricts 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 with distinct operational and configuration privileges specified in Table 20.1. The Barracuda Web Application Firewall also allows creation of custom roles with defined access privileges. Roles can be assigned to users who perform specific job functions. The ‘admin’ role, by default, is assigned to the ‘admin’ user who has permission for role management.

The RBA feature in the Barracuda Web Application Firewall uses the following components:

- Roles
- Users
- Privileges

**Roles**

A role is a set of privileges or permissions on the available system resources, created for a specific job function. The ‘admin’ role creates, modifies, and deletes roles. A role can be assigned to multiple users within an organization. Assigning a role to a user allows that user system resource privileges included in the role definition. All users who assume that role can operate in the same environment and access the same resources. For example, an administrator assigned to the 'audit-admin' role is only allowed to view logs on the system and is not permitted to access any other object.

**Predefined Roles**

The Barracuda Web Application Firewall provides a set of predefined (Factory Shipped) roles. Each predefined role has associated access privileges for system resources. These roles cannot be modified or deleted. The following table briefly describes the predefined roles:

*Table 20.1: Predefined Roles*

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>The super administrator role. The 'admin' user is allowed to perform all system operations. An admin creates and assigns roles.</td>
</tr>
<tr>
<td>audit-manager</td>
<td>This role defines auditing functions including:</td>
</tr>
<tr>
<td></td>
<td>- View logs</td>
</tr>
<tr>
<td>certificate-manager</td>
<td>This role defines certificate management capabilities including:</td>
</tr>
<tr>
<td></td>
<td>- Uploading certificates</td>
</tr>
<tr>
<td></td>
<td>- Creating certificates</td>
</tr>
<tr>
<td></td>
<td>- Uploading Trusted certificates</td>
</tr>
</tbody>
</table>
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 them to users. For more information on how to create a new role, refer to Creating A New Role below.

Users

A user is a person using the Barracuda Web Application Firewall. The set of operations that can be performed by a user is defined by the role associated with him. 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 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 ADVANCED > Admin Access Control page. If you delete a local administrator account, that user is denied access to the system. For more information on how to add a new user, refer to Creating A New Local Administrator Account below.

External Users

External administrators or users are part of an external authentication service like the Lightweight Directory Access Protocol (LDAP) or Remote Authentication Dial In User Service (RADIUS). The Barracuda Web Application Firewall enables you to configure an external authentication service, allowing authenticated external users to access the system. An external user cannot be created but is sync’ed internally from the LDAP or RADIUS 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 or RADIUS database anymore, then the user needs to be manually deleted from the Barracuda Web Application Firewall. External authentication will fail when a user is not part of LDAP or RADIUS anymore. For more information on how to create an external authentication service, refer to Creating An External Authentication Service below.

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 20.2: Object Privileges

<table>
<thead>
<tr>
<th>Object</th>
<th>Description</th>
<th>Permission</th>
</tr>
</thead>
<tbody>
<tr>
<td>Services</td>
<td>Exhibits all services that are configured on the Barracuda Web Application Firewall.</td>
<td><strong>Read</strong>: Enables the user to view the configuration of an object, but exempts from modifying the object.</td>
</tr>
<tr>
<td>Security Policies</td>
<td>Exhibits all default and customized security policies.</td>
<td><strong>Write</strong>: Enables the user to view and modify the configuration of an object, but exempts from deleting the object.</td>
</tr>
<tr>
<td>Authentication Services</td>
<td>Exhibits all authentication services such as Lightweight Directory Access Protocol (LDAP) and Remote Authentication Dial In User Service (RADIUS).</td>
<td><strong>Read All</strong>: Enables the user to view and modify the configuration of all objects, but exempts from modifying the objects. <strong>Write All</strong>: Enables the user to view and modify the configuration of all objects, but exempts from deleting the objects.</td>
</tr>
</tbody>
</table>
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, backing up the system configuration, etc. You can grant permission to perform these operations to a role. A role can only execute operations for which it has permission, 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 by a user by selecting the Primary and Secondary Tabs.

Creating A New Role

The procedure below describes how to create a new administrator role, later delegated to your user account.

To create new role:

1. Go to the 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 An External Authentication Service

The Barracuda Web Application Firewall authenticates external administrators using the configured LDAP or RADIUS servers. The procedure below describes how to add an LDAP or RADIUS authentication service:

To add an LDAP authentication service

1. Go to the ADVANCED > Admin Access Control page.
2. In External Authentication Service section, select LDAP from the drop-down list. The Add LDAP Service window appears.
3. Specify 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.

- **UID Attribute** - Specify the attributes of an LDAP object used for identifying the name of a group. For example: cn, sAMAccountName.

- **Group Filter** - Specify the LDAP filter used to retrieve the list of groups of a user.

- **GroupName Attribute** - Specify the attributes of an LDAP object used for identifying the name of a group.

- **Default Role** - Select a role from the drop-down list. This default role is assigned to all external users who are authenticated on this LDAP database.

4. Click **Add** to add the LDAP service.

**To add a RADIUS authentication service**

1. Go to the **ADVANCED > Admin Access Control** page.

2. In the **External Authentication Service** section, select **RADIUS** from the drop-down list. The **Add RADIUS Service** window appears.

3. Specify values for the following fields:

- **Realm Name** - Specifies 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** - Specifies the IP address of an external RADIUS server used for authenticating users.

- **Port** - Specifies the port number of the external RADIUS server used for authenticating users. Port 1812 is normally used for RADIUS.

- **Shared Secret** - Specifies the secret key which is shared between the Barracuda Web Application Firewall and RADIUS server.

- **Timeout** - Specifies the time in seconds the Barracuda Web Application Firewall waits for a response from the RADIUS server before retransmitting the packet.

- **Default Role** - Select a role from the drop-down list. This default role is assigned to all external users who are authenticated on this RADIUS database

4. Click **Add** to add the RADIUS service.
Creating A 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 the 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 21

Templates

This chapter provides instructions for creating Templates. It describes the use of saved templates for further custom configuration. The following topics are covered in this chapter:

- **Overview** .......................................................................................... 188
- **Creating a Template** ........................................................................ 188
- **Importing a Template** ...................................................................... 188
Overview

A template is a collection of configuration fragments arranged serially in a file. Templates are used to create/import backups of object types like Services, URL profiles, URL Policies, etc., so configuration can be exported to other Barracuda Web Application Firewall boxes in the following scenarios:

- Migrate changes from the Barracuda Web Application Firewall in front of QA servers to the Barracuda Web Application Firewall in front of production servers.
- Import the templates given out by Barracuda Web Application Firewall experts for refined policies on standard applications.
- Patch existing policies. For example, a new OWA template might need an additional “Allow Method” for a Global ACL. Or a new pattern, like sql-tautology-conditions, might require a refinement which needs patching on an existing pattern-group. An existing service might require a new keep-alive timeout, already tested and found optimal in the QA network.
- Take a backup of an application configuration.

Creating a Template

You can create a template from the existing configuration which is stored on your file system and can be distributed to others.

Use ADVANCED > Templates and select Generate Template as the Template Operation. Select a suitable Template Type and specify the Name and Description for the template. Use Exportable Objects to select the parent nodes and child nodes using check boxes. Generate to see your template displayed under Available Templates.

Importing a Template

A saved template can be imported on the configuration tree using Add or Modify. In both cases matching key parameters (see Table 21.1) are checked for each object type as follows:

- For an Add operation, the existing key parameters should not match because if the configuration already exists it is considered an error. As long as the key parameters don’t match, it adds the values of the saved template to the selected parent nodes or child nodes of the configuration tree.
- For a Modify operation, the existing configuration key parameters should match, and are blindly replaced with the new values from the saved template configuration. If they do not match, it is considered an error.
- When a service template is imported, you can specify an IP address and port on which to create a service from the template during an Add operation. Similarly, for a Modify operation, the template would modify a service existing on the box with that IP address and port. Doing this makes sense only if the source template is generated from a single service. This offers the flexibility to incrementally patch a service with template values.

Table 21.1: Object types and its Key Parameters

<table>
<thead>
<tr>
<th>Object Type</th>
<th>Key Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service</td>
<td>IP, Port</td>
</tr>
<tr>
<td>Server</td>
<td>IP, Port</td>
</tr>
</tbody>
</table>
Add

The Add operation adds a new configuration to the selected parent nodes or child nodes by using the values from the saved template. For example, suppose an object of type Server with values Server IP=192.168.128.10 and Server Port=80 already exists. Attempting to add a template with the same values is an error because the configuration already exists.

To add a new template use ADVANCED > Templates and select Import Template as the Template Operation. Select a suitable Template Type and specify the Add Operation. Use Specify to select parent nodes and child nodes you want to add to and click Add. Remove deletes a selection. Browse to locate the Template file path and Import the template file to the selected destination box.

Modify

The Modify operation modifies the existing configuration of selected parent nodes or child nodes by using the values from the saved template. For example, suppose an object of type Server with values Server IP=192.168.128.10 and Server Port=80 already exists. When you attempt to modify a template with the values Server IP=192.168.128.10 and Server Port=223, it simply replaces the Server Port value with 223 since the Server IP address value already matches.

To modify an existing template, use ADVANCED > Templates to select Import Template as the Template Operation. Select a suitable Template Type, then specify the Modify Operation. Specify the parent nodes and child nodes to which you want templates imported by clicking the Add button. Use the Remove button to delete a selection. Click Browse to select the Template file path from which to import the template file. Clicking Import patches the existing template.

Points to Remember

1. When importing an SSL based service, note that the service is imported with SSL Status set to On for the front-end and set to Off for the back-end. You need to create relevant certificates, bind them, and set SSL Status to On to complete the service creation.
2. A Modify operation blindly replaces any value of the object's parameters with the value found in the template. However, for the parameters which have multi-valued inputs (for example, Allowed Methods in SECURITY POLICIES > URL Protection), the modify operation results in a union of the existing values and the template values.

3. Template generation does not recursively copy the objects. If you have a policy bound to a service, make sure the policy exists on the destination box before importing the service on the destination box. The most common cases of objects like these within a service are: Policy, Response Pages, Certificates, Parameter Classes, Rate Control pool, Trusted Hosts.
This chapter describes how the Barracuda Web Application Firewall integrates with Web Application Vulnerability Scanners to mitigate uncovered vulnerabilities until Web site fixes are released:

- Overview ................................................................. 192
- Steps to Mitigate Web site Vulnerabilities ...................... 193
Overview

The Barracuda Web Application Firewall integrates with Web Application Vulnerability Scanners (IBM AppScan Version 7.9 is the only currently supported type) to address Web application vulnerabilities detected by the scanning tools. The vulnerable part of the Web application can be patched quickly and easily using the Barracuda Web Application Firewall, so the optimal engineering solution can be designed and incorporated through the regular code release cycle without incurring continued risk.

Administrators use vulnerability scanners which detect and report Web site vulnerabilities in a variety of report formats. Vulnerability reports can be imported using the ADVANCED > Threat Control Manager > Import Vulnerability Report section. The Barracuda Web Application Firewall uses imported reports to provide Recommendation(s) for Vulnerability Assessment, which, if applied by the administrator, modify applicable security policy settings or configuration to mitigate the reported vulnerabilities.

Managing Vulnerability Assessment Reports

Summaries of imported vulnerability assessment reports are visible, along with corresponding configuration update recommendations, using Manage Vulnerability Assessments. Select the

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import Date</td>
<td>Date and time of import of vulnerability assessment report.</td>
</tr>
<tr>
<td>Scanner Used</td>
<td>Scanner which detected vulnerabilities.</td>
</tr>
<tr>
<td>Applied Recommendations</td>
<td>Number of recommendations applied to service to mitigate threats.</td>
</tr>
<tr>
<td>Pending Recommendations</td>
<td>Number of recommendations pending, not yet applied to service.</td>
</tr>
</tbody>
</table>

assessment report from the Assessment Name drop-down list to see the summary. A detailed summary of the assessment report including security recommendations appears.

The administrator reviews and applies recommendations by first choosing the appropriate service from the Apply To drop-down list, then selecting one or more recommendations using check boxes, and clicking Apply Fix. The View drop-down list selection determines whether the visible list of recommendations includes All Recommendations, only Applied Recommendations, or only Pending Recommendations.

The default view is All Recommendations. The table Vulnerability Assessment Threat Control Recommendations details the fields in each recommendation.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the vulnerability detected.</td>
</tr>
</tbody>
</table>
Steps to Mitigate Web site Vulnerabilities

1. Scan Web application(s) using third party vulnerability scanning software.
2. Choose *.xml (the only supported format) for the exported vulnerability output file format.
3. Use ADVANCED > Import Vulnerability Report to select Assessment Name and Assessment Type (IBM AppScan Version 7.9 is the only supported type). Then Browse to select and import the vulnerability report (.xml file).
4. Submit imports the report and generates recommendations.
5. Use Manage Vulnerability Assessments to select the assessment report from the Assessment Name drop-down list.
7. Select a service from the Apply To drop-down list before applying recommendations.
8. In Recommendation(s) for Vulnerability Assessment select check boxes to apply.
9. Apply Fix to mitigate corresponding vulnerabilities.
Appendix A
Barracuda Web Application Firewall Hardware

This appendix provides hardware information for the Barracuda Web Application Firewall. The following topics are covered:

System Hardware Features .............................................................. 196
Hardware Compliance ..................................................................... 201
System Hardware Features

This section illustrates and describes the front and back panel controls, ports and LED indicators on the Barracuda Web Application Firewall.

Front Panel Features of Model 360 / 460 / 660

The following figure shows the front panel components of Model 360, 460 and 660 as described in Table A.1.

*Figure A.1: Front Panel of the 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.*

<table>
<thead>
<tr>
<th>Diagram Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>On/Off button</td>
</tr>
<tr>
<td>2</td>
<td>Reset button</td>
</tr>
<tr>
<td>3</td>
<td>Power Indicator</td>
</tr>
<tr>
<td>4</td>
<td>Disk Activity</td>
</tr>
<tr>
<td>5</td>
<td>Management Network Activity</td>
</tr>
<tr>
<td>6</td>
<td>Indicates the System Bypass Mode</td>
</tr>
<tr>
<td>7</td>
<td>Indicates the Failed Status of the System</td>
</tr>
<tr>
<td>8</td>
<td>LAN Port</td>
</tr>
<tr>
<td>9</td>
<td>WAN Port</td>
</tr>
</tbody>
</table>

Front Panel Features of Model 860 and 960

The following figure shows the front panel components of Model 860 and 960 as described in Table A.2.
The following table describes the front panel components on the Barracuda Web Application Firewall.

**Table A.2: Barracuda Web Application Firewall Front Panel Component Descriptions.**

<table>
<thead>
<tr>
<th>Diagram Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>On/Off button</td>
</tr>
<tr>
<td>2</td>
<td>Reset button</td>
</tr>
<tr>
<td>3</td>
<td>System Power Indicator</td>
</tr>
<tr>
<td>4</td>
<td>Hard Drive Activity</td>
</tr>
<tr>
<td>5</td>
<td>Management Network Activity</td>
</tr>
<tr>
<td>6</td>
<td>Redundant Management Network Activity</td>
</tr>
<tr>
<td>7</td>
<td>Indicates the System Bypass Mode</td>
</tr>
<tr>
<td>8</td>
<td>Indicates the Failed Status of the System</td>
</tr>
</tbody>
</table>

**Back Panel Features of Model 360 / 460 / 660**

The following figure shows the back panel components as described in **Table A.3**.

**Figure A.3: Back Panel of the Barracuda Web Application Firewall**

The following table describes the back panel components on the Barracuda Web Application Firewall.
Table A.3: Barracuda Web Application Firewall Back Panel Component Descriptions.

<table>
<thead>
<tr>
<th>Diagram Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Unused USB Port</td>
</tr>
<tr>
<td>2</td>
<td>Unused USB Port</td>
</tr>
<tr>
<td>3</td>
<td>Unused Network Port</td>
</tr>
<tr>
<td>4</td>
<td>Unused USB Port</td>
</tr>
<tr>
<td>5</td>
<td>Unused USB Port</td>
</tr>
<tr>
<td>6</td>
<td>VGA Display (console)</td>
</tr>
<tr>
<td>7</td>
<td>Unused Printer Port</td>
</tr>
<tr>
<td>8</td>
<td>Serial Port</td>
</tr>
<tr>
<td>9</td>
<td>Mouse</td>
</tr>
<tr>
<td>10</td>
<td>Keyboard</td>
</tr>
<tr>
<td>11</td>
<td>Redundant Power Supply</td>
</tr>
</tbody>
</table>

Back Panel Features of Model 860 and 960 - Ethernet Interface

The following figure shows the back panel components of Model 860 and 960 as described in Table A.4.

Figure A.4: Back Panel of the Barracuda Web Application Firewall

The following table describes the back panel components on the Barracuda Web Application Firewall.

Table A.4: Barracuda Web Application Firewall Back Panel Component Description

<table>
<thead>
<tr>
<th>Diagram Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>WAN Port</td>
</tr>
<tr>
<td>2</td>
<td>LAN Port</td>
</tr>
<tr>
<td>3</td>
<td>Unused Network Port</td>
</tr>
</tbody>
</table>
The following figure shows the back panel components of Model 860 and 960 as described in Table A.5.

Figure A.5: Back Panel of the Barracuda Web Application Firewall

The following table describes the back panel components on the Barracuda Web Application Firewall.

Table A.5: Barracuda Web Application Firewall Back Panel Component Descriptions.
<table>
<thead>
<tr>
<th>Diagram Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>VGA Display (console)</td>
</tr>
<tr>
<td>6</td>
<td>Unused Printer Port</td>
</tr>
<tr>
<td>7</td>
<td>Serial Port</td>
</tr>
<tr>
<td>8</td>
<td>Unused USB Port</td>
</tr>
<tr>
<td>9</td>
<td>Unused USB Port</td>
</tr>
<tr>
<td>10</td>
<td>Not Connected</td>
</tr>
<tr>
<td>11</td>
<td>Keyboard</td>
</tr>
<tr>
<td>12</td>
<td>Mouse</td>
</tr>
<tr>
<td>13</td>
<td>Redundant Power Supply</td>
</tr>
<tr>
<td>14</td>
<td>Redundant Power Supply</td>
</tr>
</tbody>
</table>
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 in 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 on an experienced radio/ television technician for help.

Notice for Canada

This apparatus compiles with the Class B limits for radio interference as specified in the Canadian Department of Communication Radio Interference Regulations.

Notice for Europe (CE Mark)

Appendix B

Extended Match and Condition Expressions

This appendix explains how extended match and condition expressions are formed to match requests and responses to configured policies on the Barracuda Web Application Firewall. The following topics are covered:

- Extended Match Rules in Barracuda Web Application Firewall..... 204
- Quick reference to Extended Match Expressions.......................... 204
Extended Match Rules in Barracuda Web Application Firewall

The Barracuda Web Application Firewall matches configured policies to incoming data to determine which elements apply. In most cases, host and URL adequately distinguish matching data. Matching in this case is determined by a best match algorithm where the best match is the rule with the longest matching host and URL keys.

When comparison of additional fields is required, Extended Match Rules are used across the Barracuda Web Application Firewall to extend the matching criteria to additional fields. Extended Match fields are only evaluated for requests matching host and URL. Extended matching uses a key comprised of the URL, host, and an extended match rule in specified sequential order. Multiple rules are evaluated in the sequence indicated by Extended Match Sequence, with lower sequential numbers matched first.

Using extended match, different elements of the request can be combined in a very flexible manner. 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 to Extended Match Expressions

- 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

An Extended match expression consists of one or more Element Matches, combined using Join operators AND and OR. 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.
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)
- **X509_OU** - This refers to the Organizational Unit (OU) stated in the X.509 certificate. Example: (X509_OU eq Engineering Division). When **Client authentication** is enabled for a HTTPS service, the certificate presented by the client is matched with the element value. If the request matches the rule, the Barracuda Web Application Firewall executes the specified action.

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.
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 reo .*\.html) && (Method eq GET)`.

Nested sub-expressions can be created using enclosed parentheses within expressions. This makes the expression more readable and unambiguous. Example: `(HTTP-Version eq HTTP/1.1) && ((Header Host eq www.example.com) || (Header Host eq Web site.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, "\e" stands for the character "e" 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 C

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 C.1: Macro Definitions

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Request Rewrites</strong></td>
<td></td>
</tr>
<tr>
<td>$SRC_ADDR</td>
<td>Inserts the source (client) IP address. You can use it for the new value (Rewrite Value parameter) when inserting or rewriting a header (see Configuring HTTP Request Rewrite on page 108 for an example).</td>
</tr>
<tr>
<td>$URI</td>
<td>Should be specified in the new value, if you are rewriting or redirecting the URI. $URI specifies the complete request URI including the query string.</td>
</tr>
<tr>
<td>$X509_VERSION</td>
<td>The client certificate's X509 version string.</td>
</tr>
<tr>
<td>$X509_SERIAL_NUMBER</td>
<td>The serial number of the client certificate.</td>
</tr>
<tr>
<td>$X509_SIGNATURE_ALGORITHM</td>
<td>The Signature Algorithm used in the client certificate.</td>
</tr>
<tr>
<td>$X509_ISSUER</td>
<td>The client certificate's issuer string.</td>
</tr>
<tr>
<td>$X509_NOT_VALID_BEFORE</td>
<td>Time from which the client certificate is valid.</td>
</tr>
<tr>
<td>$X509_NOT_VALID_AFTER</td>
<td>Time after which the client certificate is invalid.</td>
</tr>
<tr>
<td>$X509_SUBJECT</td>
<td>The client certificate's Subject string.</td>
</tr>
<tr>
<td>$X509_SUBJECT_PUBLIC_KEY_TYPE</td>
<td>The X509 Certificate Subject Key Identifier String of the client certificate.</td>
</tr>
<tr>
<td>$X509_SUBJECT_PUBLIC_KEY</td>
<td>Public Key modulus of the client certificate.</td>
</tr>
<tr>
<td>$X509_SUBJECT_PUBLIC_KEY_RSA_BITS</td>
<td>Size of the client certificate's public key, in bits.</td>
</tr>
<tr>
<td>$X509_EXTENSIONS</td>
<td>The client certificate's X509 Extensions String.</td>
</tr>
<tr>
<td>$X509_HASH</td>
<td>The X509 Hash string of the client certificate.</td>
</tr>
</tbody>
</table>
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:

```
$X509_WHOLE
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.*

*Note:
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".

---

### URL ACLs

$NONAME_PARAM
Inserts a parameter with no name (see No Name Parameters on page 210)

---

### Redirect Policy

%s Load Balancing
Represents the complete request URI.

%10sLoad Balancing
Represents the first 10 characters of the request URI.

---

### Response Page

%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 address 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 C.2: SNMP GET Command**

<table>
<thead>
<tr>
<th>Name</th>
<th>Object ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TotalApplications</td>
<td>1.3.6.1.4.1.20632.8.2</td>
<td>Total applications configured.</td>
</tr>
<tr>
<td>TotalServers</td>
<td>1.3.6.1.4.1.20632.8.3</td>
<td>Total servers configured.</td>
</tr>
<tr>
<td>TotalAttacks</td>
<td>1.3.6.1.4.1.20632.8.4</td>
<td>Count of attacks in last one hour.</td>
</tr>
<tr>
<td>ActiveApplications</td>
<td>1.3.6.1.4.1.20632.8.5</td>
<td>Total applications configured whose status is ON.</td>
</tr>
<tr>
<td>ActiveServers</td>
<td>1.3.6.1.4.1.20632.8.6</td>
<td>Total servers whose operational status is in-service.</td>
</tr>
<tr>
<td>bwsMessage</td>
<td>1.3.6.1.4.1.20632.8.7</td>
<td>System log message.</td>
</tr>
<tr>
<td>SystemLoad</td>
<td>1.3.6.1.4.1.20632.8.8</td>
<td>System load in percentage.</td>
</tr>
<tr>
<td>CPUFanSpeed</td>
<td>1.3.6.1.4.1.20632.8.9</td>
<td>CPU fan speed in rotations per min.</td>
</tr>
<tr>
<td>Name</td>
<td>Object ID</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------------------</td>
<td>------------------------------------------------------</td>
</tr>
<tr>
<td>SystemFanSpeed</td>
<td>1.3.6.1.4.1.20632.8.10</td>
<td>System fan speed in rotations per min.</td>
</tr>
<tr>
<td>CPUTemperature</td>
<td>1.3.6.1.4.1.20632.8.11</td>
<td>CPU temperature in degree celsius.</td>
</tr>
<tr>
<td>FirmwareStorage</td>
<td>1.3.6.1.4.1.20632.8.12</td>
<td>Firmware storage in percentage.</td>
</tr>
<tr>
<td>LogStorage</td>
<td>1.3.6.1.4.1.20632.8.13</td>
<td>Log Storage in percentage.</td>
</tr>
<tr>
<td>HighAvailabilityStatus</td>
<td>1.3.6.1.4.1.20632.8.14</td>
<td>High Availability Status.</td>
</tr>
<tr>
<td>OperationalMode</td>
<td>1.3.6.1.4.1.20632.8.15</td>
<td>Operation mode.</td>
</tr>
<tr>
<td>DataPathStatus</td>
<td>1.3.6.1.4.1.20632.8.16</td>
<td>Data Path Status.</td>
</tr>
<tr>
<td>LinkStatus</td>
<td>1.3.6.1.4.1.20632.8.17</td>
<td>Link Status.</td>
</tr>
</tbody>
</table>
Appendix D

WSDL Files

This appendix describes the standard WSDL syntax and provides sample code. This appendix covers the following topics:

- Elements of a WSDL file ................................................................. 214
- WSDL 1.2 Syntax ........................................................................... 214
- Sample WSDL Code .................................................................... 216
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:

```xml
<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:operation> 
  </wsdl:portType> 
</wsdl:definitions>
```
<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>
The following code is a sample WSDL file for a simple math service that adds, subtracts, multiplies, and divides.

```xml
<?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: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:sequence>
  </s:complexType>
</s:element>
</s:element>
</s:complexType>
</s:element>
</s:element name="Multiply">
<s:complexType>
  <s:sequence>
    <s:element maxOccurs="1" minOccurs="1"
      name="a" type="s:int"/>
  </s:sequence>
</s:complexType>
</s:element>
</types>
</definitions>
```
<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>
<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>
    <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>
</binding>
<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>
</definitions>
Appendix E

Evaluation Policy and Flow

This appendix describes the request and response processing flow and the evaluation policies and precedence rules that control that flow. It includes:

Request Policy Order .............................................................. 222
Response Policy Order ............................................................ 225
Evaluation Policies

The Barracuda Web Application Firewall considers a number of policies when evaluating a request or response.

Request Policy Order

The following list is the evaluation order of the policies applied to a Web application request:

1. Request limits [P1]
2. Instant SSL redirect policy [P2]
3. URL normalization [P3]
4. Rule match [P4]
5. Cookie security [P5]
7. Session Tracking [P7]
8. Local Allow Deny Rules [P8]
9. Process Profile or Default URL protection and Parameter protection [P9]
10. Advanced Security [P10]
11. Authentication and Access Control [P11]
12. Caching [P12]
13. Web address translation (WAT) [P13]
14. XML Firewall [P14]
15. Load Balancing [P15]

In addition, when an HTTP request is served, information about the request is logged once for each request. The Web log policy [P20] defines the information to be logged.

The following flowchart explains the Request Policy Order:
Response Policy Order

The following list is the evaluation order of the policies applied to a Web application response:

1. Web address translation (WAT) [P16]
2. Instant SSL rewrite [P17]
3. Caching [P18]
4. XML Firewall [P19]

The following flowchart explains the Response Policy Order:
**Execution Flow**

The following sections describe how the Barracuda Web Application Firewall processes and evaluates Web application requests and responses. (Policies are referenced by the associated number from the preceding list P1 through P16)

**HTTP Request**

The following policies are applied before a request is forwarded to the back-end server farm. The policies are executed as follows:

1. When a request is received, the Barracuda Web Application Firewall first performs a limit check [P1] on the headers. This includes checking the URL length, header length, number of headers, and request length (see Request Limits on page 61). If a request exceeds any of these limits, the request is dropped. These checks are performed as request headers are received at the application layer. Note that request headers could come in one or more TCP packets.

2. If Instant SSL redirect policy is enabled for this application [P2], the request is redirected to the corresponding HTTPS application (see Securing an HTTP Web Site with HTTPS on page 55). No other policies are applied on this request.
3. After all headers are received, the URL and domain of the request are normalized [P3] into a temporary local buffer (see URL Normalization on page 64). The normalized URL and domain strings are used by all the remaining policies.

4. The next step is to find the best matching rule group [P4]. Input to the rule match is the entire request header (see Rule Matching on page 228).

5. The next step is to decrypt the encrypted cookies by applying the cookie security policies [P5] (see Cookie Security on page 61). At this point, cookies inserted by modules such as authentication are also decrypted if they exist. If cookie decryption fails, the cookie is removed from the request. This policy never denies a request.

6. At this point global allow-deny rules [P6] are processed. If any rule matches the request, corresponding action is taken. The action can be one of allow, deny with log, deny without log or redirect. If session tracking [P7] is enabled, session is updated. After this the local allow-deny rules [P8], if any, are processed. If any rule matches the request, corresponding action is taken.

7. If profiles [P9] are enabled, perform profile validations. If a profile is matched, process the profile. If learning [P11] is configured in the profile, it is enabled for this request. If profiles are not enabled or no profile is matched, perform default URL protection [P9] and parameter protection [P9].

8. If a URL policy [P10] is matched with the request, perform the validations as per configuration. The validations may include virus scan, data theft protection, brute force prevention and rate control.

9. If access control is enabled on the request [P11], and if there is no authentication cookie, the request is dropped. If access is allowed for the user identified by the authentication cookie, the request proceeds to run the next policy. A request goes through the authentication process, if the request is for the authentication landing or login page. If authentication is successful, the Barracuda Web Application Firewall inserts an authorization cookie. None of the remaining policies are applicable to authorization requests.

10. If caching policy [P12] is enabled (see Configuring Caching on page 104) and if the request can be retrieved from the cache, the Barracuda Web Application Firewall serves the object from its cache store if it exists. If the object is found in the cache, none of the other policies are applicable on the request. The response from cache is also not subjected to any response policies. If the object is not found in cache, the request proceeds to the next step.

11. WAT policies [P13] are performed on the request (see Configuring Web Site Translation on page 108). This includes request rewrite and URL translation policies. Some portions of the request are rewritten as a result of this policy. Rewrite might also redirect the request, in which case none of the remaining policies are applied on the request. If the request is not redirected, it proceeds to run the next policy.

12. If XML Firewall [P14] is enabled request (see XML Firewall on page 134) content type is XML and if any matching protected URL is configured, XML protection validations are enforced based on the enabled configuration. The validations can include XML document checks, WS-I basic profile assertions and SOAP validations.

13. Load balancing policy [P15] (see Load Balancing on page 100) finds the appropriate server from the server farm, and the Barracuda Web Application Firewall forwards the request to a back-end server. The Barracuda Web Application Firewall might add a connection header at this step.

**HTTP Response**

The following policies are applied on a response from a back-end server.

1. WAT policies are applied [P16]:

227
1a. Response rewrite policy rewrites the response headers (see Configuring HTTP Response Rewrite on page 108). It can add a header or delete a header. This can be done conditionally based on other response headers and request URL and domain fields.

1b. URL translation policy rewrites the content (see Configuring URL Translation on page 108). If any hyperlink reference in the HTML content recognized by the HTML parsing engine matches a URL translation “inside rule,” the link is rewritten by applying the corresponding “outside rule.” If a page is translated, the response is either encoded using HTTP/1.1 Transfer Chunk Encoding scheme or the underlying TCP connection is closed if the front end used the HTTP/1.0 protocol.

2. If the Instant SSL rewrite policy is enabled [P17], some of the absolute URIs (hyperlinks with domain fields) found by the parsing engine might be rewritten to use the HTTPS protocol. This is based on the rewrite domain list in the Instant SSL policy.

3. If the caching policy [P18] is on for the request URL and if the response headers indicate that the object is cacheable, a copy of the page is stored in the cache.

4. If XML Firewall [P19] is enabled and response matches any protected URL, then XML validations are enforced.

Local Allow/Deny Rules

URL ACL rules (step 6 in “HTTP Request”) are applied in the following order. If the ACL mode is set to active, request processing is terminated when a violation of any policy in the ACL is detected. If the ACL mode is set to passive, the violation is logged and request processing continues. (The matching algorithm for URL ACLs and rule groups is the same, see Rule Matching on page 228)

1. **URL ACLs**: The incoming request is matched against the list of URL ACLs to find the best matching ACL. If no match is found or the action parameter is set to deny, the request is dropped. The request is also checked against the limits and normalization policies. The request is dropped for any violation of these policies. The match is done against the < domain, URL, header > values in that order of precedence.

2. **Header ACLs**: Each of the headers in the request is matched against the list of header ACLs. If a match is found, the header value is checked for the configured maximum length, denied metacharacters, and denied keywords for violations.

URL Policies

URL policies (step 8 in “HTTP Request”) are applied in the following order.

1. **Virus Scan**: The incoming client requests are scanned for the presence of viruses. Requests containing virus signatures are denied.

2. **Rate Control**: The Rate Control Pool is defined to limit the client requests to the Barracuda Web Application Firewall. You can add a rate control pool and set the maximum requests for that pool. Rate Controls ensures that applications are not pushed beyond their performance limits, preventing application layer Denial of Service (DoS) attacks.

3. **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.

Rule Matching

A rule consists of three patterns: host, URL, and extended match rules. The “best matching” rule is used to apply policy decisions on a request. Host and URL rules are used to match Host and URL fields, respectively. Extended match rules can be used to match any combination of HTTP headers and/or query string parameters in a request. An “*” rule (to be read as a rule consisting of URL “*”) would match any value for that parameter.
The best matching rule is the rule with the longest matching host and URL and is the first matching header rule in that hierarchical order. If there are more than one extended match rule configured with the same host and URL keys, the extended match rules are searched based on extended match sequence.

**Rule Match Algorithm**

Host and URL rules (used for both URL Policies and Rule Groups) are treated as `<prefix, suffix>` pairs by the rule-match engine:

- Prefix rule key is the part of the rule preceding the asterisk (*). The asterisk is treated as a wildcard meaning any value.
- Suffix rule key is the part of the rule succeeding the asterisk.

If a rule does not have an asterisk, its suffix rule key is NULL.

The following algorithm is used by Rule Match engine:

1. Find best matching Host Prefix Rule Key. Best match is defined as the longest rule matching the HTTP request Host header left to right. The number of characters matched is the length of the Prefix Rule Key. If no Prefix Rule matches, Rule Match engine terminates with failure and the request is dropped.

2. Find the best matching Host Suffix Rule Key. Best match is defined as the longest Suffix Rule Key matching the HTTP request Host header right to left. The number of characters matched is the length of Suffix Rule Key. If a matching rule is found, the current `<Prefix, Suffix>` pair the matching Host Rule Key and go to Step 3. If no Host suffix Rule matches, discard this Prefix Rule Key and go to Step 1 to find the next matching Host Prefix Rule.

3. Find the best matching URL Prefix Rule Key. If none found, discard this Host Rule Key and go to step 1 to find the next matching Host Rule Key. Best matching URL Prefix Key is defined as the longest URL Prefix Rule matching the HTTP Request-URI header left to right. The number of characters matched is the length of URL Prefix Rule Key.

4. Find the best matching URL Suffix Rule Key. Best match is defined as the longest Suffix Rule Key matching the HTTP Request-URI header right to left. If found, the current `<Prefix, Suffix>` pair is the matching URL Rule Key. We found the best matching Host Rule Key and URL Rule Key and continue at Step 5. If no match is found, discard this Prefix URL Rule and go back to Step 3 to find the next matching URL Prefix Rule.

5. Find the first matching Extended Match Rule. The Extended Match rules are sorted based on the extended match sequence. If a matching rule is found, Rule match engine terminates with success and the HTTP Request follows the policies defined on this rule. If no match is found, discard this Extended Match Rule and go back to Step 4 to find the next matching URL Suffix Rule Key.

A successful search terminates at step 5. An unsuccessful search terminates at step 1.

**Hierarchical Match**

The hierarchical match first matches the host header in the request against all the host matches configured, and takes the best match. Now the ACL list is reduced to this set. Next, it matches the URL path in the request against all URLs in the reduced set of ACLs. This is again a best match. If multiple ACLs match, then each extended match rule is evaluated in ascending order of extended match sequence. The first extended match rule that matches will be the ACL match.

For example, consider the requests given below and see how it is matched with the ACLs in the following table:
Table E.1: Sample Hierarchical Rule Match

<table>
<thead>
<tr>
<th>ACLs</th>
<th>Host Match</th>
<th>URL Match</th>
<th>Extended Match Rule</th>
<th>Extended Match Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><a href="http://www.abc.com">www.abc.com</a></td>
<td>/sales1/*</td>
<td>Header User-Agent co IE5.0</td>
<td>1 1</td>
</tr>
<tr>
<td>2</td>
<td><a href="http://www.abc.com">www.abc.com</a></td>
<td>/sales1/*</td>
<td>Header User-Agent co Mozilla 2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td><a href="http://www.abc.com">www.abc.com</a></td>
<td>/sales1/*</td>
<td>*</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td><a href="http://www.abc.com">www.abc.com</a></td>
<td>/sales2/*</td>
<td>Header User-Agent co wget 0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>*.abc.com</td>
<td>/sales2/*</td>
<td>*</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>*.abc.com</td>
<td>/sales3/*</td>
<td>*</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>*</td>
<td>/sales1/*</td>
<td>*</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>0</td>
</tr>
</tbody>
</table>

1. http://www.abc.com/sales1/index.html, from IE5.0
   1a. Host header is www.abc.com, therefore ACLs 1 to 4 matches this request. Also ACLs 4 to 8 matches, but this is the best match, since it is most specific.
   1b. URL Path is /sales1/index.html, therefore ACLs 1 to 3 match.
   1c. Evaluate extended match rule for ACL 1 first, since extended match sequence = 1.
   1d. As a result, ACL 1 is matched.

   2a. Host header is www.abc.com, therefore ACLs 1 to 4 match. Also ACLs 4 to 8 matches, but this is the best match, since it is most specific.
   2b. URL Path is /sales2/index.html, therefore only ACL 4 matches.
   2c. Evaluate extended match rule for ACL 4. ACL 4 does not matches, since User-Agent is expected to be wget, and it will not be true.
   2d. Thus, no matches in ACLs 1 to 4.
   2e. The next best matching ACLs on the host header are ACLs 5 to 6.
   2f. URL Path matches only ACL 5.
   2g. Evaluate extended match rule for ACL 5. * matches anything.
   2h. As a result, ACL 5 is matched.

   3a. Host header is www.abc.com, therefore ACLs 1 to 4 matches. (Also ACLs 4 to 8 matches, but this is the best match, since it is most specific.
   3b. URL Path is /sales3/index.html, which does not matches any of the URLs in ACLs 1 to 4.
   3c. The next best matching ACLs on the host header are ACLs 5 to 6
   3d. URL path matches only ACL 6.
   3e. Header rule matches ACL 6.
   3f. As a result, ACL 6 is matched.

   4a. Host header is mirror.abc.com, therefore ACLs 5 and 6 matches.
   4b. URL Path does not match any of the URLs in ACLs 5 and 6.
   4c. The next best matching ACLs on the host header are ACLs 7 and 8.
4d. URL path matches only ACL 8.
4e. Header rule matches ACL 8.
4f. As a result, ACL 8 is matched.

Sequential Match

The sequential match completely ignores the host header and URL path. Next each extended match rule is evaluated in sequential order based on the extended match sequence. The first extended match rule that matches will be the ACL match.

To explain how the rule-match engine selects the best match, consider the following Rule Match table:

Table E.2: Sample Sequential Rule Match

<table>
<thead>
<tr>
<th>ACLs</th>
<th>Host Match</th>
<th>URL Match</th>
<th>Extended Match Rule</th>
<th>Extended Match Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>*</td>
<td>*</td>
<td>Header Host eq <a href="http://www.abc.com">www.abc.com</a> &amp;&amp; Header User-Agent co IE5.0 &amp;&amp; URI req /sales1/*</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>*</td>
<td>*</td>
<td>Header Host eq <a href="http://www.abc.com">www.abc.com</a> &amp;&amp; Header User-Agent co Mozilla &amp;&amp; URI req /sales1/*</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>*</td>
<td>*</td>
<td>* ** Header Host eq <a href="http://www.abc.com">www.abc.com</a> &amp;&amp; URI req /sales1/*</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>*</td>
<td>*</td>
<td>* ** Header Host eq <a href="http://www.abc.com">www.abc.com</a> &amp;&amp; Header User-Agent co wget &amp;&amp; URI req /sales2/*</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>*</td>
<td>*</td>
<td>* ** Header Host req .abc.com &amp;&amp; URI req /sales2/*</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>*</td>
<td>*</td>
<td>* ** Header Host req <em>.abc.com &amp;&amp; URI req /sales3/</em></td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>*</td>
<td>*</td>
<td>* URI req /sales1/*</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>8</td>
</tr>
</tbody>
</table>

1. sales2http://www.abc.com/sales1/index.html, from IE5.0
   1a. ACLs 1 to 8 match host and URL keys.
   1b. Evaluate extended match rule for ACL 1 first, since extended match sequence = 1.
   1c. As a result, ACL 1 is matched.

   2a. ACLs 1 to 8 match host and URL keys.
   2b. Evaluate extended match rule for ACL 1 to 8 in order.
   2c. Extended match rule for ACL 5 matches.
   2d. As a result, ACL 5 is matched.
   3a. ACLs 1 to 8 match host and URL keys.
   3b. Evaluate header rules for ACL 1 to 8 in order.
   3c. Header rule for ACL 6 matches.
   3d. As a result, ACL 6 is matched.

   4a. ACLs 1 to 8 match host and URL keys.
   4b. Evaluate header rules for ACL 1 to 8 in order.
   4c. Header rule matches ACL 8.
   4d. As a result, ACL 8 is matched.
The following table describes some of the common Web application attack techniques and the corresponding protection employed by the Barracuda Web Application Firewall.

### Table F.1: Attack Types and Corresponding Barracuda Web Application Firewall Defense

<table>
<thead>
<tr>
<th>Technique</th>
<th>Description</th>
<th>Protection provided by Barracuda Web Application Firewall</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Injection Attacks</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQL Injection</td>
<td>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.</td>
<td>Protects against all SQL injection vulnerabilities by inspecting application traffic and blocking all methods of inserting dangerous database commands into URLs, headers, and forms.</td>
</tr>
<tr>
<td>Cross-Site Scripting</td>
<td>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.</td>
<td>Protects against cross-site scripting vulnerabilities by inspecting application traffic and blocking all methods of inserting malicious scripts into URLs, headers, and forms.</td>
</tr>
<tr>
<td>Remote File Injection</td>
<td>Remote File Inclusion attacks allow malicious users to run their own PHP code on a vulnerable Web site to access anything that the PHP program could: databases, password files, etc.</td>
<td>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.</td>
</tr>
<tr>
<td>Command Injection</td>
<td>Operating system and platform commands can allow attackers access to data or enable privileges on back-end servers to be manipulated.</td>
<td>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.</td>
</tr>
<tr>
<td>Meta-character Injection</td>
<td>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.</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Session Attacks</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cookie/Session Poisoning</td>
<td>Cookies, often used to transmit sensitive credentials, can be modified to improperly allow access or assume another user's identity.</td>
<td>Digitally encrypts, signs, and time-stamps cookies, protecting their content from tampering.</td>
</tr>
<tr>
<td>Cookie Snooping</td>
<td>Cookies, commonly used to transmit user credentials, are often encoded only with simple encoding methods like Base64. This can lead to disclosure of login credentials.</td>
<td>Digitally encrypts, signs, and time-stamps cookies, protecting their content from tampering.</td>
</tr>
</tbody>
</table>
Session Hijacking
An attacker can use a sniffer to capture a valid token session called “Session ID”, and then use the valid token 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, denying any other source who attempts to use it.

Cross-Site Request Forgery (CSRF) Attack
CSRF attack tricks a user into loading a page that executes undesired actions on another site using the user's credentials, including logout, purchase item, change account information, or retrieve account information.

Injects randomized tokens into online forms and URLs to authenticate data streams, eliminating the ability to submit unauthorized or malicious requests.

Denial of Service (DoS) Attacks
Length Attack
Requests with lengths greater than the defaults potentially perpetrate the most common kind of DoS attack, a buffer overflow attack.

Protects against buffer overflow attacks by restricting the number, name and value lengths for cookies, HTTP request URL and HTTP headers.

Rate Control Attack
The number of connection requests from a single IP address can saturate the server so it cannot respond to legitimate requests.

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
The number of sessions originating from a client IP address in a given interval of time increases beyond the capacity of the server making it unable to respond to legitimate requests.

Protects against session based attacks by limiting the number of sessions originating from a particular client IP address in a given interval of time.

Other Attacks
Reconnaissance
Hackers conduct reconnaissance to improve their chances of hiding their identity and attacking a targeted network. Successful reconnaissance is the likely precursor of an actual attack.

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
Bruteforce attacks attempt every possible code combination, or password to find the right one. For example, in a system which only allows 4 digit PIN codes, there are a maximum of 10,000 possible PIN combinations for an attacker to try before gaining access to the system.

Protects against bruteforce attacks by limiting the number of requests in error within a configured time period, blocking the requester when the threshold has been exceeded. This protects against any quick succession of malicious requests.

Cryptographic Interception
Hackers seldom attempt to break strong encryption like SSL, preferring to 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, ensuring unencrypted traffic never traverses the network. Combining all critical DMZ functionality into a single device also reduces the risk of exposure.
<table>
<thead>
<tr>
<th>Technique</th>
<th>Description</th>
<th>Protection provided by Barracuda Web Application Firewall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forceful Browsing/Directory Traversal</td>
<td>Forceful browsing attempts to access files and directories of a Web service without using the Web service provided inks. When an attacker sees directory <a href="http://a.com/guest/welcome.html">http://a.com/guest/welcome.html</a> he may try <a href="http://a.com/member/welcome.html">http://a.com/member/welcome.html</a>. This could gain access to an area of the Web site without supplying proper credentials.</td>
<td>Prevents the access of unpublished Web pages by using application profiles and blocking requests with path traversal metacharacters, thus limiting access to only those pages that the application was designed to expose.</td>
</tr>
<tr>
<td>Log Tampering</td>
<td>Erasing and tampering with transaction logs allows an attacker to cover their tracks or alter Web transaction records.</td>
<td>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.</td>
</tr>
<tr>
<td>Error Message Interception</td>
<td>Information in error messages are often rich with site-specific information, allowing an attacker to learn private application architectures.</td>
<td>The Web Site cloaking feature prevents unintended information disclosure from error messages.</td>
</tr>
<tr>
<td>Attack Obfuscation</td>
<td>Hackers frequently disguise attacks by encoding their requests with methods like URL encoding or Unicode.</td>
<td>Fully decodes URL, Unicode, and polymorphic encoding before inspection.</td>
</tr>
<tr>
<td>Application Platform Exploits</td>
<td>Well-known exploits can often be addressed through a patch, but patching is not always timely.</td>
<td>Allows for blocking of well-known attacks, effectively buying time for proper patch management.</td>
</tr>
<tr>
<td>Security Management Exploits</td>
<td>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.)</td>
<td>Has all management functions securely firewalled from production traffic and is operated through dedicated, secure management channels.</td>
</tr>
<tr>
<td>Parameter/Form Tampering</td>
<td>Parameters used in URLs, HTTP headers, and forms are often used to control and validate access to sensitive information.</td>
<td>Protects against parameter tampering by using parameter profiles for all application parameters and allowing only user requests that match the legitimate profile.</td>
</tr>
<tr>
<td>Buffer Overflow</td>
<td>Attackers attempt to flood vulnerable back-end servers with excess requests. If successful, attackers can often execute commands directly on the compromised server.</td>
<td>Automatically enforces legitimate buffer limits at the perimeter, ensuring that even vulnerable servers cannot be compromised.</td>
</tr>
</tbody>
</table>
Appendix G

Internet Protocol Version 6 (IPv6)

IP version 6 (IPv6) is a new version of the Internet Protocol, the successor of IP version 4 (IPv4). The main advantage of IPv6 is a larger address space than IPv4.

- IPv6 uses 128-bit IP addresses compared to 32-bit IP addresses used by IPv4.
- Version 6 supports varied addressing types (unicast, anycast, multicast, link-local, site-local and global).
- IPv6 addresses can be associated with an interface or a set of interfaces.

Address Notation

IPv6 addresses are represented as eight 16-bit hexadecimal block separated by colons (:


The leading zeros can be omitted within each 16-bit hexadecimal block, and written as 0 instead of 0000, 100 instead of 0100 and 10 instead of 0010. These zeros can be further compressed and replaced with double-colon (::) to make IPv6 address notation less cumbersome. Double-colon can be used only once to compress an IPv6 address, either in the beginning, middle or end.

*Example:* FEDC::FEDC:E4BF:100:10 is equivalent to FEDC:0000:0000:0000:FEDC:E4BF:100:10

IPv6 Support in Barracuda Web Application Firewall

The Barracuda Web Application Firewall supports Internet Protocol Version 6 (IPv6) along with its predecessor IPv4. The current firmware release (7.6) provides only the basic IPv6 features. Read the following sections before configuring IPv6 services:

- **Enabling IPv6**
- **Configuring the Barracuda Web Application Firewall with IPv6 address**

Enabling IPv6

By default, IPv6 is disabled on the Barracuda Web Application Firewall. To enable IPv6, go to the **BASIC > IP Configuration > Addresses** section and set **Enable IPv6** to **Yes**. Note that this section is visible only in the **expert mode**. To turn on the expert mode, the administrator is required to add “&expert=1” at the end of the URL.

*Example:* http://192.168.132.45:8000/cgi-mod/index.cgi?&user=admin&password=8d7826d5f79c&et=1305539072&auth_type=Local&locale=en_US&primary_tab=BASIC&secondary_tab=ip_config&expert=1
Configuring the Barracuda Web Application Firewall with IPv6 address

Before configuring IPv6 services, you need to assign IPv6 addresses to the interfaces on the BASIC > IP Configuration page. Only then can you connect to an IPv6 network. By default, the Barracuda Web Application Firewall accepts traffic only from IPv4 network, and drops all IPv6 traffic. When IPv6 is enabled, the Barracuda Web Application Firewall honors both IPv4 and IPv6 traffic. The IPv6 address for interfaces can be configured in the WAN IPv6 Configuration, LAN IPv6 Configuration and Management IPv6 Configuration sections. For detailed configuration instructions, see the online help.

Configuring IPv6 Service

You can create new services (IPv4 and IPv6) using the BASIC > Services > Add New Service, by selecting the appropriate version from the Version drop-down list. Specify a name, type of service desired, a Virtual IP (VIP) address, port, and one or more Real Servers. You should configure a global IPv6 address for your Virtual IP address.

The following table lists the various interfaces to services that can be used when IPv6 is enabled:

Table G.1: IPv6 Configuration

<table>
<thead>
<tr>
<th>Front-end</th>
<th>Back-end</th>
<th>Use Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv6</td>
<td>IPv6</td>
<td>Used when the complete network setup is being migrated to support IPv6 based addressing.</td>
</tr>
<tr>
<td>IPv6</td>
<td>IPv4</td>
<td>Used when you wish to publish IPv6 addresses for Web applications without changing the addressing in your internal network.</td>
</tr>
<tr>
<td>IPv4</td>
<td>IPv6</td>
<td>Used when third party applications connecting to your applications are not yet ready to communicate via IPv6.</td>
</tr>
<tr>
<td>IPv4</td>
<td>IPv4</td>
<td>Used in current deployments without any IPv6 support.</td>
</tr>
</tbody>
</table>

Limitations of IPv6 in the Barracuda Web Application Firewall

The current Barracuda Web Application Firewall IPv6 implementation has the following limitations:

- The following features do not support IPv6 based addressing:
  - Adaptive Profiling
  - Network Firewall

Note

If you use pre-configured IPv4 services, you need to REBOOT the unit after enabling IPv6. If not, the configured IPv4 services will not function.
• Trusted Hosts
• Connection to Barracuda Networks Technical Support Center via the support tunnel does not honor IPv6 addresses. If you want to establish connection to the support tunnel, make sure you have IPv4 address configured in **WAN** and **LAN IP Configuration** on the **BASIC > IP Configuration** page.
• IPv6 addresses cannot be configured via Console configuration.
• The following Network Connectivity Tests (**ADVANCED > Troubleshooting > Network Connectivity Tests**) are not supported:
  • Dig NS-lookup Device
  • Wget Web Page.
• Extended Match condition for access rules (ACLs) support only ‘equals’ Operation for the IPv6 addresses, i.e. any request or response that matches exactly with the specified value is allowed. IPv6 does not support partial matches. Example: Consider the **Element Type** is **Client-IP**: `fe40:21da:ac10:fe01:ac10:fe01:21da:fe40`, **Operation** is set to ‘equals’ and the **Value** is `fe40:21da:ac10:fe01`. This value, a partial match of the **Client-IP**, and hence denied.
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 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 ARE 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.

---

**Barracuda Networks Software License Agreement (v 2.1)**

PLEASE READ THIS SOFTWARE LICENSE AGREEMENT ("AGREEMENT") CAREFULLY BEFORE USING THE BARRACuda NETWORKS SOFTWARE. BY USING THE BARRACuda SOFTWARE YOU ARE AGREEING TO BE BOUND BY THE TERMS OF THIS LICENSE. IF YOU ARE A CORPORATION, PARTNERSHIP OR SIMILAR ENTITY, THEN THE SOFTWARE LICENSE GRANTED UNDER THIS AGREEMENT IS EXPRESSLY CONDITIONED UPON ACCEPTANCE BY A PERSON WHO IS AUTHORIZED TO SIGN FOR AND BIND THE ENTITY. IF YOU ARE NOT AUTHORIZED TO SIGN FOR AND BIND THE ENTITY OR DO NOT AGREE WITH ALL THE TERMS OF THIS AGREEMENT, 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 and documentation, whether on disk, in flash memory, 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 Agreement, 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 Software is recorded but Barracuda retains ownership of the Software itself. If you have not completed a purchase of the Software and made payment for the purchase, the Software may only be used for evaluation purposes and may not be used in any production capacity. Furthermore the Software, when used for evaluation, may not be secure and may use publically available passwords.

2. Permitted License Uses and Restrictions. If you have purchased a Barracuda Networks hardware product, this Agreement 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. You may not make a backup copy of the Software. If you have purchased a Barracuda Networks Virtual Machine you may use the software only in the licensed number of instances of the licensed sizes and
you may not exceed the licensed capacities. You may make a reasonable number of backup copies of the Software. If you have purchased client software you may install the software only on the number of licensed clients. You may make a reasonable number of backup copies of the Software. For all purchases you may not modify or create derivative works of the Software except as provided by the Open Source Licenses included below. You may not make the Software available over a network where it could be utilized by multiple devices or copied. Unless otherwise expressly provided in the documentation, your use of the 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 you may have paid Barracuda Networks the required license fee; and your use of the Software shall also be limited, as applicable and set forth in your 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 install Software), (b) concurrent users, sessions, ports, and/or issued and outstanding IP addresses, and/or (c) central processing unit cycles or instructions per second. Your use of the Software shall also be limited by any other restrictions set forth in your purchase order or in Barracuda Networks' product catalog, user documentation or Web site for the Software. The BARRACUDA SOFTWARE IS NOT INTENDED FOR USE IN THE OPERATION OF NUCLEAR FACILITIES, AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, LIFE SUPPORT MACHINES, OR OTHER EQUIPMENT IN WHICH FAILURE COULD LEAD TO DEATH, PERSONAL INJURY, OR ENVIRONMENTAL DAMAGE. YOU EXPRESSLY AGREE NOT TO USE IT IN ANY OF THESE OPERATIONS.

3. You may not transfer, rent, lease, lend, or sublicense the Software or allow a third party to do so. YOU MAY NOT OTHERWISE TRANSFER THE SOFTWARE OR ANY OF YOUR RIGHTS AND OBLIGATIONS UNDER THIS AGREEMENT. You agree that you will have no right and will not, nor will it assist others to: (i) make unauthorized copies of all or any portion of the Software; (ii) sell, sublicense, distribute, rent or lease the Software; (iii) use the Software on a service bureau, time sharing basis or other remote access system whereby third parties other than you can use or benefit from the use of the Software; (iv) disassemble, reverse engineer, modify, translate, alter, decompile or otherwise attempt to discern the source code of all or any portion of the Software; (v) utilize or run the Software on more computers than you have purchased license to; (vi) operate the Software in a fashion that exceeds the capacity or capabilities that were purchased by you.

4. THIS AGREEMENT SHALL BE EFFECTIVE UPON INSTALLATION OF THE SOFTWARE OR PRODUCT AND SHALL TERMINATE UPON THE EARLIER OF: (A) YOUR FAILURE TO COMPLY WITH ANY TERM OF THIS AGREEMENT OR (B) RETURN, DESTRUCTION OR DELETION OF ALL COPIES OF THE SOFTWARE IN YOUR POSSESSION. Rights of Barracuda Networks and your obligations shall survive any termination of this Agreement. Upon termination of this Agreement by Barracuda Networks, You shall certify in writing to Barracuda Networks that all copies of the Software have been destroyed or deleted from any of your computer libraries, storage devices, or any other location.

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, THAT CURRENT OR FUTURE VERSIONS OF ANY OPERATING SYSTEM WILL BE SUPPORTED, 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. FURTHERMORE BARRACUDA NETWORKS SHALL ASSUME NO WARRANTY FOR ERRORS/BUGS, FAILURES OR DAMAGE WHICH WERE CAUSED BY IMPROPER OPERATION, USE OF UNSUITABLE RESOURCES, ABNORMAL OPERATING CONDITIONS (IN PARTICULAR DEVIATIONS FROM THE INSTALLATION CONDITIONS) AS WELL AS BY TRANSPORTATION DAMAGE. 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. YOU EXPRESSLY ACKNOWLEDGE AND AGREE THAT YOU WILL PROVIDE AN UNLIMITED PERPETUAL ZERO COST LICENSE TO BARRACUDA FOR ANY PATENTS OR OTHER INTELLECTUAL PROPERTY RIGHTS WHICH YOU EITHER OWN OR CONTROL THAT ARE UTILIZED IN ANY BARRACUDA PRODUCT.

6. Termination and Fair Use Policy. BARRACUDA SHALL HAVE THE ABSOLUTE AND UNILATERAL RIGHT AT ITS SOLE DISCRETION TO DENY USE OF, OR ACCESS TO BARRACUDA SOFTWARE, IF YOU ARE DEEMED BY BARRACUDA TO BE USING THE SOFTWARE IN A MANNER NOT REASONABLY INTENDED BY BARRACUDA OR IN VIOLATION OF ANY LAW.

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. 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.

8. Content Restrictions. YOU MAY NOT (AND MAY NOT ALLOW A THIRD PARTY TO) COPY, REPRODUCE, CAPTURE, STORE, RETRANSMIT, DISTRIBUTE, OR BURN TO CD (OR ANY OTHER MEDIUM) ANY COPYRIGHTED CONTENT THAT YOU ACCESS OR RECEIVE THROUGH USE OF THE PRODUCT CONTAINING THE SOFTWARE. YOU ASSUME ALL RISK AND LIABILITY FOR ANY SUCH PROHIBITED USE OF COPYRIGHTED CONTENT. You agree not to publish any benchmarks, measurements, or reports on the product without Barracuda Networks’ written express approval.

9. Third Party Software. Some Software which supports Bare Metal Disaster Recovery of Microsoft Windows Vista and Microsoft Windows 2008 Operating Systems (DR6) contains and uses components of the Microsoft Windows Pre-Installation Environment (WINPE) with the following restrictions: (i) the WINPE components in the DR6 product are licensed and not sold and may only be used with the DR6 product; (ii) DR6 is provided "as is"; (iii) Barracuda and its suppliers reserve all rights not expressly granted; (iv) license to use DR6 and the WINPE components is limited to use of the product as a recovery utility program only and not for use as a general purpose operating system; (v) Reverse engineering, decompiling or disassembly of the WINPE components, except to the extent expressly permitted by applicable law, is prohibited; (vi) DR6 contains a security feature
from Microsoft that will automatically reboot the system without warning after 24 hours of continuous use; (vii) Barracuda alone will provide support for customer issues with DR6 and Microsoft and its Affiliates are released of all liability related to its use and operation; and, (viii) DR6 is subject to U.S. export jurisdiction.

10. Trademarks. Certain portions of the product and names used in this Agreement, the Software and the documentation may constitute trademarks of Barracuda Networks. You are not authorized to use any such trademarks for any purpose.

11. Export Restrictions. You may not export or re-export the Software without: (a) the prior written consent of Barracuda Networks, (b) complying with applicable export control laws, including, but not limited to, restrictions and regulations of the Department of Commerce or other United States agency or authority and the applicable EU directives, and (c) obtaining any necessary permits and licenses. In any event, you may not transfer or authorize the transfer of the Software to a prohibited territory or country or otherwise in violation of any applicable restrictions or regulations. If you are a United States Government agency the Software and documentation qualify as "commercial items", as that term is defined at Federal Acquisition Regulation ("FAR") (48 C.F.R.) 2.101, consisting of "commercial computer software" and "commercial computer software documentation" as such terms are used in FAR 12.212. Consistent with FAR 12.212 and DoD FAR Supp. 227.7202-1 through 227.7202-4, and notwithstanding any other FAR or other contractual clause to the contrary in any agreement into which this Agreement may be incorporated, Government end user will acquire the Software and documentation with only those rights set forth in this Agreement. Use of either the Software or documentation or both constitutes agreement by the Government that the Software and documentation are "commercial computer software" and "commercial computer software documentation", and constitutes acceptance of the rights and restrictions herein.

12. General. THIS AGREEMENT IS GOVERNED BY THE LAWS OF THE STATE OF CALIFORNIA, USA WITH JURISDICTION OF SANTA CLARA COUNTY, CALIFORNIA, UNLESS YOUR HEADQUARTERS IS LOCATED IN SWITZERLAND, THE EU, OR JAPAN. IF YOUR HEADQUARTERS IS LOCATED IN SWITZERLAND THE SWISS MATERIAL LAW SHALL BE USED AND THE JURISDICTION SHALL BE ZURICH. IF YOUR HEADQUARTERS IS LOCATED IN THE EU, AUSTRIAN LAW SHALL BE USED AND JURISDICTION SHALL BE INNSBRUCK. IF YOUR HEADQUARTERS IS LOCATED IN JAPAN, JAPANESE LAW SHALL BE USED AND JURISDICTION SHALL BE TOKYO. THIS AGREEMENT WILL NOT BE SUBJECT TO ANY CONFLICT-OF-LAWS PRINCIPLES IN ANY JURISDICTION. THIS AGREEMENT WILL NOT BE GOVERNED BY THE U.N. CONVENTION ON CONTRACTS FOR THE INTERNATIONAL SALES OF GOODS. This Agreement is the entire agreement between You and Barracuda Networks regarding the subject matter herein and supersedes any other communications with respect to the Software. If any provision of this Agreement is held invalid or unenforceable, the remainder of this Agreement will continue in full force and effect. Failure to prosecute a party's rights with respect to a default hereunder will not constitute a waiver of the right to enforce rights with respect to the same or any other breach.

13. Assignability. You may not assign any rights or obligations hereunder without prior written consent from Barracuda Networks.

14. Billing Issues. You must notify Barracuda of any billing problems or discrepancies within sixty (60) days after they first appear on the statement you receive from your bank, Credit Card Company, other billing company or Barracuda Networks. If you do not bring such problems or discrepancies to Barracuda Networks attention within the sixty (60) day period, you agree that you waive the right to dispute such problems or discrepancies.

15. Collection of Data. You agree to allow Barracuda Networks to collect information ("Statistics") from the Software in order to fight spam, virus, and other threats as well as optimize and monitor the Software. Information will be collected electronically and automatically. Statistics include, but are not limited to, the number of messages processed, the number of messages that are categorized as
spam, the number of virus and types, IP addresses of the largest spam senders, the number of emails
classified for Bayesian analysis, capacity and usage, and other statistics. Your data will be kept private
and will only be reported in aggregate by Barracuda Networks.

16. Subscriptions. Software updates and subscription information provided by Barracuda Energize
Updates or other services may be necessary for the continued operation of the Software. You
acknowledge that such a subscription may be necessary. Furthermore some functionality may only
be available with additional subscription purchases. Obtaining Software updates on systems where
no valid subscription has been purchased or obtaining functionality where subscription has not been
purchased is strictly forbidden and in violation of this Agreement. All initial subscriptions commence
at the time of activation and all renewals commence at the expiration of the previous valid
subscription. Unless otherwise expressly provided in the documentation, you shall use the Energize
Updates Service and other subscriptions 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 you. All subscriptions are
non-transferrable. Barracuda Networks makes no warranty that subscriptions will continue un-
interrupted. Subscription may be terminated without notice by Barracuda Networks for lack of full
payment.

17. Auto Renewals. If your Software purchase is a time based license, includes software
maintenance, or includes a subscription, you hereby agree to automatically renew this purchase when
it expires unless you notify Barracuda 15 days before the renewal date. Barracuda Networks will
automatically bill you or charge you unless notified 15 days before the renewal date.

18. Time Base License. If your Software purchase is a time based license you expressly acknowledge
that the Software will stop functioning at the time the license expires. You expressly indemnify and
hold harmless Barracuda Networks for any and all damages that may occur because of this.

19. Support. Telephone, email and other forms of support will be provided to you if you have
purchased a product that includes support. The hours of support vary based on country and the type
of support purchased. Barracuda Networks Energize Updates typically include Basic support.

20. Changes. Barracuda Networks reserves the right at any time not to release or to discontinue release
of any Software or Subscription and to alter prices, features, specifications, capabilities, functions,
licensing terms, release dates, general availability or other characteristics of any future releases of the
Software or Subscriptions.

21. Open Source Licensing. Barracuda Networks products may include programs that are covered
by the GNU General Public License (GPL) or other Open Source license agreements, in particular the
Linux operating system. It is expressly put on record that the Software does not constitute an edited
version or further development of the operating system. 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. Further details may be provided in
an appendix to this agreement where the licenses are re-printed. Barracuda Networks makes available
the source code used to build Barracuda products available at source.barracuda.com. This directory
includes all the programs that are distributed on the Barracuda products. Obviously not all of these
programs are utilized, but since they are distributed on the Barracuda product we are required to make
the source code available.
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.

**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 GNU TERMS AND CONDITIONS

Barracuda Networks 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 Networks 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 Networks 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: Redis...
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, AND 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 Networks Software may include programs that are covered by the Apache License or other Open Source license agreements. The Apache 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.

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

Barracuda Networks makes available the source code used to build Barracuda products available at source.barracuda.com. This directory includes all the programs that are distributed on the Barracuda products. Obviously not all of these programs are utilized, but since they are distributed on the Barracuda product we are required to make the source code available.
A
Action Policy 65
Adaptive Security 85
  Recommended Usage 85
Administration 170
Advanced Security 88
  Brute Force 88
  Data Masking 89
  Rate Control Pool 89
  Session Tracking 89
  URL Policy 88
Appearance 170
Attack Protection 233
  Buffer overflow 61
  Cross site scripting 62
  Form POST parameters 63
  Remote file inclusion 62
  URL query string 63
Authentication 114
  SSO 115
Authentication Services
  LDAP 113
  RADIUS 113
Authorization 114
B
Backup 173
Backup Server 102
Barracuda Control Center 143
  Connect one or more Appliances 143
  Disconnect from 143
Bridge Path 38
Brute Force Protections 88
C
Caching 104
  Object Freshness 104
Certificate
  Extracting the key 125
  Third party signing 125
  Upload a Trusted Certificate 127
  Upload in PEM Format 126
  Upload PKCS12 Token Format 126
Certificate Components 124
  CA Certificate 124
  Distinguished Name 124
  Key Pair 124
  Token 124
Client Certificates 117
Compression 105
Content Rule 102
Creating a High Availability (HA) Environment 162, 192
Creating Session Identifiers 95
Custom Data Creation
  Attack Types 93
  Identity Theft Patterns 93
  Input Types 94
  Parameter Class 94
  Response Pages 95
Custom Response Pages 95
Custom Service 56
Custom SSL Service 56, 57
D
Denial of Service (DoS) 26
E
Energize Updates 174
Extended Match 103, 204
  Allow/Deny URLs 70
  Global ACLs 65
  Matching to URL Profiles 68
  URL Policy 88
  Use in Traffic Management 103, 204
F
Failback 165
Failover 164
Firmware Update 173
FTP Service 56
G
Global ACLs 65
H
Header Name For Actual Client IP 48
HTTP service 45
HTTPS service 46
I
Initial Setup 40
Instant SSL service 55
Interface Routes 153
L
Load Balancing
  HTTP only using Content Rule 100
Load balancing
  Proxy Mode all TCP traffic 100
Logs 145
Access Logs 146
Audit Logs 146
Export Logs 147
FTP Web Logs 148
Search Logs 146
Web Firewall Logs 50, 145

Managing Multiple Systems With Barracuda Control Center 143

Network Access Control List (ACL) 160
Network Firewall 73, 156, 180, 188

Passive Mode 68
Performance Statistics
Viewing 140
Policy Tuner 50
Preferred Clients
Exempting from Rate Control 97

Rate Control
Exempting Preferred Clients 97
Rate Control Pool
Benefits 96
Considerations 96
Scheduling algorithm 96
Recovery Mode 177
Redirect service 56
Reload/Shutdown 175
Removing the units for RMA 167
Removing units from a cluster 166
Reports 150
Aggregated for multiple appliances 143
Request Limits 61
Reverse Proxy 36

Secure Socket Layer (SSL) Encryption
Implementation 122
Session Identifiers
Extracting 95
Session Recording 176
Session Tracking 89
Signed Certificate 126
SNATs 158

SSL
Barracuda Web Application Firewall to Server 123
Client to Barracuda Web Application Firewall 123
Implementation and Configuration 123
SSL for the administration Web interface 171
Static Routes 152
Syslog
Monitor 147
System Alerts 143
System Configuration 177
System Tasks
Viewing 141

Templates 188
Threat Control Manager 192
Time Zone 170
Traffic Management 100
Trap Messages 143
Troubleshooting 174

URL ACLs 70
URL Policy 88

Viewing Health Indicators 141
Viewing Performance Statistics 140
Viewing Task Errors 141
VLAN 153
Vulnerability Assessment
Import Report 192
Management 192
Recommendations 192

Web Customization 25
Web Service Protections
See XML Firewall 132

XML Firewall 134
Import of WSDL file 134
Import Schema/WSDL 135
SOAP Validations 134
XML Validation Settings 134

Zero Day Attacks
Defined 22