Roger Jordan

CS1699

Project 2

Access Control Language

**W0:** For this assignment I chose to use a version of SD3. SD3 supports the use of groups letting it support indirection. SD3 also has the built in $ notation, allowing it to support delegation. Role inheritance and \_SADPLASDPSLD is also supported in this version of SD3.

**W1:** The syntax used to write the policy is as follows:

**NOTE:** Options under notation that are in *italics* are optional

|  |  |  |  |
| --- | --- | --- | --- |
| **Policy** | **Notation (LHS)** | **Delim** | **Notation (RHS)** |
| **Users** | PKD(“username”, userkey) | :- | ; |
|  | PKD(user, key) | :- | UserKey$PKD(user, key); |
| **Groups** | Group( “groupname”, “user-to-add”) | :- | ; |
| **Permissions** | Perms(“user”, “filename”, “Permission*”)* | :- | ; |
| (All user of group have permission for file) | Perms(user, “filename”, “Permission”) | :- | Group(“group”, user); |
| The user is trusted to edit permissions. | Perms(user, resource, op) | :- | PKD("username", key); |
| If the “username” has the permission for resource on RHS then permission of resource is granted to LHS user | Perms(“user”, “resource”, “Permission”) | :- | PKD(“username”, “key”), “username”$perm(user, “resource”, “Permission”); |
| **Group Hierarchy** | SubGroup(“parentgroup”, “childgroup”) | :- | ; |
| **Trust**  ( | T(“user”, “file”) | :- | Key$E(“user”, “file”); |
| (x trusts y if x trust z and z trust y) where | T(x, y) | :- | T(x, z), T(z, y); |