Role Explosion

Role-Based Access Control (RBAC) is a guiding model for many identity management deployments. The problem is that traditional static RBAC model does not scale. RBAC is fine for few tens of roles. But as the number of connected systems grow, the number of roles grows as well. Organization with thousand employees can easily end up with few of thousands of roles. The difficult problem of managing thousand employees will be transformed to even more difficult problem of managing few thousands of roles.

The reasons are quite understandable but they are far from being obvious:

- **Cartesian product:** Let's have roles for bank teller, supervisor and branch director. Let's operate in London, Berlin and Bratislava. We need a role for "teller in London", "teller in Berlin", ... 9 roles total. The number of roles grows very quickly.
- **Atomization:** Cartesian product can be sometimes avoided by decomposing roles to re-usable re-combinable units. Combine them to a "higher-order" roles using role hierarchy. This gives us 6 roles instead of 9. Unless the teller shares part of the access rights with janitor. Then we need "basic branch office access" role, "teller's IT access" role and a "teller" role that combines these two. Now we have 8 roles. The number of roles still grows quite quickly.
  Uniqueness: Many employees have one-of-a-kind set of access rights. Any attempt to fit them into the roles will inevitably result in a number of roles that is equal or greater than the number of such employees. This is quite pointless.

This is known as *role explosion*. It is a severe disease of most IDM projects. A project that started with good intention to simplify user management will end up with a role structure that is much more difficult to maintain then it was before the project.

Static RBAC model usually cannot be used to efficiently handle role explosion. There are some solutions, but none of it is a panacea:

- **ABAC - Attribute-Based Access Control:** The principle is that instead of having roles there is a expression that computes access rights. The expression takes user attributes as parameters. It seems to be very flexible. But it is a new concept and has yet to be proven. Also, it replaces formal structure of roles with logic. That means programming instead of configuration. It is also very difficult to apply to provisioning. ABAC may solve the problem of role management, but does it also solve the generic problem of policy complexity and management?
- **RBAC, but forget about least privilege:** Create "bigger" roles than what would the principle of least privilege dictate. Create roles that grant much more permissions than they should grant. This improves role re-use and keeps the number of roles manageable. Although it results in a less secure situation it may be good approach for some environments.
- **Spice up roles with some logic:** That usually means adding *rules* to make roles more generic. E.g. instead of creating "teller in London" create a parametric role "teller" that takes locality from user profile as a parameter and computes the path to the home directory. This is somewhere between ABAC and static RBAC. It has good features from both approaches and seems to be the most popular choice for provisioning systems.
- **Manage the chaos:** Let users request access rights, let others approve it. The obvious problem is how to revoke the unneeded rights. There needs to be yet another process to review and revoke them, usually called re-certification or attestation. This is a compromise between security and usability as there is always someone who have approved what he shouldn't have. Especially considering the fact that the approvals are often done by "I just look at it and if it seems OK I approve". It is difficult to enforce good scrutiny during approvals especially if someone has to approve hundreds of requests each day.

Practical enterprise IDM solution will most likely need all of these mechanisms, not just one. Dynamic roles and approvals are the two most critical features when fighting a role explosion with a provisioning system.

External links

- What is midPoint Open Source Identity & Access Management
- Evolveum - Team of IAM professionals who developed midPoint