# Detect Discrepancy in Permission Assignments

Bala Sundaresan
Security Consultant
Digital Security Group
Cognizant Technology Solutions
65/2, Baghmane Tech Park
C.V.Raman Nagar
Byrasandra
Bangalore, 560 093, India

Anand Upendra
Security Advisor
Digital Security Group
Cognizant Technology Solutions
Glenpointe Centre West
500 Frank W. Burr Blvd
New Jersey NJ 07666 USA

Rama Subramanian Sundaram
Security Architect
Digital Security Group
Cognizant Technology Solutions
Glenpointe Centre West
500 Frank W. Burr Blvd
New Jersey NJ 07666 USA

## ABSTRACT
Most enterprise legacy applications tend to lack supportive or detailed documentation about the user access permissions defined in their systems. When a need arises to clean up data with some or no input about the permissions, a considerable amount of time and effort is consumed in understanding permissions and the discrepancy of their assignment in the system.

This paper proposes an approach which would help mine the discrepancies in permission assignments in a given dataset with useful data representations leading to less time required for the user-access clean-up process. The present work also discusses the implementation of the approach through a case study.

## Keywords
Permission assignment discrepancy, user privilege analysis, RBAC, Role Based Access Control, entitlement discrepancy, user authorization analysis

## 1. INTRODUCTION
Role Based Access Control (RBAC) is one of the most popular methods used in enterprises to manage user authorizations. In RBAC, permissions are assigned to roles which in turn are given to a user based on his need to perform his duties on the system. RBAC authorizes users' access based on his role. Roles can be referred as logical grouping of permissions which are assigned to users. Different roles are created based on the job functions available in the enterprise. RBAC provides a powerful way of managing the authorization requirements for different users based on their line of work.

One of the challenges in RBAC systems is to identify the logical group of permissions for each role. Permissions assigned to a role could result in conflict depending on the permission type. Permissions can be classified based on their purpose and usage.

- Critical permissions which are not to be shared between roles are defined as Disjoint permissions. Disjoint permissions are not shared between Segregation of Duties (SOD) roles. This prevents creation of a senior role with permissions common to roles defined as SOD roles.
- Permissions which exhibit conflict of interests based on the authorizations they provide are categorized as Conflicting permissions. Business rules would aid in defining permissions which are conflicting in nature. Roles defined as SOD roles do not to share the conflicting permissions.
- Certain permissions cannot exist by themselves. Say to access a text box in a page; the role would also need access to the page. It is meaningless to assign the authorization of only the text box to a role without the access to the page. Such permissions are referred as pre-requisite or dependent permissions.
- Certain permissions are assigned only to a single role. Say a Database Administrator (DBA) role alone can have super user access to the databases. Such permissions are referred to as Permission Assigned to a Single Role(PASR).

In this paper we present a case study that was performed to identify the possible conflicts between the Pre-requisite Permission Assignments in a RBAC system. This case study is not extended to check other permission assignment discrepancies. The write-up is organized into three sections. The first section has the methodology proposed to identify the conflicts. The second section details the observations from its applicability on real-time data. Conclusions are presented in the last section.

The intended audience for this work would include, but not restricted to, Security Architects, Design and Development Engineers, Application Managers, IT Security Managers mainly responsible for Access Control and compliance

## 2. METHODOLOGY
We propose a methodology for identifying the discrepancies in pre-requisite permission assignment based on a data visualization technique called Dendrogram. Dendrogram is a tree diagram which depicts the data in clusters. The clusters are further studied to analyze the permission groupings.

Dendrogram is constructed by applying clustering algorithm and Linkage method on Proximity Matrix. Proximity matrix represents the similarity between the entities in the raw data calculated using the math function called similarity/dissimilarity measure. Clustering algorithm defines the logic used for deriving the entity clusters from the proximity matrix. Linkage method defines the logic to connect the clusters. This is illustrated in Figure 1.

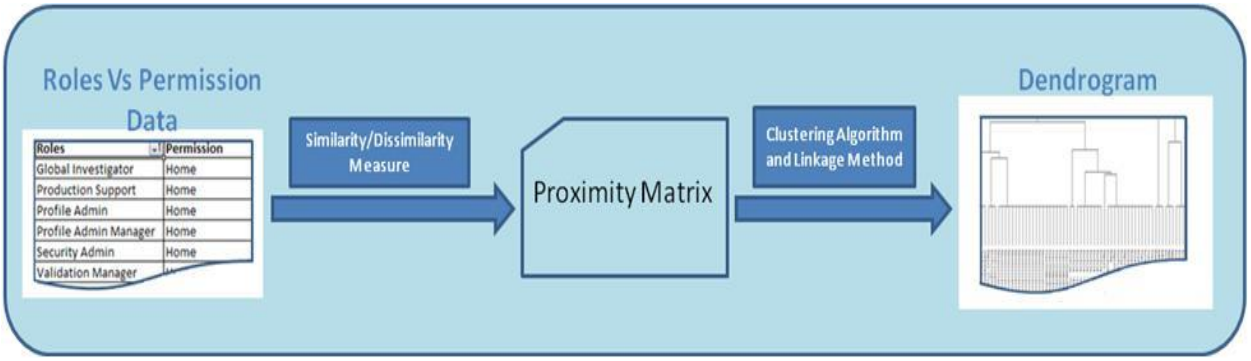


**Figure 1 - Data Visualization using dendrogram**

1. Get the data set defining RolesVsPermissions
2. Construct a dendrogram
3. Calculate the Proximity Matrix using the Similarity/Dissimilarity measure
4. Cluster the output using the Clustering algorithm and Linkage Method
5. Traverse from the most probable cluster to the root.
6. Verify the requisite permissions of any permission say p. If p is at nth level of the cluster then all its requisite permissions are available at nth or <nth level of the cluster

Any violation at step 4 would mean that p is at > nth level in the cluster and hence a discrepancy in the assignment of the dependent permission.

The rationale behind this, for any entitlement,

> Number of child assignments will be <= Number of parent assignments

As a result the pre-requisite permission always occurs before the child permissions in the clusters, when moving from the basic cluster to the root.

## 3. CASE STUDY FOR CONFLICT DETECTION

This case study illustrates the following two scenarios:

1. Data set with discrepancies in Permission Assignments
2. Data set with discrepancies eliminated

The Roles Vs. Permission data set is taken from a sample application that manages RBAC in a financial enterprise.

To plot the Dendrogram the Similarity Measure used is "Similarity ratio" and clustering algorithm follows "Hierarchical Clustering".

### 3.1 Scenario 1 - Permission Assignment Discrepancy

Figure 2 represents the dendrogram plotted to view the results on a data set in which certain roles do not have the pre-requisite permissions assigned. Traversal path in the dendrogram is from the most probable cluster represented in the dendrogram as 1 to the last cluster represented in dendrogram as 11.

Discrepancies identified through the dendrogram,

1. Cluster 1 has entitlement "Profile – ViewProfileMemberDetailsForUser", but its pre-requisite permission "Profile" is at cluster 3.
2. Cluster 1 has entitlement "Report-OpenReport", but its pre-requisite permission "Report" is at cluster 2
3. Cluster 6 has entitlements of Reconciliation, but its pre-requisite permission "Reconciliation" is at cluster 10
4. Entitlement Home does not indicate any improper assignments though it would be expected to appear at cluster 1

### 3.2 Scenario 2 - Permission Assignment Discrepancy Eliminated

The above data set is modified to correct the identified discrepancies by adding the requisite permissions to the roles

- Entitlement "Profile" assigned to role "Profile Admin"
- Entitlement "Report" assigned to role "Profile Admin Manager"
- Entitlement "Reconciliation" assigned to roles "Profile Admin Manager" and "Global Investigator"
- Entitlement "Home" assigned to roles "Global Investigator" , "Profile Admin"

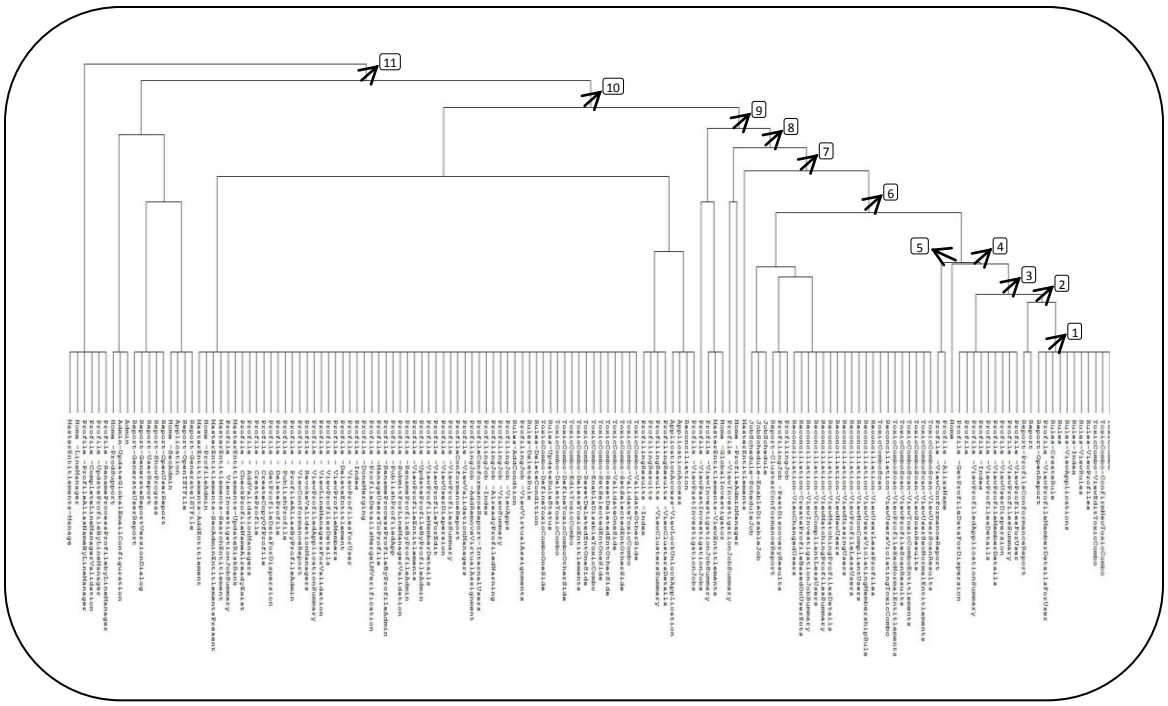


**Figure 2 – Dendrogram with improper assignments [Entities listed in Appendix B]**

Figure 3 represents the dendrogram plotted to view the results on a Data set with no discrepancies in Permission Assignments. Traversal path is taken from the most probable cluster marked 1 to the root cluster marked 11. All the permission assignments are found to be consistent.

## 3.3 SOD Conflict Detection

This approach can be extended to identify SOD violations by plotting dendrogram of roles.

Any two roles grouped together depict the commonness in the permissions shared between them.

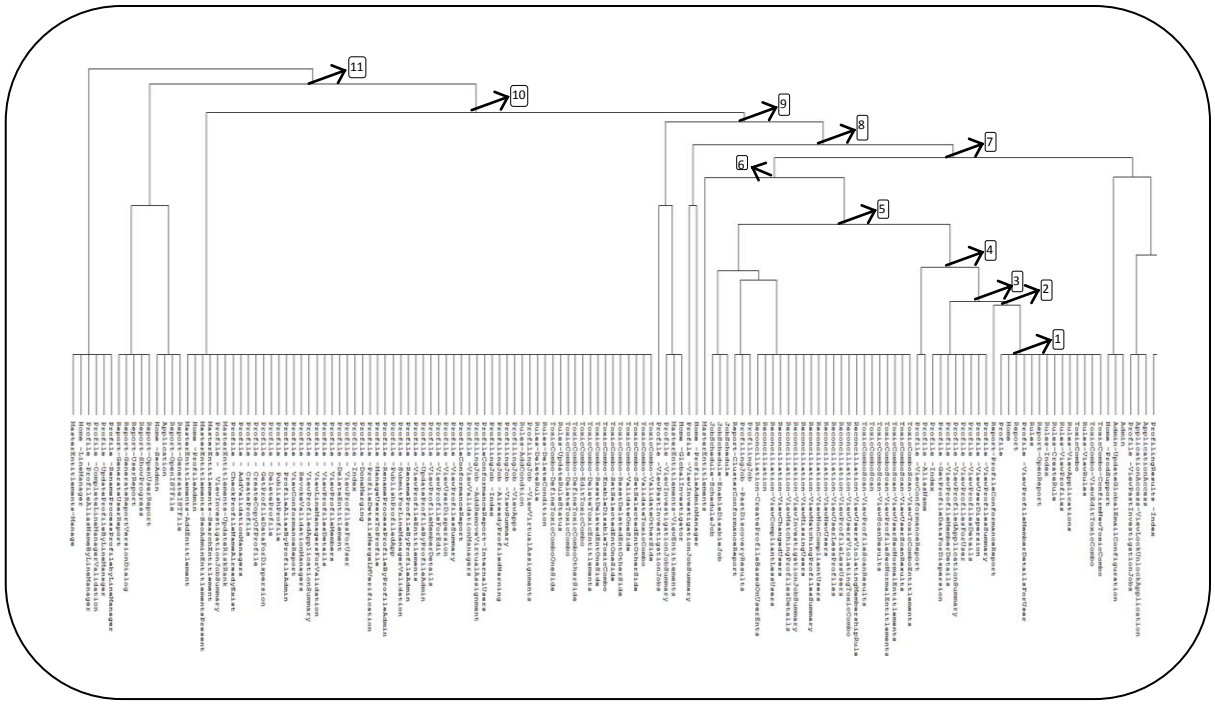


**Figure 3 – Dendrogram with proper assignments [Entities listed in Appendix B]**

Shorter the distance, greater the number of shared permissions

the same pattern of distribution across the roles, they are grouped under one cluster when the dendrogram is plotted. This would help revisit and validate their distribution across the roles. The basic clusters have been numbered from 1 to 14 in the

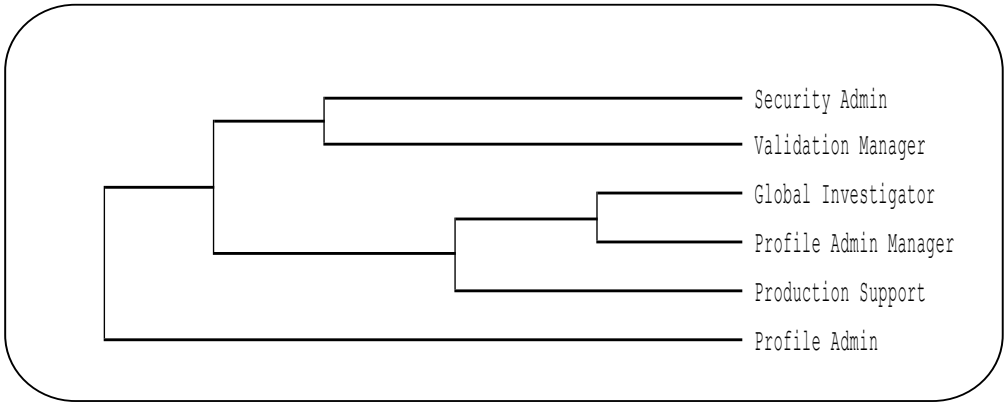


**Figure 4 – Dendrogram of Roles**

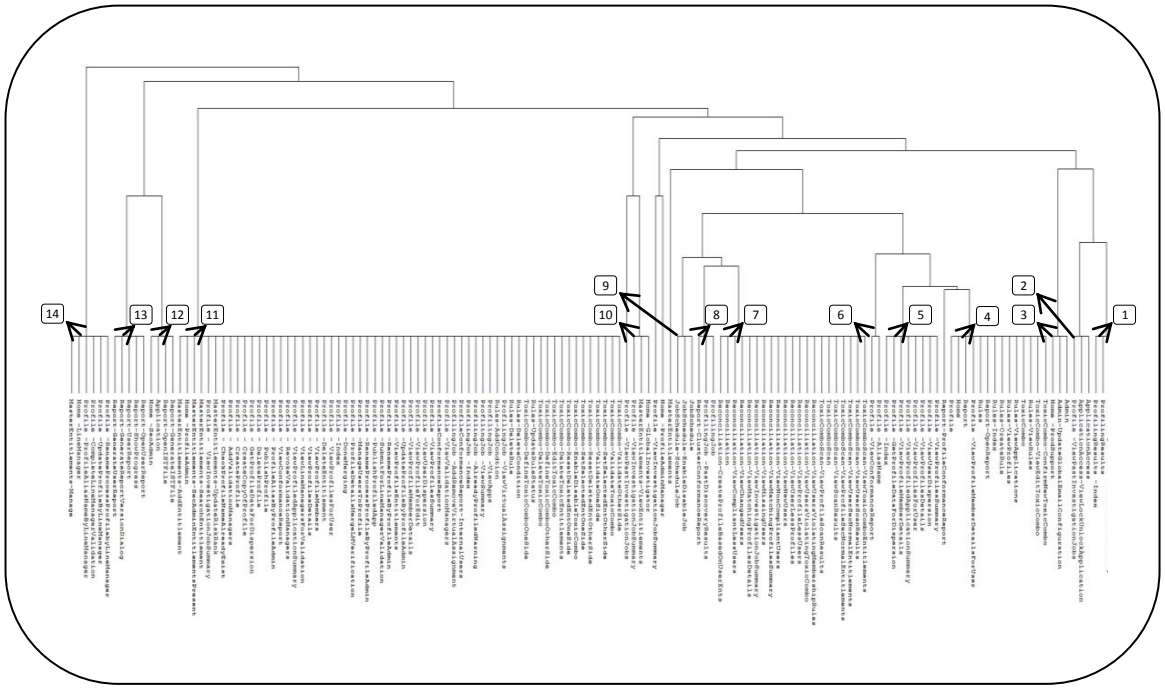


**Figure 5 - Basic grouping of permissions [Entities listed in Appendix B]**

From the dendrogram in figure 4, we see that "Global Investigator" has entitlements common with "Profile Admin Manager" and hence these roles cannot be SOD. This process will need the SOD roles as input to detect the violation.

## 3.4 Bottom Up Permission Analysis

Dendrogram aids in bottom up permission analysis. One useful finding using this approach would include identifying the basic group of permission which occurs in the clusters together. Say p and q are two permissions defined in the system. If p and q have

dendrogram shown in Figure 5.

# 4. CONCLUSION

This paper explains how the proposed methodology was adopted to check the discrepancies in a data set with improper permission assignments. The traversal from the basic to the root cluster aided in identifying the pre-requisite permission discrepancies. The identified discrepancies were corrected and the same reflected in the dendrogram plotted with the corrected

values. The methodology facilitates identification of discrepancies and depends on the business inputs to identify the violations.

The proposed approach combines statistical inputs to visualize the relation between the entities defined in the system. The visual representation of data provides the user with an easy representation to analyze the data, thereby reducing the time required to identify the discrepancies. The approach identifies discrepancies in pre-requisite permission assignment and can be extended to SOD roles analysis.

# 5. REFERENCES

[1] Chang-Joo Moon, Woojin Paik, Young-Gab Kim and Ju-Hum Kwon. "The Conflict Detection Between Permission Assignment Constraints in Role-Based Access Control" in Proc. Information Security and Cryptology Lecture Notes in Computer Science, 2005, Volume 3822/2005,pp. 265-278.

[2] Lengler R., Eppler M. "Towards A Periodic Table of Visualization Methods for Management," in Proc IASTED Proceedings of the Conference on Graphics and Visualization in Engineering, 2007

[3] "Cluster Analysis." Internet: http://www.statsoft.com/textbook/cluster-analysis/?button=1 [Jan. 05 2011]

[4] "Find Clusters." Internet: http://reference.wolfram.com/mathematica/ref/FindClusters.html [Feb 16 2011]

[5] "Hierarchical Clustering Algorithms." Internet: http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/hierarchical.html [Mar. 15 2011]

# 6. APPENDICES

## 6.1 Appendix A : Application Background

The entitlements used in the case study are from a custom developed application that manages the roles of a financial services enterprise. The application supports creation of profiles, analysis of clusters, approval workflows, generation of report based on user needs. The key functionalities used in the application are given below.

1) Home
2) ProfilingJob
3) ProfilingResults
4) Profile
5) Report
6) ProfileConformanceReport
7) MasterEntitlements
8) Assignments
9) Application
10) Admin
11) ApplicationAccess
12) JobSchedule
13) Rules
14) ToxicComboScan
15) ToxicCombo
16) Reconciliation

The application consists of six roles with access to the entitlements.

1) Global Investigator
2) Production Support
3) Profile Admin
4) Profile Admin Manager
5) Security Admin
6) Validation Manager

## 6.2 Appendix B : Entitlement Details

Entities from right to left in "FIGURE 2 – Dendrogram with improper assignments".

1) ToxicCombo
2) ToxicCombo-ConfirmNewToxicCombo
3) ToxicCombo-ViewEditToxicCombo
4) Rules-ViewProfiles
5) Rules-ViewRules
6) Rules-Index
7) Rules-ViewApplications
8) Rules
9) Rules-CreateRule
10) Profile -ViewProfileMemberDetailsForUser
11) Report-OpenReport
12) Report
13) Report-ProfileConformanceReport
14) Profile -ViewProfilesForUser
15) Profile -ViewProfilesSummary
16) Profile -ViewUserDispersion
17) Profile -ViewProfileMemberDetails
18) Profile -ViewProfileDetails
19) Profile -Index
20) Profile -ViewProfiledApplicationSummary
21) Profile
22) Profile -GetProfileDataForDispersion
23) Home
24) Profile -AliasName
25) Profile -ViewConformanceReport
26) ToxicComboScan-ViewUserScanResults
27) ToxicComboScan-ViewUserSecNormalEntitlements
28) ToxicComboScan-ViewScanResults
29) ToxicComboScan-ViewToxicComboEntitlements
30) ToxicComboScan-ViewProfileScanResults
31) ToxicComboScan-ViewProfieSecNormalEntitlements
32) Reconciliation-ViewUsersViolatingToxicCombo
33) ToxicComboScan
34) Reconciliation-ViewUserLessProfiles
35) Reconciliation-ViewUsersViolatingMembershipRules
36) Reconciliation-ViewNonCompliantUsers
37) Reconciliation-ViewProfileLessUsers
38) Reconciliation-VIewMissingUsers
39) Reconciliation-ViewNewUsers
40) Reconciliation-ViewMatchingProfilesDetails
41) Reconciliation-ViewMatchingProfilesSummary
42) Reconciliation-ViewCompliantLessUsers
43) Reconciliation-ViewInvestigationJobSummary
44) Reconciliation-CreateProfileBasedOnUserEnts
45) Reconciliation-ViewChangedUsers
46) ProfilingJob

47) ProfilingJob -PastDiscoveryResults
48) Report-ClusterConformanceReport
49) JobSchedule
50) JobSchedule-EnableDisableJob
51) JobSchedule-ScheduleJob
52) MasterEntitlements
53) Home -ProfileAdminManager
54) Profile -ViewInvestigationJobSummary
55) Home -GlobalInvestigator
56) MasterEntitlements-ViewEntitlements
57) Profile -ViewInvestigationJobSummary
58) Profile -ViewPastInvestigationJobs
59) Profile -ViewPastInvestigationJobs
60) Reconciliation
61) ApplicationAccess
62) ApplicationAccess-ViewLockUnlockApplication
63) ProfilingResults -ViewClustersDetails
64) ProfilingResults -ViewClustersSummary
65) ProfilingResults
66) ProfilingResults -Index
67) ToxicCombo-ValidateOtherSide
68) ToxicCombo-ValidateToxicCombo
69) ToxicCombo-SetSelectedEntOtherSide
70) ToxicCombo-ValidateOneSide
71) ToxicCombo-ResetDeletedEntOtherSide
72) ToxicCombo-SetSelectedEntOneSide
73) ToxicCombo-EnableDisableToxicCombo
74) ToxicCombo-ResetDeletedEntOneSide
75) ToxicCombo-DeleteToxicEntitlements
76) ToxicCombo-EditToxicCombo
77) ToxicCombo-DefineToxicComboOtherSide
78) ToxicCombo-DeleteToxicCombo
79) Rules-UpdateRuleStatus
80) ProfilingJob -ViewVirtualAssignments
81) Rules-AddCondition
82) ProfilingJob -ViewApps
83) ProfilingJob -ViewSummary
84) ProfilingJob -AlreadyProfiledWarning
85) ProfilingJob -Index
86) ProfileConformanceReport-InternalUsers
87) ProfilingJob -AddRemoveVirtualAssignment
88) Profile -ViewValidationManagers
89) ProfileConformanceReport
90) Profile -ViewProfileSummary
91) Profile -ViewUserDispersion
92) Profile -ViewProfileForEdit
93) Profile -ViewProfileMemeberDetails
94) Profile -UpdateProfileByProfileAdmin
95) Profile -ViewProfileEntitlements
96) Profile -RenameProfileByProfileAdmin
97) Profile -SubmitForLineManagerValidation
98) Profile -PublishProfiledApp
99) Profile -RenameProcessProfileByProfileAdmin
100) Profile -ManageUsersToProfile
101) Profile -ProfileDetailsMergeLMVerification
102) Profile -DoneMerging
103) Profile -Index
104) Profile - ViewProfilesForUser
105) Profile -DeleteEntitlement
106) Profile - ViewProfileMembers
107) Profiles - ViewProfilesDetails
108) Profile - ViewLineManagersForValidation
109) Profile - ViewProfiledApplicationSummary
110) Profile - RevokeValidationManagers
111) Profile - ViewConformanceReport
112) Profile - ProfileAliasByProfileAdmin
113) Profile - PublishProfile
114) Profile - DeleteProfile
115) Profile - GetProfileDataForDispersion
116) Profile - CreateCopyOfProfile
117) Profile - CreateProfile
118) Profile - AddValidationManagers
119) Profile - CheckProfileNameAlreadyExist
120) MasterEntitlements-UpdateRiskRank
121) Profile - ViewInvestigationJobSummary
122) MasterEntitlement-SearchEntitlement
123) MasterEntitlements-SecAdminEntitlementsPresnt
124) Home -ProfileAdmin
125) MasterEntitlement-AddEntitlement
126) Report-GenerateISTFile
127) Report-OpenISTFile
128) Application
129) Home -SecAdmin
130) Report-OpenUserReport
131) Report-ShowProgress
132) Report-UserReport
133) Report-GenerateReportVersionDialog
134) Report-GenerateUserReport
135) Admin
136) Admin-UpdateGlobalEmailConfiguration
137) Hoem -ProdSupprot
138) Profile -RenameProcessProfilebyLineManager
139) Profile - UpdateProfileByLineManager
140) Profile -CompleteLineManagerValidation
141) Profile -ProfileAliasNameByLineManager
142) Home -LineManage
143) MasterEntitlements-Manage

Entities from right to left in "FIGURE 3 – Dendrogram with proper assignments" and "FIGURE 5 – Basic grouping of permissions".

1) ProfilingResults -ViewClustersDetails
2) ProfilingResults -ViewClustersSummary
3) ProfilingResults
4) ProfilingResults -Index
5) ApplicationAccess
6) ApplicationAccess-ViewLockUnlockApplication
7) Profile -ViewPastInvestigationJobs
8) Admin
9) Admin-UpdateGlobalEmailConfiguration
10) Home -ProdSupport
11) ToxicCombo-ConfirmNewToxicCombo
12) ToxicCombo-ViewEditToxicCombo
13) Rules-ViewRules
14) ToxicCombo
15) Rules-ViewApplications
16) Rules-ViewProfiles
17) Rules-CreateRule
18) Rules-Index
19) Report-OpenReport
20) Rules
21) Profile -ViewProfileMemberDetailsForUser
22) Report

23) Home
24) Profile
25) Report-ProfileConformanceReport
26) Profile -ViewProfilesSummary
27) Profile -ViewUserDispersion
28) Profile -ViewProfilesDetails
29) Profile -ViewProfilesForUser
30) Profile -ViewProfiledApplicationSummary
31) Profile -ViewProfileMemberDetails
32) Profile -GetProfileDataForDispersion
33) Profile -Index
34) Profile -AliasName
35) Profile -ViewConformanceReport
36) ToxicComboScan-ViewToxicComboEntitlements
37) ToxicComboScan-ViewUserScanResults
38) ToxicComboScan-ViewUserSecNormalEntitlements
39) ToxicComboScan-ViewProfileSecNormalEntitlements
40) ToxicComboScan-ViewScanResults
41) ToxicComboScan
42) ToxicComboScan-ViewProfileScanResults
43) Reconciliation-ViewUsersViolatingMembershipRules
44) Reconciliation-ViewUsersViolatingToxicCombo
45) Reconciliation-ViewProfileLessUsers
46) Reconciliation-ViewUserLessProfiles
47) Reconciliation-ViewNewUsers
48) Reconciliation-ViewNonCompliantUsers
49) Reconciliation-ViewMatchingProfilesSummary
50) Reconciliation-ViewMissingUsers
51) Reconciliation-ViewInvestigationJobSummary
52) Reconciliation-ViewMatchingProfilesDetails
53) Reconciliation-ViewChangedUsers
54) Reconciliation-ViewCompliantLessUsers
55) Reconciliation
56) Reconciliation-CreateProfileBasedOnUserEnts
57) ProfilingJob
58) ProfilingJob -PastDiscoveryResults
59) Report-ClusterComformanceReport
60) JobSchedule
61) JobSchedule-EnableDisableJob
62) JobSchedule-ScheduleJob
63) MasterEntitlements
64) Home -ProfileAdminManager
65) Profile -ViewInvestigationJobSummary
66) Home -GlobalInvestigator
67) MasterEntitlements-ViewEntitlements
68) Profile -ViewInvestigationJobSummary
69) Profile -ViewPastInvestigationJobs
70) ToxicCombo-ValidateOtherSide
71) ToxicCombo-ValidateToxicCOmbo
72) ToxicCombo-SetSelectedEntOtherSide
73) ToxicCombo-ValidateOneSide
74) ToxicCombo-ResetDeletedEntOtherSide
75) ToxicCombo-SetSelectedEntOneSide
76) ToxicCombo-EnableDisableToxicCombo
77) ToxicCombo-ResetDeletedEntOneSide
78) ToxicCombo-DeleteToxicEntitlements
79) ToxicCombo-EditToxicCombo
80) ToxicCombo-DefineToxicComboOtherSide
81) ToxicCombo-DeleteToxicCombo
82) Rules-UpdateRuleStatus
83) ToxicCombo-DefineToxicComboOneSide
84) Rules-DeleteCondition
85) Rules-DeleteRule
86) ProfilingJob -ViewVirtualAssignments
87) Rules-AddCondtion
88) ProfilingJob -ViewApps
89) ProfilingJob -ViewSummary
90) ProfilingJob -AlreadyProfiledWarning
91) ProfilingJob -Index
92) ProfileConformanceReport-InternalUsers
93) ProfilingJob -AddRemoveVirtualAssignment
94) Profile -ViewValidationManagers
95) ProfileConformanceReport
96) Profile -ViewProfilesSummary
97) Profile -ViewUserDispersion
98) Profile -ViewProfileForEdit
99) Profile -ViewProfileMemberDetails
100) Profile -UpdateProfileByProfileAdmin
101) Profile -ViewProfileEntitlements
102) Profile -RenameProfileByProfileAdmin
103) Profile -SubmitForLineManagerValidation
104) Profile -PublishProfiledApp
105) Profile -RenameProcessProfileByProfileAdmin
106) Profile -ManageUsersToProfile
107) Profile -ProfileDetailsMergeLMVerification
108) Profile -DoneMerging
109) Profile -Index
110) Profile - ViewProfilesForUser
111) Profile -DeleteEntitlement
112) Profile - ViewProfileMembers
113) Profile - ViewProfilesDetails
114) Profile - ViewLineManagersForValidation
115) Profile - ViewProfiledApplicationSummary
116) Profile - RevokeValidationManagers
117) Profile - ViewConformanceReport
118) Profile - ProfileAliasByProfileAdmin
119) Profile - PublishProfile
120) Profile - DeleteProfile
121) Profile - GetProfileDataForDispersion
122) Profile - CreateCopyOfProfile
123) Profile - CreateProfile
124) Profile - AddValidationManagers
125) Profile - CheckProfileNameAdreadyExist
126) MasterEntitlements-UpdateRiskRank
127) Profile - ViewInvestigationJobSummary
128) MasterEntitlements-SearchEntitlement
129) MasterEntitlements-SecAdminEntitlementsPresent
130) Home -ProfileAdmin
131) MasterEntitlements-AddEntitlement
132) Report-GenerateISTFile
133) Report-OpenISTFile
134) Application
135) Home -SecAdmin
136) Report-OpenUserReport
137) Report-ShowProgress
138) Report-UserReport
139) Report-GenerateReportVersionDialog
140) Report-GenerateUserReport
141) Profile -RenameProcessProfileByLineManager
142) Profile -CompleteLineManagerValidation
143) Profile -ProfileAliasNameByLineManager
144) Home -LineManager
145) MasterEntitlements-Manage

## 6.3 Appendix C : Glossary Of Technical Terms

1. Clustering : Clustering is the process of grouping similar or dissimilar objects into clusters.

2. Dendrogram : A tree diagram used to illustrate the arrangement of clusters.

3. Proximity Matrix : A square matrix in which the entry in cell (j, k) is some measure of the similarity (or distance) between the items to which row j and column k correspond.

4. Linkage Method : Linkage methods cluster the objects based on the distance between them.