# Salesforce Security Analysis Tool

Last Update: 11/13/2017

## Overview

This tool was designed to analyze Orgs from the standpoint of:

* Users
  + Object, Field, and Application Permissions in Permission-Sets by User
  + Inactive Users
  + Active Users
* Profiles
  + Object, Field, and Application Permissions in Profile by Profile
  + Profiles with “Suspicious” permissions
* Permission Sets
  + Permissions Sets with “Suspicious” permissions (by assignment)
* Roles
  + Unused Roles
* Share Rules
  + Redundant share records
  + Shares (Manual, Apex, Rule) Analysis Report

The tool supports analyzing a significant number of orgs (e.g. more than a couple and up to 100 or so if memory permits) from various perspectives. The tool is a web application built using Java 8 JEE 7. It requires a single configuration file (called “orgs.txt”) where all the various Orgs are configured. More about this in the “Authentication into Your Orgs” section below.

The tool currently uses OAuth 2.0 JWT to authenticate into your Salesforce Orgs. However, if using this method of authentication becomes inconvenient, we can enhance the tool to support an additional form of authentication.

The local directory name tells the tool where any share-rules, profile, and permission-set metadata exists on the local file system (downloaded via Force.com Migration Tool). The object-list tells the tool which objects to look for in this directory.

## Authentication into Your Orgs

1. First, since we are using OAuth JWT, we need to create a new public/private key pair on your local machine:
   1. keytool -genkey -alias myoauth -keyalg RSA -validity 364 -keystore myoauth
   2. keytool -exportcert -alias myoauth -file myoauth.cer   
      -keystore myoauth
2. Second, in Salesforce Setup, create a new Connected App (e.g. called “sfsectool”)
   1. Enable OAuth Settings
   2. Callback URL:   
      https://login.salesforce.com/services/outh2/callback
   3. Enable “Use Digital Signatures” and specify/upload your certificate file (e.g. myoauth.cer)
   4. Select these 2 “Selected OAuth Scopes”:  
      Access and Manage Your Data (API)  
      Full Access
   5. Start URL:  
      <Your landing page URL>
3. Next, click on “Manage” for this Connected App to update these 2 settings:
   1. Permitted Users:  
      Admin approved users are pre-authorized
   2. Refresh Token Policy  
      Refresh token is valid until revoked
4. Create an “orgs.txt” file to configure a list of the org-specific parameters for the sfsectool to use in order to authenticate and control its behavior. Specifically, the file is in the below format, and it has one line for each Org you want to authorize the tool to analyze:

<admin userid>,<oauth consumer key>,<login url>,<local directory name>,<object-list>

An example of this file is as follows:  
  
don@gmail.com,3MVG9yZ.WNe6byQDbcaQvmckmQApQtRU9umnAxN\_AJbAWi.ocdtA7aZdey6Ql6.65Ul7g8.5qVy8L.PXJJUuz,https://login.salesforce.com,hands,Account:Opportunity:Case

pfendt@dev.com.adm,3MVG98XJQQAccJQe\_9h.KwAg9GRdd.vijrOKpZUYMDOwsMwwl5tY1JUYjxcQmlLaj3gzFhJqiV2ZmyJvSRTuU,https://login.salesforce.com,dev,Account:Opportunity:Case

Note that the <object-list> at the end consists of one or more object names (object API names) separated by colons.

## Downloading and Using the Tool

Download the tool by going to this URL and downloading the binary WAR file inside the “web” directory:

<https://github.com/sfsectool/sfsectool>

To download the WAR file, you can just GIT CLONE this Repo:

git clone https://github.com/sfsectool/sfsectool.git

## Downloading a JEE App Server

You can go to this URL to download the Tomcat JEE app server:

<http://tomee.apache.org/downloads>

Then download the “zip” of the most recent WebProfile version of tomcat. Unzip this file and change directories into the apache-tomee-webprofile-X.Y.Z directory (where X.Y.Z is the tomcat version). Example URL:

<http://repo.maven.apache.org/maven2/org/apache/tomee/apache-tomee/7.0.2/apache-tomee-7.0.2-webprofile.zip>

Add the CATALINA\_HOME and CATALINA\_BASE environment variables to your Unix/Windows shell environment. CATALINA\_HOME should point to the tomee directory where you unzipped the download into. Then create a “tomcatapps” directory as a peer to this directory and set CATALINA\_BASE to point to this tomcatapps directory.

cd $CATALINA\_HOME/../tomcatapps

export CATALINA\_BASE=$PWD

Next, create the following subdirectories under CATALINA\_BASE:

conf

lib

logs

bin

temp

work

webapps

Copy the downloaded “sfsectool.war” file into this $CATALINA\_BASE/webapps directory, and create/copy the orgs.txt file into this same webapps directory.

You now need to edit the “orgs.txt” file that is inside the WAR file. So edit the “sample\_orgs.txt” file to include your Org-specifics, and then rename the file to “orgs.txt” and place it inside this $CATALINA\_BASE/webapps directory. Finally, run the following command (while inside this webapps directory) to update this file inside of the WAR file:

zip -u sfsectool.war

This will update the WAR file with the contents of the orgs.txt file.

Next, copy the server.xml file from $CATALINA\_HOME/conf/server.xml into the $CATALINA\_BASE/conf directory. Then you need to remove the SSL sections/clauses (sections containing “<Connector port=”8443”…/>”. This is an example of one of the clauses you need to remove:

<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol”   
maxThreads="150" SSLEnabled="true"> <SSLHostConfig> <Certificate certificateKeystoreFile="conf/localhost-rsa.jks" type="RSA" xpoweredBy="false" server="Apache TomEE" /> </SSLHostConfig>

</Connector>

Next, add the $CATALINA\_HOME/bin directory to your PATH environment variable:

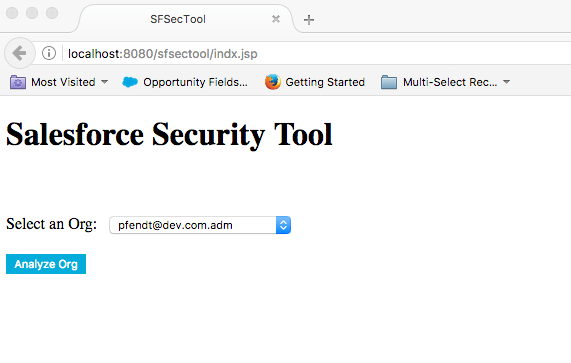
export PATH=$PATH:$CATALINA\_HOME/bin

Finally, you can run “startup.sh” (startup.bat on Windows) to start the tomcat app server. To see the log file while the server is starting, change directories to $CATALINA\_BASE/logs directory and run this command:

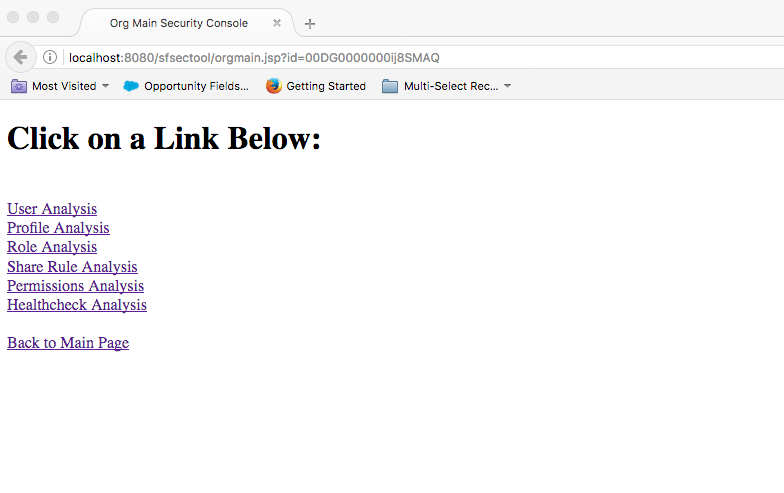
tail -f catalina.out

## Using the Tool

Once the tool starts up successfully, it will have already authenticated to the various orgs you specified in orgs.txt. The tool then presents the user with a list of Orgs (listed/identified by authenticated admin user-names) and the user can select which org to analyze and click on the “Analyze Org” button.

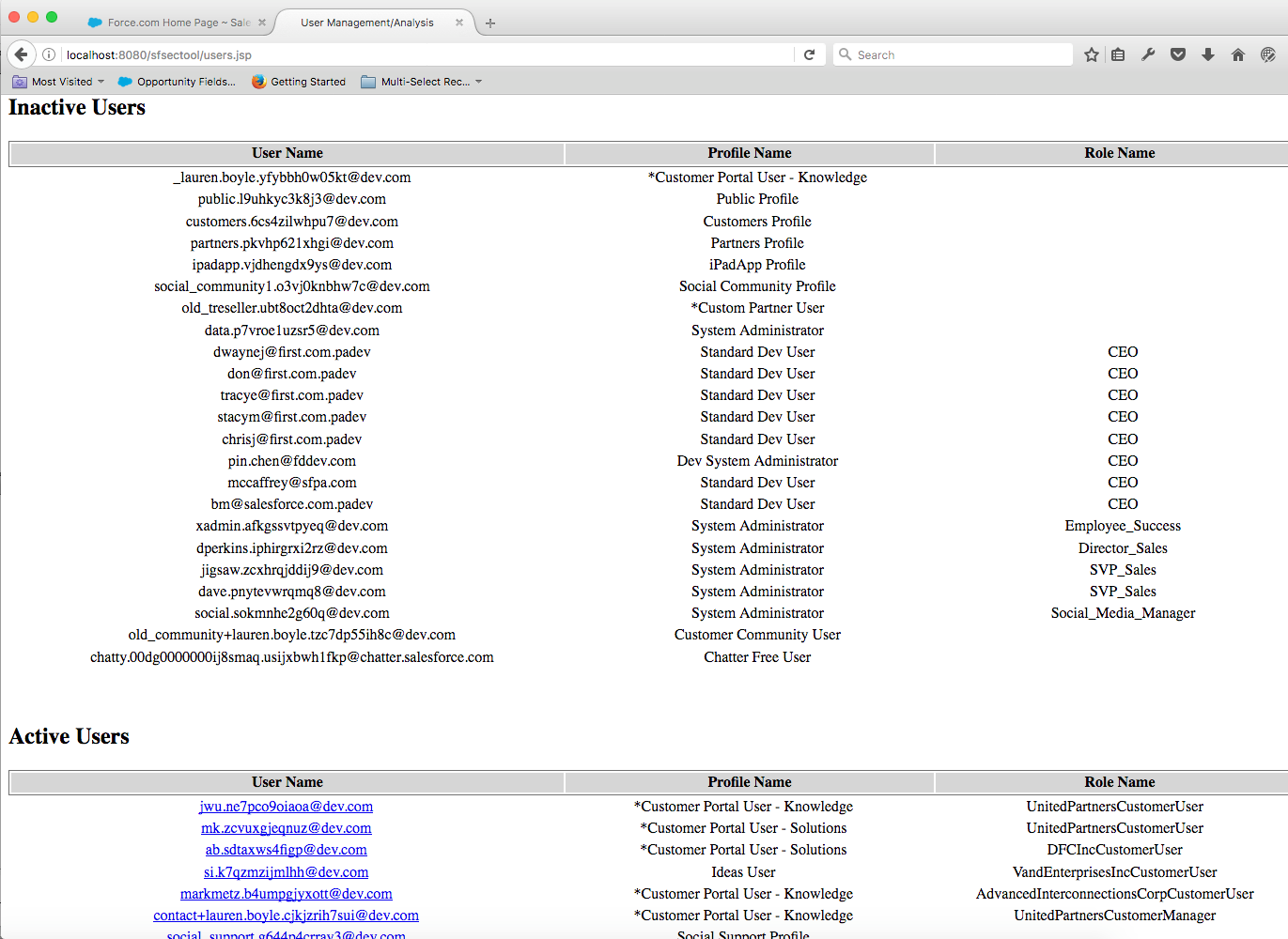


Next, the user is presented with a list of areas to explore – where each area is manifested as a HTML/JSP page:



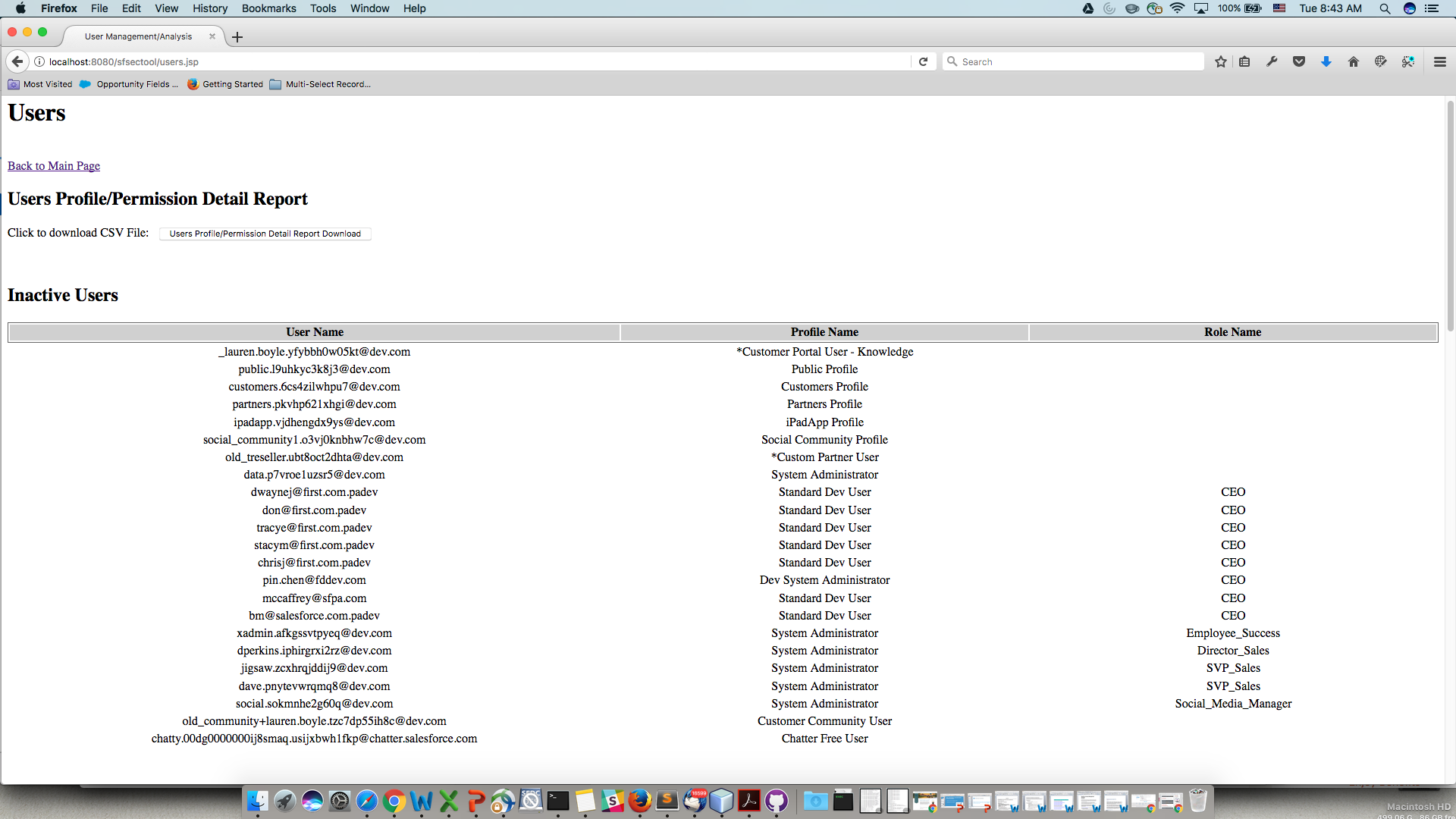
What follows are pictures of these pages.

## User Analysis Page

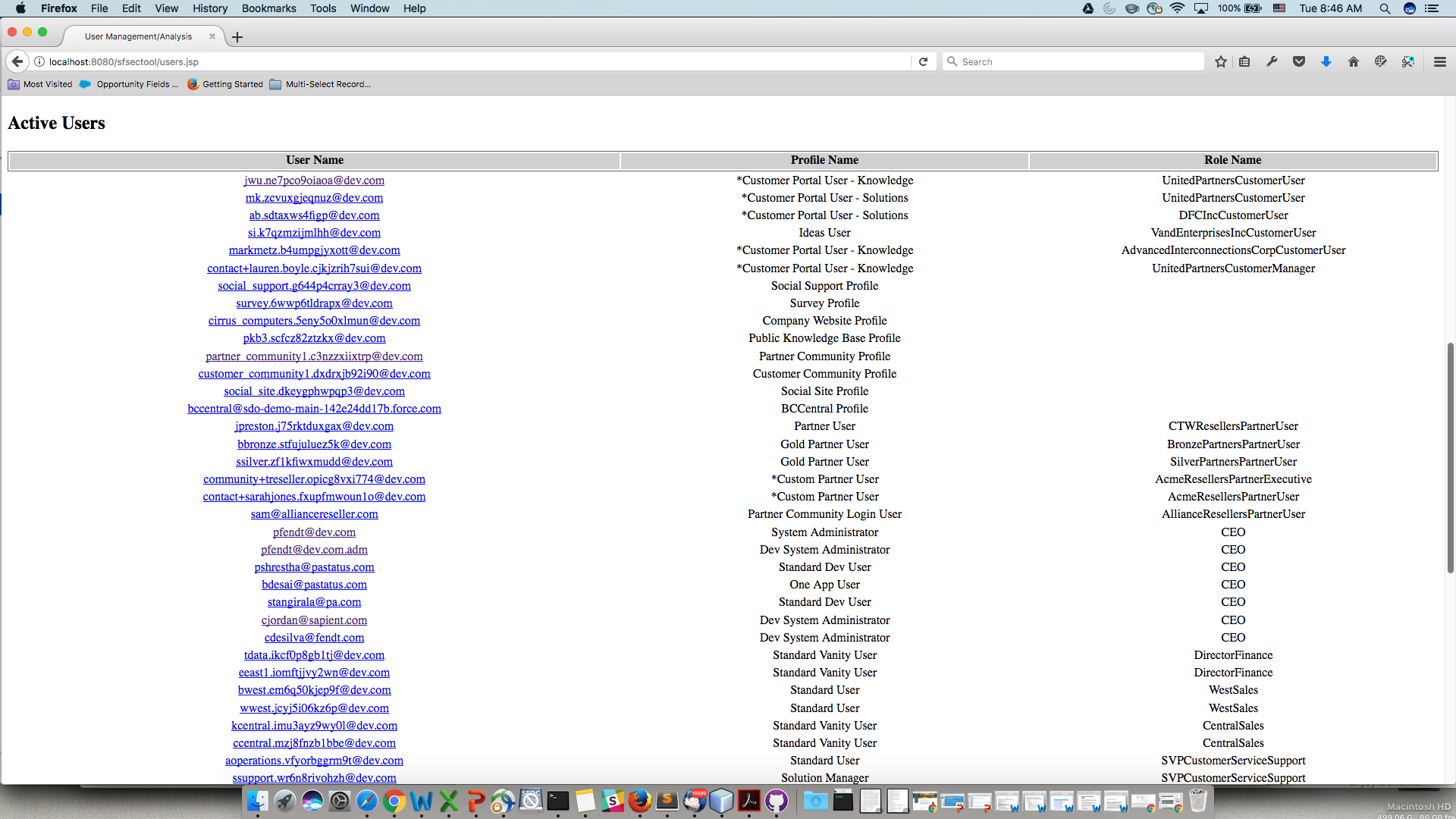


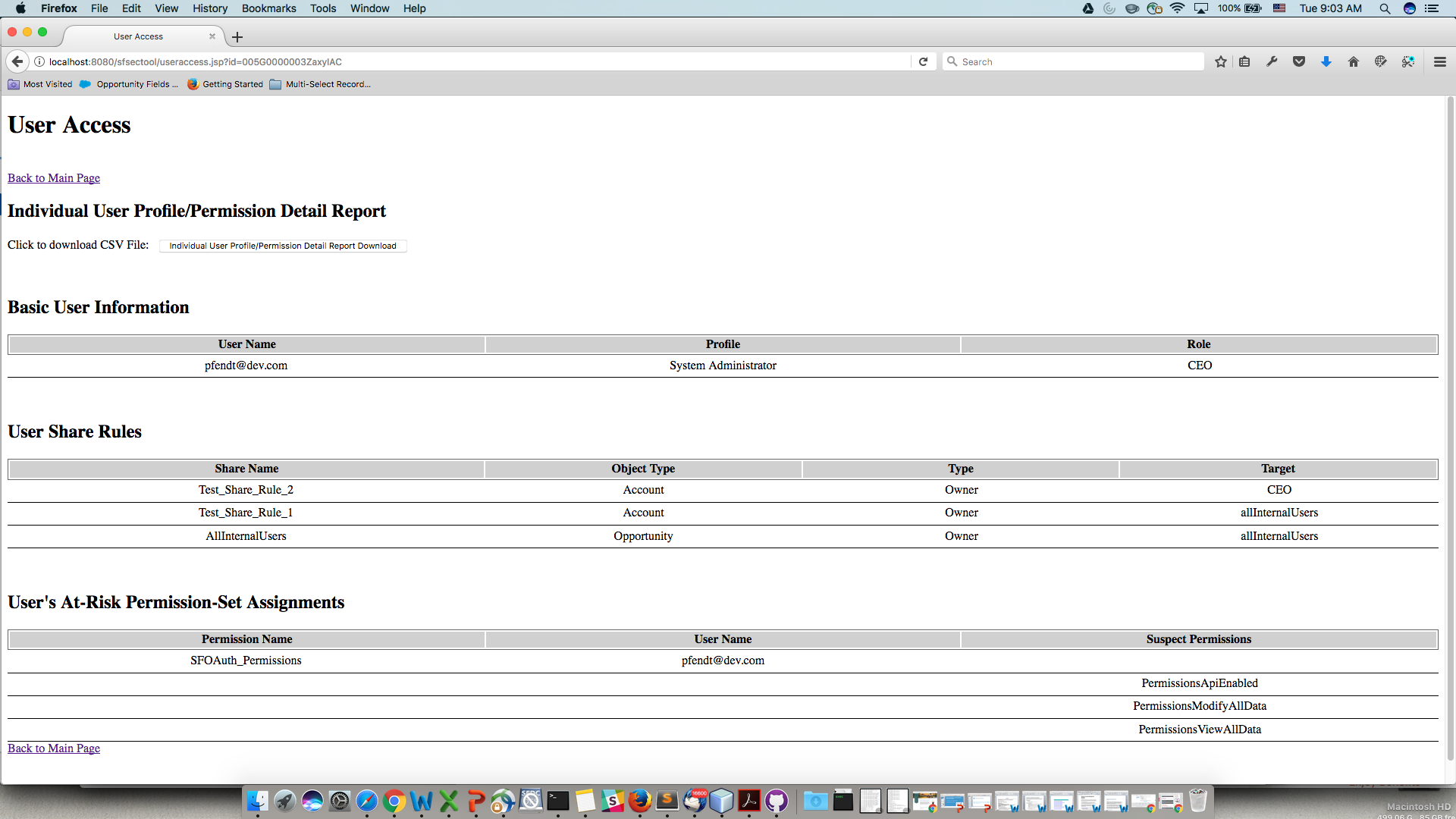
When you click on one of the Active users, you are then taken to the ***User Access*** page providing detailed information on what objects/records a user can see.

## User Access Page

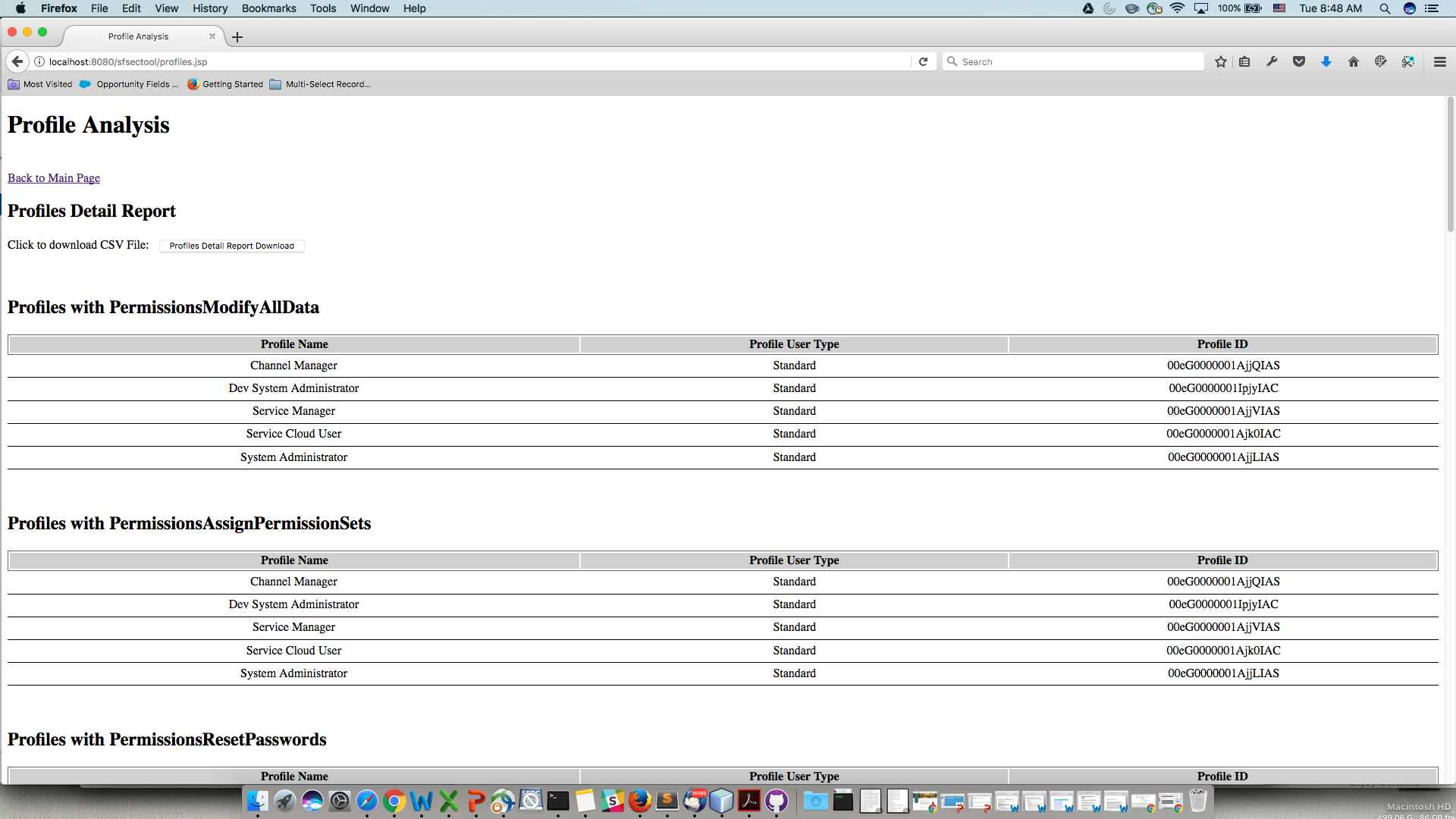


And you can click on individual user below to take you to the User Access page:

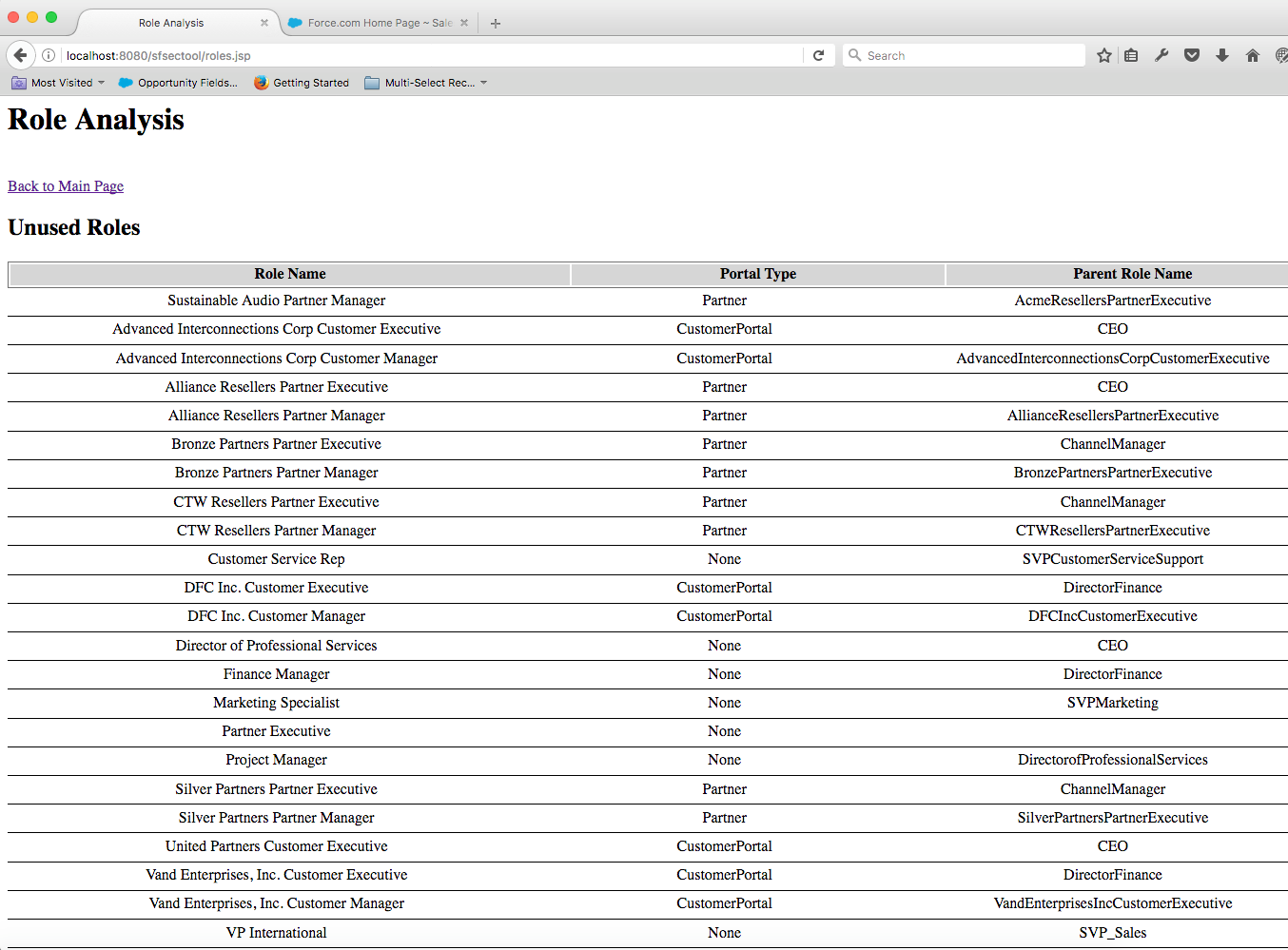




## Profile Analysis Page

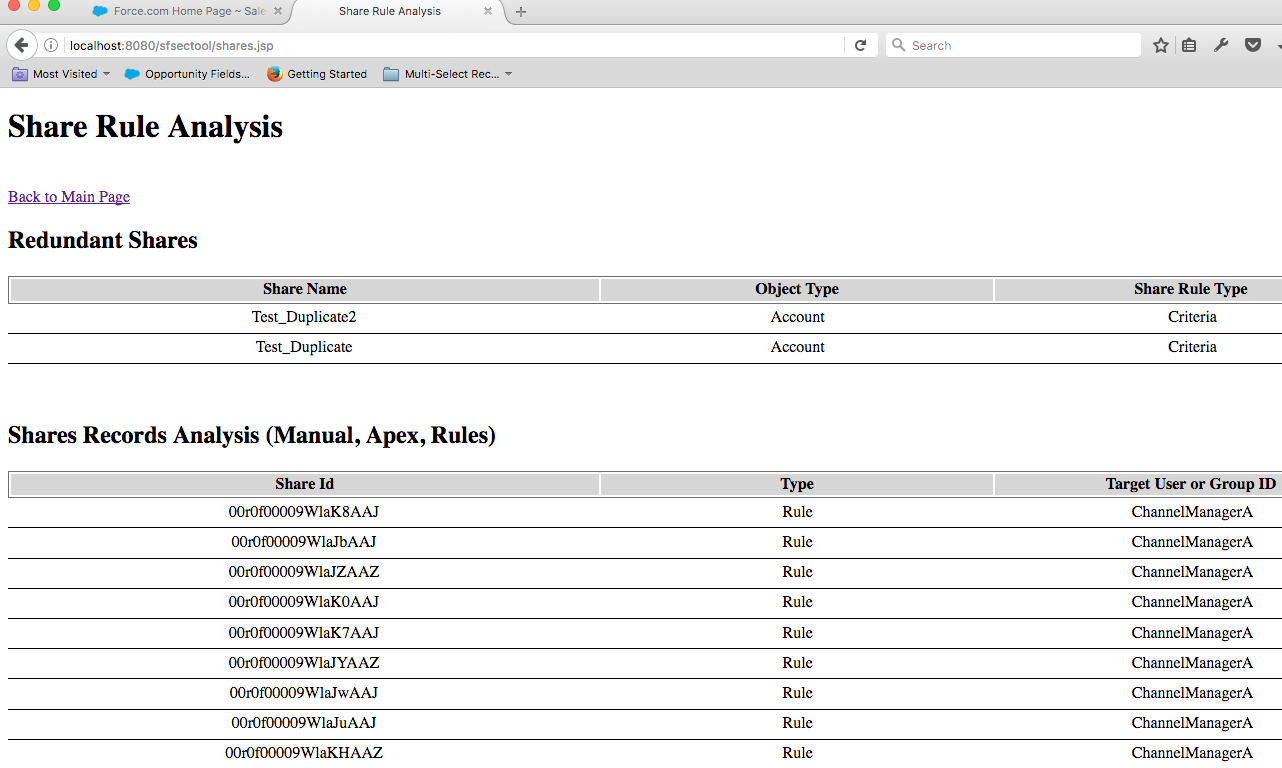


## Role Analysis Page



## Share Rule Analysis Page

More features are coming later this week. For starters, the Share Name rather than ID needs to be displayed, and the User/Group name rather than ID needs to be displayed:



## Permissions Analysis Page

This page highlights elevated or suspicious permissions and which users have them. (There is currently only 1 suspicious PermissionSet assignment in this Org, but page will display N such assignments.)

## 

## Health Check Analysis Page

