hierarchy breakout session notes

Todd from University of Illinois reported out on this hierarchy breakout:

The group started talking about University of Illinois at Urbana-Champaign.

Case: Developing authorization system for networking objects such as switchers, ports, routers, VLANS.

Have database system that contains that for the campus networks. Need to authorize campus users to query the info they need. Using groups as grantees. Privileges are create, update, read.

Important Question:

- is there an accepted model for how to assign these priv?

- where should the business logic reside?

- where is the most appropriate place to put this logic?

- How should inheritance and potential collisions be handled?

A lot of what was covered the 1st couple of days of this CAMP was very important.

- start off with use cases.
- Use natural language as much as possible
- Identify the main components of the authorization system
- What are resources, grantor, grantee, etc ?
- Next portion is modeling those relationships
- Separation of relationships between grantees and the resources themselves
- Don't know if we can use a strict hierarchical model, but can use a mostly hierarchical model

We want to assign permissions to network ports and VLANS. So a user can only modify a port if it's on a VLAN they have permission to.

Do we combine those things and have port to VLAN relationship or do we have each have the unit and have business logic, such as "if user X wants to do something they must have this priv AND this priv" ?

Consensus is to assign priv to objects themselves. Have business logic navigate the relationships.

How to navigate the inheretance of the privileges themselves?

A couple of options. Depends on application situation.

1. Priv can be inherited from a parent resources with inheretance computed at runtime by navigating a permissions tree

OR

Alternate solution

2. Instantiate all privileges for all objects themselves so at runtime don't need to do a lot of computation

But when there's an update, then the business logic needs to recalibrate for child objects

1. it's on query time

and

2 it's on update time

PaulH: It sounds like inheritance is applied to resources. In perMIT, inheritance is applied to the scope, not the resource. It's more about the data.

Todd: in our case at Univ. of Illinois, there is a Layer 3 network and there are VLANs that are a part of that. Ports on that VLAN.

Model says "this VLAN is in this network." Inheretence could be such that we assign privilege to the network so that
privilege cascades.