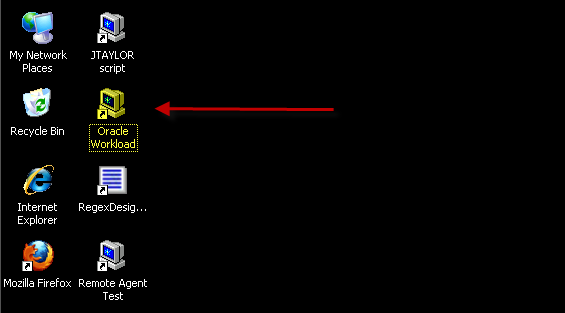
## LAB 3: Configuring Policies

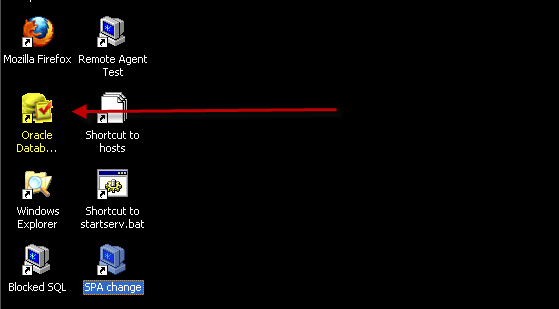
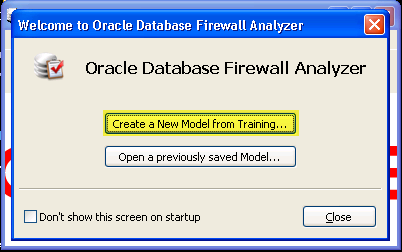
### Start Collection Workload

1. In the Windows Client, Run Oracle Workload.

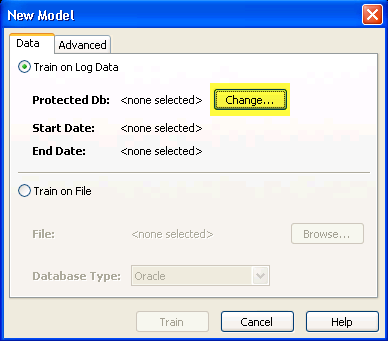


### Create Policy

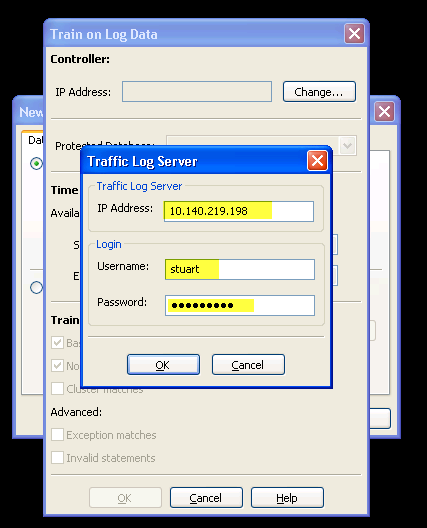
1. In the Windows Client, start the Analyzer and click on Create a New Model from Training.

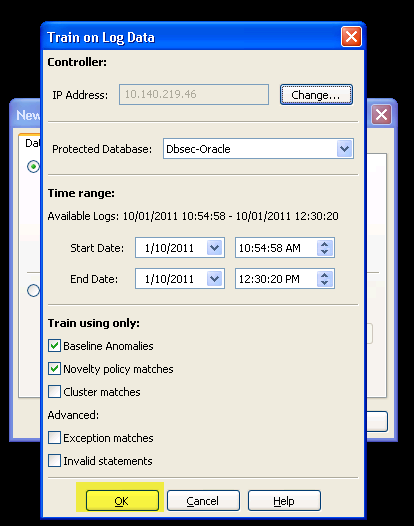
1. Select Train on Log Data and click on Change…



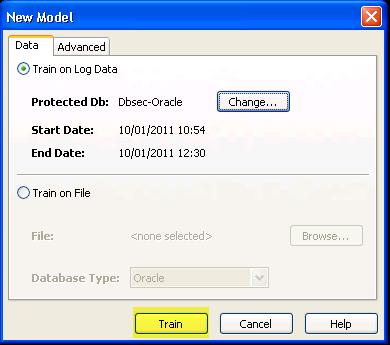
1. Enter the IP address of the firewall, and the username and password you created earlier. Click on OK.



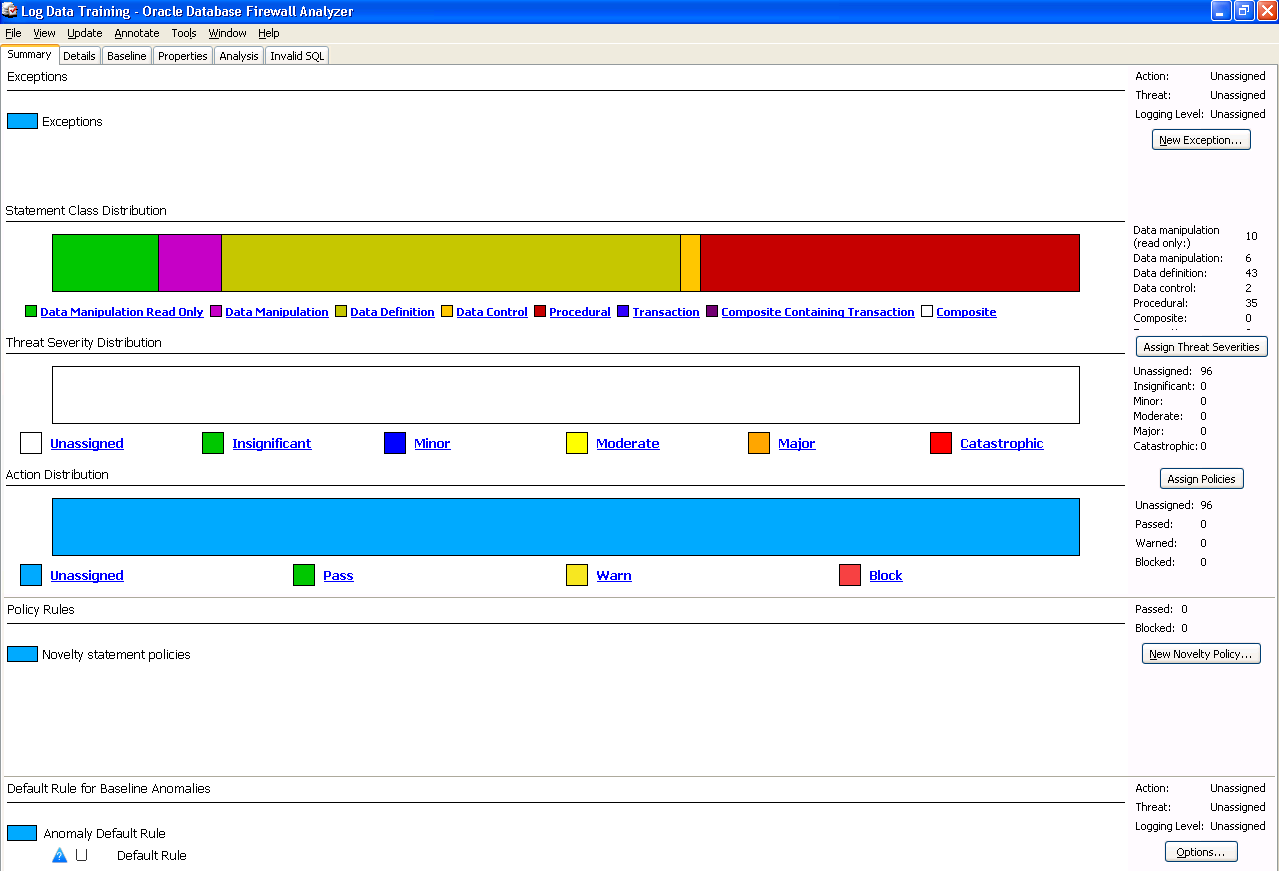
1. You will see the Protected Database you defined earlier and the time range of available data to analyze. Click on OK.



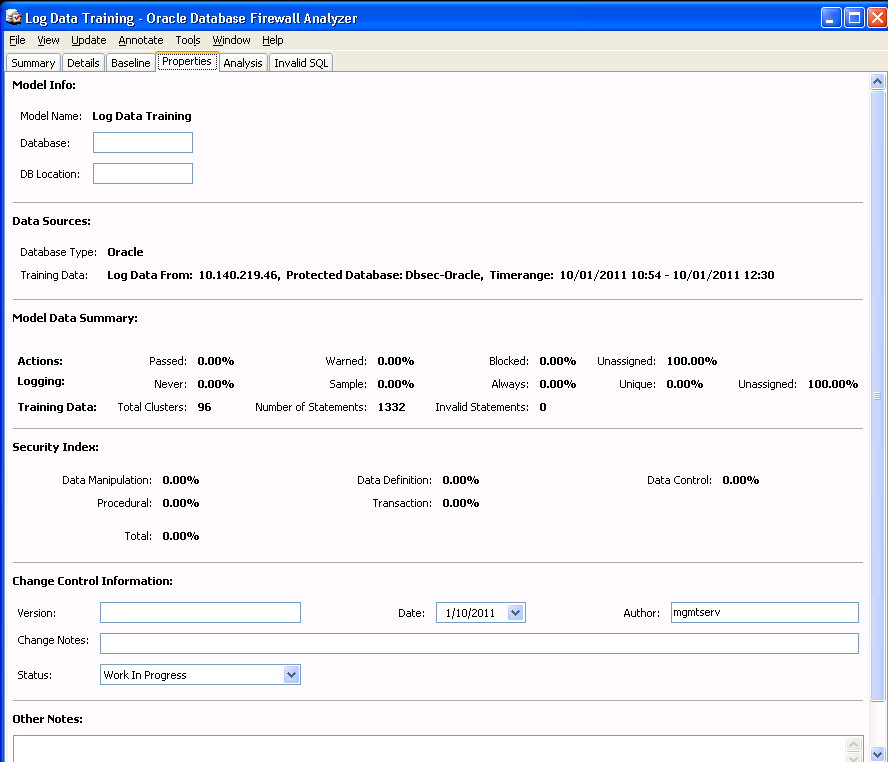
1. Click on Train.



1. You will then be presented with the Summary tab.

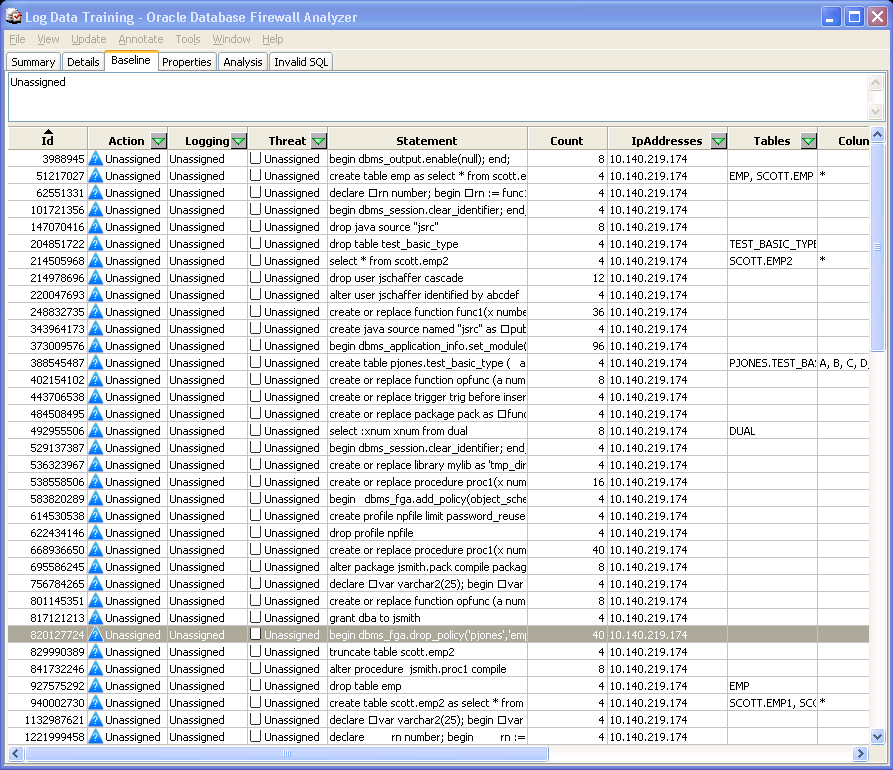


1. Click on the Properties tab to view summary statistics

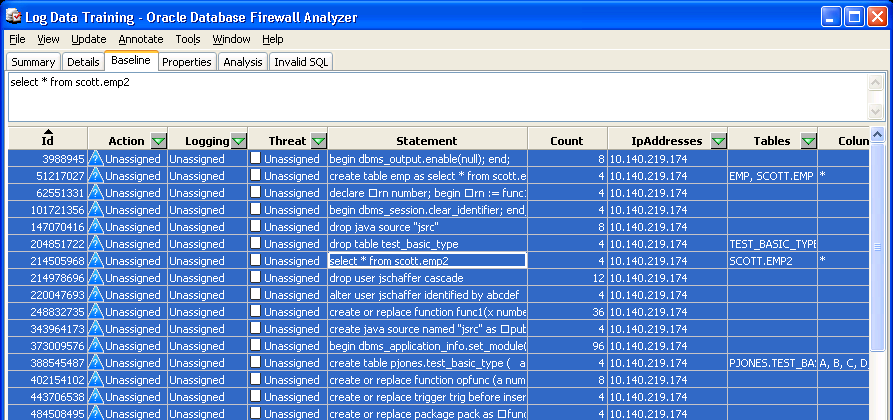


### Creating a Basic White List with WARN on UNSEEN

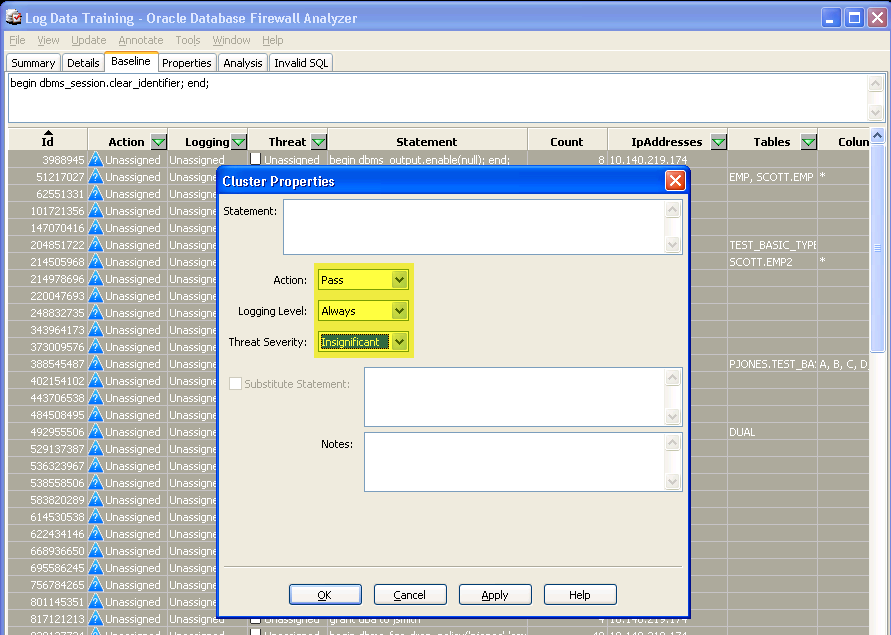
1. Click on the Baseline tab, click on the middle of the screen.



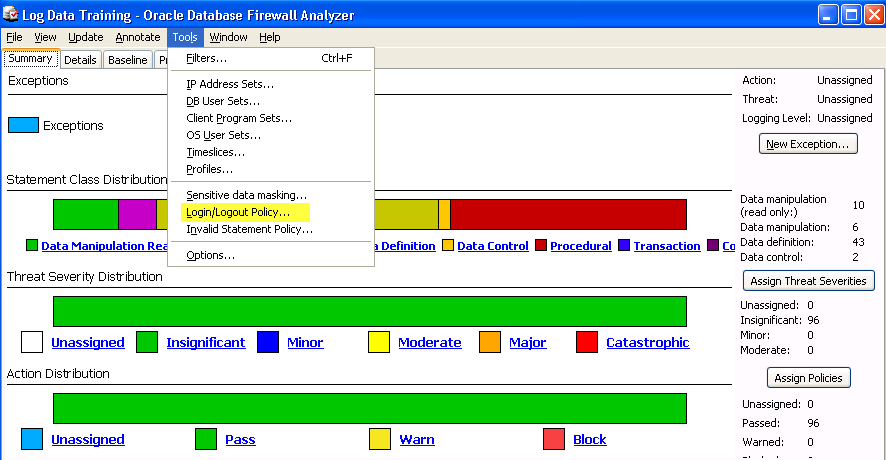
1. Click on any row and press Ctrl-A to select all clusters. These clusters are the list of SQL statements that expected activity and will be used as your white list. Right-click on the selected clusters and choose Properties.



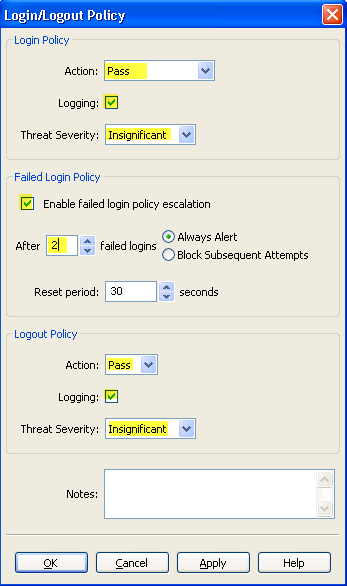
1. Set the Action to Pass, Logging Level to Always and Threat Severity to Insignificant and click on OK. In a production environment, Logging Level would not usually be set to ‘Always’ since these statements are expected, acceptable activity. The aim of a white list security policy is to identify record and potentially block out-of-policy activity.



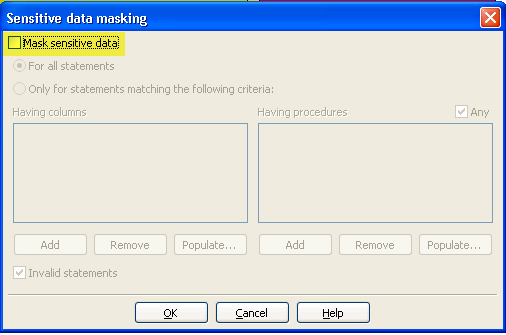
1. Click on the Tools menu and select Login/Logout Policy.



1. Set both the Login and Logout Policies to Action Pass, with Logging selected and Threat Severity Insignificant. Enable failed login escalation and set the threshold to 2 with Always Alert selected. This will raise an alert any time there are more than 2 failed logins from the same source within the reset period (in this case, within 30 seconds).

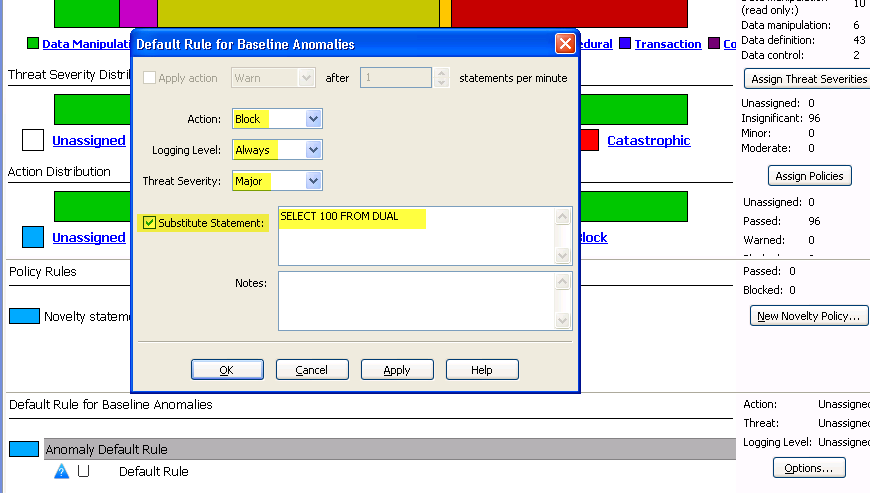


1. Sensitive Data Masking obscures parameter values in SQL statements to prevent sensitive data from being recorded by the Database Firewall. For the purpose of our labs, this will be turned off. Click on the Tools menu and select Sensitive Data Masking. Untick the Mask sensitive data option and click on OK.

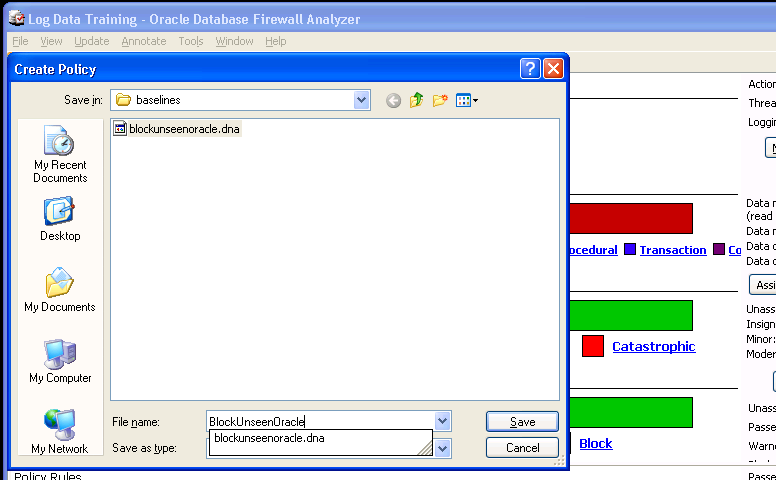


1. Click on the Summary tab, right click on Anomaly Default Rule at the bottom of the window and click on Properties. This will determine the action to be taken for any statements that fall outside our white list. Set the Action to Block, Logging to Always and Threat Severity to Major.  
     
   Select Statement Substitution and enter:  
     
    SELECT 100 FROM dual  
     
   (do not enter a semi-colon)

Click OK



1. Click on the File menu, select Create Policy. Create a new folder called policies. Enter the name BlockUnseenOracle and click on Save. This policy file will be uploaded to the Firewall via the web interface.

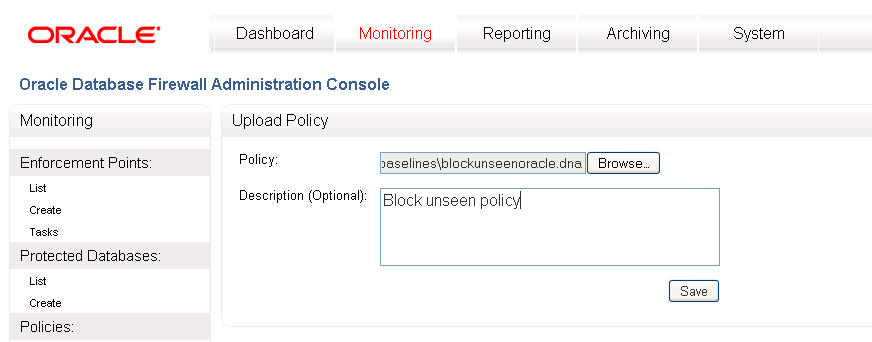


### Upload and Apply the Block Unseen policy

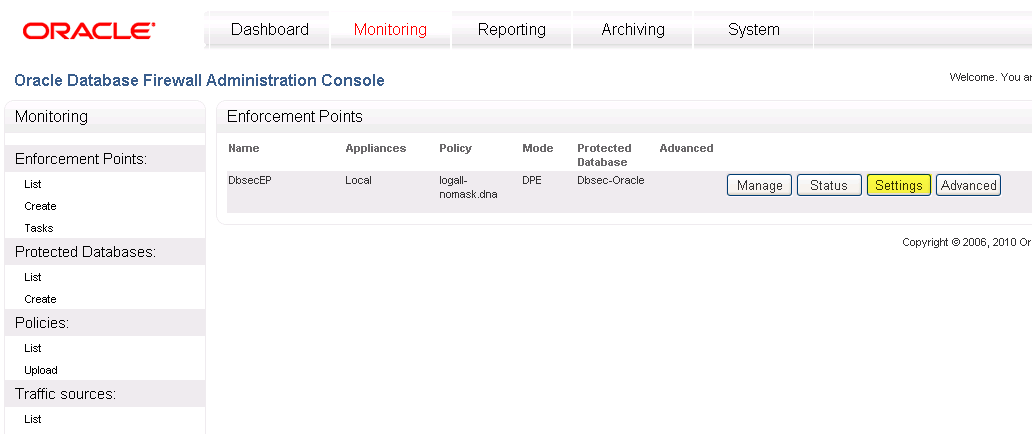
1. On the Windows server, go to the Firewall web interface, click on the Monitoring tab. On the left menu under Policies, click on Upload.

Select the BlockUnseenOracle.dna policy file.

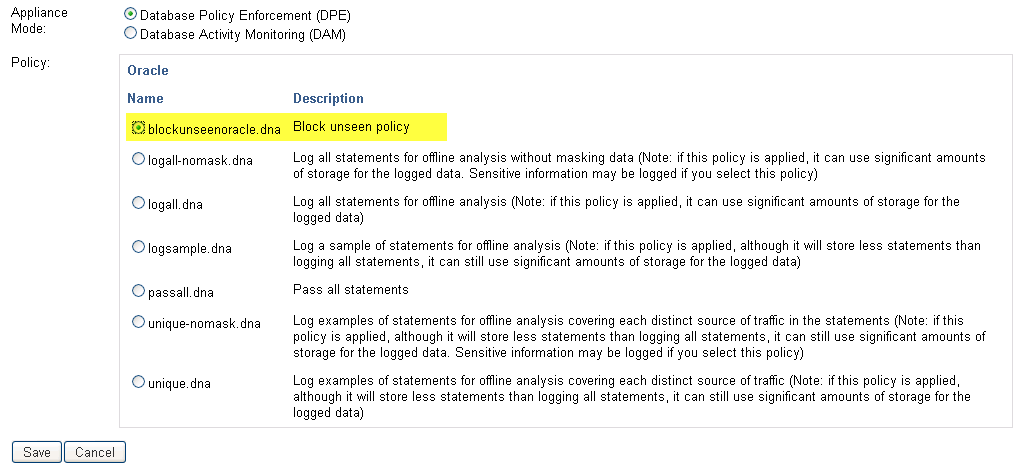
Click on Save.



1. Click on List under the Enforcement Point menu on the left. Select Settings for your Enforcement Point.

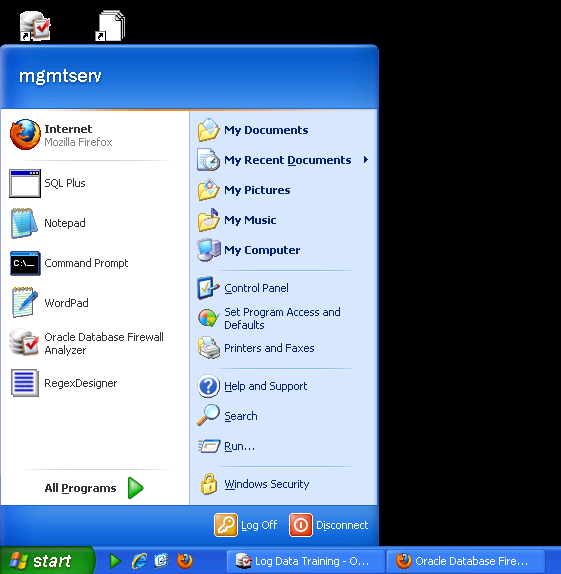


1. Scroll down to the list of policies and select BlockUnseenOracle. Click on Save.

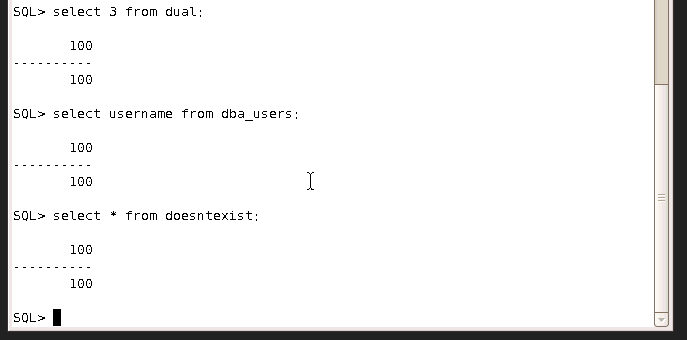


### Run sample commands

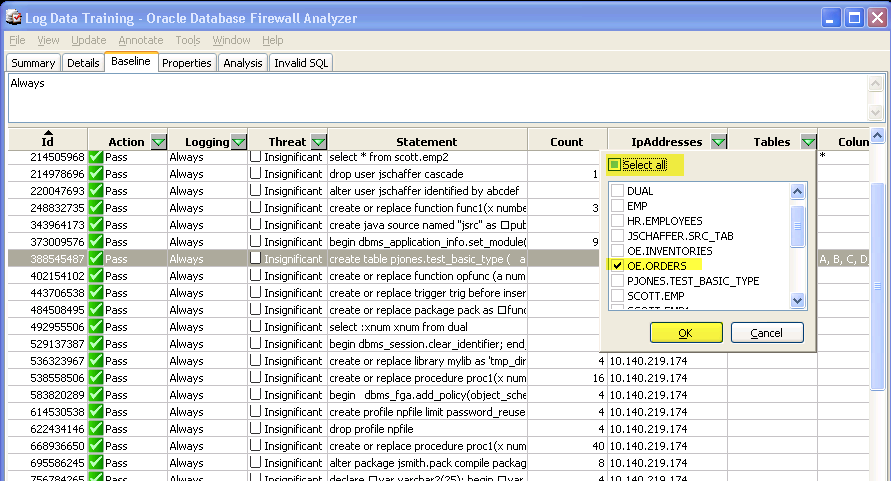
1. Return to the terminal window on the Oracle server.
2. Close any existing connections to the database.
3. Start SqlPlus and connect as system/oracle1
4. Type the following sample statements:  
     
   select 3 from dual;  
     
   select username from dba\_users;  
     
   select \* from doesntexist;  
     
   You will see results similar to the following:  
     
   SQL> select 3 from dual;  
    3  
   -----------  
    3  
   SQL> select username from dba\_users;  
   ….  
   ANONYMOUS  
   SYSTEM  
   SYS  
   MGMT\_VIEW  
   DBSNMP  
     
   76 rows selected.  
   SQL> select \* from doesntexist;  
   select \* from doesntexist  
    \*  
   ERROR at line1:  
   ORA-00942: table or view does not exist
5. Return to the Windows Client and start SQLPLUS from the Start menu.



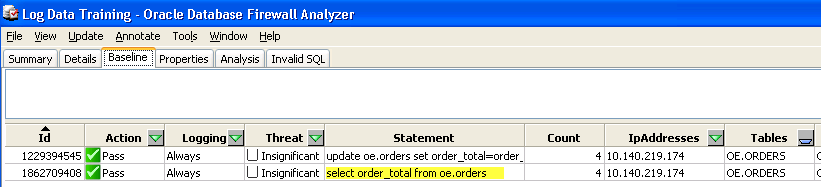
1. Login as system/oracle1. Run the same commands again. Note that these statements are not in our white list. Because we configured the policy to substitute any statement outside our white list with ‘select 100 from dual’, the result will always be ‘100’.



1. Go to the Analyzer on the WindowsClient and click on the Baseline tab. Filter by the oe.orders table by clicking on the triangle to the right of the Tables column title and unticking Select All. Scroll down to the oe.orders table and select it and click OK.

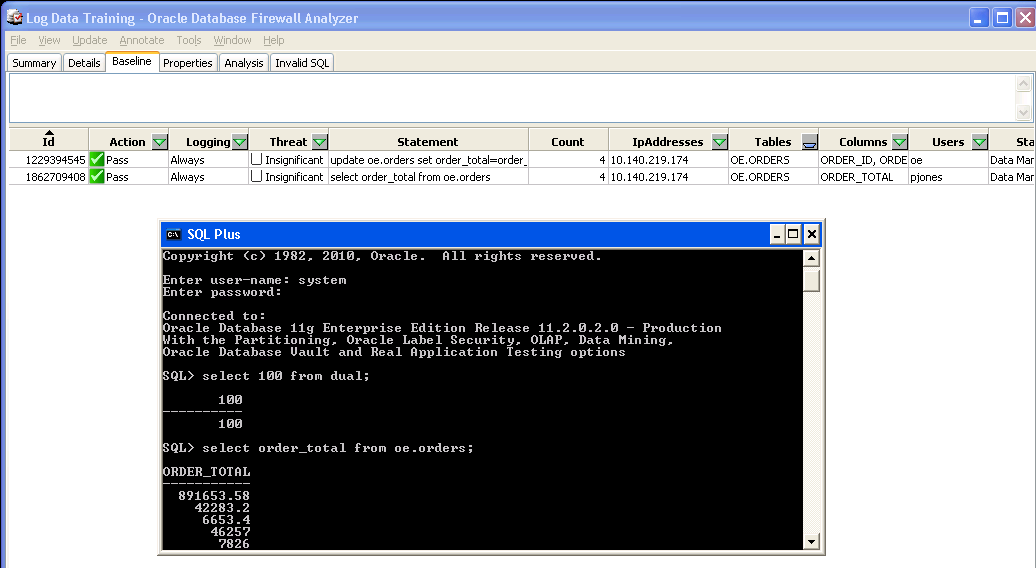


1. Confirm that the following query is in your white list:  
     
   select order\_total from oe.orders;



1. From SQLPLUS on the Windows Client, type the query from the white list to confirm that it is allowed to pass:

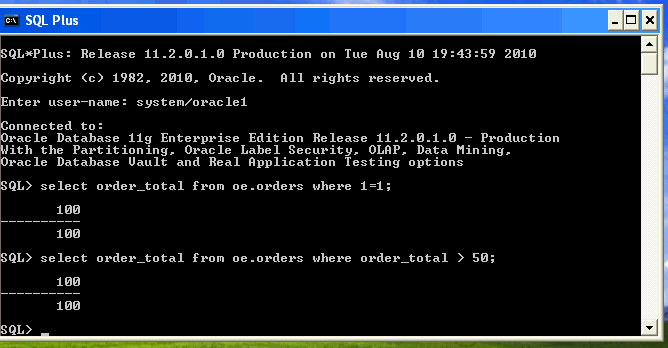
select order\_total from oe.orders;



1. Alter the query slightly to simulate an SQL injection attack:  
     
   select order\_total from oe.orders where 1=1;   
   (Note: this would normally return all rows, since 1=1 is true for all records)

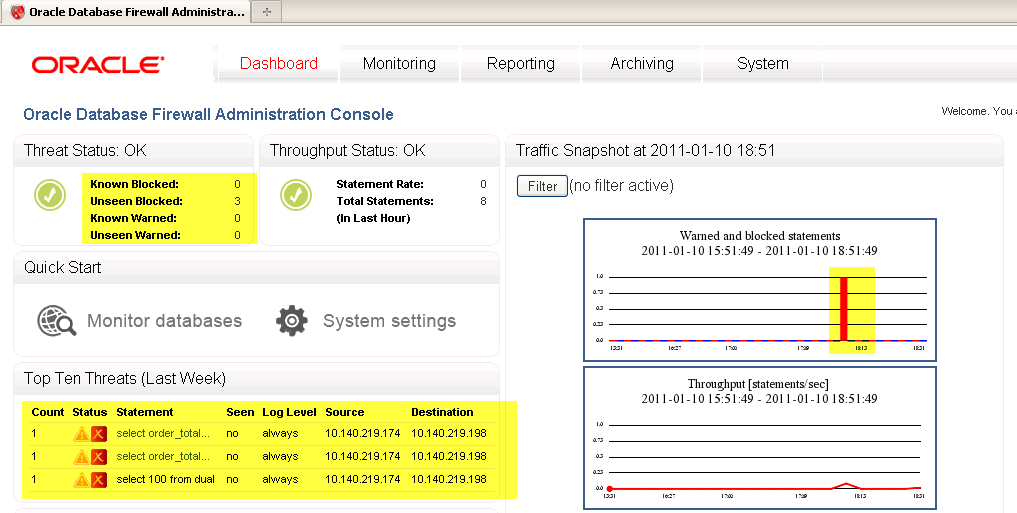
select order\_total from oe.orders where order\_total > 50;

Confirm that the statement is blocked and substituted with ‘select 100 from dual’.

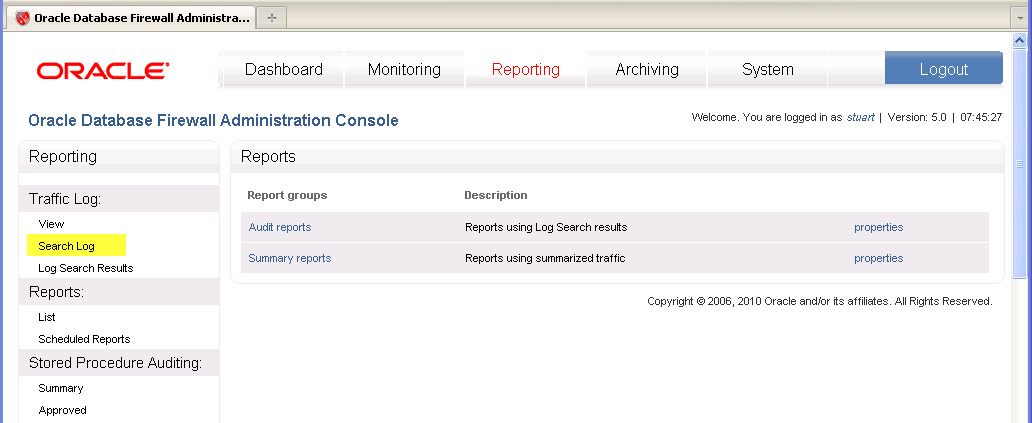


### Forensic Analysis

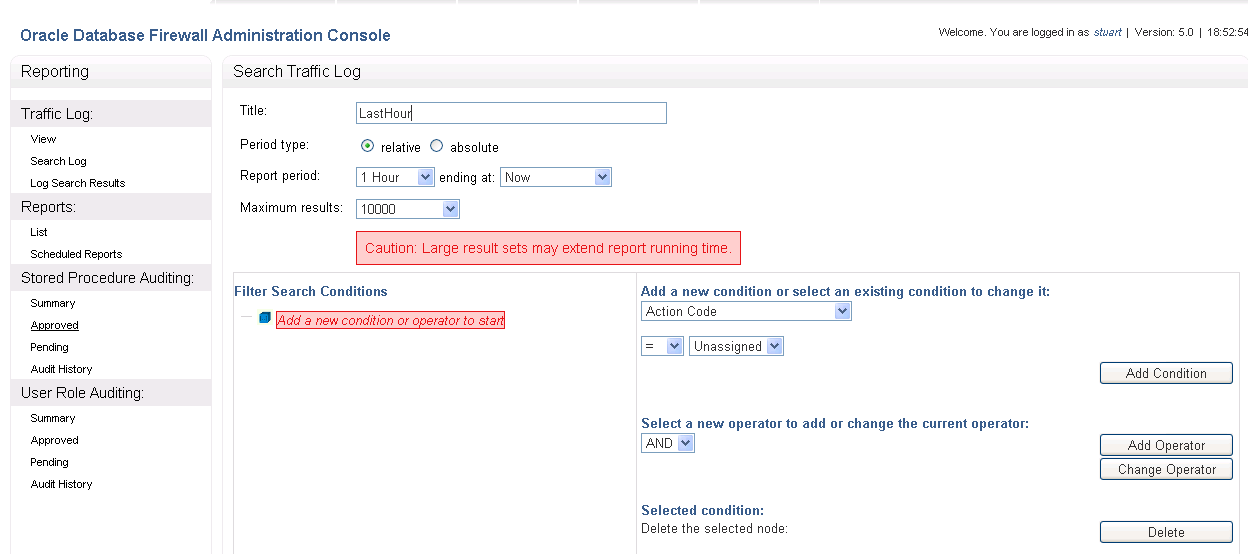
1. Return to the Firewall web interface. Confirm that the Block count has increased. Some of the blocked or warned statements will appear under Top Ten Threats.



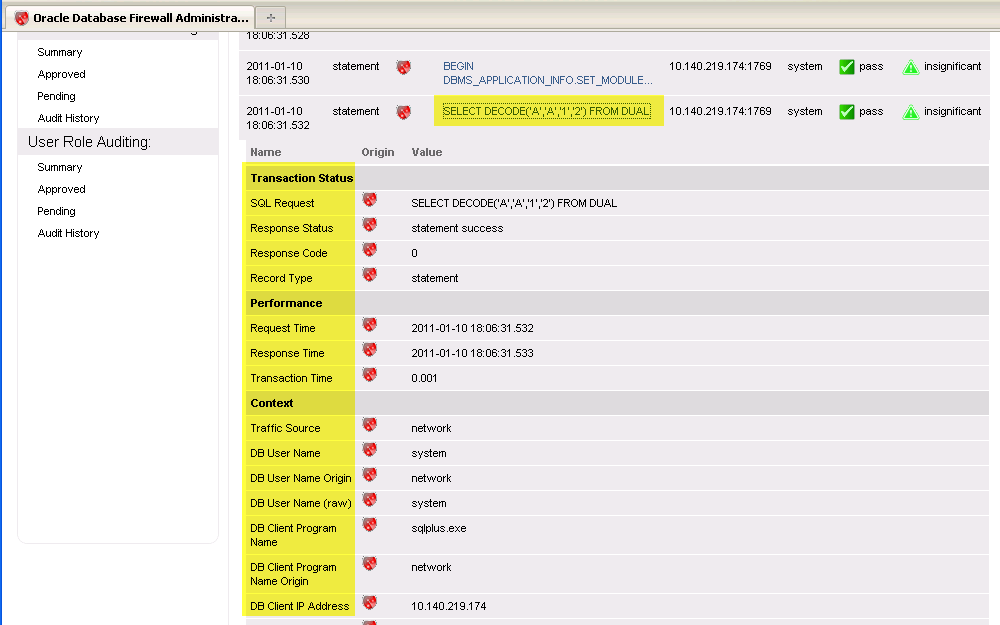
1. Click on the Reporting tab.
2. Click on Search Log under Traffic Log on the left.



1. Enter the title LastHour, choose a Report Period of 1 hour and select a maximum result set of 10,000. Click on Search.



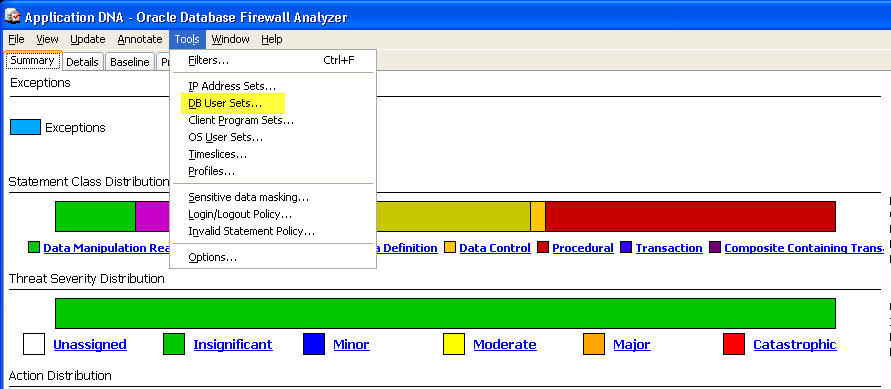
1. Click on the Search Result name LastHour to view the statements. Click on a statement to see the detailed view.



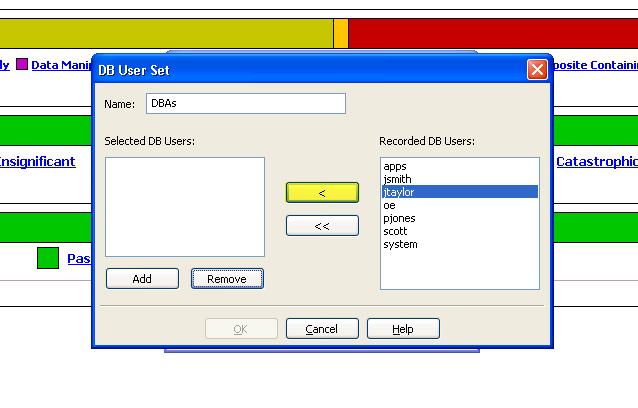
### Adding an Exceptions Policy

Exceptions provide a means for bypassing policy rules. This is most often used for DBAs, whose activity is usually not routine and cannot therefore be easily contained within a white list. In this example, we will add one DBA user who will be permitted to do anything, but whose activity will be fully audited.

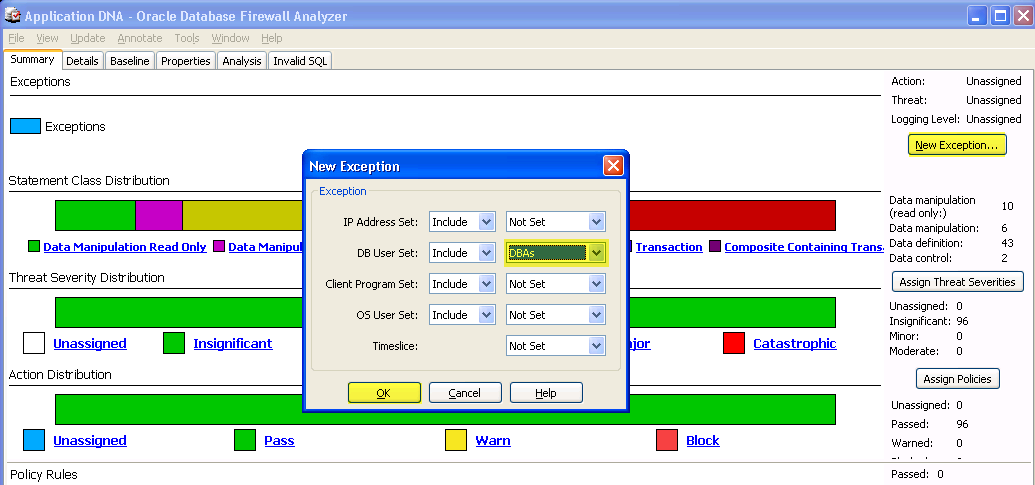
1. Return to Analyzer on the Windows server. Click on the Tools menu and select DB User Sets.



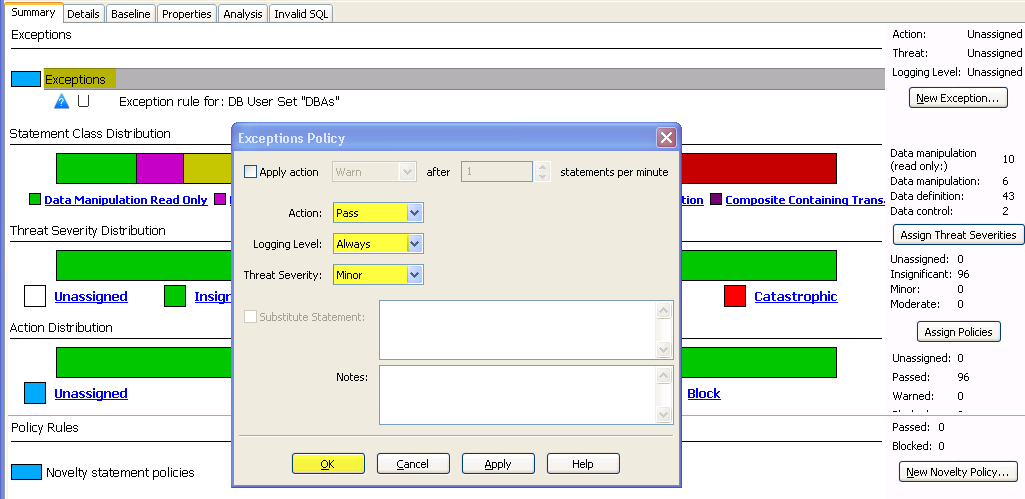
1. Click on Add. Enter the Name DBAs, then click on the name jtaylor from the list on the right. Click on the left-pointing single arrow to add jtaylor to the DB Users list. Click on OK twice.



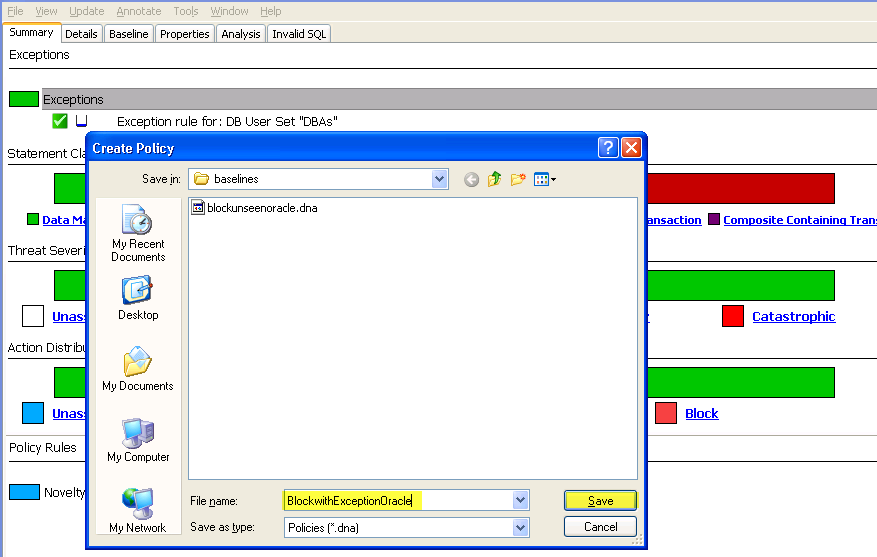
1. Click on the Summary tab and on the New Exception button on the right. From the drop-down list to the right of DB User Set, select DBAs and click on OK.



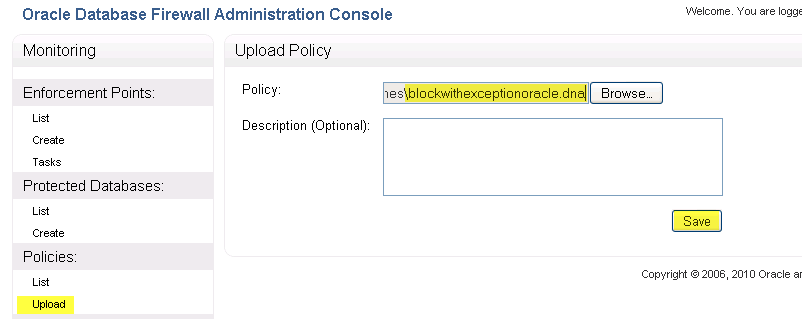
1. Right click on Exceptions and select Properties. Select Action Pass, Logging Level Always and Threat Severity Minor. Press OK.



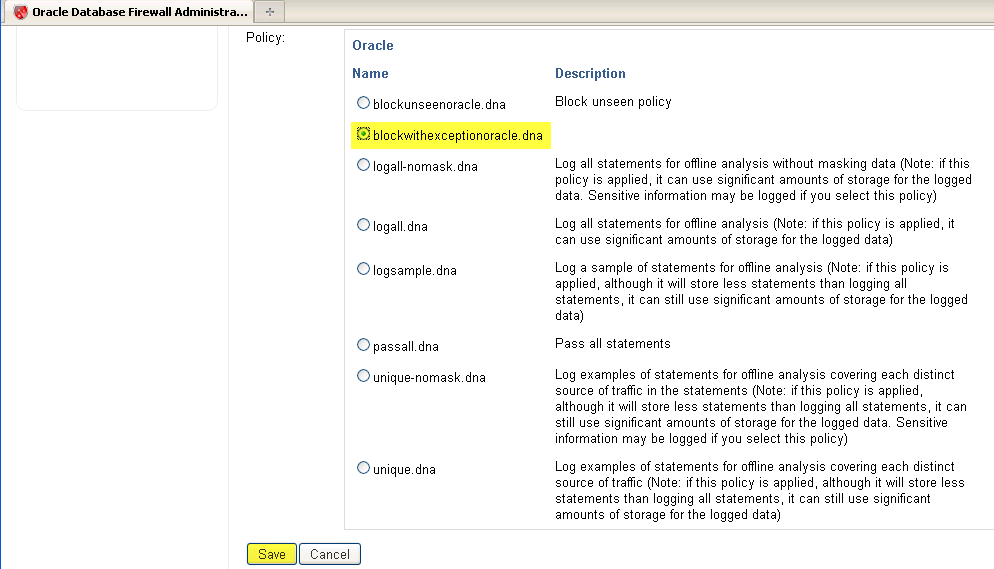
1. Click on the File menu and Create Policy. Use the name:  
     
    BlockwithExceptionOracle



1. Go to the Firewall browser session on the Windows server.
2. Click on the Monitoring tab and Upload under Policies on the left. Upload the Blockwith ExceptionOracle.dna file.



1. Click on the Settings button on the Enforcement Point. Scroll down to the list of policies and select the BlockwithExceptionOracle baseline. Click on Save.



### Run sample commands as a DBA

1. Start a sqlplus session from the Windows Client and log in as jtaylor:  
     
   sqlplus jtaylor/jtaylor
2. Run the following statement not in the white list as the DBA. With the previous policy, these commands returned 100:  
     
   select order\_total from oe.orders where 1=1;

select order\_total from oe.orders where order\_total > 50;

