

Oracle® Database Firewall

Security Management Guide

Release 5.0

E18696-06

September 2011

Oracle Database Firewall Security Management Guide, Release 5.0

E18696-06

Copyright © 2003, 2011, Oracle and/or its affiliates. All rights reserved.

Contributors: Tammy Bednar, Paul Betteridge, K. Karun, Valarie Moore, Steve Moyle, Stuart Sharp, James Spooner, James Wilson

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface	ix
Audience	ix
Documentation Accessibility	ix
Related Documents	ix
Conventions	x
1 Introducing the Oracle Database Firewall System	
Downloading the Latest Version of This Manual	1-1
About the Oracle Database Firewall System	1-1
What Is the Oracle Database Firewall System?	1-1
The Oracle Database Firewall Approach to Database Security	1-2
Oracle Database Firewall Architecture	1-2
Using Policy Files to Create Data Policies	1-4
Creating Policy Files	1-4
Using Policy Files	1-4
Oracle Database Firewall Operational Modes	1-4
How the Oracle Database Firewall Logging Feature Works	1-5
What Is the Purpose of Logging?	1-5
What Logs Does Oracle Database Firewall Maintain?	1-5
Oracle Database Firewall Applications	1-6
Oracle Database Firewall Analyzer	1-6
Oracle Database Firewall Administration Console	1-6
About the Administration Console	1-7
Types of Reports Generated from the Administration Console	1-7
Stored Procedure Auditing and User Role Auditing	1-7
Planning the Protection Level for Your Databases	1-7
2 Using the Administration Console	
About the Administration Console	2-1
Accessing the Administration Console	2-2
Who Can Log in to the Administration Console	2-2
Logging in to the Administration Console	2-2
Using the Dashboard	2-3
Parts of the Dashboard	2-3

3 Using Oracle Database Firewall Analyzer

Overview of the Oracle Database Firewall Analyzer	3-1
About the Analyzer.....	3-1
The Concept of Clustering SQL Statements in the Analyzer.....	3-1
The Process of Developing a Policy.....	3-2
Creating a Model	3-2
About Creating a Model.....	3-2
Supplying Data to Train the Analyzer	3-3
Enabling Log Unique Policies to Provide Logging Data	3-3
Creating a New Model in the Analyzer	3-4
Creating a New Model from Training on Log Data	3-4
Creating a New Model from Training on a SQL Statement File.....	3-6
Opening an Existing Model.....	3-7
Viewing and Analyzing Data in the Model	3-8
About Analyzing Data.....	3-8
The Analyzer Main Window	3-8
Elements of the Analyzer Summary Tab.....	3-9
Other Analyzer Tabs.....	3-10
Viewing Clusters by Cluster Groups	3-10
Viewing Clusters in the Analysis tab.....	3-10
Viewing Cluster Groups in the Details Tab	3-12
Viewing Data by Database Tables	3-13
Viewing Data by Database Columns	3-14
Filtering Data in the Details and Analysis Tabs	3-14
Viewing and Filtering Data in the Baseline Tab	3-15
Viewing Data by Profile	3-16
Viewing the Properties of a Model.....	3-16
Designing the Policy	3-17
About Designing the Policy	3-17
Creating a Policy Automatically	3-18
Manually Setting the Action, Logging Level, and Threat Severity.....	3-20
Managing Traffic Encrypted with Oracle Database Advanced Security Option	3-20
Creating Exceptions, Novelty Policies, and a Default Rule	3-21
About Exceptions, Novelty Policies, and the Default Rule	3-21
Creating Exceptions.....	3-21
Creating Novelty Policies	3-22
Customizing the Default Rule.....	3-24
Blocking SQL and Creating Substitute Statements	3-24
Creating Login and Logout Policies for Database Users.....	3-25
Using Profiles to Display and Set Policy Rules for Specific Data	3-26
Creating a Profile	3-27
Using Profiles in the Analysis and Details Tabs.....	3-27
Defining Sets of Factors to Use in Profiles and Exceptions.....	3-28
Creating a Policy File and Uploading it into the Database Firewall	3-29
Creating a Policy File in the Analyzer.....	3-29
Uploading and Enabling a Policy in the Database Firewall or Management Server.....	3-30
Improving and Refining the Policy with new Data	3-31

Refining the Policy Interactively	3-31
Refreshing the Analyzer with Updated Data from the Monitored Database	3-32
Analyzing the Updated Data.....	3-32
Assigning Policy Rules to the New Data and Updating Your Policy	3-33
Additional Features	3-34
Sensitive Data Masking	3-34
Exporting the Data in a Model as HTML	3-34
Creating a Model from a Policy File.....	3-34
Dividing the Screen into Two Screens	3-35

4 Auditing Stored Procedures and Roles

About Auditing Stored Procedures and Roles	4-1
Viewing and Approving Changes to Stored Procedures	4-1
About Viewing and Approving Changes to Stored Procedures.....	4-1
Running a Manual Stored Procedure Audit	4-2
Approving Changes Made to a Stored Procedure	4-2
Filtering Options for Approving Changes in Stored Procedures	4-4
Viewing and Approving Changes to User Roles	4-5
About Viewing and Approving Changes to User Roles	4-5
Running a Manual User Role Audit.....	4-6
Approving Changes Made to a User Role.....	4-6
Filtering Options for Approving Changes in User Roles.....	4-8

5 Accessing and Viewing the Traffic Log

Accessing the Traffic Log	5-1
Accessing Traffic Logs.....	5-1
Viewing Logged Traffic.....	5-2
Searching for Traffic Logs	5-2
Viewing the Log Search Results.....	5-4
Log Search Results and Scheduled Reports	5-5
Viewing the Traffic Log for Database Response Monitoring	5-5

6 Generating Oracle Database Firewall Reports

About Oracle Database Firewall Reports	6-1
Reports Generated from the Administration Console.....	6-1
Generating Reports	6-1
Generating Audit and Summary Reports	6-2
Options in the Reports Menu	6-4
Adding Your Own Reports	6-4
Scheduling Reports	6-5
How the Security Index Formula Is Calculated	6-6

Index

List of Figures

1-1	Oracle Database Firewall Architecture	1-3
2-1	The Management Server Administration Console: Dashboard Tab	2-2
3-1	Clustering into Semantically Similar Statements	3-2
3-2	Analyzer Main Window	3-9
3-3	Displaying a Cluster Group	3-10
3-4	Contents of a Cluster Group.....	3-11
3-5	Statements within a Cluster Group in the Analysis Tab.....	3-11
3-6	Finding the Percentage of Statements in a Cluster in the Analysis Tab	3-11
3-7	Indicator Showing the Percentage of Statements in the Analysis Tab.....	3-12
3-8	Threat Security Indicator	3-12
3-9	Example Data Grouped by Shape in the Details Tab	3-13
3-10	Contents of a Cluster Group in the Details Tab	3-13
3-11	Selected Tables	3-14
3-12	Tabular View of the Generated Clusters	3-15
3-13	Finding General Information About a Selected Model	3-17
3-14	Creating an Initial Policy	3-19
3-15	Changing the Action in the Details Tab.....	3-20
3-16	Cluster Properties Dialog Box.....	3-25
3-17	Login and Logout Policies for Database Users.....	3-26
3-18	Iterative Development Cycle of the Policy	3-31
3-19	Example Data Showing Additional Clusters Created	3-33
3-20	Additional Clusters Created from New Data in the Details Tab.....	3-33
3-21	Setting Up Rules for Automatic Masking of Sensitive Data.....	3-34
5-1	Accessing the Traffic Log.....	5-1
5-2	Searching for a Traffic Log	5-2
5-3	Traffic Log Search Using the AND Operator Condition.....	5-3
5-4	Traffic Log Search Using the OR Operator Condition	5-3
5-5	Traffic Log Search Using the OR and AND Conditions	5-3
5-6	Viewing Log Search Results	5-4
6-1	Audit Reports Page of the Management Server Administration Console	6-2

Preface

Welcome to *Oracle Database Firewall Security Management Guide*.

This section contains:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

Audience

This guide has been written for users who are responsible for creating Oracle Database Firewall policies and monitoring SQL statement traffic. It includes the following topics:

- An introduction to the concepts and components of Oracle Database Firewall
- Details of how to plan and design a Oracle Database Firewall system
- Procedures for creating policies and monitoring SQL statement traffic
- Information on how to get the most from your system

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Database Firewall documentation set:

- *Oracle Database Firewall Release Notes*

- *Oracle Database Firewall Installation Guide*
- *Oracle Database Firewall Administration Guide*
- *Oracle Database Firewall Licensing Information*
- Oracle Database Firewall Analyzer Online Help

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Introducing the Oracle Database Firewall System

This chapter contains:

- [Downloading the Latest Version of This Manual](#)
- [About the Oracle Database Firewall System](#)
- [Oracle Database Firewall Applications](#)
- [Planning the Protection Level for Your Databases](#)

Downloading the Latest Version of This Manual

You can download the latest version of this manual from the Oracle Database Firewall Web site, which is in the Database section of Oracle Technology Network. The URL is as follows:

<http://www.oracle.com/technetwork/indexes/documentation/index.html>

About the Oracle Database Firewall System

This section contains:

- [What Is the Oracle Database Firewall System?](#)
- [The Oracle Database Firewall Approach to Database Security](#)
- [Oracle Database Firewall Architecture](#)
- [Using Policy Files to Create Data Policies](#)
- [Oracle Database Firewall Operational Modes](#)
- [How the Oracle Database Firewall Logging Feature Works](#)

What Is the Oracle Database Firewall System?

The Oracle Database Firewall system secures and protects data in Oracle, Microsoft SQL Server, Sybase Adaptive Server Enterprise (ASE), Sybase SQL Anywhere SQL, and IBM DB2 SQL (Linux, UNIX, and Microsoft Windows) databases. It blocks attempted attacks, logs activity, and produces related warnings. It provides tools to assess vulnerabilities and enhances existing database security features, such as encryption and user authentication.

Traditional systems usually test the syntax of statements passed to the database, recognizing predefined expressions. Creating a set of rules using this technique

requires a hand-crafted approach and can be very time-consuming and complex, even for someone very knowledgeable about the database. Even if significant resources create satisfactory protection for known threats, little protection may be offered for unknown threats. The Database Firewall addresses these challenges.

The Oracle Database Firewall Approach to Database Security

The Oracle Database Firewall system works by analyzing the *meaning* of the SQL statements that database clients send to the database. This provides a much higher degree of protection than traditional database firewalls, because it does not depend on the source of an attack or recognition of syntax of known security threats.

There is no limit to the length of SQL statements that the Analyzer can analyze; it displays up to the first 2000 characters of a statements. Also, it looks at all types of SQL statements, not just product-specific SQL. For example, it looks at regular ANSI SQL as well as Oracle PL/SQL.

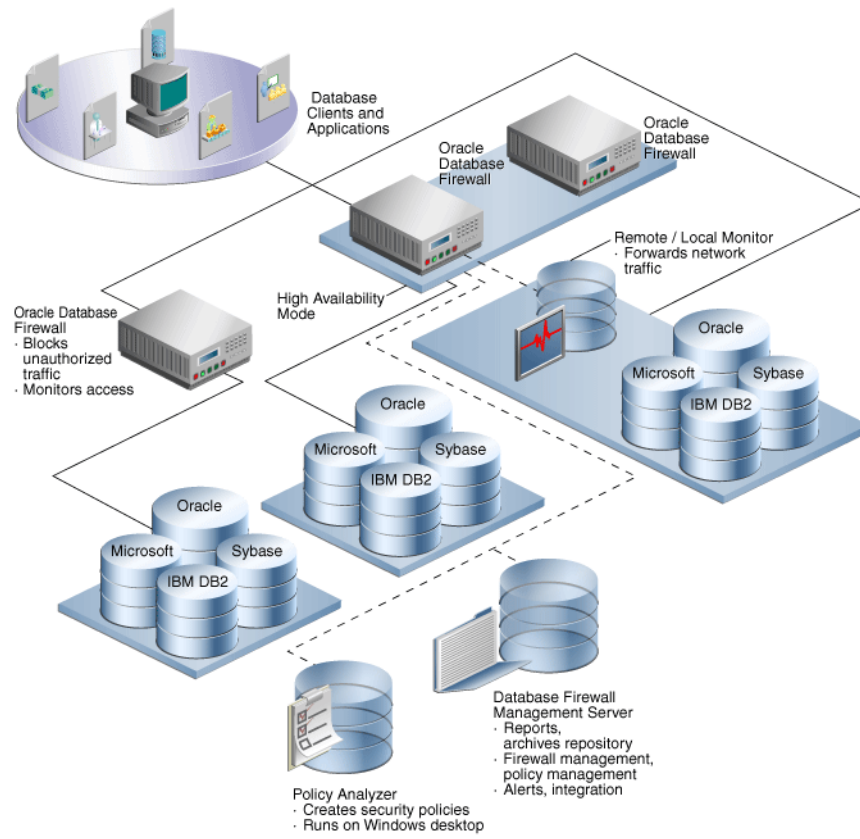
The database firewall can block previously unseen attacks (known as "zero-day" attacks), including those targeted individually against your organization. Zero-day attacks are becoming more widespread, and there is a great need to protect databases against such attacks. The database firewall also blocks blind SQL injection attacks

The Oracle Database Firewall protects the database server without affecting the performance of the database server or its client applications. The system protects from attacks originating from inside firewalls, as well as from external sources.

Oracle Database Firewall Architecture

The central feature of an Oracle Database Firewall system is the ability to scan and log SQL traffic to and from the monitored databases. The Database Firewall system scans all SQL statements passed to the databases in real time. You can configure enforcement points to monitor traffic, generate warnings of potential attacks, and block harmful statements.

[Figure 1-1](#) shows a typical deployment of Oracle Database Firewall.

Figure 1–1 Oracle Database Firewall Architecture

The main components are as follows:

- **Database clients and applications:** These represent the end users of your applications.
- **Oracle Database Firewalls:** The Database Firewall enforces data policies that you create in Oracle Database Firewall Analyzer. Oracle Database Firewall employs at least one Database Firewall for up to 20 protected databases. Each Database Firewall collects the SQL data from these databases, sends it to the Management Server, and then deletes their SQL data locally. The Database Firewall handles real-time recording and analysis of SQL transaction requests and responses from a protected database. The Database Firewall can use the remote and local monitoring features to forward network traffic from remote or local databases to Database Firewall.
- **Protected databases:** You can protect Oracle, Microsoft SQL Server, Sybase Adaptive Server Enterprise (ASE), Sybase SQL Anywhere SQL, and IBM DB2 SQL (Linux, UNIX, and Microsoft Windows) databases.
- **Oracle Database Firewall Analyzer:** You use the Analyzer to create data policies so that you can monitor your site's protected databases. See "[Oracle Database Firewall Analyzer](#)" on page 1-6 and "[Using Policy Files to Create Data Policies](#)" on page 1-4 for more information.
- **Oracle Database Firewall Management Server:** The Database Firewall Management Server manages all Database Firewalls that are connected to it. It accumulates SQL from these firewalls, stores and manages log files, provides business reports, and integrates with third-party applications as needed. You, as the administrator responsible for managing policies, use the Management Server

to upload policies, set the monitoring mode, and so on. See ["Oracle Database Firewall Administration Console"](#) on page 1-6 for more information.

Using Policy Files to Create Data Policies

A **policy file** is a set of rules the Database Firewall uses when it monitors SQL traffic to a database. Using a policy file, the Database Firewall compares incoming SQL statements and determines what actions to take when intercepting a SQL statement. Actions can be allowing, blocking, or producing a warning for each SQL statement that the Database Firewall passes to the database. The policy file can produce an alert when a specific database user logs in or out, and block database users who make a specified number of unsuccessful logins attempts.

You can use Oracle Database Firewall Analyzer to customize the policy file to the exact needs of the protected database, detecting any unusual system activity, from both inside or outside of the organization. The policy file can, for example, generate a warning when previously unseen classes of SQL statements occur, or block injected SQL statements that attempt to delete or access information in specified tables.

Creating Policy Files

You use the Analyzer to create an initial policy file from SQL statements that the Database Firewall logs while monitoring normal database traffic. The Database Firewall system automatically trains itself with knowledge of the protected database. This process produces a set of logged SQL statements that reflect actual usage of the database to provide a *model* of normal operation. This model stores all the settings used to create a policy and the results of testing logged SQL statements against the previous set of data. The Analyzer then generates a policy from these logged statements.

See Also: ["Oracle Database Firewall Operational Modes"](#) on page 1-4 for further information about Training Mode

The Analyzer also can generate a policy file from a train file or a server trace file. See ["Using Oracle Database Firewall Analyzer"](#) on page 3-1 for further information.

In all cases, you can further customize the policy file to the specific requirements of the site and the database being protected.

Using Policy Files

The Analyzer uses its knowledge of the SQL language to group the logged SQL statements into **clusters**, defined by similar semantics. The rules to allow, block, or produce a warning for a statement are defined not at the statement level, but at the cluster level, which may encompass many different individual statements.

Oracle Database Firewall Operational Modes

Depending on operational needs (which are reflected in the design of the policy), a Database Firewall can operate in either a monitoring or blocking mode. See ["Planning the Protection Level for Your Databases"](#) on page 1-7 for more information.

How the Oracle Database Firewall Logging Feature Works

This section contains:

- [What Is the Purpose of Logging?](#)
- [What Logs Does Oracle Database Firewall Maintain?](#)

What Is the Purpose of Logging?

You can use the information logged by a Database Firewall for the following purposes:

- **System monitoring and report generation:** Enables you to monitor the system for possible attacks and then generate reports based on these findings.
- **Analysis and comparison against the data used to create the policy:** Enables the Analyzer to detect behavior changes and improve the policy.

The process of analyzing new log data can expose new and undesirable features of database client applications, such as unexpected permission changes, which may occur over time. In this way, a Database Firewall can help to maintain focus on security over the entire life of the database.

- **Independent logging of database activity for audit or compliance purposes:** Logged information can, for example, be used for forensic analysis to determine how a database was accessed and by whom.

The Database Firewall can perform targeted logging, such as only logging statements that match specific clusters or only logging statements that do not match previously seen clusters. Targeted logging reduces log storage requirements and ensures that only required information is saved. The logging rules are stored in the policy.

Note: For informational reasons, you should always enable logging for actions that you want to block.

A Database Firewall digitally signs all logged data to indicate the authenticity and origin of the data.

What Logs Does Oracle Database Firewall Maintain?

A Database Firewall maintains three logs:

- **Traffic Log:** Stores all SQL statements and database login and logout events that the policy requires.

Each logged statement and event can include a set of attributes that provide additional information about the originator, including:

- The database user login name
- The IP address of the database client
- The user's operating system login name
- The name of the client program

If the information about the originator is not available from the SQL traffic directly, a direct database interrogation (DDI) feature enables a Database Firewall to query the database to obtain the information. DDI can be enabled or disabled as required. DDI can only be used with Microsoft SQL Server and Sybase SQL Anywhere.

See Also: *Oracle Database Firewall Administration Guide* for further information

In addition, you can enable a database response monitoring feature, which stores all responses that the protected database makes to SQL statements and login and logout requests, in the traffic log.

- **Event Log:** Stores system events that are not directly related to the Database Firewall software, such as operating system warnings.
- **Administration Log:** Stores the login ID of any user who changes configurations for system actions such as shutdowns, restarts, and policy uploads, in the Administration Console.

Oracle Database Firewall Applications

This section describes the Database Firewall applications that you will use to create Database Firewall policies.

- [Oracle Database Firewall Analyzer](#)
- [Oracle Database Firewall Administration Console](#)

Oracle Database Firewall Analyzer

You use the Oracle Database Firewall Analyzer to create the policy that the Database Firewalls use to block, alert, log or permit SQL statements for the database. The Analyzer does this by reading logs that the Database Firewalls create. It enables users who have little knowledge of SQL to develop policies automatically, while enabling users who have detailed knowledge of SQL to customize policies.

As part of the process of developing a policy, the Analyzer tests the logged data against previous sets of data. This enables you to identify new potential threats and to further improve the policy.

Oracle Database Firewall Analyzer is installed on a Microsoft Windows client computer, and uses secure communications to the Database Firewall or Database Firewall Management Server.

See [Chapter 3, "Using Oracle Database Firewall Analyzer,"](#) for details on using the analyzer and creating policies.

Note: Oracle Database Firewall Analyzer has Online Help that you can access using the **F1** key.

Oracle Database Firewall Administration Console

This section contains:

- [About the Administration Console](#)
- [Types of Reports Generated from the Administration Console](#)
- [Stored Procedure Auditing and User Role Auditing](#)

About the Administration Console

The Administration Console is a Web browser-based application for configuring, managing, and monitoring the system. You display it by logging into a Database Firewall or Database Firewall Management Server from a Web browser.

See [Chapter 2, "Using the Administration Console,"](#) for more information on using the Administration Console.

Types of Reports Generated from the Administration Console

You can produce a variety of different types of reports directly from the Administration Console. The reports can be generated and displayed as Adobe Acrobat PDF documents or Microsoft Excel spreadsheets.

You can schedule reports to run automatically at defined intervals, such as every day, week, or month. Scheduled reports are automatically forwarded to a nominated e-mail address.

See [Chapter 6, "Generating Oracle Database Firewall Reports,"](#) for details about Oracle Database Firewall reports.

Stored Procedure Auditing and User Role Auditing

Stored procedure auditing and user role auditing are part of the Database Firewall reporting system.

- Stored procedure audits record any changes or additions to stored procedures on a specified database server.
- User role audits record any new or changed user roles. The user role audits capture information about nested roles as well as directly-granted roles.

You can use options in the Administration Console to audit and approve changes to stored procedures and user roles in the databases on a specified database server.

See *Oracle Database Firewall Administration Guide* for information about configuring stored procedure and user role auditing.

See [Chapter 4, "Auditing Stored Procedures and Roles,"](#) for information about viewing and approving changes to stored procedure and user role auditing

Planning the Protection Level for Your Databases

Depending on operational needs, Database Firewall can operate in one of the following modes:

- **Database Activity Monitoring (DAM):** The system detects and logs unusual activity, and produces warnings, but does not block potential threats. This is useful during the early stages of deployment while you are developing and refining a policy. It is also known as monitoring mode.
- **Database Policy Enforcement (DPE):** The system performs all the actions of database activity monitoring and blocks potential attacks. It is also known as blocking mode.

These operations can continue independently of the Analyzer or other Database Firewall applications. For example, one Database Firewall can simultaneously monitor one protected database while blocking another.

Consider what you want Oracle Database Firewall to achieve for you. Do you want:

- Audit logging?

- Warnings of potential attacks?
- Blocking of potential attacks?

In general, implementing only audit logging requires the least up-front development time to create a satisfactory policy. SQL statement blocking requires the most development time, but provides the greatest protection.

One strategy for deployment is to start with audit logging only, and then deploy a policy file that can protect the database against potential attacks at a later date. This helps you to become familiar with the deployment.

If you want to block SQL statements eventually, then using Database Activity Monitoring (DAM) can build confidence before you deploy the system fully.

When DAM mode is set, policies can include block action levels, *but* statements with specified action levels pass straight through. Syslog events and reports show the statements as blocked, while in reality, the statements passed through normally. This enables you to evaluate the system in a live environment before you switch on statement blocking. During this evaluation phase, you can change the policy to modify the system responses.

Using the Administration Console

This chapter contains:

- [About the Administration Console](#)
- [Accessing the Administration Console](#)
- [Using the Dashboard](#)

About the Administration Console

The Administration Console is a Web browser-based application for configuring, managing, and monitoring the system. You display it by logging into a Database Firewall or Database Firewall Management Server from a Web browser.

The Administration Console provides access to the following variations of Oracle Database Firewall:

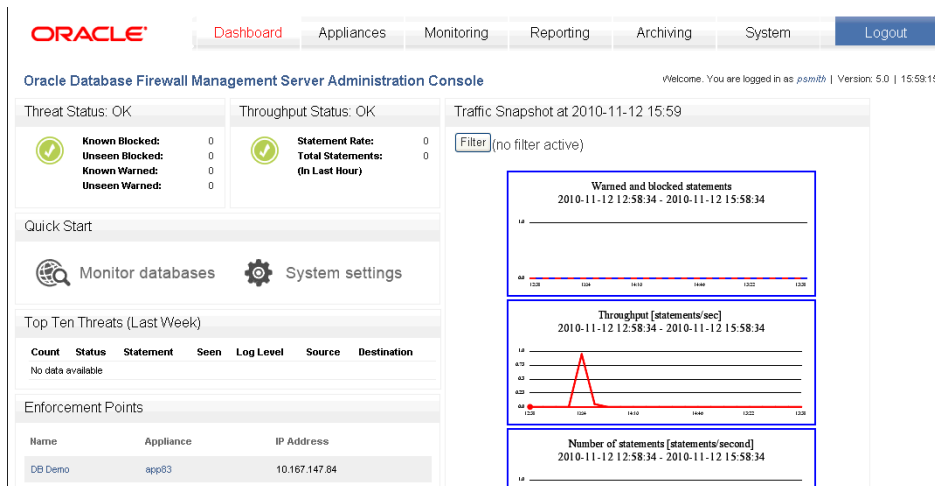
- **A Database Firewall Management Server:** Manages one or more Database Firewalls.
- **A managed Database Firewall:** A Database Firewall that has been configured to be managed by a Management Server.
- **A standalone Database Firewall:** This is a Database Firewall that operates independently, that is, it is not managed by a Database Firewall. In most cases, you will configure it to be a managed Database Firewall.

For a full list of the tasks that you can perform with each of these variations, see *Oracle Database Firewall Administration Guide*.

As a user responsible for policy management, you will use the Administration Console to quickly find high level information about the database you must protect, generate and manage reports, and audit SQL database stored procedures and user roles. The Administration Console is also used by network or system administrators responsible for IT systems deployment, maintenance, and monitoring.

[Figure 2-1](#) shows the Dashboard tab of the Management Server Administration Console.

Figure 2–1 The Management Server Administration Console: Dashboard Tab



Accessing the Administration Console

This section contains:

- [Who Can Log in to the Administration Console](#)
- [Logging in to the Administration Console](#)

Who Can Log in to the Administration Console

All users of the Administration Console must enter a valid login ID and password before access is granted. The following user roles are available:

- **System Administrator:** This user controls the entire Database Firewall system. The default user `admin`, created when you install Database Firewall, has this role.
- **Log Administrator:** This user is responsible for archiving the traffic logs.
- **View-only User:** This user can run reports but cannot make changes to policies or other settings.

A user who has been granted the System Administrator role can use the Administration Console to create and manage user accounts with these roles. (Note that these user accounts are not stored in the database.)

Because the Administration Console is a browser-based application, you can use it from any computer that has a supported Web browser, although access can be restricted by IP address.

For better security and separation of duty, you should assign these roles to trusted users and only use the `admin` user account as a back-up account. See *Oracle Database Administrator's Guide* for more information about configuring users.

Logging in to the Administration Console

To log in to the Administration console:

1. Open a Web browser from any computer that has network access to Oracle Database Firewall.
2. Enter the following URL:

`https://ip_address/user/login`

Provide the IP address for the server on which Oracle Database Firewall is installed. For example:

```
https://192.0.2.206/user/login
```

If you change the user interface port number (by using the System Settings page of the Administration Console), then you must also include this port number in the URL. Use the following syntax:

```
https://ip_address:port/user/login
```

For example:

```
https://192.0.2.206:444/user/login
```

Add this address to your **Favorites** to make it easy to access.

See *Oracle Database Firewall Administration Guide* for information about changing the Administration Console port number.

3. If you are prompted to choose a digital certificate, click **OK**.
4. If you see a message claiming that there is a problem with the Web site security certificate, then click the **Continue to this website** link.
5. In the Login page, enter the user name and password for an account that has System Administrator privileges
6. Click **Login**.

Using the Dashboard

When you are connected to a Database Firewall Management Server, the Administration Console includes the Dashboard tab. (See [Figure 2-1](#) on page 2-2.) The Dashboard provides a high-level view of important information about the databases being protected, such as the threat status, throughput, and top ten threats. Charts display key indicators for viewing by IT and security managers responsible for day-to-day monitoring of the system.

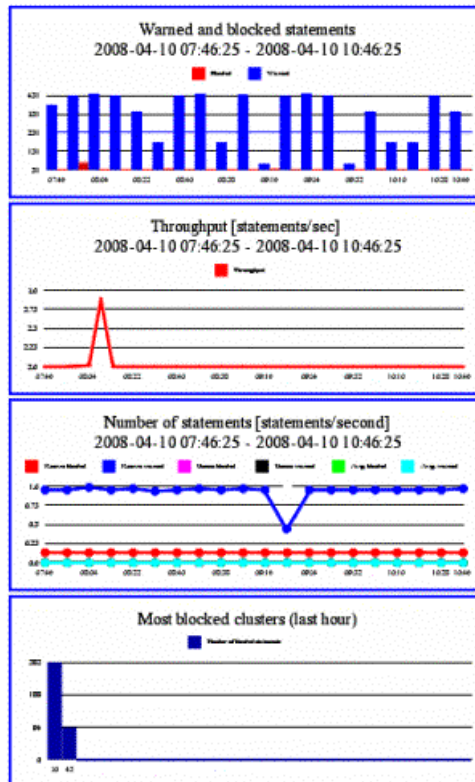
The Dashboard also provides **Quick Start** options that allow you to set up the system configuration settings with ease.

Parts of the Dashboard

The Dashboard contains the following sections:

- **Threat Status:** Provides statistics about the number of statements that have been blocked or caused a warning. Separate counts are provided for known and anomaly statements; unseen statements are those that match none of the clusters in the baseline policy.
- **Throughput Status:** Gives the number of statements per second and the total number of statements in the last hour.
- **Quick Start:** Provides wizards that help you to configure your system quickly and easily.
- **Top Ten Threats:** Lists the most significant threats over the indicated period of time.

- **Enforcement Points:** Gives details of the enforcement points configured in the Administration Console.
- **Traffic Snapshot:** Provides statistics about the performance of Oracle Database Firewall and the actions it has taken. Security managers who are responsible for day-to-day monitoring of the system may want to view this information at frequent intervals. The following is an example.



These examples are described in order as follows:

- Shows the number of SQL statements that were blocked or caused a warning over the last three hours. Clicking the chart zooms in.
- Shows the number of SQL statements processed per second over the last three hours. Clicking the chart zooms in.
- Shows by statement class, the number of SQL statements processed per second over the last three hours. Clicking the chart zooms in.
- Shows the SQL cluster IDs that were most blocked in the last hour. Clicking the chart displays additional information.

Note: When you zoom in, Oracle Database Firewall displays controls that enable you to zoom in further and navigate along the horizontal axis.

A **Filter** button is provided, which you can use to filter the displayed information. If required, you can apply more than one filter. The operators are self-explanatory, except for the following:

>= (greater than or equal to) <= (less than or equal to) <> (not equal to)

Using Oracle Database Firewall Analyzer

This section contains:

- [Overview of the Oracle Database Firewall Analyzer](#)
- [Creating a Model](#)
- [Viewing and Analyzing Data in the Model](#)
- [Designing the Policy](#)
- [Creating a Policy File and Uploading it into the Database Firewall](#)
- [Improving and Refining the Policy with new Data](#)
- [Additional Features](#)

Overview of the Oracle Database Firewall Analyzer

This section contains:

- [About the Analyzer](#)
- [The Concept of Clustering SQL Statements in the Analyzer](#)
- [The Process of Developing a Policy](#)

About the Analyzer

Oracle Database Firewall Analyzer enables you to design policies quickly and efficiently. Successful deployment of a Database Firewall system depends on an effective policy. Policy rules can depend on any combination of the SQL statement type, time of day, name of the database user, IP address of the database client, operating system user name, client program name, or any exceptions you specify.

Developing a policy is an iterative process that keeps refining and improving the policy with new data.

Note: The Oracle Database Firewall Analyzer has extensive Online Help available by pressing the F1 key.

The Concept of Clustering SQL Statements in the Analyzer

Clustering is an important tool that the Analyzer uses to categorize the SQL statements it reads into sets of semantically similar statements called clusters.

The Analyzer further groups clusters into a cluster group if they have the same SQL grammar pattern. [Figure 3–1](#) shows a cluster group of nine clusters that match the pattern `select <column> from <table>`.

Figure 3–1 Clustering into Semantically Similar Statements

```
select * from scott.dept
select * from scott.emp
select * from session_roles
select * from dual
select * from hr.salary
select * from sum$
select * from scott.bonus
select * from hr.company
select * from hr.employee
```

When you develop a policy, you specify the actions that the Oracle Database Firewall should take for each cluster, rather than for each individual SQL statement.

The Process of Developing a Policy

Developing a policy consists of these main steps:

1. Create a model in the Analyzer to use for designing your policy. The model is created by training the Analyzer using log data from Database Firewall or by uploading a file of SQL statements. See ["Creating a Model"](#) on page 3-2.
2. Analyze the data in the model. See ["Viewing and Analyzing Data in the Model"](#) on page 3-8.
3. Design your policy by setting policy actions and rules. See ["Designing the Policy"](#) on page 3-17.
4. Create a policy file in the Analyzer, upload this file into the Database Firewall, and select this policy in a configured Enforcement Point. See ["Creating a Policy File and Uploading it into the Database Firewall"](#) on page 3-29.
5. Refine the policy with new data in the Analyzer. See ["Improving and Refining the Policy with new Data"](#) on page 3-31

Creating a Model

This section contains:

- [About Creating a Model](#)
- [Supplying Data to Train the Analyzer](#)
- [Creating a New Model in the Analyzer](#)
- [Opening an Existing Model](#)

About Creating a Model

A model is a file that stores data needed to create a policy. You can supply data for a model by "training" the Analyzer in two ways:

- Using traffic log data from the Database Firewall
- Uploading a SQL statements file

You can create any number of models. Typically, you create different models for different databases or for different analyses of the same database.

Creating a new model results in two files, *filename.smdl* and *filename.smdl_data*. If you want to open a model in a different Analyzer installation, then you must copy both files to the computer that has the Analyzer software.

Supplying Data to Train the Analyzer

You can supply logged data or SQL statement files to the Analyzer in one of these ways:

- **Directly from the traffic log of the Management Server or a standalone Database Firewall:** This is the recommended method of supplying log data.

Use the Management Server or standalone Database Firewall Administration Console to enable log unique policies while the system monitors normal day-to-day database traffic. See ["Enabling Log Unique Policies to Provide Logging Data"](#) on page 3-3.

Log unique policies enable you to log statements for offline analysis that include each distinct source of SQL traffic. Be aware that if you apply this policy, even though it stores fewer statements than if you had chosen to log all statements, it can still use a significant amount of storage for the logged data.

Log unique policies log SQL traffic specifically for developing a new policy. The logged data enables the Analyzer to understand how client applications use the database and enables rapid development of a policy that reflects actual use of the database and its client applications.

- **From a Database Firewall train file:** This is a text file generally written by a developer. It contains a list of SQL statements, one line for each statement. See the Online Help for the required syntax.
- **(Microsoft SQL Server Only) From a Server trace file:** A binary log file created on Microsoft SQL Server. It contains a list of SQL statements.

The database that you are monitoring must be running, but you do not need to configure any additional settings for it, such as enabling auditing.

Enabling Log Unique Policies to Provide Logging Data

To enable log unique policies:

1. Log in to the standalone Database Firewall or Management Server Administration Console.
See ["Logging in to the Administration Console"](#) on page 2-2 for more information.
2. In the Login page, enter the user name and password for the user who has been granted the System Administrator role. Then click **Login**.
3. Select the **Monitoring** tab.
4. From the **Enforcement Points** menu, select **List**.
This menu item should be selected by default.
5. In the Enforcement Points page, find the enforcement point for the database whose data you want to analyze and then select the **Settings** button.
6. In the Monitoring Settings page, scroll down to the Policy area.
7. In the Policy area, select the **unique.dna** option.

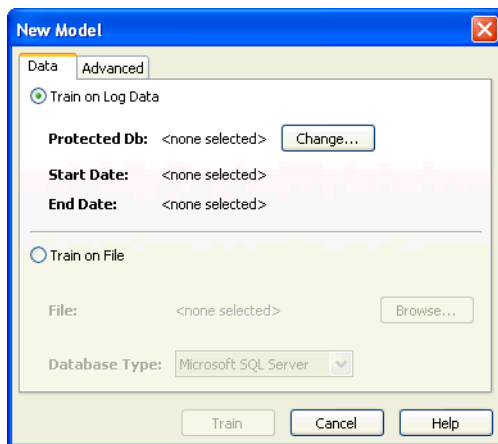
8. Click the **Save** button.
The new setting takes effect immediately.

Creating a New Model in the Analyzer

To create a new model:

1. On your desktop, double-click the **Oracle Database Firewall Analyzer** icon.
If you have not previously selected **Don't show this screen on startup**, the Welcome to Oracle Database Firewall Analyzer page appears.
2. Do one of the following:
 - From the Welcome page, select **Create a New Model from Training**.
 - From the **File** menu, select **New**.

The New Model dialog box appears.

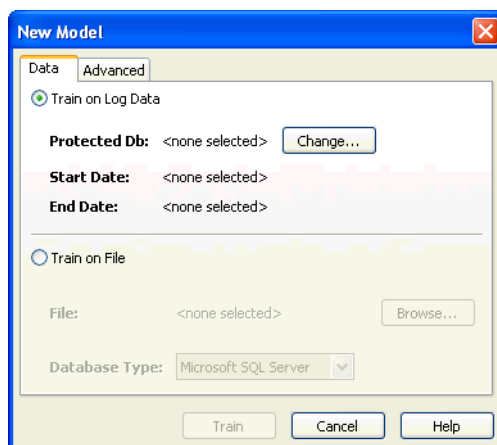


3. Select one of the following:
 - **Train on Log Data** - to provide training data from the Data Base Firewall traffic log. See "[Creating a New Model from Training on Log Data](#)" on page 3-4.
 - **Train on File** - to upload a training file containing SQL statements. See "[Creating a New Model from Training on a SQL Statement File](#)" on page 3-6.

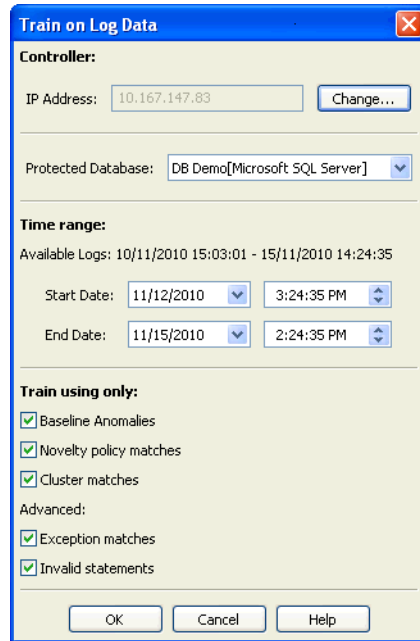
Creating a New Model from Training on Log Data

To create a new model from training on logged data:

1. Follow the steps in "[Creating a New Model in the Analyzer](#)" on page 3-4 and select **Train on Log Data** in Step 3.



2. Click **Change**.
3. In the Traffic Log Server dialog box, enter the IP address (and port number, if necessary), username, and password of the Oracle Database Firewall or Management Server system administrator, and then click **OK**.
4. In the Train on Log Data dialog box, select the following:
 - a. In the **Protected Database** menu, select a named database to retrieve its data.
If your databases are all of the same type (for example, Oracle), you also have the option to select **All Databases** to retrieve data logged for all your databases. Choose **All Databases** only if you are intending to use the same policy for all of them.
Protected databases must be those set up by your System Administrator.
 - b. Specify the date range of the log data to read.
By the default, the end date is today's date.
 - c. Select the types of logged statements to import or all check boxes to import all statements. Press **F1** to see Online Help for descriptions of various options, if needed.



5. Click **OK** in the Train on Log Data dialog box.
The New Model dialog appears.
6. Ensure that **Train on Log Data** is selected, and then click **Train**.
The model appears in the main Analyzer window.
7. Select **Save** or **Save As** in the **File** menu to save the model to the hard disk of your computer.
The Analyzer creates two files using the name you provide, with the extensions `.smdl` and `.smdl_data`.

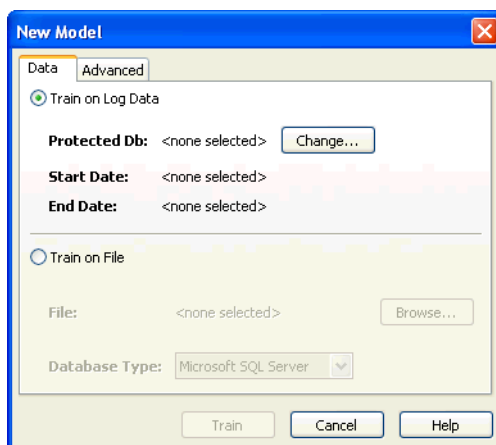
In a moment, the main window of the Analyzer appears, similar to [Figure 3-2](#) on page 3-9.

Note: If you do not see links under the Statement Class Distribution, Threat Severity Distribution, and Action Distribution areas, then resize the application window.

Creating a New Model from Training on a SQL Statement File

To create a new model from training on a SQL statement file:

1. Follow the steps in "[Creating a New Model in the Analyzer](#)" on page 3-4, and select **Train on File** in Step 3.



2. Click **Browse** and select the file you want to use.
3. Select the correct **Database Type** from the list.
4. Ensure that **Train on File** is selected, and then click **Train**.

The main Analyzer window appears, with the name of the .train or .trc file at the top to indicate that the model is based on this file.

5. Select **Save** or **Save As** in the **File** menu to save the model to the hard disk of your computer.

The Analyzer creates two files using the name of the base file, with the extensions .smdl and .smdl_data.

In a moment, the main window of the Analyzer appears, similar to [Figure 3-2](#) on page 3-9.

Note: If you do not see links under the Statement Class Distribution, Threat Severity Distribution, and Action Distribution areas, then resize the application window.

Opening an Existing Model

To open an existing model:

1. Start the Analyzer.
2. In the Welcome to Oracle Database Firewall Analyzer window, select **Open a previously saved Model**.
3. In the Open dialog box, navigate to the directory where you saved the previous model file.

The model has the file extension .smdl. (You cannot open the .smdl_data file; it is only used to store the data of the model.)

4. Select the model and then click **Open**.

The model appears in the Analyzer, with the Summary page displayed.

If you do not see links under the Statement Class Distribution, Threat Severity Distribution, and Action Distribution areas, then resize the application window.

Viewing and Analyzing Data in the Model

This section contains:

- [About Analyzing Data](#)
- [The Analyzer Main Window](#)
- [Viewing Clusters by Cluster Groups](#)
- [Viewing Data by Database Tables](#)
- [Viewing Data by Database Columns](#)
- [Filtering Data in the Details and Analysis Tabs](#)
- [Viewing and Filtering Data in the Baseline Tab](#)
- [Viewing Data by Profile](#)
- [Viewing the Properties of a Model](#)

About Analyzing Data

After you have created a model by providing training data to the Analyzer, you can use it to analyze the SQL statements in that data.

This section describes various ways of viewing the data in the Analyzer by using different tabs and viewing options.

You can analyze the data before or after assigning policy rules (See "[Designing the Policy](#)" on page 3-17). Before assigning policy rules, the Analyzer will give you information on the training data you have provided, allowing you to filter it by statement type, database table, and column. After you assign policy rules, you will have more filtering options based on those policy rules.

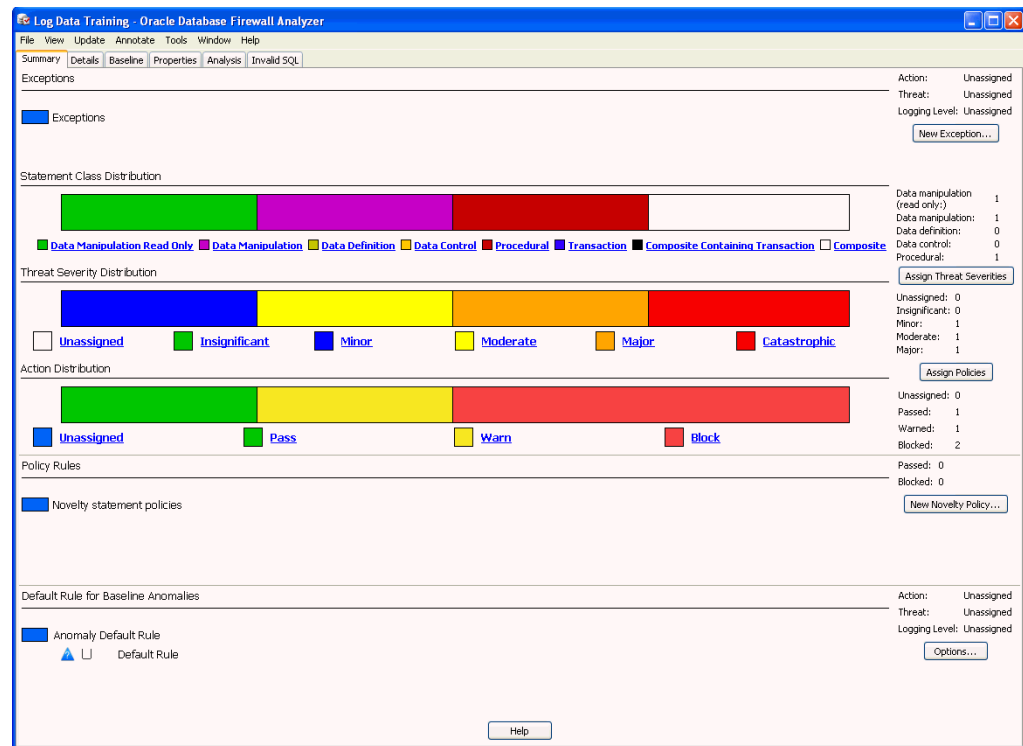
The Analyzer Main Window

The **Summary** tab is the main interface of the Analyzer. The remaining tabs provide additional settings that you can modify, and analysis and information that you can view. These tabs are listed below and are discussed in this chapter.

Note: If you do not see links under the bar charts, resize your application.

[Figure 3–2](#) shows the main window of the Analyzer.

Figure 3–2 Analyzer Main Window



Elements of the Analyzer Summary Tab

The **Summary** tab shows a graphical representation of the policy rules that are being applied to the statement types (clusters) being currently analyzed, as well as exceptions and other rules that may apply. This tab enables you to generate a policy automatically, set novelty policies, and filter the information that appears in the **Details** tab.

The **Summary** tab is divided into these areas:

- **Exceptions** - Lists exceptions you have created. The rules that you have assigned to clusters in the model will not apply to these exceptions. You can specify one rule to be applied to all the exceptions listed here.
- **Graphical Distribution of Statements** - Bar charts display the distribution of clusters in terms of different statement classes, threat severities, and action levels currently set in the policy.

When you click a link under a bar chart, the **Details** tab displays a subset of clusters based on that link. For example if you click **Pass** under Action Distribution, the **Details** tab displays the statement clusters that are set to Pass in this policy.

- **Novelty Policies** - Lists special policies you have created for specific statement classes and/or specific tables in your protected database.
- **Default Rule** - Shows the default rule for any statement anomalies that are not covered by the rules set for clusters seen in the model, Exceptions, or Novelty Policies.

Other Analyzer Tabs

In addition to the **Summary** tab, the Analyzer main window contains the following tabs.

Note: The tabs and the menus are described in detail in the Online Help.

- **Details:** Provides a different way to view the SQL data by organizing clusters into cluster groups. It enables you to customize the policy manually.
- **Baseline:** An alternative to using the **Details** tab. It displays clusters in a tabular format and shows the attributes of each policy. This tab also provides multiple ways of filtering clusters and enables you to customize the policy manually.
- **Properties:** Contains general information about the model, such as the original data sources for the model, statistics, change control information, and notes.
- **Analysis:** Enables you to analyze the SQL statements the Analyzer has scanned.
- **Invalid SQL:** Displays any SQL statements that the Analyzer did not recognize, such as statements that do not conform to the SQL syntax.

Viewing Clusters by Cluster Groups

A cluster group is a set of clusters grouped by the Analyzer according to statement meaning, for example, `select <column> from <table>`. Statements that match this pattern, but that may have different values for the column and table, will be in this cluster group.

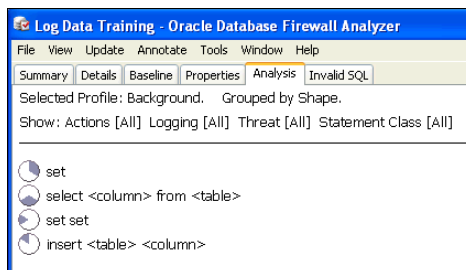
Viewing Clusters in the Analysis tab

To view clusters by cluster groups in the **Analysis** tab:

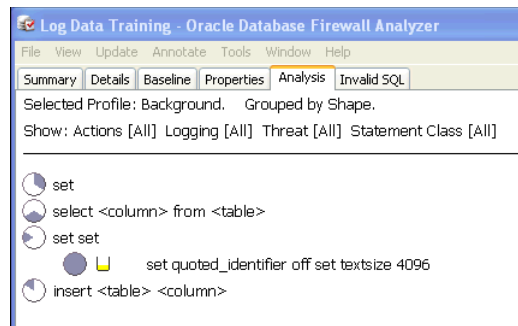
1. In the Analyzer, click the **Analysis** tab.
2. From the **View** menu, select **Group by Shape**.

A hierarchical view of the SQL data appears. At the top level, the Analyzer organizes all the clusters it has defined into a number of cluster groups. [Figure 3–3](#) shows four different cluster groups.

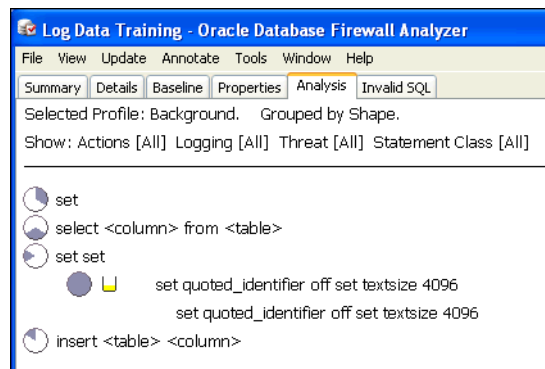
Figure 3–3 *Displaying a Cluster Group*



Double-clicking a cluster group reveals the clusters it contains. [Figure 3–4](#) shows an example of the contents of a cluster group.

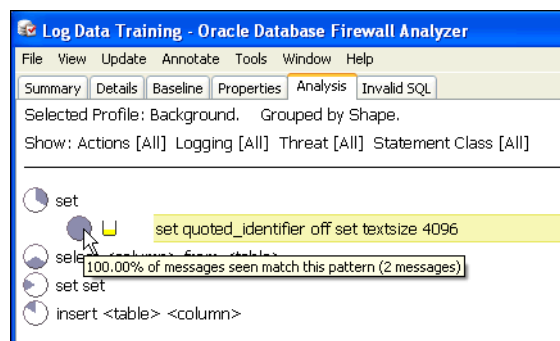
Figure 3–4 Contents of a Cluster Group

Double-clicking this cluster reveals all statements in the cluster. [Figure 3–5](#) shows a statement within a cluster group. Cluster groups can contain multiple statements.

Figure 3–5 Statements within a Cluster Group in the Analysis Tab

Cluster Indicators in the Analysis Tab

The **Cluster** indicator on the left side of a cluster (a pie-shaped icon) shows the percentage of statements in the cluster group that are in the cluster. Positioning the mouse pointer on the indicator gives the percentage to two decimal places, as shown in [Figure 3–6](#). The tooltip also shows that there are two statements (messages) in the cluster.

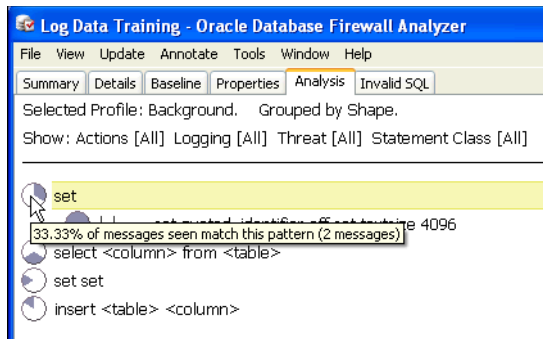
Figure 3–6 Finding the Percentage of Statements in a Cluster in the Analysis Tab

The clusters in a cluster group are ordered by percentage.

Cluster Group Indicator

The **Cluster Group** indicator on the left side of a cluster group shows the percentage of statements in the model that are in the cluster group. [Figure 3-7](#) shows that the cluster group accounts for 33.33 percent of all statements in the model.

Figure 3-7 Indicator Showing the Percentage of Statements in the Analysis Tab

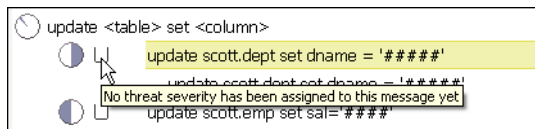


The cluster groups are ordered by percentage.

Threat Severity Indicator in the Analysis Tab

The **Threat Severity** indicator for the cluster highlighted below shows the threat severity. The indicator is a vessel-shaped icon that varies from empty to full. [Figure 3-8](#) shows an empty vessel indicator, with a message that no threat severity has been assigned yet.

Figure 3-8 Threat Security Indicator



This example also shows sensitive data, which is masked with #### symbols:














```
update scott.dept set dname = '####'
```

See "[Sensitive Data Masking](#)" on page 3-34 for further information.

Viewing Cluster Groups in the Details Tab

The **Details** tab provides the same **Group by Shape**, **Table**, or **Column** views that are available for the **Analysis** tab. [Figure 3-9](#) shows sample data displayed when **Group by Shape** is selected in the **Details** tab.

Figure 3–9 Example Data Grouped by Shape in the Details Tab

	select <column> from <table>
	update <table> set <column>
	select <function> from <table>
	dbms_application_info.set_module
	begin dbms_application_info.set_module end
	dbms_output.disable
	begin dbms_output.disable end
	select <function> from <table>
	commit
	commit
	update <table> set <column>
	update scott.dept set dname = '#####'
	update scott.emp set sal='####'











The data is organized in a similar way as in the **Analysis** tab. At the top level are the cluster groups. Each cluster group has an indicator, which shows the proportion of its clusters that have each action level. For example, at the end of the list in [Figure 3–9](#), the update <table> set <column> icon indicates that 50 percent of the clusters in the cluster group have an "Unassigned" action level (blue), and 50 percent have a "Warn" action level (yellow).

Oracle Database Firewall has masked sensitive data in these two clusters. See ["Sensitive Data Masking"](#) on page 3-34.

Double-click a cluster group to see the clusters it contains. The indicators next to a cluster show the currently selected action and threat severity.

The statement shown at the cluster level is an example of a statement in the cluster. [Figure 3–10](#) shows an example of contents of a cluster group.

Figure 3–10 Contents of a Cluster Group in the Details Tab

	select <column> from <table>
	select * from scott.dept
	select * from scott.emp
	select * from session_roles
	select * from dual
	select * from hr.salary
	select * from sum\$
	select * from scott.bonus
	select * from hr.company
	select * from hr.employee

Viewing Data by Database Tables

The Analyzer can organize the data by selected database tables.

To view data by database tables:

1. In the Analyzer, click the **Analysis** tab.
2. From the **View** menu, select **Group by Table**.

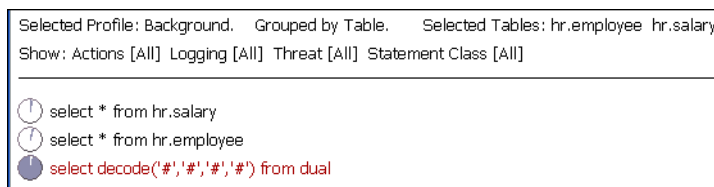
The Select Tables dialog appears automatically the first time you select **Group by Table**. Select the tables that you want to display, and then click **OK**.



3. To select different tables, from the **View** menu, select **Change Tables**, select tables, and then click **OK**.

Figure 3–11 shows that two tables have been selected: **hr.employee** and **hr.salary**.

Figure 3–11 Selected Tables



The SQL statements are grouped according to table. The statement `select decode('#','#','#') from dual` is for statements that do not refer to either of these selected tables.

Viewing Data by Database Columns

The **Group by Column** view is similar to **Group by Table**, described under "[Viewing Data by Database Tables](#)" on page 3-13, but the Analyzer organizes the data by selected columns.

To view data by database column:

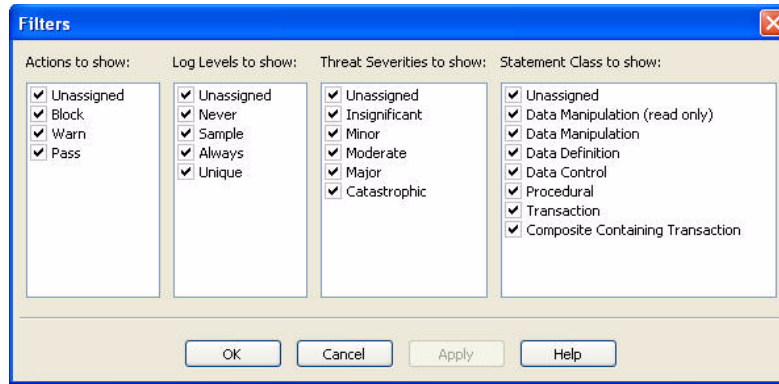
1. In the Analyzer, click the **Analysis** tab.
2. From the **View** menu, select **Group by Column**.
3. Select the columns you want to use to filter data, and then click **OK**.
4. To select different columns, from the **View** menu, select **Change Columns**, select columns, and then click **OK**.

Filtering Data in the Details and Analysis Tabs

To show specific types of clusters in the **Details** or **Analysis** tabs:

1. In the Analyzer Details or Analysis tab, from the Tools menu, select **Filters**.

The Filters dialog is displayed:



2. Select which clusters to show by selecting the **Actions**, **Log Levels**, **Threat Severities**, and/or **Statement Classes** for the clusters.

For example, if you select the action **Warn**, and the logging levels **Sample** and **Always**, only those clusters that have a **Warn** action and either a **Sample** or **Always** logging level are displayed.

Viewing and Filtering Data in the Baseline Tab

The Analyzer **Baseline** tab provides a tabular view of the data. Figure 3–12 shows a portion of this tab.

Figure 3–12 Tabular View of the Generated Clusters

Summary Details Baseline Properties Analysis Invalid SQL						
Block						
Id	Action	Logging	Threat	Statement	Count	
130857805	Unassigned	Unassigned	Unassigned	select * from sum\$	1	10
202666050	Block	Always	Major	select * from scott.bonus	1	10
211048546	Pass	Never	Insignificant	select * from scott.dept	2	10
314680859	Warn	Unique	Moderate	select * from hr.company	1	10
317744773	Pass	Never	Unassigned	begin dbms_application_info.set_module	5	10
643745415	Pass	Never	Unassigned	begin dbms_application_info.set_module	5	10
1084188303	Block	Always	Major	update scott.dept set dname = '#####'	1	10
1120310753	Unassigned	Unassigned	Unassigned	update ecott.emp set comm = 0	1	10
1228321332	Unassigned	Unassigned	Unassigned	update scott.comp set comm = 0	2	10
1353347896	Block	Always	Major	select * from hr.salary	1	10
1536480440	Unassigned	Unassigned	Unassigned	delete from scott.emp where empno = '9'	1	10

The **Baseline** tab displays one row per cluster and has multiple columns that can be filtered. You can filter data by:

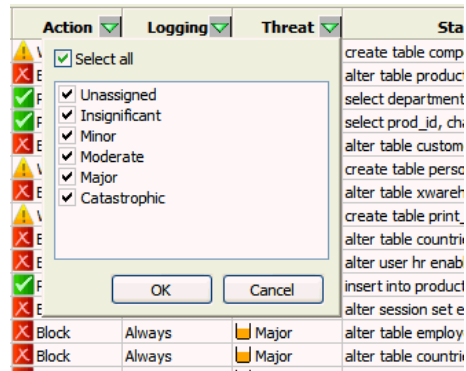
- Action
- Logging level
- Threat level
- IP address
- Tables
- Columns
- Users
- Statement type
- Client program

- OS users

To use the Baseline tab filters:

1. In the Analyzer, click the Baseline tab.
2. Click the filter icon (a down-arrow) at the top of any column that has a filter.

A filter dialog box similar to this appears:



Viewing Data by Profile

A Profile allows you to define a set of characteristics to use in filtering statements. See ["Using Profiles to Display and Set Policy Rules for Specific Data"](#) on page 3-26. If you have created Profiles, you can view data by profile.

To view data by profile:

1. In the Analyzer, click the **Details**, **Baseline**, or **Analysis** tab.
2. From the **View** menu, select **Profile**, or **Change Profile** if you have previously selected one.
3. In the Select Profile dialog, select the Profile you want to view, and then click **OK**.
4. To return to viewing all data, from the **View** menu, select **Background**.

Viewing the Properties of a Model

The **Properties** tab contains general information about the selected model, such as the original source of the data for the model, statistics, change control information, and notes. If you enter data in any of the fields, such as **Database** or **DB Location**, the Analyzer saves this information when you save the model. [Figure 3-13](#) shows general information about a selected model.

Figure 3–13 Finding General Information About a Selected Model

The screenshot displays the Oracle Database Firewall Analyzer interface for a selected model named 'Log Data Training'. The interface includes a menu bar (File, View, Update, Annotate, Tools, Window, Help) and a tabbed view with 'Summary', 'Details', 'Baseline', 'Properties', 'Analysis', and 'Invalid SQL' tabs. The 'Summary' tab is active, showing the following information:

- Model Info:** Model Name: Log Data Training; Database: [text box]; DB Location: [text box].
- Data Sources:** Database Type: Microsoft SQL Server; Training Data: Log Data From: 10.167.147.83, Protected Database: DB Demo, Timerange: 12/11/2010 15:24 - 15/11/2010 14:24.
- Model Data Summary:**
 - Actions:** Passed: 25.00%, Warned: 50.00%, Blocked: 25.00%, Unassigned: 0.00%
 - Logging:** Never: 25.00%, Sample: 0.00%, Always: 75.00%, Unique: 0.00%, Unassigned: 0.00%
 - Training Data:** Total Clusters: 4, Number of Statements: 6, Invalid Statements: 0
- Security Index:**
 - Data Manipulation: 36.00%, Data Definition: 0.00%, Data Control: 0.00%
 - Procedural: 24.00%, Transaction: 0.00%
 - Total: 60.00%
- Change Control Information:**
 - Version: [text box], Date: 11/16/2010 [dropdown], Author: phuey [text box]
 - Change Notes: [text box]
 - Status: Work In Progress [dropdown]
- Other Notes:** [text area]

A 'Help' button is located at the bottom center of the interface.

Designing the Policy

This section contains:

- [About Designing the Policy](#)
- [Creating a Policy Automatically](#)
- [Manually Setting the Action, Logging Level, and Threat Severity](#)
- [Blocking SQL and Creating Substitute Statements](#)
- [Creating Login and Logout Policies for Database Users](#)
- [Creating Exceptions, Novelty Policies, and a Default Rule](#)
- [Using Profiles to Display and Set Policy Rules for Specific Data](#)
- [Defining Sets of Factors to Use in Profiles and Exceptions](#)

About Designing the Policy

To successfully deploy an Oracle Database Firewall system you must develop an effective policy. Using the Analyzer you can design and refine a policy efficiently in minimum time.

Note: In blocking mode, by default the Database Firewall blocks all IPv6 traffic regardless of the policies in place.

Designing a policy includes:

- Specifying these settings for each cluster in the model:

- **Action level:** Whether or not Oracle Database Firewall permits, blocks, or produces a warning when it encounters a statement that matches the cluster.
- **Logging level:** Whether Oracle Database Firewall never logs, logs all statements, or logs statements that have a unique combination of cluster, source IP address, database username, operating system username, and client program name. (See the Online Help for further information.) You can use logging as an independent record of database activity, which may, for example, be used for future audit or forensic purposes

Consider the amount of logging carefully, because increasing the data logged directly impacts required disk space. The frequency for the sample logging is every tenth statement for the cluster.

Oracle recommends that you use log unique policies or the initial policy because it guarantees one of each type. It efficiently samples traffic without logging all statements.

- **Threat Severity:** The anticipated threat from statements in a cluster. There are six threat severity settings, ranging from Unassigned (vessel empty) to Catastrophic (vessel filled bright red). When Oracle Database Firewall logs a statement, the threat severity of the statement is also logged. You can use third-party reports and syslogs to display SQL statements based on the logged threat severity.

You can let the Analyzer automatically assign these settings when you provide the training data for a model, then adjust the settings as needed. See "[Creating a Policy Automatically](#)" on page 3-18.

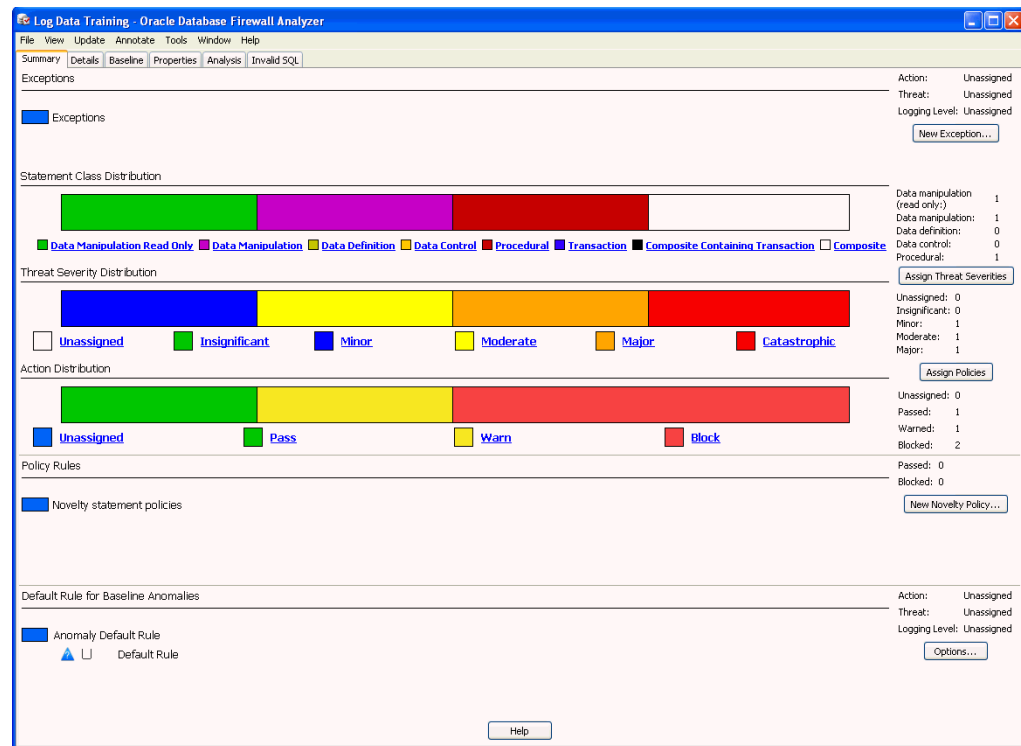
- Creating Exceptions to the policy settings
- Adding Novelty Policies (or rules) that are triggered when specific statement types are encountered and/or selected tables are called
- Creating Profiles to filter data and set policy rules based on specific criteria (such as client IP address)

Creating a Policy Automatically

You create an initial policy automatically from the **Summary** tab.

[Figure 3-14](#) shows a partial view of the **Summary** tab window.

Figure 3–14 Creating an Initial Policy



The **Summary** tab provides the primary interaction with the policy. From the **Summary** tab, you can generate a policy automatically and view distribution charts of statement classes, threat severities, and action levels currently in the policy. Using the links that follow the charts, you can filter the contents of the **Details** tab while manually customizing the policy (for example, to display only policies with an Unassigned action level). An unassigned statement is a SQL statement that you have not yet categorized (for example, assigned it a threat level).

To create an initial policy:

1. Select the **Summary** tab.
2. On the right side of the Summary page, select the following buttons:
 - **Assign Threat Severities:** This setting automatically assigns a threat severity to each cluster that has an "Unassigned" threat severity. The threat severity assigned is based on the perceived risks.
 - **Assign Policies:** This setting automatically assigns logging and action levels to each cluster based on its threat severity. (See the Online Help.)

Users who have view-only privileges can create a policy but they cannot apply it to a Database Firewall. Only Database Firewall system administrators can upload and apply new policy to a Database Firewall.

3. Optionally, set the following functionality:
 - **Exceptions**
 - **New Novelty Policy**
 - **Options**

See "Creating Exceptions, Novelty Policies, and a Default Rule" on page 3-21.

After you create an initial policy, you can customize the policy using the **Details** or **Baseline** tab.

Manually Setting the Action, Logging Level, and Threat Severity

You can set or change the action, logging level, and threat severity of a cluster from the right-click menu in the **Details** or **Baseline** tab. The following shows an example of the right-click menu in the **Details** tab.

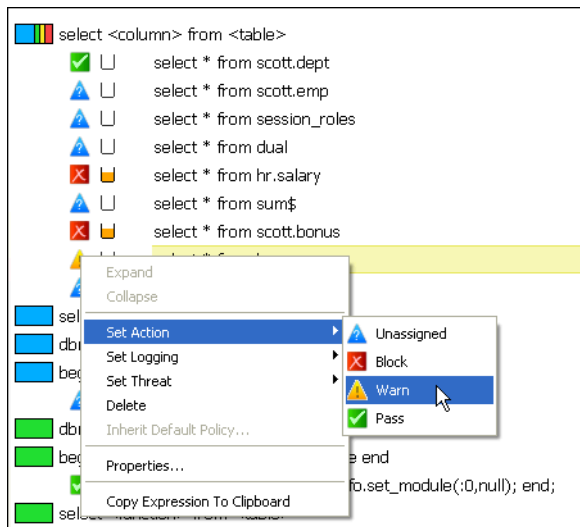
To set or change the Action, Logging, or Threat Severity for a cluster:

1. In the **Details** or **Baseline** tab, right-click a cluster.

Note: You can select several clusters in the **Baseline** tab by selecting the first cluster, and then pressing the **Ctrl** key to select more clusters.
2. In the **Set Action**, **Set Logging**, or **Set Threat** sub-menus, select the desired setting.

Figure 3–15 shows the right-click sub-menu for setting the action.

Figure 3–15 Changing the Action in the Details Tab



See Also: ["Oracle Database Firewall Operational Modes"](#) on page 1-4 for a complete description of Database Policy Enforcement (DPE)

Managing Traffic Encrypted with Oracle Database Advanced Security Option

Oracle Database provides the Advanced Security Option (ASO). For Oracle databases configured to use ASO, this option automatically encrypts network traffic. Oracle Database Firewall indicates the presence of ASO-encrypted network traffic in the reports and enables you to block this traffic through a policy.

If the Database Firewall encounters ASO-encrypted traffic, it enters the following string into the log file:

```
extracted_from_protocol encrypted
```

After you create or refresh a model from an ASO-enabled database that has been generating ASO traffic (see ["Creating a New Model from Training on Log Data"](#) on page 3-4), a cluster with the text extracted from protocol encrypted appears in the

list of other clusters generated in the model. From here, you can create policies on this cluster, such as setting a threat level or blocking it.

See Also: *Oracle Database Advanced Security Administrator's Guide* for detailed information about ASO

Creating Exceptions, Novelty Policies, and a Default Rule

This section contains:

- [About Exceptions, Novelty Policies, and the Default Rule](#)
- [Creating Exceptions](#)
- [Creating Novelty Policies](#)
- [Customizing the Default Rule](#)

About Exceptions, Novelty Policies, and the Default Rule

You create Exceptions and Novelty Policies to set rules for specific conditions and types of statements that occur in your database traffic.

You customize the Default Rule to handle any statement that is an anomaly, and therefore, not covered by any of your policy settings, Exceptions, or Novelty Policies.

Creating Exceptions

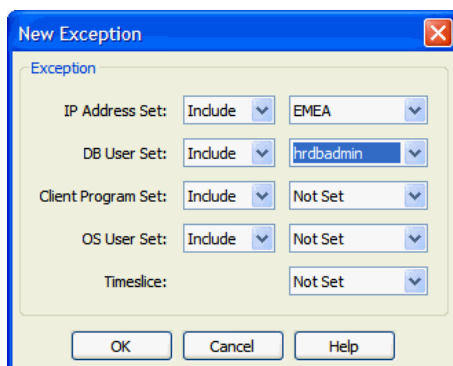
An exception determines the action, logging level, and threat severity to use when certain session data is encountered. For example, an exception could specify rules for statements that occur during specific times of the day, or originate (or do not originate) from selected client IP addresses or user names.

Exceptions override all other policy rules. For example, you may want to override standard policy rules if SQL statements originate from an administrator, or if they originate from anywhere other than a specific IP address.

In order to create an Exception, you must already have defined the sets of factors to be used in defining it. See "[Defining Sets of Factors to Use in Profiles and Exceptions](#)" on page 3-28.

To create an Exception:

1. In the Analyzer **Summary** tab, click **New Exception**.
2. In the **New Exception** dialog, select to include or exclude sets in your Exception. All of the criteria for sets in this exception must be met for the rules to take effect.

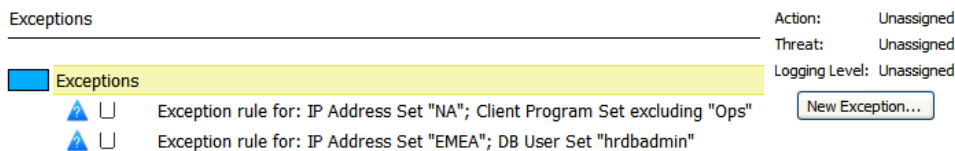


In the above example, you are specifying that this Exception applies to statements originating from the EMEA IP Address Set AND from the hrdbadmin DB User Set. If you were to **Exclude** hrdbadmin in DB User Set, the exception would apply to statements originating from EMEA IP address AND *not* originating from the hradmin DB User Set.

3. Click **OK**.

The Exception appears in **Summary** tab at the top. In the example below, there are two exceptions:

- The first applies to statements originating from the NA IP Address set and that are *not* originating from the Ops Client Program Set.
- The second applies to statements originating from the EMEA IP Address Set and that are from users in the hrdbadmin DB User Set.



4. In the **Summary** tab, right-click the Exceptions (above your list of Exceptions).
5. Set the **Action**, **Logging**, and **Threat** levels that apply if ANY of your Exceptions apply.

The icons next to all your Exceptions change depending on your settings.

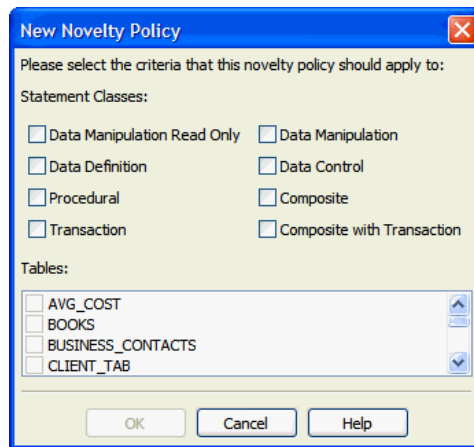
Creating Novelty Policies

Novelty policies specify the action, logging level, and threat severity to use for specific types of statements and/or statements that operate on selected tables. Novelty policies can be used to loosen or tighten your normal policy rules if certain statements are encountered.

For example, if the normal policy action for a certain statement type is Warn, you may want to set up a novelty policy that applies a Pass action if this statement type operates on tables containing public information. Alternatively, you may want to set up a novelty policy that blocks all statements that operate on tables containing sensitive information.

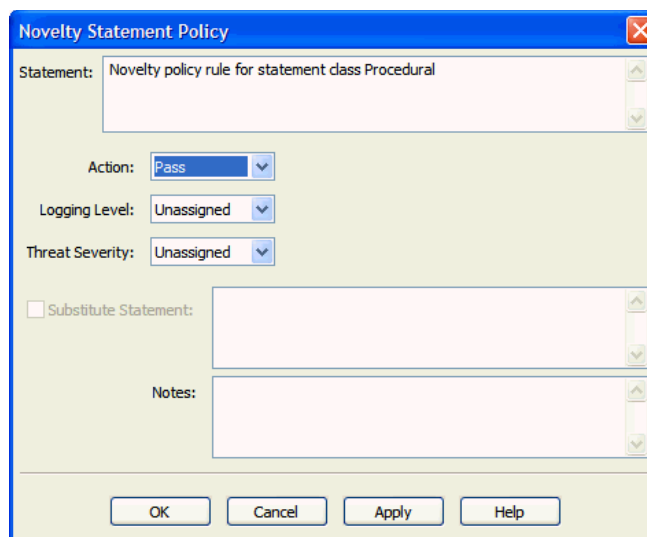
To Create a Novelty Policy:

1. In the Analyzer **Summary** tab, click **New Novelty Policy**.
2. In the **New Novelty Policy** dialog, select one or more classes of statements and/or one or more database tables that will trigger this policy.



3. Click **OK**.
4. In the **Summary** tab, under **Novelty statement policies**, right-click the new Novelty Policy, and select **Properties**.

The Novelty Statement Policy dialog appears:

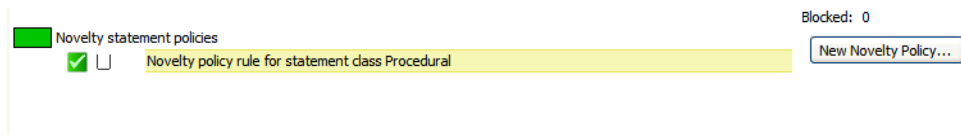


5. Set the following to define the rules for this policy:
 - **Action, Logging Level, and Threat Severity**
 - Optionally select **Substitute Statement** and enter a statement that will be substituted if the specified type of statement in this Novelty Policy is encountered. Make sure when providing a substitute statement that the statement can be handled by your client applications.

Note: A Novelty Policy may only have a Pass or Block action. Where the action is Pass, a statement will be passed only if all of it triggers the Novelty Policy rule. Where the action is Block, a statement will be blocked if any part of it triggers the Novelty Policy rule.

6. Click **OK**.

Your new Novelty Policy is listed in the Summary tab, with icons matching your settings.



If a statement matches more than one Novelty Policy, the worst-case policy is used. For example, a policy that blocks takes priority over a policy that warns.

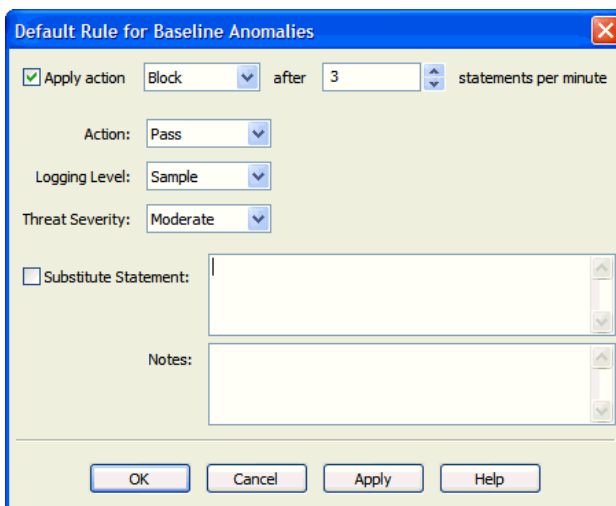
Customizing the Default Rule

For statement anomalies (that is, statements that do not fall into any of your other policy rules), the Analyzer lets you specify the default settings for logging, threshold action rest time, syntax, and case sensitivity.

To Customize the Default Rule:

1. In the Analyzer **Summary** tab, in the Default Rule for Baseline Anomalies section, right-click the Default Rule, and select **Properties**.

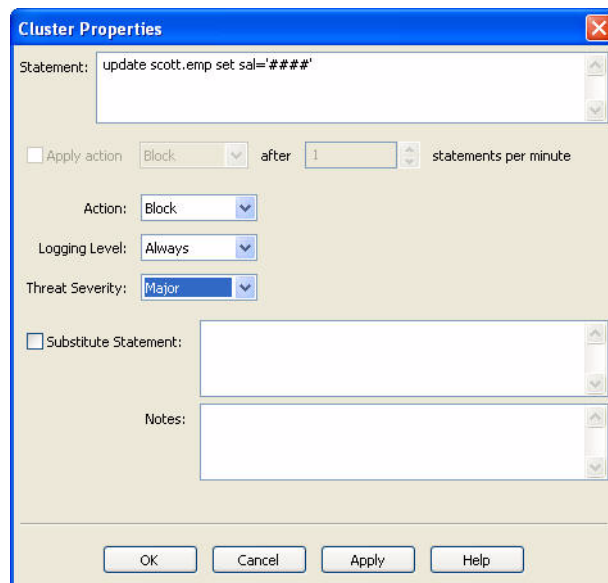
The Default Rules for Baseline Anomalies dialog appears.



2. Set the following:
 - **Action, Logging Level, Threat Severity**
 - (Optional) To apply a different action after a certain number of anomalous statements per minute are encountered, select the **Apply action** check box at the top, and set the counter to the number of **statements per minute**.
 - Optionally, check **Substitute Statement** and enter a substitute statement when this default rule is triggered. Be careful to write a statement that can be handled by your client applications.
3. Click **OK**.

Blocking SQL and Creating Substitute Statements

You can find and change the properties of a cluster listed in the **Baseline** tab by right-clicking the cluster, and selecting **Properties** from the menu.

Figure 3–16 Cluster Properties Dialog Box

This dialog box provides the following additional features:

- **Blocking SQL statements or producing warnings:** You can choose to block the SQL statement or produce a warning if a statement that matches the selected cluster occurs more frequently than a specified number of times in one minute. Remember that you should always enable logging for blocked statements.
- **Creating substitute SQL statements:** In Database Policy Enforcement (DPE) mode only, you can define a **Substitute Statement** for any cluster that has a blocked action. A substitute statement may be necessary to ensure that the database client is presented with an appropriate error message or response.

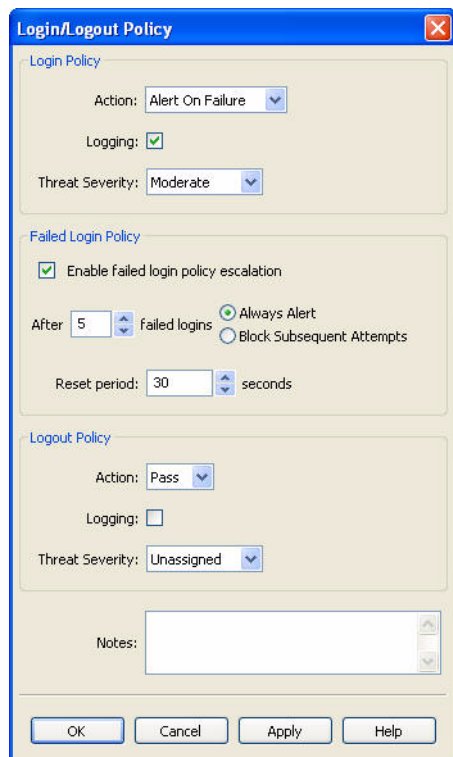
The following is an example of a good substitute statement that you can use for an Oracle database, one that is harmless and does not return any values or affect performance.

```
SELECT 100 FROM DUAL
```

Creating Login and Logout Policies for Database Users

You can use **Login/Logout Policy** from the **Tools** menu to specify the login and logout policies for database users. Login and Logout policies send alerts when the policy has been violated. This is useful in the case of automated attacks on the database.

[Figure 3–17](#) shows the Login/Logout Policy dialog box.

Figure 3–17 Login and Logout Policies for Database Users

The Login/Logout Policy dialog box contains three sections:

- **Login Policy:** Specify the action level and threat severity to use for successful or unsuccessful database user logins, and whether or not to log logins.
- **Failed Login Policy:** Specify the failed login policy, for example, to block a client or generate an alert after a specified number of consecutive unsuccessful logins (an "alert" being a "warn" action level). If triggered, login blocking continues for the specified **Reset period**; after this period, the database client can attempt to log in again.
- **Logout Policy:** Specify the action level and threat severity to use for database user logouts, and whether or not to log logouts.

Using Profiles to Display and Set Policy Rules for Specific Data

A profile is a type of filter that can display data in various ways and to set up policy rules for specific database users, IP addresses, operating system users, client programs, and times of day.

You can, for example, decide to create a profile that allows you to set up different policy rules for a certain set of database users who access the database during the night. When a user in the set accesses the database during the specified time, the profile policy rules are used and override the standard "background" rules.

A profile is any combination of the following sets of factors that are used as filters (see ["Defining Sets of Factors to Use in Profiles and Exceptions"](#) on page 3-28).

- **IP addresses**
- **Database user login names**
- **Client Program names** (for example, SQL*Plus)

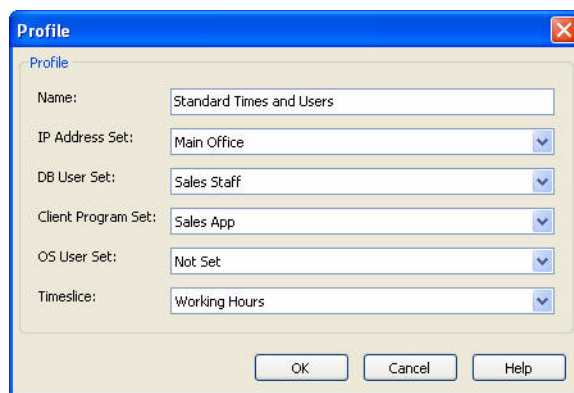
- **Operating System user names**
- **Timeslice** (for example, 9 a.m. to 5 p.m., Monday through Friday)

Creating a Profile

In order to create a profile, there must be sets of factors defined to use for filtering purposes. See ["Defining Sets of Factors to Use in Profiles and Exceptions"](#) on page 3-28.

To create a Profile:

1. From the **Tools** menu, select **Profiles**.
2. In the Profiles dialog box, select **Add**.
3. In the Profile dialog box, enter the following settings:
 - **Name:** Enter a name for the profile.
 - **IP Address Set:** From the list, from the available IP address sets, or leave it at **Not Set**.
 - **DB User Set:** From the list, select from the available database user sets, or leave it at **Not Set**.
 - **Client Program Set:** From the list, select from the available client program sets, or leave it at **Not Set**.
 - **OS User Set:** From the list, select from the available operating system user sets, or leave it at **Not Set**.
 - **Timeslice:** From the list, select from the available timeslices, or leave it at **Not Set**.



4. Click **OK**.

Using Profiles in the Analysis and Details Tabs

When the **Analysis** tab is displayed, the first time you select **Profile** from the **View** menu, you are prompted to select a profile. Subsequently, you can change the profile by using **View, Change Profile**.

The **Analysis** tab then displays only those clusters with SQL statements that have originated from the sources and times that match the selected profile. If, for example, the profile includes only a database user set, the **Analysis** tab displays only those clusters with SQL statements that have originated from the database users in the DB user set. If the profile includes both a database user set and a timeslice, then the **Analysis** tab only displays clusters with statements from that user set, and from that timeslice.

When viewing a profile, you can set up policy rules for that profile. These override the background rules. Note that with a profile selected, you still can change the background action level of a cluster by right-clicking.

In the **Details** tab, selecting a profile does not change the clusters displayed. You will still see all clusters, however, if you have selected a profile, you can set up policy rules for both that profile and all other statements (background rules).

A SQL statement can match more than one profile. In this case, Oracle Database Firewall uses the most severe action, logging level, and threat severity of all matching profiles.

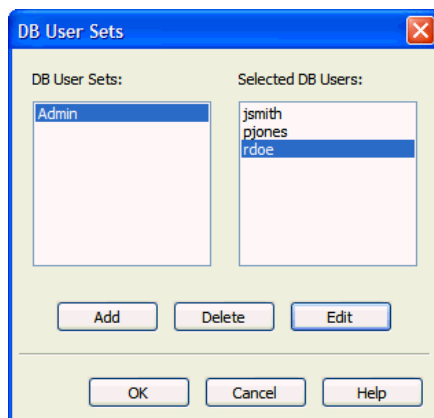
Defining Sets of Factors to Use in Profiles and Exceptions

Sets are used in defining the filters used in Profiles and Exceptions. Profiles and Exceptions are defined using the following sets:

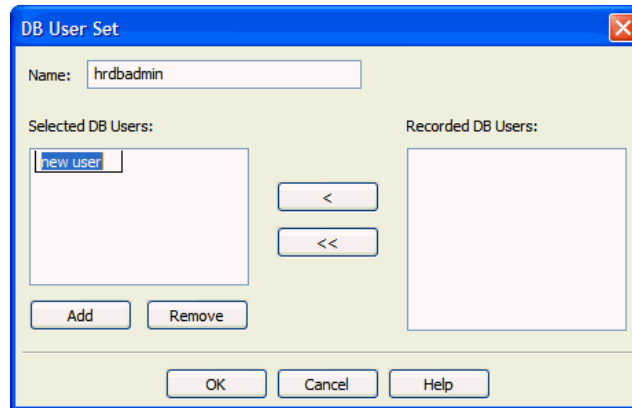
- **IP address set:** A specified list of IP addresses of database clients
- **DB user set:** A specified list of database user login names
- **Client Program set:** A specified list of client programs, for example SQL*Plus.
- **OS User set:** A specified list of operating system user names
- **Timeslice:** A specified set of hours in a week. For example, a timeslice may be 9 a.m. to 5 p.m., Monday through Friday. Time is based on the Database Firewall that you are using to monitor the database.

To define sets:

1. From the **Tools** menu, select from the following options:
 - **IP Address Sets**
 - **DB User Sets**
 - **Client Program Sets**
 - **OS User Sets**
 - **Timeslices**
2. In the dialog that appears (for example, DB User Sets), click **Add** to create a new set.



3. In the next dialog box, enter the set name in the **Name** field, for example, hrdbadmin.



4. Do one or both of the following:
 - From the **Recorded** list on the right, select the items you want, and then click the left angle bracket to move them to the **Selected** list.
 - To specify a new member of the set, click **Add** and type the name of the new item in the field in the **Selected** list.

You can use two wild cards for members of a set (but not for IP addresses): a question mark (?), which matches a single character, and an asterisk (*), which matches any number of characters.

Repeat this step as necessary to add items to this set.
5. Click **OK**, and then click **OK** again.
6. Repeat these steps for as many sets as necessary. You can then use these sets in defining Profiles or Exceptions.

Creating a Policy File and Uploading it into the Database Firewall

After you have designed a policy, you must save the policy to a file and then deploy this file to a Database Firewall.

This section contains:

- [Creating a Policy File in the Analyzer](#)
- [Uploading and Enabling a Policy in the Database Firewall or Management Server](#)

Creating a Policy File in the Analyzer

To create a policy file in the Analyzer:

1. In the Analyzer, from the **File** menu, select **Create Policy**.
2. In the Create Policy dialog box, navigate to the location of your Database Firewall model files (assuming that you want to keep the models and policies together).
3. Enter a file name and then click **Save**.

The name of the current training model is offered by default, with the file extension `.dna`. Oracle Database Firewall saves the policy `.dna` file in the `sm1` directory by default.

Uploading and Enabling a Policy in the Database Firewall or Management Server

To upload and enable a policy in the Database Firewall or Management Server:

1. Log into the Standalone Database Firewall or Management Server Administration Console and select the **Monitoring** tab.
2. In the **Policies** menu, select **Upload**.

The Upload Policy page appears.

3. In the Upload Policy page, do the following:
 - a. Click the **Browse** button to find the correct policy file (extension .dna).
 - b. Optionally, in the **Description** field, enter a description for the policy.
 - c. Click the **Save** button.

The Policies page notifies you that the policy has been uploaded, and displays a list of the currently uploaded policies. From here, you can edit or delete the policy if you want.

Policy	Created	Database Type	Description
sales_db_policy.dna	2010-11-17	Microsoft SQL Server	Policies for the sales database

4. To enable the policy:

- a. From the **Monitoring** tab, under the **Enforcement Points** menu, select the **List** button.
- b. Select the **Settings** button for the enforcement point that you want to use for the policy.
- c. In the Monitoring Settings page, scroll to the Policy area.
The policy that you uploaded is listed with the default policies.
- d. Select the policy that you uploaded.
- e. Click the **Save** button.

If you are using the Database Firewall Management Server, it automatically distributes the policy to the appropriate Oracle Database Firewalls.

To list preconfigured and uploaded policies:

1. In the Database Firewall or Management Server Administration Console, select the **Monitoring** tab.

2. In the Policies menu, select **List**.

The page displays preconfigured policies at the top and any policies that have been uploaded.

Preconfigured policies are those that are supplied by default. You cannot delete preconfigured policies.

Policies that are listed with a blue background are currently used by an enforcement point.

3. (Optional) From this page:

To delete an uploaded policy, click **Delete**.

To edit a policy's description, click **Edit**.

Improving and Refining the Policy with new Data

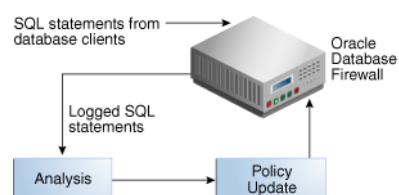
This section contains:

- [Refining the Policy Interactively](#)
- [Refreshing the Analyzer with Updated Data from the Monitored Database](#)
- [Analyzing the Updated Data](#)
- [Assigning Policy Rules to the New Data and Updating Your Policy](#)

Refining the Policy Interactively

You can refine the policy with new data at any time. [Figure 3–18](#) illustrates how the development of the policy is an iterative process.

Figure 3–18 Iterative Development Cycle of the Policy



The iterative policy development cycle is as follows:

1. Log unique policies are enabled in order to collect SQL traffic for the policy. See ["Enabling Log Unique Policies to Provide Logging Data"](#) on page 3-3.
2. The Database Firewall logs SQL statements according to the initial policy used, for example, the Log Unique policies or the policy you have developed.
3. The Analyzer allocates the new statements to the appropriate clusters, and when necessary, creates new clusters. If new clusters have been created, action and logging levels for these should be assigned, either automatically or manually.
New session factors (such as client IP addresses, database or OS user names, or client application names) should also be allocated to appropriate sets, and Profile definitions updated accordingly. See ["Defining Sets of Factors to Use in Profiles and Exceptions"](#) on page 3-28.
4. After you have made modifications, you can deploy the policy.

Refreshing the Analyzer with Updated Data from the Monitored Database

Development of the policy is an iterative process. Use unique log policies after you have deployed the initial policy. This enables Oracle Database Firewall to log new SQL statements, which you then can import into the Analyzer for analysis against the statements used to build the current policy.

Unique log policies also enable you to detect policy anomalies. This way, you can identify possible security vulnerabilities and to improve the policy further. You can repeat this process as many times as required.

To refresh the Analyzer with updated data from the monitored database:

1. From the **Update** menu, select **Update with Log Data**.
2. In the Traffic Log Server dialog box, enter the IP address of the Database Firewall and the credentials of a valid Database Firewall system administrator.
The default IP address and credentials are offered.
3. In the Update with Log Data dialog box, specify the appropriate settings and then click **OK**.
See Step 4 under ["Creating a New Model from Training on Log Data"](#) on page 3-4 for more information about these settings.

Alternatively, in Step 1, you can also select:

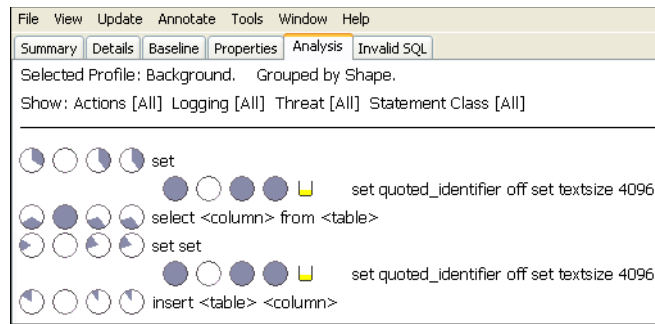
- **Update with File** to test the model against data from a train or trace file
- **Test Single Statement** to test the model against a single SQL statement

Analyzing the Updated Data

After you have refreshed the Analyzer with updated data from the monitored database, the Analyzer reads each SQL statement in the updated data and assigns it to a cluster for you to analyze in the **Details**, **Baseline**, and **Analysis** tabs.

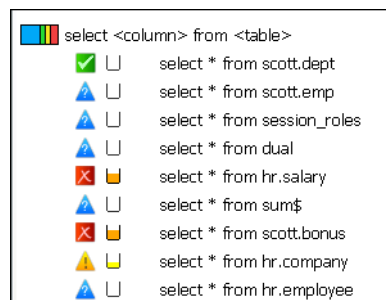
Some statements in the new data may generate additional clusters. These can be easily identified in the **Details** and **Baseline** tabs, because they have the default "Unassigned" action icon, which is triangle that contains a question mark.

[Figure 3-19](#) shows an example of the results displayed when using **Group by Shapes** in the **Analysis** tab.

Figure 3–19 Example Data Showing Additional Clusters Created

In this example, additional sets of pie-shaped indicators appear. The first indicator on each line (on the left) describes the statements in the original data used to create the model. The second indicator describes the statements in the first set of test data. A new set of indicators appears each time you test the model. See "[Cluster Indicators in the Analysis Tab](#)" on page 3-11.

[Figure 3–20](#) shows an example of additional clusters that have been created from new data in the **Details** tab.

Figure 3–20 Additional Clusters Created from New Data in the Details Tab

Assigning Policy Rules to the New Data and Updating Your Policy

After refreshing the Analyzer with updated data, you can assign rules to the new clusters in your model.

To assign policy rules to the new data and update the policy:

1. Do one of the following:
 - To assign policy settings to the new data automatically, in the **Summary** tab, click **Assign Threat Severities** and **Assign Policies**. You can change the settings assigned later.
 - To assign policy settings manually, find the new clusters in the **Baseline** or **Details** tab and assign settings for Action, Logging Level, and Threat Severity. See "[Manually Setting the Action, Logging Level, and Threat Severity](#)" on page 3-20.
2. From the **File** menu, select **Create Policy**.

In the Create Policy dialog box, you can either create a new policy with a different name, or select your current policy and replace it with your updates.

When you have created the policy file, you must upload it to the Management Server using the Administration Console. See "[Creating a Policy File and Uploading it into the Database Firewall](#)" on page 3-29.

Additional Features

This section contains:

- [Sensitive Data Masking](#)
- [Exporting the Data in a Model as HTML](#)
- [Creating a Model from a Policy File](#)
- [Dividing the Screen into Two Screens](#)

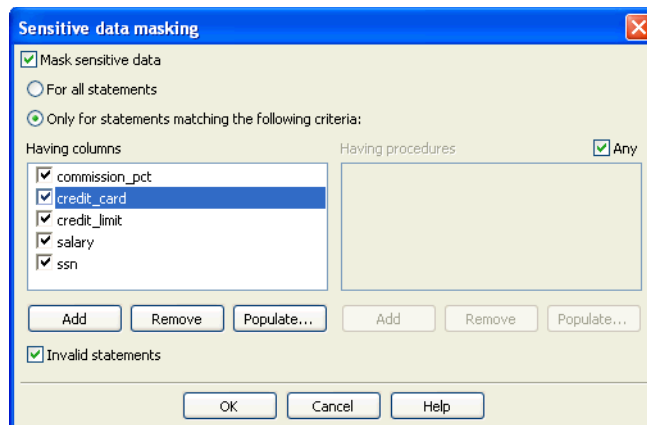
Sensitive Data Masking

Sensitive data masking prevents sensitive data, such as credit card numbers, from appearing in log files.

Selecting **Sensitive data masking** in the **Tools** menu enables you to set up rules for automatically masking sensitive data in log files.

[Figure 3–21](#) shows the **Sensitive data masking** dialog box.

Figure 3–21 *Setting Up Rules for Automatic Masking of Sensitive Data*



If a logged statement matches the masking policy set up in this dialog, the policy automatically replaces all user data in that statement (such as, string constants, integer constants, hexadecimal constants, and float constants) with alternative characters. The characters used depend on the data type.

Exporting the Data in a Model as HTML

To create an HTML summary of the data in a model, from the **File** menu, select **Export as HTML**. You may want to use this feature for reporting purposes.

Creating a Model from a Policy File

You may wish to experiment with a new model based on an existing policy file.

To create a model from a policy .dna file:

1. From the **File** menu, select **Load Policy**.

2. Select the .dna policy file.
3. From the **File** menu, select **Save**.

The file is saved as a .smdl model file.

Dividing the Screen into Two Screens

Selecting **Split** in the **Window** menu divides the screen into two, which enable you to view two tabs at the same time. To revert to a single screen, from the **Window** menu, select **Remove Split**.

Auditing Stored Procedures and Roles

This chapter contains:

- [About Auditing Stored Procedures and Roles](#)
- [Viewing and Approving Changes to Stored Procedures](#)
- [Viewing and Approving Changes to User Roles](#)

About Auditing Stored Procedures and Roles

You can audit and approve changes to stored procedures and user roles in the databases on a specified database server. Oracle Database Firewall connects to the database server at scheduled intervals and determines which changes or additions (if any) have been made to stored procedures. Stored procedure auditing and user role auditing are supported for Oracle, Microsoft SQL Server, Sybase ASE, Sybase SQL Anywhere, and IBM DB2 SQL (Linux, UNIX, and Microsoft Windows) databases.

Before you can audit stored procedures and roles, you must configure the Database Firewall-protected database to enable stored procedure and role auditing. See *Oracle Database Firewall Administration Guide* for more information.

Viewing and Approving Changes to Stored Procedures

This section contains.

- [About Viewing and Approving Changes to Stored Procedures](#)
- [Running a Manual Stored Procedure Audit](#)
- [Approving Changes Made to a Stored Procedure](#)
- [Filtering Options for Approving Changes in Stored Procedures](#)

About Viewing and Approving Changes to Stored Procedures

After you have configured stored procedure auditing, you can begin to monitor changes to stored procedures being run on the protected database right away. You can run a manual audit on a stored procedure at any time, in addition generating reports that run automatically according to the schedule set up in the SPA enforcement point settings. After the audit process is complete, you can approve changes made to the stored procedure.

You can perform the following types of stored procedure auditing activities:

- View all additions or changes made to the stored procedures

- Determine which changes are pending approval
- Approve changes
- View all approvals made
- Examine a history of previous approvals

Note: For Oracle databases, privileges such as invoker's right or definer's rights do not affect stored procedure auditing.

See Also:

- [Chapter 6, "Generating Oracle Database Firewall Reports,"](#) for information about running reports
- *Oracle Database Firewall Administration Guide* for information about configuring stored procedure auditing

Running a Manual Stored Procedure Audit

To run a manual audit for stored procedures:

1. Log in to the standalone Database Firewall or Management Server Administration Console.

See "[Logging in to the Administration Console](#)" on page 2-2 for more information.

2. Select the **Monitoring** tab.
3. Under **Enforcement Points**, click **List**.

The Enforcement Points page appears, and lists the available enforcement points.

4. For the enforcement point that is responsible for the stored procedure audit, click **Manage**.
5. In the Manage Enforcement Point page, scroll down to Stored Procedure Auditing Control.
6. Click the **Run Now** button.

Database Firewall displays a message letting you know that the audit has been started.

Approving Changes Made to a Stored Procedure

To approve or decline approval for changes made to a stored procedure:

1. If you are not already logged in, log in to the standalone Database Firewall or Management Server Administration Console.

See "[Logging in to the Administration Console](#)" on page 2-2 for more information.

2. Select the **Reporting** tab.
3. Under **Stored Procedure Auditing**, select **Pending**.

See "[Filtering Options for Approving Changes in Stored Procedures](#)" on page 4-4 for descriptions of all the options.

The Pending Approvals for Stored Procedures page appears, with a list of audited stored procedures.

4. Check each stored procedure by clicking the area just below the stored procedure link, as shown in the following screen.

Enforcement Point	Stored Procedure Name	Class	Modifications	Status	By	Last Modification Date
DB01	AVSRCUSER1 DBMS_SRC_STREAMS_UTILITY_BODY	user	New	Pending		2008-09-25 21:41:28

The area expands to show a modification history and notes for the stored procedure:

Event	By	Date
New	user	2008-09-25 21:41:28

Note	Auditor	Date

Add Note

5. Check the modification history of the stored procedure.

The Event column provides a history of all changed events for the stored procedure. (If there are no changed events, only the **New** link is shown under Event.) To find how the stored procedure changed, click the **Show Differences** link, which appears after the listed events for the stored procedure. A separate window appears, showing how the SQL for the stored procedure was modified.

Note: The **Show Differences** link appears only after a stored procedure has been approved.

If you want to see the original SQL text that was used to create the stored procedure, then click on the link for the stored procedure itself.

The following example, shows how an original stored procedure appears. The red text indicates keywords.

```

Modification For Stored Procedure "AVSRCUSER1.DBMS_SRC_STREAMS_UTILITY_BODY"
Print this Page

Action:          New          Last Approval Date: [Never Approved]
Object Class:    User          Tags:
Last Modification Date: 2008-09-25 21:41:28  MetaDataInteraction,InjectionRisk
Monitoring Point: DB01

----- PRIVATE PROCEDURES -----
-- NAME: EXTRACT DB VERSION
--
-- DESCRIPTION:
-- This procedure extracts each component of the database version,
-- (separated by '.' ) passed into it.
-- In simple terms, its something like StringToNumber for version
-- PARAMETERS
-- ver_string (IN) - Version of the Database
-- ver_comp1 (OUT) - First component of version [LEFTMOST]
-- ver_comp2 (OUT) - Second component of version
-- ver_comp3 (OUT) - Third component of version
-- ver_comp4 (OUT) - Fourth component of version
-- ver_comp5 (OUT) - Fifth component of version
--
-- EXAMPLE:
-- extract_db_version('10.2.0.0.0', vc1, vc2, vc3, vc4, vc5);
-- returns:
-- vc1 = 10, vc2 = 2, vc3 = 0, vc4 = 1, vc5 = 0

```

6. In the Notes area of the Modification History area for the stored procedure, optionally add a note and then click the **Add Note** button.
7. If you approve the modifications to the stored procedure, then click **Accept**; if you disapprove, click **Decline**.

If you click **Decline**, then the stored procedure changes are declined immediately.

8. If you click **Accept**, the Accept Changes for *name* area appears; enter a note in the **Approval Comment** field.

For example:

Changes to stored procedure AVSRCUSER1.DBMS_SRC_STREAMS_UTILITY_BODY approved by LBouligny, 8/17/10

Click **Accept**. The list of audited stored procedures re-appears.

You can add notes from this main list by clicking on any of the stored procedure settings, such as user or authorization. The screen expands to display an **Add Note** field. To remove the display of this field, click it again.

If you click the stored procedure name, which is a link, it displays the text of the stored procedure.

Filtering Options for Approving Changes in Stored Procedures

You can use any of the following filtering options to view the stored procedure audit report.

- **Summary:** Lists each enforcement point that has Stored Procedure Auditing enabled in the enforcement point settings. For each enforcement point, the page lists the number of stored procedures that have been fully approved, the number that are pending at least one approval, and the total number of records in the audit history.
- **Approved:** Lists each stored procedure that has at least one approval. A **Filter** button is available to filter the results (see "Searching for Traffic Logs" on page 5-2 for details of how to set up filter search conditions). Clicking the name of a stored procedure shows the modification detail, including when it was first approved, and any approvals that have been granted for subsequent modifications. The text shown in the **Tags** column is highlighted in red in the detail. The tags are generated by Oracle Database Firewall itself, based on preset rules.

- **Pending:** Lists each stored procedure that matches the **Filter** settings and is awaiting at least one approval. The type of change, such as **New** or **Modify**, is displayed in the **Modifications** column. Clicking the name of the stored procedure shows the content of the procedure after the change. Clicking anywhere along the green bar displays the modification detail and a box to enter notes, such as details of the actions that need to be investigated before approval can be granted. If the change is **Modify**, also displayed is a **Show Difference** link, which you can use to identify the changes made. The text shown in the **Tags** column is highlighted in red in the detail.

On the right side of the page, you will see **Decline** and **Accept** buttons for each stored procedure. Clicking the **Accept** button approves all changes that are pending approval for the stored procedure. Clicking **Decline** prevents the changes from being approved when **Approve All** is selected.

Clicking **Approve All** near the top of the page approves all changes made to all stored procedures that match the currently-selected **Filter** and have not been declined.

- **Audit History:** Lists all previous approvals and all pending approvals that match the **Filter** settings. Each time a procedure is approved, the transaction is recorded in the audit history. Click anywhere along the green bar to see more detail.

Viewing and Approving Changes to User Roles

This section contains:

- [About Viewing and Approving Changes to User Roles](#)
- [Running a Manual User Role Audit](#)
- [Approving Changes Made to a User Role](#)

About Viewing and Approving Changes to User Roles

After you have configured user role auditing, you can begin to monitor changes to user roles being used on the protected database right away. You can run a manual audit on a user role at any time, in addition to generating reports that run automatically according to the schedule set up in the URA enforcement point settings. After the audit process is complete, you can approve the changes made to the user role.

You can perform the following types of user role auditing activities:

- View all additions or changes made to the user roles.
- Approve the changes.
- Determine which changes are pending approval.
- View all approvals made.
- Examine a history of previous approvals.

See Also:

- [Chapter 6, "Generating Oracle Database Firewall Reports,"](#) for information about running reports
- *Oracle Database Administrator's Guide* for information about configuring user role auditing

Running a Manual User Role Audit

To run a manual audit for user roles:

1. Log in to the standalone Database Firewall or Management Server Administration Console.

See "[Logging in to the Administration Console](#)" on page 2-2 for more information.

2. Select the Monitoring tab. the **Monitoring** tab.
3. Under **Enforcement Points**, click **List**.

The Enforcement Points page appears, and lists the available enforcement points.

4. For the enforcement point that is responsible for the user role audit, click **Manage**.
5. In the Manage Enforcement Point page, scroll down to User Auditing Control.
6. Click the **Run Now** button.

Database Firewall displays a message letting you know that the audit has been started. The audit process should last a couple of minutes.

Approving Changes Made to a User Role

To approve or decline approval for changes made to stored procedures:

1. If you are not already logged in, log in to the standalone Database Firewall or Management Server Administration Console.

See "[Logging in to the Administration Console](#)" on page 2-2 for more information.

2. Select the **Reporting** tab.
3. Under **User Role Auditing**, select the filtering option **Pending**.

See "[Filtering Options for Approving Changes in User Roles](#)" on page 4-8 for descriptions of all the options.

The Pending Approvals for User Roles page appears, with a list of audited user roles.

4. Check each user role by clicking the area just below the user role link, as shown in the following screen.

Enforcement Point	User Role Name	Class	Modifications	Status	By	Last Modification Date	Tags
DB01	AVSRCUSR1	user	1 modification	Pending		2010-08-26 13:46:04	
DB01	AVSYS	user	1 modification	Pending		2010-08-26 13:46:04	SYSTEM

The area expands to show a modification history and notes for the user role.

DB01 AVSYS user 1 modification Pending 2010-08-26 13:46:04 SYSTEM [Decline](#) [Accept](#)

Modification History

Event	By	Date
Last Approved	admin	2010-08-26 13:42:41
Modify		2010-08-26 13:46:04

[Show Difference](#)

Notes

Note	Auditor	Date

[Add Note](#)

5. Check the modification history of the user role.

The Event column provides a history of all changed events for the user role. (If there are no changed events, only the **New** link is shown under Event.) To find how the user role changed, click the **Show Differences** link, which appears following the events for that user role. A separate window appears, showing how the SQL for the user role was modified.

If you want to see the original SQL text that was used to create the user role, then click the link for the user role itself.

The following example shows how the AVUSR role was changed:

Difference for User Role "AVSYS"

Key: Unmodified Text **New Text** **Inserted Text** **Replaced Text** **Deleted Text**

[Print this Page](#)

Enforcement Point: DB01
 User Role Name: AVSYS
 Class: User
 Edit Summary: 1 modification
 Last Approved: 2010-08-26 13:42:41 UTC
 Modified: 2010-08-26 13:46:04 UTC

```

Username: AVSYS
Authentication method: DATABASE

Roles:
AQ_ADMINISTRATOR_ROLE : (parent: AV_ADMIN)
AQ_USER_ROLE : (parent: AV_SOURCE)
AV_ADMIN
AV_AGENT : (parent: AV_ADMIN)
AV_SOURCE
CONNECT : (parent: DV_ACCIMGR)
DV_ACCIMGR
HS_ADMIN_ROLE : (parent: SELECT_CATALOG_ROLE)
RESOURCE
SELECT_CATALOG_ROLE : (parent: AV_ADMIN)
XDBADMIN : (parent: AV_ADMIN)
XDBWEBSERVICES : (parent: XDBADMIN)

Privileges:
ALTER ANY RULE
ALTER PROFILE : (parent: DV_ACCIMGR)
ALTER SYSTEM
ALTER_USER : (parent: DV_ACCIMGR)
CREATE ANY RULE
CREATE ANY VIEW : (parent: AV_AGENT)
  
```

6. In the Notes area of the Modification History area for the user role, optionally add a note and then click the **Add Note** button.

7. If you approve the modifications to the user role, then click **Accept**; if you disapprove, click **Decline**.

8. If you click **Accept**, the Accept Changes for *name* area appears; enter a note in the **Approval Comment** field.

For example:

Changes to user role AVUSER approved by LBouligny, 8/17/10

Click **Accept**. The list of audited user roles re-appears.

You can add notes from this main list by clicking on any of the stored procedure settings, such as user or authorization. The screen expands to display an **Add Note** field. To remove the display of this field, click it again.

If you click the stored procedure name, which is a link, it displays the text of the stored procedure.

Filtering Options for Approving Changes in User Roles

You can use any of the following filtering options to view the stored procedure audit report

- **Summary:** Lists each enforcement point that has User Role Auditing enabled in the enforcement point settings. For each enforcement point, the page lists the number of user roles that have been fully approved, the number that are pending at least one approval, and the total number of records in the audit history.
- **Approved:** Lists each user role that has at least one approval. A **Filter** button is available to filter the results (see "[Searching for Traffic Logs](#)" on page 5-2 for details of how to set up filter search conditions). Clicking the name of a user role shows the modification detail, including when it was first approved, and any approvals that have been granted for subsequent modifications. The text shown in the **Tags** column is highlighted in red in the detail. The tags are generated by Oracle Database Firewall itself, based on preset rules.
- **Pending:** Lists each user role that matches the **Filter** settings and is awaiting at least one approval. The type of change, such as **New** or **Modify**, is displayed in the **Modifications** column. Clicking the name of the user role shows the content of the user role after the change. Clicking anywhere along the green bar displays the modification detail and a box to enter notes, such as details of the actions that need to be investigated before approval can be granted. If the change is **Modify**, also displayed is a **Show Difference** link, which you can use to identify the changes made. The text shown in the **Tags** column is highlighted in red in the detail.

On the right side of the page, you will see **Decline** and **Accept** buttons for each user role. Clicking the **Accept** button approves all changes that are pending approval for that user role. Clicking **Decline** prevents the changes from being approved when **Approve All** is selected.

Clicking **Approve All** near the top of the page approves all changes made to all user roles that match the currently-selected **Filter** and have not been declined.

- **Audit History:** Lists all previous approvals and all pending approvals that match the **Filter** settings. Each time a user role is approved, the transaction is recorded in the audit history. Click anywhere along the green bar to see more detail.

Accessing and Viewing the Traffic Log

This chapter contains:

- [Accessing the Traffic Log](#)
- [Viewing the Traffic Log for Database Response Monitoring](#)

Accessing the Traffic Log

This section contains:

- [Accessing Traffic Logs](#)
- [Viewing Logged Traffic](#)
- [Searching for Traffic Logs](#)
- [Viewing the Log Search Results](#)

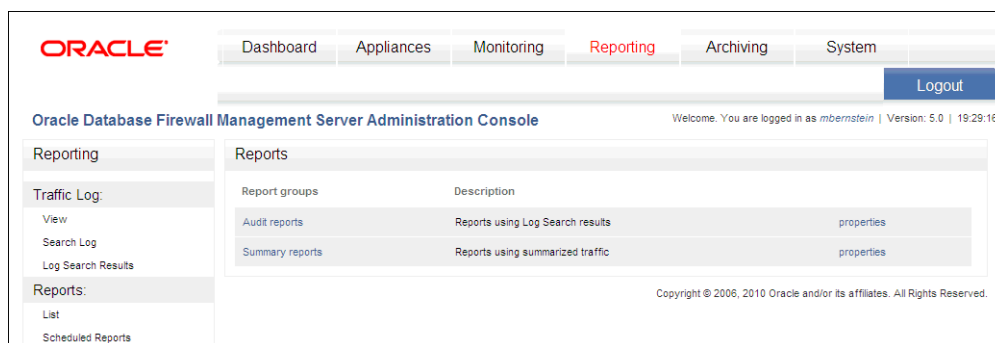
Accessing Traffic Logs

From time to time, you may want to recover data from the traffic log for auditing purposes, or to investigate possible attempted attacks. The traffic log stores details of all logged SQL statements.

To do so, log in to the standalone Database Firewall or the Management Server Administration Console, select the **Reporting** tab, and use the **Traffic Log** menu in the Reporting page to view reports, search logs, and find log search results.

[Figure 5–1](#) shows the Traffic Log page of the Administration Console.

Figure 5–1 Accessing the Traffic Log



The **Traffic Log** menu contains three options: **View**, **Search Log** and **Log Search Results**, as described in the following sections. To learn how to log in to the Administration Console, see *Oracle Database Firewall Administration Guide*.

Viewing Logged Traffic

Clicking the **View** button, followed by clicking **Start**, displays logged traffic (the latest information may take up to five minutes to display). The feature is automatically switched off after one hour to prevent loss of performance. A **Filter** button is available to filter the results.

Searching for Traffic Logs

Figure 5–2 shows the Search Traffic Log page of the Administration Console. You can use this option to retrieve a range of records from the traffic log for reporting purposes.

Figure 5–2 Searching for a Traffic Log

- **Title:** Enter a title for the report (for example, Traffic 1st-2nd March).
- **Period Type:** Choose **relative** if you want to retrieve a set of records that occurred with a period that is relative to the current date and time. Choose **absolute** if you want to retrieve a set of records that occurred within a fixed period.

The following is displayed if you choose **relative**:

- **Report Period:** If, for example, you choose **1 Week** and **Now - 1 Hour**, and the report is generated at 18:00, all records for one week prior to 17:00 will be retrieved. You may want to use the **relative** option for scheduled reports (see ["Generating Audit and Summary Reports"](#) on page 6-2), because the period of report is automatically adjusted according to the time that the report is generated.

The following is displayed if you choose **absolute**:

- **Timerange begin/Timerange end:** Use these options to specify the fixed time period.

- **Maximum results:** You can limit the number of results to return. This can help to reduce the length of time required to retrieve records. The earliest records are retrieved if the limit is reached.
- **Search Conditions:** You can use the **Search Conditions** panel to filter the records to retrieve and reduce the time taken for the process to complete. For example, you could choose to return only records that have a threat severity greater than "moderate", or those that have a threat severity greater than "moderate" and belong to the "sales" database. The panel offers a high degree of flexibility to customize the search conditions to your exact requirements. All results are returned if no filter is specified.

You can add search conditions by using the menus, options and fields in the right-hand side of the panel. The tree view on the left-hand side of the panel shows the search conditions that are set up and defines the logical operations between those conditions.

Each operator (AND, OR, NOT) in the tree view operates on the conditions at the next level below.

Figure 5–3 shows how the AND operator appears for a traffic log search condition. In this example, only records for the protected database `sales_db` that have a threat severity greater than moderate are retrieved.

Figure 5–3 Traffic Log Search Using the AND Operator Condition

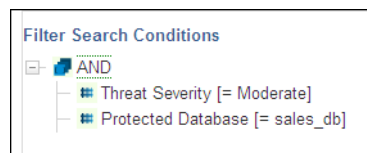


Figure 5–4 shows the OR operator search condition, which retrieves records that have a threat severity greater than moderate, and refer to the `sales_db` database.

Figure 5–4 Traffic Log Search Using the OR Operator Condition

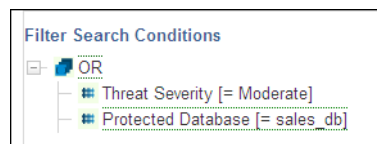
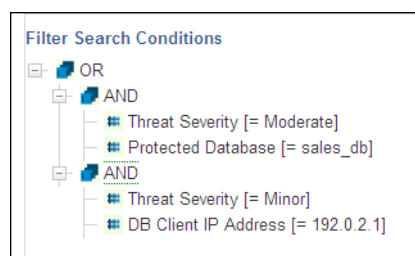


Figure 5–5 shows the OR and AND operator search conditions, which retrieve records that reveal a moderate threat severity level for the `sales_db` database and a minor threat severity level for a database client IP address.

Figure 5–5 Traffic Log Search Using the OR and AND Conditions



To define a search condition:

1. Log in to the standalone Database Firewall or Management Server Administration Console.
See ["Logging in to the Administration Console"](#) on page 2-2 for more information.
2. Select the **Reporting** tab to display the Reporting page.
3. From the **Traffic Log** menu, select **Search Log**.
4. In the Filter Search Conditions area, select an operator (for example, **AND**) from the **Select a new operator to add or change the current operator** list, and then click **Add Operator**.
5. If you want nested conditions, as shown in [Figure 5-5](#), select the top level condition (in this case, **OR**), select a new operator, and then click **Add Operator**.
6. To add one or more conditions under each operator:
 - a. Select the operator to which you want to add a condition.
 - b. In the **Add a new condition or select an existing condition to change it** list, select the condition (for example, **Threat Severity**), and then set the operator (for example, **Moderate**).
 - c. Click **Add Condition**.
7. To start the search, click the **Search** button.

Clicking **Search** displays the Searches page, which shows the current progress and details of the search. The Searches page is also accessible by clicking **Log Search Results** in the **Traffic Log** menu, as described next.

Viewing the Log Search Results

[Figure 5-6](#) shows an example of the results from clicking the **Log Search Results** button in the **Traffic Log** menu in the Reporting page.

Figure 5-6 Viewing Log Search Results

Reporting		Searches						
Traffic Log:		Title	Time Range	Started	Finished	Progress	Results	Status
View	Search Log	Last Day Events	2010-07-20 17:20:25 - 2010-07-21 17:20:25	2010-07-21 17:20:30	2010-07-21 17:20:30	100.0%	39	completed
Log Search Results	Reports:	sys activity	2010-07-20 17:39:39 - 2010-07-21 17:39:39	2010-07-21 17:39:50	2010-07-21 17:39:50	100.0%	11	completed
List	Scheduled Reports	system activity	2010-07-20 19:53:57 - 2010-07-21 19:53:57	2010-07-21 19:56:48	2010-07-21 19:56:48	100.0%	39	completed

The displayed **Status** updates automatically. For example, from "running" to "completed". The page can list multiple searches.

Clicking a title displays the statements included in that search. You can produce an audit report of the results by clicking the **Report** button. (A **Filter** button is available to filter the results.) A list of available audit reports is displayed; selecting one of these generates the report using only the data included in the log search results. You also can generate audit reports from the **Reports** menu. See [Chapter 6, "Generating Oracle Database Firewall Reports,"](#) for more information.

In the traffic log, you can expand each record to display attributes such as the action code, logging level, database type, cluster type and the origin of the attribute values (Oracle Database Firewall and/or F5 system).

Symbols to the right of the word **statement** indicate whether the statement contains attribute values that have originated from the Oracle Database Firewall system, F5 system, or both.

The value is the attribute value and the symbol under **origin** indicates that the attribute value has originated from the Oracle Database Firewall system.

Oracle Database Firewall Administration Guide describes the traffic log attributes in detail.

Log Search Results and Scheduled Reports

If you schedule multiple audit reports based on the same set of log search results, you should follow these guidelines to obtain meaningful data as well as make sure that reports are generated as scheduled:

- To have consistent data against which to compare multiple reports, schedule reports that use the same log search results to run at the same time. This ensures that the same time period and log data is used for these reports.
- Generating log search results can take a long time. Keep this in mind when scheduling reports to run at different times, since log search results will be regenerated before each unique scheduled time for reports that use them. If you have too many scheduled time periods for running reports, they may not be generated due to the time it takes to run log search results.

Viewing the Traffic Log for Database Response Monitoring

You can view the database response information by opening the traffic log (see ["Accessing the Traffic Log"](#) on page 5-1) and examining, in particular, the **Transaction Status** section.

The **Failure Count** attribute in the **Database Firewall Analysis** section indicates that this is the first consecutively-failed login attempt.

See Also: *Oracle Database Firewall Administration Guide* for detailed information about the traffic log attributes

The **Response Text** attribute shows the detailed error message generated by the database. The text in this example refers to a table not found. Note that this attribute contains responses only if **Full error message annotation** is selected in the database Response Monitoring settings. (To change this setting, you must reconfigure database response monitoring. See *Oracle Database Firewall Administration Guide*.)

Generating Oracle Database Firewall Reports

This chapter contains:

- [About Oracle Database Firewall Reports](#)
- [Generating Audit and Summary Reports](#)
- [Options in the Reports Menu](#)
- [Adding Your Own Reports](#)
- [Scheduling Reports](#)
- [How the Security Index Formula Is Calculated](#)

About Oracle Database Firewall Reports

This section contains:

- [Reports Generated from the Administration Console](#)
- [Generating Reports](#)

Reports Generated from the Administration Console

From the Administration Console, you can produce Sarbanes-Oxley (SOX), Payment Card Industry (PCI), Data Protection Act (DPA), Gramm-Leach-Bliley Act (GLBA), and Health Insurance Portability and Accountability Act (HIPAA) reports. These reports are provided by default, giving full traceability of all essential information over a selected date and time range.

You can specify which of these reports are required for a protected database. To do so, display the **Monitoring** page in the Administration Console, click **List** in the **Protected Databases** menu, click the database name, and then select the required check boxes.

Where appropriate, information is displayed graphically. This improves clarity, highlights anomalies, and enables easy interpretation of trends.

Generating Reports

You can generate reports using the **Reports** menu in the Reporting tab. The reports can be displayed as a PDF document or Excel spreadsheet. If you schedule a report, Oracle Database Firewall enables you to e-mail the report to one or more recipients. You can schedule the report to be sent to e-mail recipients at specific times, for example, once a day. You can configure a reporting user account, which is only allowed to log in to the Management Server Administration Console and run reports.

Other than this user, all valid Database Firewall system administrators can generate reports.

Figure 6–1 shows the Audit reports page, accessible from the Reports page of the Management Server Administration Console.

Figure 6–1 Audit Reports Page of the Management Server Administration Console

Reports within group: Audit reports

Audit Reports depend on Log Search Results. To create search task click here:

Report groups	Description	
[up]		
Access	Reports showing access to the databases	properties
Error conditions	Reports showing traffic causing errors	properties
FS	Shows FS related events	properties
Forensic	Reports showing all the statements	properties
SPA	Stored procedure auditing	properties
URA	User role auditing	properties

Reports	Type	Description		
Database Traffic Analysis by Application Name Detail	rtf	Audit details for statements grouped by protected database and application name	customized	properties
Database Traffic Analysis by Client IP Detail	rtf	Audit details for statements grouped by protected database and client IP address	customized	properties
Database Traffic Analysis by OS User Detail	rtf	Audit details for statements grouped by protected database and OS user	customized	properties
Database Traffic Analysis by User Detail	rtf	Audit details for statements grouped by protected database and database user	customized	properties

Generating Audit and Summary Reports

Clicking **List** displays two top-level report groups:

- Audit reports:** These are reports that include only the data included in a selected log search (see "Accessing the Traffic Log" on page 5-1). Click **Audit reports**, then the **customized** link to choose the log search to use for the report. Click the name of the report to generate the report. Audit reports are refreshed each time they are run.
- Summary reports:** These are reports that extract the required information from the traffic log while the report is being produced. Only "summarized" data (see the next section) is used.

There are many more summary reports than audit reports. Reports can take longer to generate depending on the data included.

To generate a report:

- Log in to the standalone Database Firewall or Management Server Administration Console.
See "Logging in to the Administration Console" on page 2-2 for more information.
- Select the **Reporting** tab.
- From the **Reports** menu, select **List**.
The Reports page displays the top-level set of report groups. Each group can contain reports and other groups. The **Description** column explains the types of report that the group contains.
- Drill down through the report groups until the report you want to produce is listed in the **Reports** column of the page.

The following screen shows the contents of the **Summary reports, General reports, Data access** group.

Report groups				
[up]				
Reports	Type	Description		
Access traffic analysis report by IP	rtf	grouping data by database client IP address	customized	properties
Access traffic analysis report by user	rtf	grouping data by database user name	customized	properties
Admin command usage	rtf		customized	properties
DCL command detail	rtf		customized	properties
DDL command count	rtf		customized	properties
DDL command detail	rtf		customized	properties
DML statement executions	rtf		customized	properties
Sessions by server type	rtf	List of sessions by database type	customized	properties
Top-level traffic analysis report by IP	rtf	grouping data by database client IP address	customized	properties
Top-level traffic analysis report by user	rtf	grouping data by database user name	customized	properties
Unseen traffic report	rtf	grouping data by database client IP address	customized	properties

If you want the most recent data to be made available for reporting purposes, click **Summarize Now**. This makes the data in the traffic log files available for reporting. Automatic summarizing takes place every hour.

Clicking **[up]** displays the previous report group.

5. If you want to produce a report using default parameters, such as a reporting period of one week from the current time, click the name of the report in the **Reports** column.

Alternatively, if you want to specify the reporting period or other parameters (depending on the report type), click **customized**.

The **retained reports** link is displayed if a copy of the report has been saved on the Oracle Database Firewall Management Server using the **Retain** button. The link enables you to view or delete retained reports of that type.

You can use the **properties** link to change the title or description of the report, upload a new report template, or download the existing one.

6. The report is displayed, as shown next.

The Oracle Database Firewall Management Server caches (that is, temporarily stores) the report. If you generate the report again within half an hour, the cached report is displayed.

The following four buttons are available on the page:

- **Retain:** Retains a copy of the report on the Oracle Database Firewall Management Server. You can view or remove a retained report by clicking the **retained reports** link (see the preceding section). Retained reports are included in any configuration archives.
- **Schedule:** Allows you to schedule the report to be created automatically at regular intervals (see "[Scheduling Reports](#)" on page 6-5).
- **Customize:** Allows you to change the reporting period or other parameters. Parameters depend on the report being generated.
- **Refresh:** Generates the same report again. This button is active when you access the report from the list after the report has been generated.
- **Update report:** Generates the report with any new parameters you selected.

7. Select the report parameters:
 - The parameters are different depending on the report selected.
 - For all free form parameters, you can use POSIX extended regular expressions to define the parameters. Here are some examples:
 - `ee` returns any data containing the characters `ee` (Green, Lee, Feeney, etc.)
 - `^Steven$` returns data with an exact match (Steven)
 - `Steven | Roger` returns data containing either Steven or Roger
 - By default, the report is displayed as a PDF document. To generate the report in XLS format, select **Microsoft Excel 2007 Worksheet (XLSX)** from the **Report format** drop-down list, then click **Update report**. Clicking the `<report name>.xlsx` link in the bottom-left corner of the screen allows you to view or save the report, depending on your browser settings.

Options in the Reports Menu

The following options can be displayed in the **Reports** menu on the left side of the screen:

- **Main Group:** Displays the top-level report group.
- **List:** Displays the contents of the last group visited.
- **Add Report:** Lets you add a custom report
- **Display Report:** Displays the selected report
- **Retained:** Displays retained reports of the currently-selected type.
- **Properties:** Enables you to change the title or description of the report.
- **Scheduled Reports:** Lists all scheduled reports that have been set up.

Adding Your Own Reports

You can add your own custom reports using Oracle Database Firewall and Oracle BI Publisher included with the Database Firewall installation. You will need a data definition file (XML format) and a report template (RTF format). This section describes how to extract these files from an existing Database Firewall report and use them for your own report. You will need to refer to Oracle Business Intelligence Publisher documentation for how to customize the report template.

Note: You can use Oracle Business Intelligence Publisher embedded within Database Firewall to run or modify the layout of existing reports. However, in order to add your own reports, you must have a Full Use license for Oracle Business Intelligence Publisher.

To add a report starting from existing data definition and template files:

1. Click the **Reporting** tab.
2. Drill down to an existing report, and click its **properties** link.
3. At the bottom of the **properties** page, right-click the **Report Data Definition** and **Report Template** links to save both files on your computer.

4. Customize the data definition file (an XML file) as necessary. (You will customize the report template later.)
5. Click the **Reporting** tab, then click a report group (such as Summary Reports), or drill down through the groups until you get to a group where you want to add a new report.
6. In the **Reports** menu on the left, click **Add Report**, enter a title and optional description, and then click **Add**.
7. Click the **upload** link for the **Report Data Definition**, and upload your data definition file into the new report.
8. To generate sample data to use for customizing the report template, in the **Reports** menu on the left, click **Display Report**, and then click **Generate Sample Data**.
Sample data for the new report is generated based on the data definition file you uploaded. A link to the sample data file appears at the bottom of the page.
9. In the new report, right-click the sample data file link and save it to your computer.
10. Use Oracle BI Publisher to customize the report template you downloaded from an existing report, using the sample data you generated in the new report.
Refer to Oracle BI Publisher documentation available from this page:
<http://www.oracle.com/technetwork/documentation/index.html>.
11. To upload the custom report template into the new report, locate it in the report list in Database Firewall, and then click its **properties** link.
12. Click the **upload** link for the **Report Template**, upload the template, and then click **Save**.

Scheduling Reports

A scheduled report is an audit or summary report that is generated automatically at a specified time. Optionally, the report can be set up to run automatically every hour, day, week, etc. A scheduled report is sent as a PDF document or Excel spreadsheet to specified e-mail addresses. The settings can be different for each report you set up.

To schedule a report:

1. Log in to the standalone Database Firewall or Management Server Administration Console.
See "[Logging in to the Administration Console](#)" on page 2-2 for more information.
2. Select the **System** tab.
3. Select **Email Configuration** to ensure that the SMTP e-mail settings are configured.
See *Oracle Database Firewall Administration Guide* for more information about configuring the system settings.
4. Generate the required report, as described previously. See "[Generating Audit and Summary Reports](#)" on page 6-2 for details of how to do this.
5. Select the report parameters, including the report period and the format of the report.
See "[Generating Audit and Summary Reports](#)" on page 6-2.

- Click the **Schedule** button displayed at the top of the report. The following page is displayed.

- Complete all fields, and click **Schedule**.

You must enter at least one email address. Separate several email addresses with spaces.

The **Title** is displayed in the list of scheduled reports that are set up and in the title of the report e-mail. The report will now automatically run according to the defined schedule.

- You can display a list of scheduled reports that have been set up by selecting **Scheduled Reports** in the Reports menu. For example:

Title	Next Run Time	Recurring	Last Run Time	Status
Admin command usage	2010-08-17 07:00:00	Yes	Not yet run	Not yet run

- Clicking the name of a report allows you to delete or edit the report schedule.

How the Security Index Formula Is Calculated

For reports that display a security index, the index is calculated as follows:

$$\text{Security Index} = \Sigma (\text{Threat severity (cid)} \times \text{Frequency (cid)}) / 5$$

In this specification:

- Threat severity** is the threat severity of the cluster ID, as set in the Analyzer (range 0 to 5).
- cid** is the cluster ID. All clusters that occur over the specified time period are included in the calculation.
- Frequency** is the percentage of all statements recorded over the specified period that match the cluster.

A

action level
 defined, 3-18
 novelty policy, 3-23
 setting level in policy, 3-20

Administration Console
 about, 1-7, 2-1
 auditing, 1-7
 Dashboard tab, 1-7, 2-1, 2-3
 logging in, 2-2
 Reports page, 6-2
 Search Traffic Log page, 5-2
 Traffic Log page, 5-1
 users who can log in, 2-2

administration log, 1-6

Advanced Security Option (ASO), 3-20

Analysis tab
 cluster group percentages, 3-11
 new data pie chart indicators, 3-33
 pie chart indicators, 3-11
 threat severity indicator, 3-12
 using profiles in, 3-27

Analyzer
 about, 1-6, 3-1
 Analysis tab, 3-10
 Baseline tab, 3-10, 3-15, 3-16
 creating policy file, 3-29
 Details tab, 3-12, 3-28
 how it uses clusters, 3-1
 main window, 3-9
 model data analysis, 3-8
 Properties tab, 3-16
 Summary tab, 3-9
 supplying training data for, 3-3
 tabs, 3-10
 using policy file SQL statements, 1-4

anomalies
 of statements, default rule for, 3-24

applications
 in Database Firewall, 1-6

architecture
 Oracle Database Firewall, 1-2

ASO (Advanced Security Option), 3-20

assign policies
 procedure for, 3-19

assign threat severities
 procedure for, 3-19

attacks, 1-2
 See security attacks

audit reports
 manual audit for stored procedures, 4-2
 manual audit for user roles, 4-6

auditing
 about, 1-7

automated attack, 3-25

B

Baseline tab
 filters, 3-16

blocking
 in cluster properties, 3-24
 See Database Policy Enforcement

C

cluster groups
 example contents, 3-11
 viewing data by, 3-10
 viewing in Details tab, 3-12

clusters
 about, 3-1
 action level, 3-18
 displaying data in Baseline tab, 3-15
 encrypted traffic, 3-20
 finding properties of, 3-24
 how used by Analyzer, 3-1
 logging level, 3-18
 percentage of statements in cluster group, 3-11, 3-12
 threat severity, 3-18

creating policy files, 1-4

D

DAM
 see Database Activity Monitoring

Dashboard contents
 enforcement points, 2-4
 Quick Start, 2-3
 threat status, 2-3

- throughput status, 2-3
- top ten threats, 2-3
- traffic snapshot example, 2-4
- Dashboard tab, 1-7, 2-1
 - Filter button, 2-4
- data
 - analyzing in model, 3-8
 - exporting as HTML, 3-34
 - filtering in Details and Analysis tabs, 3-14
 - masked data example, 3-12
 - masking sensitive data, 3-34
 - new, assigning policy rules to, 3-33
 - new, refining policies with, 3-31
 - updated, analyzing, 3-32
 - viewing by
 - cluster group, 3-10
 - database columns, 3-14
 - database tables, 3-13
 - profile, 3-16
- data definition file
 - in reports, 6-4
 - upload to report, 6-5
- data masking
 - example statement, Analysis tab, 3-12
 - feature, 3-34
- Database Activity Monitoring (DAM)
 - about, 1-7
 - strategy for using, 1-8
- Database Policy Enforcement (DPE)
 - about, 1-7
 - IPv6, traffic blocked, 3-17
 - setting blocking, 1-8
 - substitute statements, 3-25
- databases
 - state of in order to monitor, 3-3
- Default Rule
 - customizing, 3-24
- Details tab
 - using profiles in, 3-28
 - viewing cluster groups, 3-12
- display
 - dividing screen into two, 3-35
- DPE
 - see* Database Policy Enforcement

E

- encrypted traffic, 3-20
- enforcement points
 - dashboard display, 2-4
- event log
 - about, 1-6
- examples
 - traffic snapshot, 2-4
- Exceptions
 - creating as part of policy, 3-21
 - defining sets for, 3-28
 - using Exclude in definition, 3-22
- Exclude
 - in Exception definition, 3-22

Index-2

F

- filtering data
 - by using profiles, 3-26
 - in Baseline tab, 3-15
 - in policies, 3-14

H

- hackers
 - See* security attacks
- HTML, exporting data as, 3-34

I

- injected SQL
 - security attacks, 1-4
- IPv6
 - traffic blocked, 3-17

L

- log search results
 - and scheduling reports, 5-5
- log unique policies
 - about, 3-3
 - enabling, 3-3
 - storage of SQL data, 3-3
 - using, 3-32
- logging
 - about, 1-5
 - blocking SQL statements, 1-7
 - location of logging rules, 1-5
 - purpose, 1-5
 - setting level in policy, 3-20
 - targeted, 1-5
 - types available, 1-5
- logging level
 - defined, 3-18
- login policies for database users, 3-25
- logout policies for database users, 3-25
- long SQL statements, 1-2

M

- Microsoft SQL Server
 - using server trace file for training Analyzer, 3-3
- models
 - about creating, 3-2
 - creating, 3-2
 - creating from policy file, 3-34
 - opening existing, 3-7
 - procedure for creating, 3-4
- models and policy files
 - storing setting in model, 1-4

N

- Novelty Policy
 - creating, 3-22
 - statement matches multiple, 3-24

substitute statement, 3-23

O

operational modes

about, 1-4
defined, 1-7

Oracle Database Firewall

about, 1-1
advantages over other firewall products, 1-2
architecture, 1-2
scanning SQL traffic, 1-2
typical deployment, 1-2

Oracle Database Firewall Analyzer

See Analyzer

P

pie charts

indicators for new data sets, 3-33
indicators in Analysis tab, 3-11

planning Oracle Database Firewall system, 1-7

policies

action level, setting, 3-20
creating a model for, 3-2
creating automatically, 3-18
creating Exceptions, 3-21
creating file in Analyzer, 3-29
creating model from policy file, 3-34
designing, 3-17
development process, 3-2
exporting as HTML, 3-34
filtering
data displayed, 3-14
data displayed (profiles), 3-26
displayed clusters, 3-15
finding cluster properties, 3-24
IPv6, traffic blocked, 3-17
iterative development cycle, 3-32
listing in Management Server, 3-31
logging level, setting, 3-20
logins for database users, 3-25
logouts for database users, 3-25
masking sensitive data, 3-34
Novelty Policy, 3-23
operational modes, 1-4
procedure for automatic creation, 3-19
profiles, 3-26
refreshing with updated data, 3-32
See also Analyzer
supplying training data for, 3-3
threat severity, setting, 3-20
threat status, 2-3
updated data, analyzing, 3-32
uploading and deploying, about, 3-29
uploading and enabling in Database Firewall, 3-30
viewing general properties of, 3-16

policy files

about, 1-4

clusters, 1-4
creating, 1-4

profiles

about, 3-26
creating, 3-27
defining sets for, 3-28
using in Analysis tab, 3-27
using in Details tab, 3-28
viewing data by, 3-16

properties

of clusters, changing, 3-24

Properties tab, 3-16

protection level

planning, 1-7

Q

Quick Start Dashboard option, 2-3

R

reports

adding your own, 6-4
defining parameters, 6-4
menu options, 6-4
scheduling, 6-5
scheduling and log search results, 5-5

S

screen, dividing into two screens, 3-35

security attacks, 1-4

blind SQL injection attacks, 1-2
external, 1-2
internal, 1-2
zero-day attacks, 1-2

sets

factors used in profiles and exceptions, 3-28
procedure for defining, 3-28

SQL statements

default rule for anomalies, 3-24
finding percentage in a cluster, 3-11
injected SQL, 1-4
long, 1-2
match more than one Novelty Policy, 3-24
types, 1-2
viewing by
cluster groups, 3-10
database columns, 3-14
database table, 3-13
profile, 3-16

stored procedure auditing (SPA)

about, 4-1
approving changes to, 4-2
filtering options, 4-4
general approval process, 4-1
running manual audit, 4-2

stored procedures

auditing, 1-7

substitute statements

in cluster properties, 3-24

- in Novelty Policy, 3-23

Summary tab

- creating a policy automatically, 3-18
- elements of, 3-9

T

template

- for reports, 6-4
- upload to report, 6-5

threat severity

- defined, 3-18
- indicator, 3-12
- setting level in policy, 3-20

threat status, 2-3

throughput status, 2-3

top ten threats, 2-3

traffic log

- about, 1-5
- for training data, 3-3, 3-4
- log search results and scheduled reports, 5-5
- viewing, 5-5

training data

- enabling log unique policies for, 3-3
- from file, defined, 3-3
- from file, procedure for, 3-6
- from traffic log, 3-3, 3-4
- supplying to analyzer, 3-3

U

user role auditing (URA)

- about, 4-1
- approving changes to, 4-6
- filtering options, 4-8
- general approval process, 4-5
- running manual audit, 4-6

V

view

- dividing screen into two, 3-35

W

warnings

- specifying in cluster properties, 3-24