# The Fundamentals of AD tiering

itm8.com/articles/fundamentals-ad-tiering

itm8

**14** min read

Tech Blog

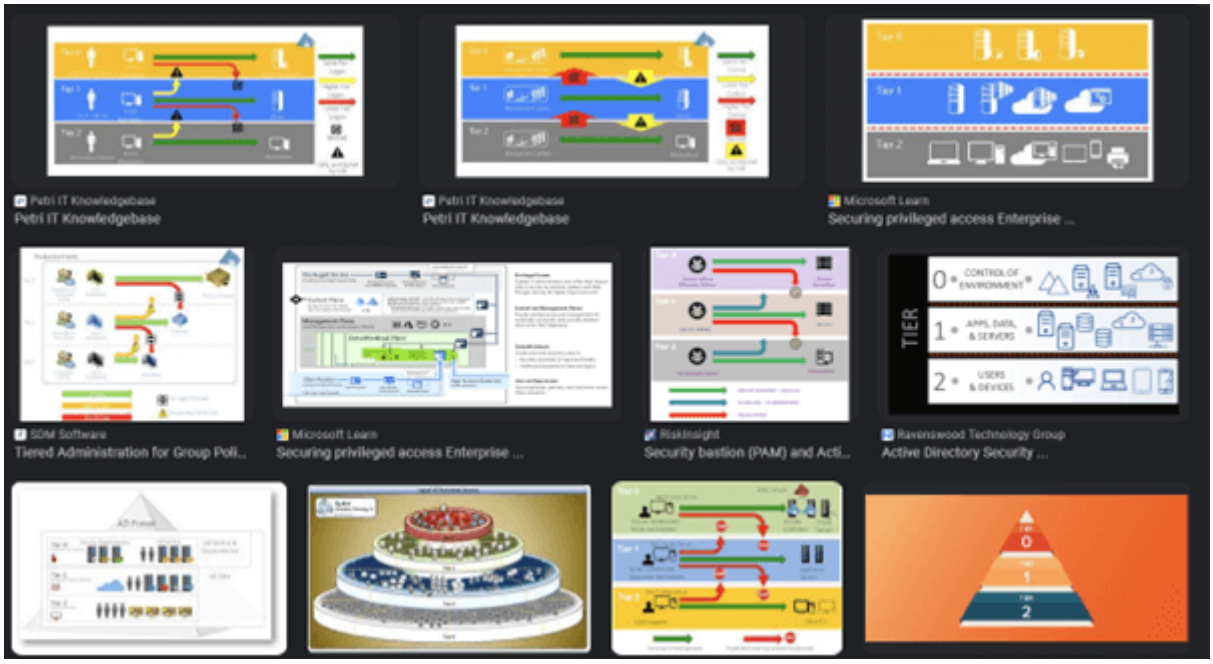
This blog post was inspired by the lack of available information on how to implement tiering on a technical level.

Many blogs and articles found online describe AD tiering and its importance in broad terms, but few go into the technicalities. This blog aims to do just that.

Tiering is a conceptual security model created by Microsoft, which is one of the issues with it. It is a concept and not a firmly defined model.

Probably the reason for that is that no one solution fits all, since every AD is unique and has different edge cases. If Microsoft had said definitively how to implement tiering, it would cause a lot of issues.

This also means that if you do an image search for Active Directory tiering you will see a lot of similar but different diagrams describing different types of tiering, as tiering is a concept with no clear standard on how to do it.



We will seek to remedy that with this blog post by going through how to do bare-bones tiering. The following topics will be explored (If you are only interested in how to do the technical implementation and not the theory, then skip to section 4):

## Table of contents



## What is tiering?

The Microsoft Active Directory tiering model is Micorsoft's reponse to the rise of infosec threats.

As Active Directory is designed to be a centralized way to manage assets, it presents a tempting target for threat actors. The abudance of information that needs to be available to the employees can result in bad practices and a central point from which the AD can be taken over.

The solution that Microsoft developed was to keep management centralized but limit what any one user can access, using least privilege as the core concept.

The traditional tiering model consists of three tiers, tiers 0, 1, and 2.

Tier 0 consists of the most critical computers.  'Critical' being defined not as 'business critical' i.e.  critical for the business to run around financially, but critical in the way that if any one of the computers or users that belong to tier 0 gets compromised then the entire domain or forest is at risk of being taken over.

SpecterOps has made an excellent list of built-in tier 0 assets that can be found here: https://github.com/SpecterOps/TierZeroTable/

Only a small number of administrators need access to tier 0 and even fewer need to be domain administrators.

The administrators who do have access to tier 0 need to use a tier 0 account to access tier 0 servers. The tier 0 user can only be used for accessing tier 0 servers as it will be blocked elsewhere.

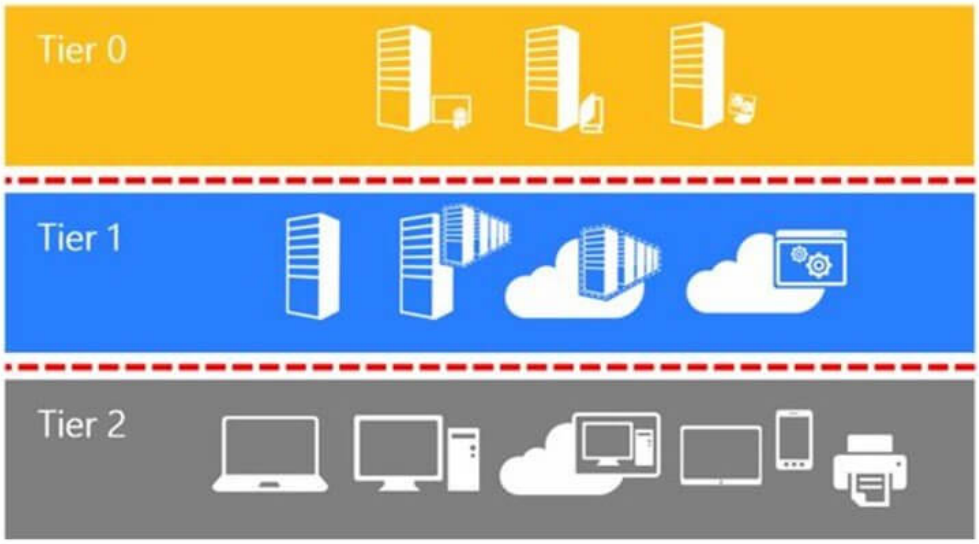
Most day-to-day administration can be done using a management computer and a tier 0 account instead of using the domain controller and a domain administrator account.

The same goes for tier 1. The only accounts that can access tier 1 servers should belong to tier 1.

Tier 1 usually consists of the computers that are not critical enough to be placed in Tier 0, also called member servers.

Tier 2 is the tier that contains workstations and tier 2 users are the normal everyday users, the ones most likely to get phished, click a link, plug in a USB found in the parking lot, scan a QR code, etc.

Tier 2 can also contain servers. If a server runs an application that all or most employees need to access, something like time registration for example, or a Citrix server that everyone logs on, then it should be in tier 2 and not in tier 1 even though it is a computer and not a workstation.




It is important to note that the reason for the three tiers is mostly tradition. It is entirely possible to only implement tiering on tier 0 and block that from all users who are not tier 0.

It is also possible to have five or seven tiers instead of three. It just depends on how you want to separate the resources in the organization. There might be OT in your domain that you want to keep separate or if your workplace works with retail, you might have

store PCs that you wish to segregate. Perhaps developers should have their own tier so that their access to the rest of the domain is limited but they are not hindered in their work by security.

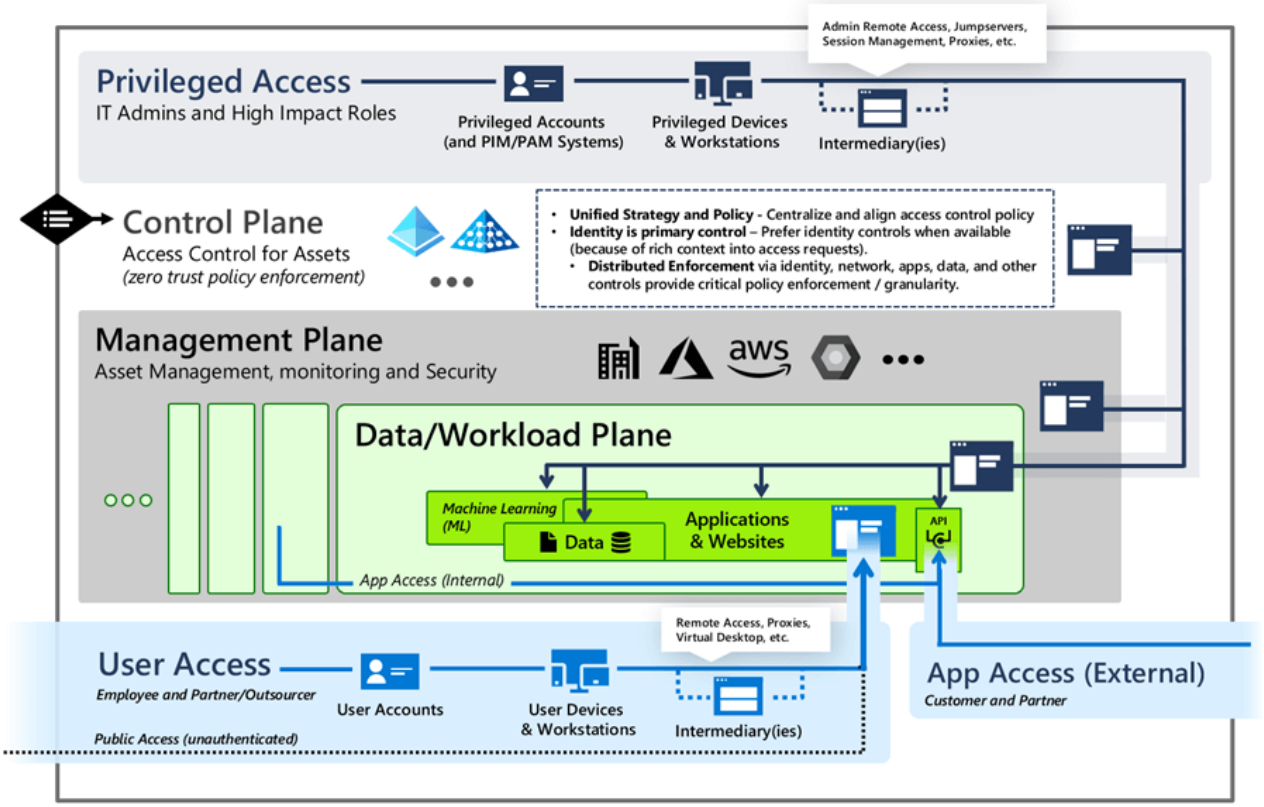
## What is the difference between the legacy model and the enterprise access model?

As you might have guessed from the name the legacy model came first. It is a simpler model than the enterprise access model though it's built on the same principles.

The concept is similar to AD tiering but where the legacy model focused on on-prem AD the enterprise access model also includes networking, OT, Azure, and other cloud providers and adds the need for secure privileged access like Just-In-Time (JIT) access and conditional access policies.

The enterprise access model consists of 5 tiers but calls them planes instead:

1. Control plane: Used for administrative access control, and identity systems.
2. Management plane: Used for managing data, applications, and services.
3. Data/Workload plane: Handles user access for both internal and external and management of devices.
4. App access: The access setup for customers and partners to access the necessary resources in the organization.
5. User access: The systems set up for users to access organization resources, called user access pathways.




Both the legacy model and the enterprise access model recommend dedicated administrator accounts, using PAWs for administrative tasks together with MFA.

The biggest difference is that the enterprise access model incorporates cloud services. The focus is still on secure dedicated administrative accounts and the principle of least privilege.

You can read more about the enterprise access model here: https://learn.microsoft.com/en-us/security/privileged-access-workstations/privileged-access-access-model

## Why do we want tiering?

The main reason to implement tiering is to prevent credential theft and to get rid of overprivileged control of AD objects.

When tiering is implemented credential theft is mitigated simply by enforcing that no credentials that can be dumped are attractive. The credentials available can only be used for moving horizontally on the same tier instead of moving laterally, leaving the attacker stuck at that tier.

Credentials can still be extracted, but if tiering is implemented properly then they all belong to the same tier. This does mean that yes, the attacker might be able to compromise an entire tier, which is not good. It is however better than them being able to

compromise the entire domain. This severely limits the damage an attacker can do to the organization,.

The removal of overprivileged control of AD objects is achieved by ensuring that permissions are kept inside each tier. A tier 2 account is not allowed to have rights to a tier 1 account.

Tier 0 does have ownership of the rest of the tiers due to them containing the Domain Admins group but that is also why it is the most restricted tier.

The owners of all objects should be the Domain Admins group. If that is not the case in your domain, you should look into that.

A tier 2 account should only have read permissions on objects on the other tiers, and then only because that is the Active Directory default (the read permissions can be removed if you want to do security by obscurity, though it should be done selectively).

A tier 2 object should in no way have any control over any tier 1 or tier 0 object. That goes for users, groups, OUs, and GPOs.

In the process of separating objects into tiers, overly privileged control will also be dealt with and removed, resulting in an AD that is more contained in a privileged sense.

If you create a new tiering OU structure and move all your objects to that then as a side effect a cleanup of the AD happens at the same time as you go through all the enabled computer and user objects in your domain.

This is not to say that tiering is the magical pill that will solve all security issues your organization faces.

Tiering is one aspect of hardening and should be treated as such. It doesn't matter if tiering is implemented and enforced if the attacker can just abuse a misconfigured PKI solution, RDP into a domain controller that is internet exposed, or kerberoast an SPN.

In the second to last section, I have suggested a couple of common hardening tasks to look into.

## How to implement tiering?

If tiering is implemented while the domain is being built it is easy to do. Or at least not overly difficult.

There you create a tiering OU structure, an OU for each tier, and create the objects in there.

Add the GPOs that restrict one tier from the others (we go through how to do this later) and assign permission in your preferred way.

It is more common that tiering is implemented in a well-used domain, so I will go through the steps needed to accomplish that. One that is built organically, full of messy ACLs.

This method of tiering will also only concern itself with how to do bare-bones tiering, meaning only blocking users across tiers. I won't go into detail on how to assign access as there in all organizations already is an established way of granting access.

As a computer object can be both a workstation or a computer, I will be referring to both computers and workstations as a computer, as it after all a computer object.

## Step 1. Make a tiering OU structure

In an AD that has been in use for a while, it will often be messy. Different administrators will have been through and will have done things their way. Users, computers, and groups will all be spread among several different OUs. There might be disabled users/computers here and there along with forgotten objects.

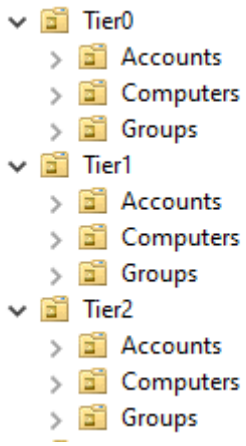
So, create a tiering OU structure. This makes for a clear overview. It will be easy to see where each object belongs by looking at its distinguishedname.

It also eases administration in the future as all objects are kept in a very structured way.

An example of a tiering OU structure is Figure 4. The most important thing when making a tiering OU structure is to have a clear overview of what objects are in the domain and for that reason, we recommend you create an OU for each tier at the root of the AD.

You can use sub-OUs if you want or you can put everything in under the root tier OU. That is a question of personal preferences.

Creating and using a tiering OU structure is not strictly necessary but it does make tiering and administration easier, so we recommend it.
If you don't want to move all your objects away from their current location then you can keep them there and apply the following steps to the existing structure, but there is a risk that it will add complications.




## Step 2. Classify users, computers and groups

### Computers

I prefer using PowerShell to get a list of all computer objects in the domain.

*Get-ADComputer -Filter * -Properties Operatingsystem | Select Name,Operatingsystem,Enabled*

Go through all the computers listed. If a computer has permissions that can be used to take over the domain it is tier 0.

What permissions a computer has will depend on the role it has. Here I assume that the it team will have an overview of what the different computers are used for.

Some computers that should always be in tier 0 are for example Domain controllers, ADFS, ADCS, backup, etc.

As stated, earlier SpecterOps have a very nice table listing what built-in objects they consider to be tier 0: https://github.com/SpecterOps/TierZeroTable/

The tier 2 computers are the ones that normal users log on to every day and work on. If there are computers that a large number of unprivileged users access often then that should be classified as tier 2 as well. Better to classify it as tier 2 than to either break tiering by allowing tier 2 users to access a tier 1 computer or to make a lot more tier 1 users to let the users access that specific computer (which might result in a lot of password reuse).

Tier 1 computers are all the ones that fall in between. It is the computers that cannot be used to take over the domain and the ones that normal users don't log on to.

**Users**

Most users will be tier 2 users. The ones that normal users already use should be classified as tier 2 users.

Domain Admins are tier 0 users and other administrative users may be either tier 0 or tier 1 users. That depends on what computers they are used to administrate and what tier those computers end up in.

Go through the built-in privileged groups such as Account Administrators, Backup Administrators, DNS Admins, and others (again, see the table from SpecterOps). The members of these groups should only be tier 0 users. Many built-in groups offer a path that leads to the takeover of the domain and it is for a reason that by default some groups have no members. You can read more about which groups here. https://learn.microsoft.com/en-us/windows-computer/identity/ad-ds/manage/understand-security-groups - account-operators

A lot of new administrative accounts will most likely need to be created as system administrators will need to have three accounts, one for each tier.

**Groups**

The groups that the users on each tier are members of should also be moved to that tier (except for built-in groups).

There are some groups that all users will most likely be a member of, such as a VPN group. But if you access tier 1 or tier 0 then it should be done through a jumphost with MFA for tier 1 and a PAW for tier 0, so those users would not need to be a member of the VPN group.

## Step 3. Move the users, groups, and computers/computers to the tiering OU structure

Once the users, computers, and groups have been classified they should be moved to the tiering OU structure.

This is a task by itself because there quite often is a lot of hardcoded paths.

Scripts, product configurations, shortcuts, and the like often rely on hardcoded paths. Some of you might say that we stopped using that a long time ago, but a lot of organizations have been around for a long time and have a lot of legacy stuff left. Not only old operating systems and products but also scripts and configurations created using best practices from another decade or millennium.

So, before the actual tiering starts, the computers, users, and group objects need to be moved.

Before you move them you should look at what ACEs there are in the current OU structure and on the objects themselves, and either remove them if they are not necessary or move them to the new tiering structure. If Citrix for example needs to be able to create new VDAs then that is an ACE that should be kept.

If there are any ACEs on the objects that break tiering, by giving rights over objects from one tier to objects from other tiers then they need to be removed, or reassigned if they are needed.

Adalanche, Bloodhound, and Forest Druid are all free tools that can help you explore what ACEs that are configured on objects and whether they will break tiering.

When moving the objects, conflicts will likely happen, and resolving them before tiering is implemented makes troubleshooting easier.

If you have a lot of computers, then I suggest writing a PowerShell script that imports a CSV with all the names of the computers of a tier that then moves them to their new OU.

```
$csvPath = "C:\Path\To\Your\ServerList.csv"
$targetOU = "OU=TierOU,DC=YourDomain,DC=com"

$servers = Import-Csv -Path $csvPath

foreach ($server in $servers) {
    $serverName = $server.ServerName
    $serverDN = Get-ADComputer -Filter {Name -eq $serverName} | Select-Object -ExpandProperty DistinguishedName
    Move-ADObject -Identity $serverDN -TargetPath $targetOU
    Write-Host "Server $serverName moved to $targetOU"
}

Write-Host "Server move operation completed."
```

The same approach, and script, can be used for moving the groups and users to the selected Ous. If you decide to use a script for moving users, computers, and groups then the same script can be used with some slight modification depending on the type of object being moved.

## Step 4. Make new users

The administrators that need access to the different tiers should have users created for each tier and access assigned before the tiering starts so that the normal workflow is not interrupted.

Some tier 2 users will most likely also need a tier 1 account in larger organizations as it is not only administrators who work on computers.

## Step 5. Add users to groups

Now that all the users, computers, and groups for each OU are collected in one place we need to add the users for each tier to a tiering group.

The groups can for example be called:

- Tier0Users
- Tier1Users
- Tier2Users

These groups are going to be used to block users on one tier to access the others.

This means that their membership needs to be dynamic, so when a new user is created, they automatically get added to the group that makes sure they are blocked from other tiers.

This can be accomplished with PowerShell and a scheduled task and is made easy by the fact that all users will be in the same OU.

```
$ouPath = "OU=TierOU,DC=YourDomain,DC=com"
$group = "TierGroup"

$users = Get-ADUser -Filter * -SearchBase $ouPath

foreach ($user in $users) {
    $userName = $user.SamAccountName

    if (-not (Get-ADUser $userName -Properties MemberOf).MemberOf -contains (Get-ADGroup
$group).DistinguishedName) {
        Add-ADGroupMember -Identity $group -Members $userName
        Write-Host "User $userName added to group $group"
    }
}

Write-Host "User group check and update completed."
```

## Step 6. Create tiering GPOs

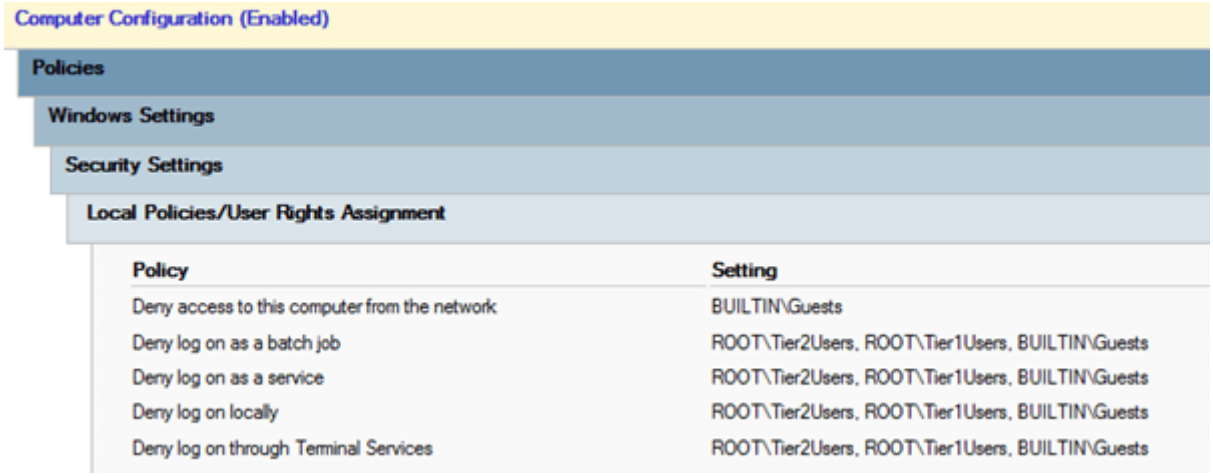
Now that we have populated the groups, we are going to use to block access we need to make rules we can enforce the blocking with. This can be done using GPOs.

We need to create three GPOs, one for each tier.

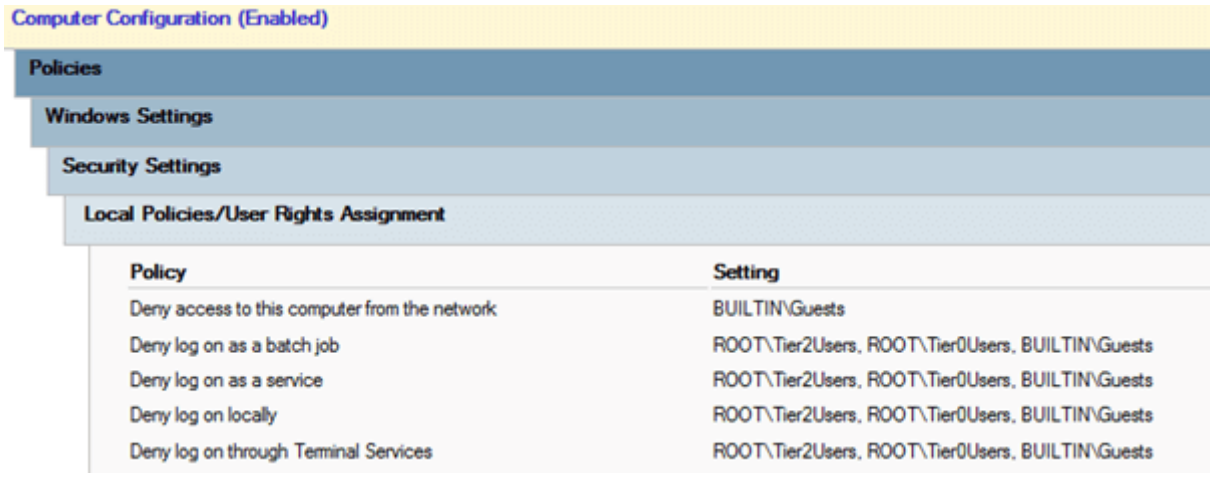
10 different permissions can be set using GPOs that are relevant for tiering. The 5 different types of logons for which there are both allow and deny permissions. We are only going to concern ourselves with the deny permissions:

- Deny access to this computer from the network
- Deny log on as a batch job
- Deny log on as a service
- Deny log on locally
- Deny log on through Terminal Services

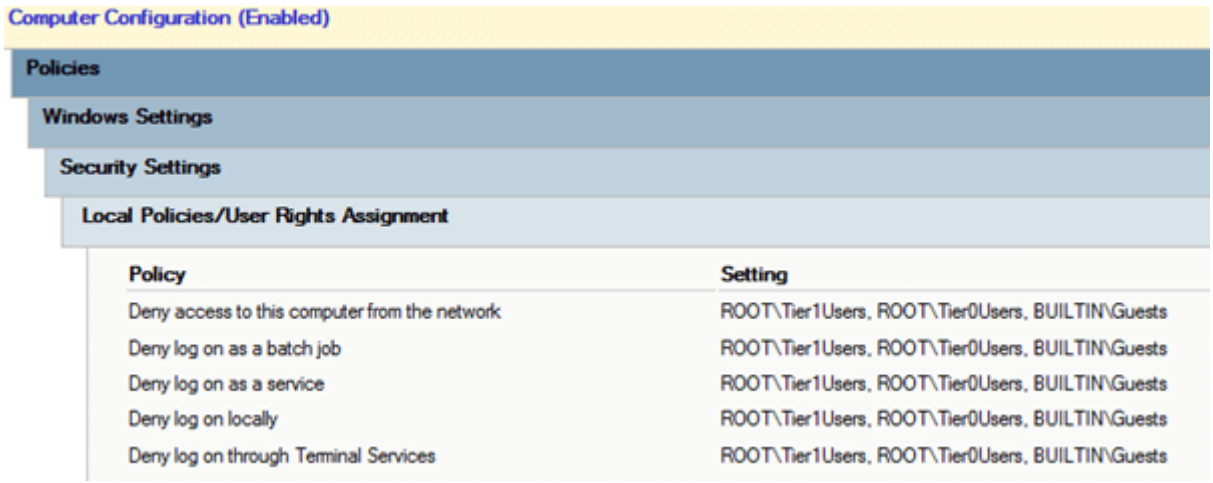
Make a GPO called “Tier 0” and configure it so that as shown in Figure 7.




Make a GPO called “Tier 1” and configure it so that as shown in Figure 8.




Make a GPO called “Tier 2” and configure it so that as shown in Figure 9.




On tier 2 make sure to block the groups Tier0Users and Tier1Users with “Deny access to this computer from the network”. The reason for that is tier 2 authenticate using the network when using the domain, but the same is not true in the other direction.

## Step 7. Resolve GPO conflicts

Look through the other GPOs in the domain and make sure that there are no other GPOs that define the same settings as the tiering GPOs.

If there are then check whether those settings should be preserved and included in the tiering GPOs or can be discarded because they are old or will be covered by the new blocking.

If they are not needed, then they should be removed from the other GPOs as there is otherwise a risk that the link order can be changed, and the tiering GPOs can be overwritten.

## Step 8. Analysis of systems to be tiered

At this point, we have all the computers, users, and groups in their new OUs. We have made GPOs that are ready to be applied and block access.

The last thing we need to do before we can tier is to analyze the computers in question.

We have added all the users where they need to be. But where they have previously been, is very relevant as well. Because service accounts, gMSAs, and regular user accounts are most likely running services and batch jobs all over the place.

So, on every computer, this needs to be looked at. Is there any service or batch job running on a tier 0 computer that is configured with an account that is also running services on tier 1?

If there are accounts that run services and batch jobs across tiers, then they need to be found and the services and batch jobs need to be reconfigured with a new service account (If you are creating a new service account then consider making a gMSA).

It won't be possible to use accounts across tiers as the deny permission takes precedence, so all computers should be looked through and services and batch jobs checked that they are configured with an account that will not be blocked once the computer is tiered.

Normal workstations shouldn’t have services or scheduled tasks running that could be impacted by this, but they should still be analyzed to ensure that they don’t.

If you have a lot of computers, I suggest you either figure out a way to automate the data collection or get some interns or student workers.

## Step 9. Tier the system

Now that the analysis is done the computer is ready to get tiered.

By this point, you should have made sure all services and batch jobs can keep running once the computers are tiered and that the users/administrators that need access have an account in the same tier as the computer they need access to.

If all that is in place, then simply apply the GPO and either wait until it applies or run a gpupdate on the computer.
Then do a reboot of the computer, and check that everything works and that the tier users have access.

I do suggest though that you do it in batches and not do all the computers at the same time.

## How to maintain tiering

Once tiering is implemented it is important to make sure it is maintained.

As PowerShell has been the answer to many things so far, it is once again the answer.

Write a script that checks whether any users or computers are outside the tiering OU structure and if any users are not members of the tiering groups. Either make it write a log that is checked, write to the event log and forward the logs, or send an email or a notification.

If any user, computer, or groups reside outside the tiering OU structure then they are not blocked and can break tiering.

There do need to be some exceptions, as the domain controllers, krbtgt, the administrator account, and some built-in groups should stay outside the tiering OU structure.

```
$ouPath = "OU=TierOU,DC=YourDomain,DC=com"
$group = "TierGroup"
$logPath = "C:\Path\To\Your\Log\file.txt"

# Check users outside TierOU
$allUsers = Get-ADUser -Filter *
$outsideUsers = $allUsers | Where-Object { $_.DistinguishedName -notlike "*,$ouPath" }

foreach ($user in $outsideUsers) {
    $userName = $user.SamAccountName
    Add-Content -Path $logPath -Value "User $userName is outside TierOU"
    Write-Host "User $userName logged as outside TierOU"
}

# Check users in TierOU not in TierGroup
$insideUsers = Get-ADUser -Filter * -SearchBase $ouPath

foreach ($user in $insideUsers) {
    $userName = $user.SamAccountName

    if (-not (Get-ADUser $userName -Properties MemberOf).MemberOf -contains (Get-ADGroup
$group).DistinguishedName) {
        Add-Content -Path $logPath -Value "User $userName in TierOU is not a member of $group"
        Write-Host "User $userName logged as not in $group"
    }
}

Write-Host "User checks and log completed."
```

As mentioned in step 5, make sure there is a script that adds users to the tiering groups.

And make sure the scripts run on a scheduled task that suits your needs.

More scripts will likely turn out to be necessary. The two checks described here are the most basic, but depending on how you decide to implement tiering in your environment there will likely be some more things that you want to keep tabs on.

You should also use Adalanche, Bloodhound, Forest Druid or Improhound regularly to make sure that there are no tiering violations, and that no violations are introduced as you continue to work in your AD, creating new objects and introducing new products.

## Futher hardening to be done

Once you have tiered your domain that doesn't mean that everything is secure, and you can lean back and relax. The following is a list of things that can (should) also be done.

Make sure you use them. Configure them for tier 0 users.

Use the Protected Users group for tier 0 users.

Use MFA. Use it together with your PAWs and everywhere else you can use it. It is very effective.

Make use of hardening baselines. CIS and Microsoft both offer free baselines that will make your computers and workstations much more secure.
https://www.cisecurity.org/cis-benchmarks
https://learn.microsoft.com/en-us/windows/security/operating-system-security/device-management/windows-security-configuration-framework/windows-security-baselines

Many free tools let you explore the vulnerabilities in your domain. Use for example Bloodhound, Forest Druid, Adalanche, PingCastle, and Purple Knight

Make use of password auditing. Check the strength of the passwords in your organization. Password Analysis.

This is NOT an exhaustive list!

## Do you want to talk?

Fill out the form, and we will contact you.

Itm8 is committed to protecting and respecting your privacy, and we'll only use your personal information to administer your account and to provide the products and services you requested from us. From time to time, we would like to contact you about our products and services, as well as other content that may be of interest to you. If you consent to us contacting you for this purpose, please tick below to say how you would like us to contact you:

You can unsubscribe from these communications at any time. For more information on how to unsubscribe, our privacy practices, and how we are committed to protecting and respecting your privacy, please review our Privacy Policy.

By clicking submit below, you consent to allow Itm8 to store and process the personal information submitted above to provide you the content requested.