Security Analysis Process

- Overview
- Security Design Methodology
- Analysis Approach

Overview

top
Before deploying Yellowfin to all your users you should determine the security management profiles that you wish to deploy.

The following sections give an overview of how you should approach the security of your Yellowfin application and the access of users to your business critical information.

Security Design Methodology

top
The security design methodology described in this guide consists of one planning stage, and two implementation phases:

1. Analysis of business needs and planning the security solution
2. Designing the security framework
3. Implementing your security framework

Each implementation phase is based on an assumption that you have completed an initial planning phase. The planning phase can be done without using administrator, and is the decisive phase for the success or failure of your security. A poorly planned security framework that is not based on a study of your business needs will be difficult to maintain and may enable unauthorized access to sensitive data.
Each of these phases is described as follows:

1. **Plan the your security framework before you start using Administrator**
   Before starting the first phase, you should spend time understanding your businesses security requirements and how they related to the data that will be exposed to the business through Yellowfin.
   You must analyse the security need of the target audience for each data source and view to be implemented. The structures that you use to manage security should be based on a clearly defined user need to access the data contained in those tables and columns and stay consistent with the overall security strategy of your business.
2. **Designing the security framework**
   You create a security framework by understanding the needs of your users. You have choices to limit users to be report consumers only, limit access to database views, or limit the ability for users to publish reports to the Public repository.
3. **Implementing your Security Framework**
   Create the user roles, groups, report categories, and provide access to date sources and views to ensure your security requirements are met. Test these requirements against a sub set of users that have various levels of access.

The table below outlines the major phases in a typical view development cycle:

Development phase	Description
Prepare	Identify the target data source and become familiar with its structure. Know what data is contained within each table of each of the target databases. Understand the joins. Identify the cardinality. Know what is possible.
Analyse	Identify the user population and how it is structured; for example is the user group structured by department or by task. Identify what information the users need. Identify what standard reports they require. Familiarise yourself with their business terminology so that you can name items sensibly. Plan Identify a project strategy. For example, how many views should be created and which ones should have the capacity to be linked and to what level.
Implement	Implement your physical view SQL on the target database Build the Yellowfin view using Administrator. This manual covers this part of the view development cycle, the actual use of the tool. Test frequently during the build process for validity and reliability of inferred SQL.
Test	Form a small group of users, preferably power users who have some knowledge of what information they expect to get from the view. Pre-Release the view to these users by adding them the access security list for the view. Ask the users to perform thorough tests simulating live usage of the view(s).
Deploy	Change access security of the view so that it is available to the target user base.
Evolve	Update and maintain the view as the data sources and user requirements change and grow.

Analysis Approach

top

The following questions and responses may assist you to define your security framework and strategy.

Do all my users need report writing access?	If no then use your **reader role** to only allow users with read access to the system
Is the data in my source systems sensitive?	If yes then you will need to apply security to your data source. This will stop unauthorized access to SQL report writers and users that have admin access to the product.
View Security – Is data in my view sensitive. Can all report writers have access to the data contained in it?	If some report writers do not have access to data in the view then view security is required. The security will stop unauthorized reports being written.
The majority of the view is not sensitive but only 1 or 2 columns are.	Define view columns as secure.
*If publishing Public reports from the **same view** will some contain sensitive data and other reports not. For example an HR report containing salaries could be written from the same view as an HR report containing Headcount*	If some users should be restricted from the salary data you should create two categories for report saving – one for general access which is unsecure and one for secure access. This assumes that people with report writing access do not have EDIT access to the view – if they do then they could edit a report and add the sensitive data.
Will I be reporting from source systems with completely different subject areas?	Best to set up different categories

top