NETWORK CREDENTIAL MANAGEMENT

BLOG POST 3: THE PYRAMIDS OF DELEGATION

SAM BREJCHA
@SAMBREJCHA
JANUARY 2016

THE PYRAMIDS OF DELEGATION

Securing an enterprise against credential theft is, I believe, a painfully dull topic to discuss, yet I also believe this is one of cybersecurity's most important subjects. Delegation models lay the foundation of what permissions users should have and where those permissions are valid. Instead of deploying the latest and greatest threat analytical platform to watch alerts flood your screen, creating delegation models is mostly about establishing written policies. Eventually, a company will use technologies that enforce these policies, but again, most of the work is creating and maintaining documentation. Frankly, most people in the cybersecurity world hate dealing with IT and Cybersecurity policies, but delegation models serve as a critical foundation for your identity management program and offer the best defense against credential theft.

WHAT IS A "DELEGATION MODEL"?

A delegation model is the formal structure of all your administrative groups: which user accounts are members of what administrative groups and what powers those groups have. It becomes a policy map that your network defenders and auditors can use to visualize who has been authorized certain powers.

It is important for network owners to take the time to establish effective delegation models. If not, adversaries will more easily compromise these domains. Almost all network owners will use the default "Domain Admins" as their default administrator which is not a sound delegation model. In most cases while these administrators may delegate control to other active directory groups, "Domain Admins" still has full access to every system on the domain. Relying on the Microsoft default allows adversaries to gain access to highly-sensitive credentials scattered across a domain by free-roaming admin.

To make matters worse, network owners don't realize this situation exists until after a compromise. In the wake of a compromise, network administrators must figure out how to lock down their "Domain Admins" groups, granting permissions to only a handful of systems, and map out where these permissions have been delegated.

Formalizing a list of which users have administrative rights and where their access has been granted aids the network defenders in knowing when such privileges are out of place. Let's say that an active directory group named "CEO Exempted" was added to the administrators group on a handful of laptops. How can your network defenders know if that was an authorized change or not? Are they going to ask Jon from the AD team every time they have a question, or should a document exist stating that "CEO Exempted" was granted the following privileges to these laptops?

DELEGATION MODEL FOUNDATION

1. Do not use any Microsoft built-in groups if it can be helped. Attackers go after those groups because they are the most commonly used.

    a. Remove groups like "Domain Admins" from local administrator on all non-domain controllers

2. Split the entire enterprise administration into tiers with assets featuring higher sensitivities towards the top and workstations at the bottom. (See the Pyramid of Delegation below.)

    a. A great starting structure is a three-tiered approach. Domain Controllers and equally sensitive systems (Ex: Privileged Access Workstations) in Tier 0, Servers in Tier 1, and all others in Tier 2. Some network owners may opt for more than three tiers and this foundation will support.

    b. Administrators in each tier will only have access to systems in their tier. Access should never cross a tier boundary.

        i. Ex: Domain Administrator should not be allowed to log on to any workstation except those identified in Tier 0 that are used for domain level administration.

3. Document authorized users in sensitive administration groups

    a. Ex: A document signed by management level X states that Jon, Bob, Jill, and Service Account X are the only authorized members of "Domain Admins".

4. Never issue privileges directly to a single user. Use role based administration with groups and add the user to the group. Document what permissions that group has and who the authorized members are.

    a. Ex: Instead of adding CEO Steve to the administrators group on his laptop, add the "CEO Exempted" group to administrators and enroll CEO Steve into the "CEO Exempted" group.

5. Review and relinquish privileges as often as possible. Formalize the process to happen every few months.

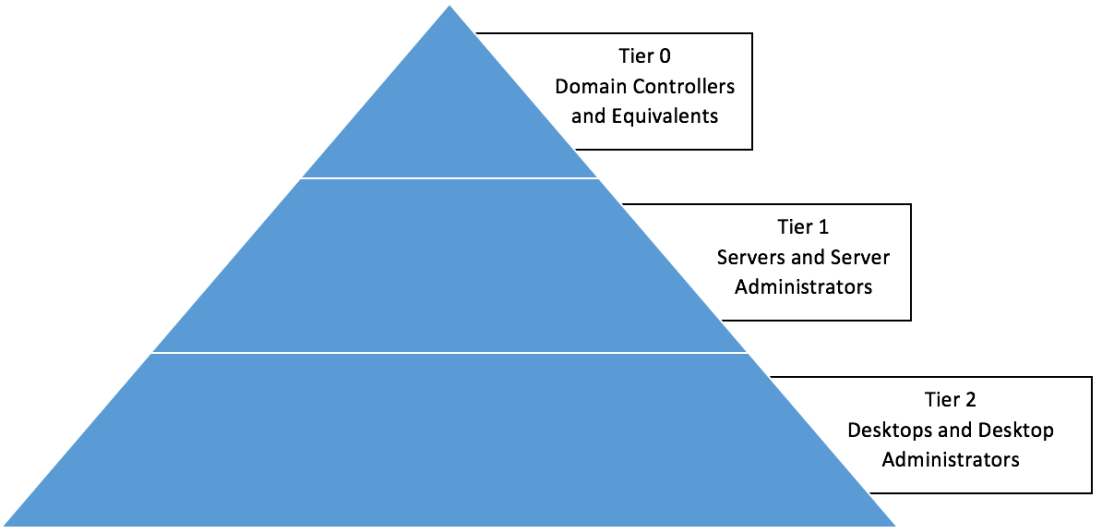
    a. Privileges should not be granted and forgotten. Proactively combing through active directories for obsolete permissions will help remove threats of forgotten accounts.

    b. This is where network defenders and auditors can compare sign authority documents (Bullet #3) against what exists in the active directory.

PYRAMID OF DELEGATION

These five bullet points are just the start of formalizing delegation models; the process can be as simple or complex as desired. (I only listed these points as a starting point, as it will take a significant effort to simply put these in place when a network owner is starting from scratch.) Linked below is the official Microsoft theory of delegation models which provides a lot more detail on how to formalize this process. (The article covers situations such as legitimate uses of crossing a tier boundary under the right security configuration and more.) Again, the steps I provided are a general, fundamental approach necessary to getting an organization started.

(The first time I saw an illustration of a delegation model, it was represented in a pyramid shape. I liked it so much, that I decided to share it with you. In the "Pyramid of Delegation," Tier 0 sits at the top with the fewest assets, and each additional tier features additional elements with most of the network, including workstations, residing in the bottom tier.)



https://technet.microsoft.com/en-us/windows-server-docs/security/securing-privileged-access/securing-privileged-access-reference-material

In my next post, I'll continue building on what I listed here and share some techniques on how to enforce your delegation models.