LAB 5: Advanced White List Configuration

White List Updates

White lists work effectively in environments where there are regular patterns of data access and manipulation. Database access via an application server will usually fall into this category. To ensure a Firewall white list is complete, it is important to capture a representative range of application usage. One method of creating a white list will involve observing live application-database traffic. White lists based on observing live access will limit future access to queries that have been run in the past. The two drawbacks to relying on live data are the need to sanitize the white lists to ensure they do not included any unwanted queries, and the need to have effective exception and novelty policies to ensure blocking policies do not interfere with unseen legitimate database access.

A second way to create a white list is to develop a cluster list in the User Acceptance testing environment. By running through a full range of application usage, a white list will be comprehensive before it is ever deployed on a production system. This is also useful for adding clusters to an existing baseline when new features are tested in a UAT environment.

White lists developed in testing environments, however, will often include queries that are not used in practice. The most secure white lists will include only the clusters used in live, legitimate access and would not contain any unused clusters.

Exceptions Policies

Not all access to databases follows an established pattern. DBA activity, by its very nature, can seldom be pre-defined based on past patterns of access. The most common approach to handling DBA and some other types of privileged user access is to monitor and report on the complete range of their activity without applying blocking policies to their access.

The simplest way to achieve this is through the use of an Exception policy. An Exception policy is based on session parameters irrespective of the actually queries run. Exception policies can be defined based on:

- Source IP address
- User name
- Client Application name
- OS User name
- Time of day

The most common Exception rule will apply a 'Log All' policy to a list of DBA user names combined with their desktop IP addresses. All DBA activity will therefore be logged (and reported), without interfering with their access.

The key to establishing a comprehensive white list policy is to capture a full range of acceptable behaviour.

White List based on Live Usage

1. Initial Training Period
   The initial training period should cover a standard business cycle, including month-end reporting.
2. Implement initial policy with alerting, not blocking.
3. Capture infrequent events.
4. Move into blocking mode

White List based on Application Testing

1. Pre-production Testing
2. Live usage updates. Blocking can be part of Change Control, requiring new patterns of access to have to be approved before access is permited.
3. Pre-release Policy Updates

Moving from Alerting to Blocking

In a production environment, a ‘log unique’ or ‘log all unseen’ policy would be in place for several months before deploying a blocking policy.

Reports should be run on a regular basis to monitor the rate of new clusters. The ideal situation occurs when no new clusters have been observed for a complete business cycle (e.g. one month).

For the sake of efficient logging and reporting, set clusters in a white list to log never (except for clusters referencing sensitive tables when monitoring all access to sensitive tables)

Sensitive Table White List

There are two approaches to developing a baseline that only restrict access to sensitive tables.

1. Sensitive Table Access Restricted by White List
   - Cluster list containing only queries related to sensitive tables
   - Novelty policy blocking all other access to sensitive tables
   - Unseen policy passing all other queries

2. Sensitive Table Access Restricted by Query Type
   - Populate sensitive table list with ‘dummy’ query
   - Delete all other clusters
   - Novelty Policy restricting access to sensitive tables
   - Unseen policy passing all other queries

Handling Out-of-Policy Events

The main methods for handling out-of-policy events are:

- Manual policy switch-over
- Exceptions
- Novelty Policies
- Channel-specific White List

White List Examples

Start up the Firewall, Windows and Oracle 11g images: DBFW, WINCLIENT, DBSEC

On the WINCLIENT, start up the Analyzer

Train on all log data, following the instructions in the Security Management Guide documentation, starting on pages 3-6 to 3-8.

Save the model file as *DBSEC-Initial*.

Preparing a Baseline

1. **Create Sets**
    a. Users: App Users, Support Users, Sys Users. Leave pjones out of any list to use as an example of an unauthorized user.
    b. IP addresses: Apps IP
    c. Client program: Privileged User Programs, Application Programs
    d. OS user: Privileged OS Users
    e. Time slice: Oracle Hours
2. **Create Channels**
    a. Application: App Users, Apps IP, Application Programs
    b. Support: Support Users, Privileged User Programs, Privileged OS Users, Working Hours
    c. Sys Admin Channel: Sys Users, Privileged User Programs, Privileged OS Users, Oracle Hours
3. **Application Channel**
    a. Filter by App Users
    b. Set all filtered clusters to: Pass, Log Never, Insignificant
4. **Support Channel**
    a. Filter by Support Users and Privileged User Programs
    b. Set all filtered clusters to: Pass, Log Always, Insignificant
5. **Sys Admin Channel**
    a. Filter by Sys Users, Privileged User Programs
    b. Select all clusters run only by Sys Users and delete.

Save the model files as *DBSEC-Channels*.

Baseline 1

Application White List with Privileged User Exception

Save the model as *DBSEC-App White List*.

1. Create two new exceptions:
    a. Sys Users, Privileged User Programs, Privileged OS Users, Oracle Hours
    b. Support Users, Privileged User Programs, Privileged OS Users, Working Hours
2. Set the Exception policy to be:
    a. Action: Pass, Logging Level: Always, Threat: Insignificant
3. Go to the Clusters tab and view the Application channel.
4. Filter by the Application users, the Application IP address and the Application Program.
    a. Select all filtered clusters.
    b. Set the policy to:
    c. Action: Pass, Logging Level: Never, Threat: Insiginificant
5. View the Background clusters
6. Select all clusters and set the policy to:
    - Action: Block, Logging Level: Always, Threat: Major
7. On the Baseline tab, click on Unseen statements and set the policy to:
    - Action: Block, Logging Level: Always, Threat: Major

Save the model.

Create a baseline called *DBSEC-App White List*.

Upload the baseline and apply it to the enforcement point.

Run sample statements.

Baseline 2

Application White List with Privileged User Exception

Save the *DBSEC-APP White List* model as *DBSEC-Support White List*.

1. Delete the Support Users exception so that only the Sys Users exception remains:
    a. Sys Users, Privileged User Programs, Privileged OS Users, Oracle Hours
2. Go to the Clusters tab and view the Support channel.
3. Filter by the Support users, Privileged User Programs and Privileged OS Users.
    a. Select all filtered clusters.
    b. Set the policy to:
    c. Action: Pass, Logging Level: Unique, Threat: Insiginificant

Save the model.

Create a baseline called *DBSEC-Support White List*.

Upload the baseline and apply it to the enforcement point.

Run sample statements.