Jonas Bülow Knudsen September 14, 2023

Posts from SpecterOps team members on various topics relating information security

This is Part 2 of our webinar and blog post series *Defining the Undefined: What is Tier Zero.*

In Part 1, we gave an introduction to the topic and explained why the reader should care. We discussed Microsoft's original list of Tier Zero Active Directory (AD) groups, and we defined what we classify as Tier Zero:

Tier Zero is a set of assets in control of enterprise identities and their security dependencies

You can watch the recording of Part 1 here: Defining the Undefined: What is Tier Zero, or you can read the Part 1 blog post here: What is Tier Zero — Part 1.

We will in this episode cover some of the additional Tier Zero AD objects suggested by community members. We have added all the objects we cover in this episode to the TierZeroTable on GitHub. Alexander Schmitt and Elad Shamir helped produce the content for this blog post. If you are more of a listener than a reader, check out the Part 2 webinar here: Defining the Undefined: What is Tier Zero Part II.

Community suggestions

We have received two sets of suggestions for additional assets that we should classify as Tier Zero on the TierZeroTable GitHub repo after the first episode of the series: https://github.com/BloodHoundAD/TierZeroTable/issues.

The suggestions were the following assets:

- Domain root object
- AdminSDHolder object
- krbtgt user account
- RID-500 account
- OUs
- GPOs
- Read-Only Domain Controllers
- TrustedDomain objects
- AAD Connect object(s)

All the above are great suggestions and interesting topics. We will in this episode cover the suggested assets except the last two which we will cover in future episodes.

If there is anything you would like us to discuss in a future episode then do not hesitate to submit your suggestion on the TierZeroTable GitHub page. We value any contributions!

The suggestions *Domain root object*, *AdminSDHolder object*, and *OUs* are all AD containers (in some way). So let us zoom out a bit and discuss AD containers in general first.

An AD container is a logical container of AD objects. The objects within a container are known as *child objects* and the container is known as the *parent*. The container is also an AD object, and it can contain nested containers. There exists an object class in AD called a *Container*, but many other objects are containers in AD. In fact, you can configure any object in AD to be a container, as you can read about here: https://learn.microsoft.com/en-us/windows/win32/ad/containers-and-leaves.

An AD container is rarely a target for an attacker. We care about containers because you can abuse permissions configured on a container to compromise the child objects, which could be Tier Zero groups, users, computers, etc.

Example of top-level containers in the Default Naming Context

Common for all types of containers is that you can compromise the child objects through Discretionary Access Control List (DACL) inheritance. An attacker with write access to the DACL of a container can create an Access Control Entry (ACE) that grants full control inherited down to the child objects. This will give full control of the child objects unless the child objects have ACL inheritance disabled.

There exist other attack techniques to compromise child objects, that only work for containers and child objects of certain object types. For example, it is possible to compromise users and computer objects by linking a Group Policy Object (GPO) to a parent Organizational Unit (OU) or the domain root object. However, this technique does not work for containers of the Container object type and for child object types such as groups.

It is difficult to remember all the conditions for when you can compromise child objects of a container. Therefore, we suggest a simple definition for when a container should be Tier Zero: **A container is Tier Zero if it contains Tier Zero objects**. This rule may lead to a container being classified as Tier Zero even though you cannot compromise Tier Zero with control over the container. However, we believe this is better than risking misclassifying a Tier Zero container as non-Tier Zero. In general, we recommend creating dedicated Tier Zero containers with only Tier Zero child objects for ease of management and operations. For more recommendations on how you can protect your containers in AD, check out our presentation from this year’s Troopers conference: Hidden Pathways: Exploring the Anatomy of ACL-Based Active Directory Attacks and Building Strong Defenses.

Let us dive into some specific containers.

Domain root object: Tier Zero✅

The domain root object contains all principals of the domain including Tier Zero principals. So it is Tier Zero by our container rule. Special for the domain root object is that its DACL holds the privileges to perform DC replication. An attacker with control over the domain root object can therefore compromise and take over Tier Zero with a DCSync attack. Additionally, it is possible to perform a GPO attack on the domain root object to compromise Tier Zero.

AdminSDHolder: Tier Zero✅

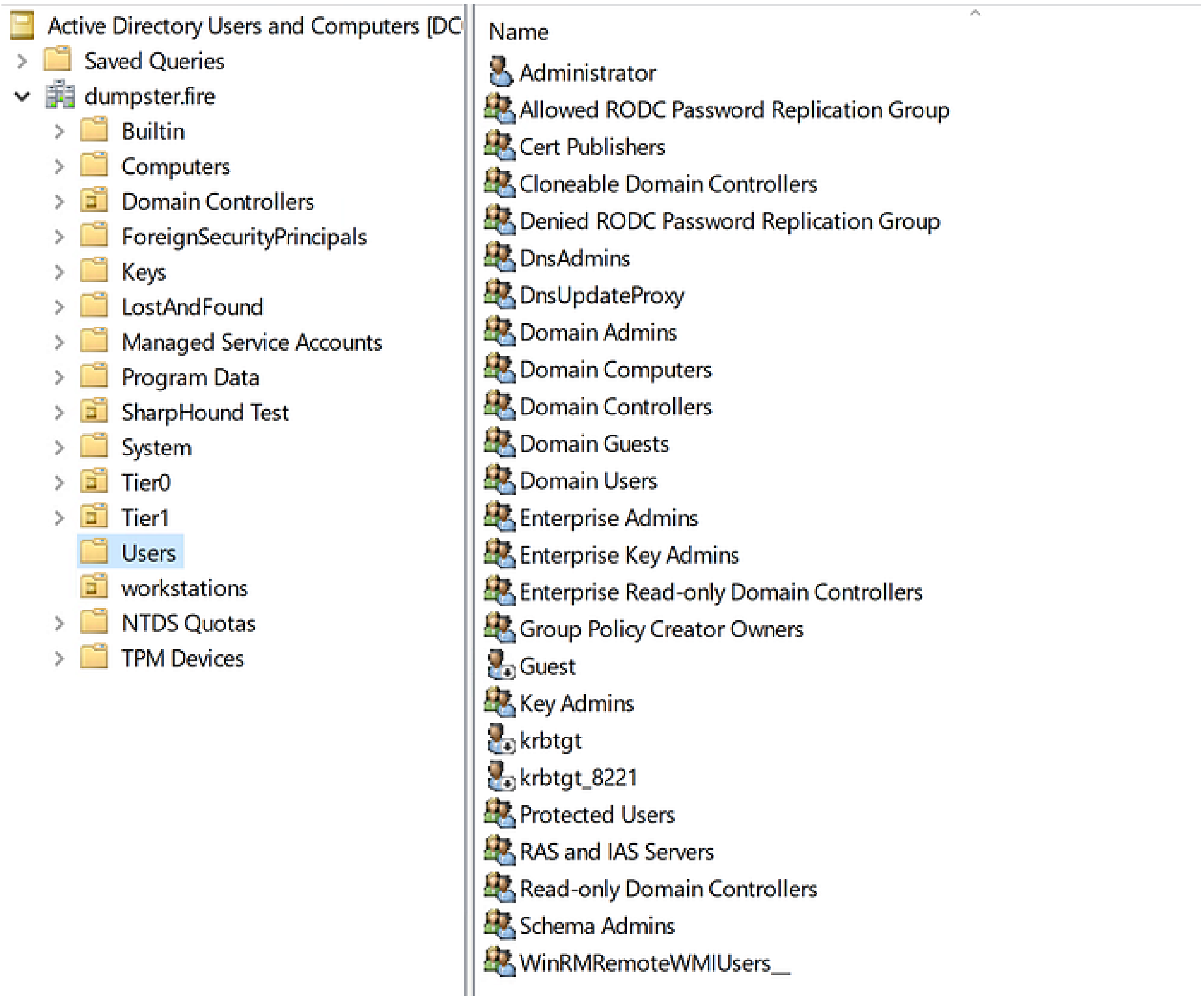
The AdminSDHolder object is a container that exists under the System container, and it has no child objects. It is a template for the security descriptors of Protected Accounts and Groups. A process that runs every hour called SDProp will apply the security descriptor of AdminSDHolder to principals of Protected Accounts and Groups, which include Tier Zero groups such as Domain Admins. Control over AdminSDHolder will therefore automatically provide you with control over Tier Zero. The AdminSDHolder container is a Tier Zero object for this reason. The AdminSDHolder has inheritance disabled.

Domain Controllers (OU): Tier Zero✅

The Domain Controllers OU contains all Domain Controller (DC) computer objects including Read-Only Domain Controller (RODC) computer objects. DCs and RODCs do not have inheritance disabled by default, which means they can inherit permissions placed on the Domain Controllers OU. An attacker could thereby grant themselves GenericAll on DCs and RODCs, which enables the attacker to perform a full domain takeover compromise. If the attacker has the privilege to create or modify GPOs, the attacker could compromise DCs with a malicious GPO. For these reasons, the Domain Controllers OU is Tier Zero.

Users (Container): Tier Zero✅

The Users container contains multiple default Tier Zero objects by default and is therefore considered Tier Zero. AdminSDHolder protects the most privileged ones like Domain Admins, so control over the Users container does not enable compromise of these objects. But AdminSDHolder does not protect some Tier Zero objects such as Cert Publishers and DnsAdmins and they have inheritance enabled, which means an attacker with control over the Users container can compromise them. However, these Tier Zero objects can only disrupt Tier Zero operations. There are no publicly known abuse techniques to take over the control of Tier Zero to our knowledge.



Active Directory Users and Computers [DC
- Saved Queries
- dumpster.fire
  - Builtin
  - Computers
  - Domain Controllers
  - ForeignSecurityPrincipals
  - Keys
  - LostAndFound
  - Managed Service Accounts
  - Program Data
  - SharpHound Test
  - System
  - Tier0
  - Tier1
  - Users
  - workstations
  - NTDS Quotas
  - TPM Devices

Name

- Administrator
- Allowed RODC Password Replication Group
- Cert Publishers
- Cloneable Domain Controllers
- Denied RODC Password Replication Group
- DnsAdmins
- DnsUpdateProxy
- Domain Admins
- Domain Computers
- Domain Controllers
- Domain Guests
- Domain Users
- Enterprise Admins
- Enterprise Key Admins
- Enterprise Read-only Domain Controllers
- Group Policy Creator Owners
- Guest
- Key Admins
- krbtgt
- krbtgt_8221
- Protected Users
- RAS and IAS Servers
- Read-only Domain Controllers
- Schema Admins
- WinRMRemoteWMIUsers__

Child objects of the Users container by default

It is worth noticing that the Users container contains both Tier Zero objects but also non-Tier Zero objects, like the Domain Users group. This is not a security risk per se, but it is better in general to have containers dedicated to a specific tier to make the security boundaries in the environment clear.

Group Policy Objects (GPOs)

It is possible to link a GPO to the root domain, OU containers, and Site containers. The GPO settings apply to computer and user objects. An attacker can abuse a GPO to compromise a user or computer in many ways. On a computer, an attacker could for example add a compromised user to the local Administrators group. For a target user, the attacker could configure a logon script that opens a reverse shell every time the user logs in on a computer.

Security Filtering and WMI Filtering can block GPO settings from applying to specific users or computers, but these configurations are set on the GPO object. So control over the GPO allows the attacker to disable those. However, it is possible to block a GPO linked to a parent container from applying to child objects of an OU by disabling GPO inheritance on the OU. You can bypass this with GPO enforcement, but that requires write access on the container and not just control over the GPO.

As for containers, we recommend to keep things simple. Therefore, **a GPO is Tier Zero if it is linked to a Tier Zero container**.

Special Users

Administrator: Tier Zero✅

The built-in Administrator user account (RID 500) is by default member of the Administrators group and has full control over DC hosts. It is therefore a Tier Zero object.

Krbtgt: Tier Zero✅

AD uses the user krbtgt's credentials to encrypt Kerberos Ticket-Granting-Tickets (TGT). The TGT holds what groups the given principal is a member of. Therefore, if you compromise the credentials of this account, then you can craft your own TGT with any membership as any user, a so-called *Golden Ticket*. The Golden Ticket allows you to access any Kerberos service as the principals you provided in the ticket, hence you can compromise any Tier Zero principal. We therefore consider the krbtgt user as Tier Zero.

There is currently no known privilege on the object to obtain the credentials or to compromise the account in any other way. When you reset the password of krbtgt, AD will ignore your password input and use a random string instead. So, the reset password privilege does not work for a compromise. An attacker could use the reset password privilege to disrupt Tier Zero, as a double password reset causes all Kerberos TGTs in the domain to become invalid.

Read-Only Domain Controller (RODC)

Elad Shamir published a great blog post earlier this year on the subject: At the Edge of Tier Zero: The Curious Case of the RODC. I recommend reading this deep dive into the subject.

We will cover these different aspects of RODCs:

1. RODC computer object — The AD computer object for the RODC.
2. RODC host — The computer host and OS for the RODC.
3. RODC krbtgt — The krbtgt account for RODC.
4. RODC AD groups (multiple).

Note: We classified the *Read-only Domain Controllers* group as non-Tier Zero asset in Part 1 of the series.

RODC computer object: Tier Zero✅

An attacker with control over a RODC computer object can compromise Tier Zero principals. The attacker can modify the msDS-RevealOnDemandGroup and msDS-NeverRevealGroup attributes of the RODC computer object such that the RODC can retrieve the credentials of a targeted Tier Zero principal. The attacker can obtain admin access to the RODC host through the managedBy attribute, from where they can obtain the credentials of the RODC krbtgt account. With that, the attacker can create a RODC golden ticket for the target principal. The attacker can convert this RODC golden ticket to a real golden ticket as the msDS-RevealOnDemandGroup attribute covers the target and the msDS-NeverRevealGroup attribute does not. Therefore, the RODC computer object is Tier Zero.

RODC host: Not Tier Zero❌

An attacker with admin access to the OS of a RODC computer can compromise any user or computer which the RODC can retrieve the credentials for. A RODC should never be able to retrieve the credentials of Tier Zero principals. The RODC host is not Tier Zero if that practice is followed. This allows non-Tier Zero users who belong to the remote office of the RODC computer to log in on the RODC computer with admin access. We recommend to configure the RODC computer objects' msDS-NeverRevealGroup attributes to cover all Tier Zero principals to prevent a potential compromise of Tier Zero.

There may be other misconfigurations that could allow for a compromise. For example, as suggested in the TierZeroTable GitHub repo, the Directory Services Restore Mode (DSRM) password is sometimes configured to be the same as for writable DCs. Another example is that it is possible to remove attributes from the Filtered Attribute Set (FAS), which holds attributes not replicated to RODCs. There exist at least a couple of online guides on how you can remove the LAPS password attribute from FAS, such that the RODCs also have the LAPS password. If there are Tier Zero computers with LAPS, then there is an attack path to compromise Tier Zero.

RODC krbtgt: Not Tier Zero❌

The RODC krbtgt's credentials allow one to obtain a golden ticket for any account the RODC can retrieve credentials for. As already mentioned, aRODC should never be able to retrieve the credentials of Tier Zero principals. The RODC krbtgt account is not Tier Zero when this practice is followed.

Allowed RODC Password Replication Group: Not Tier Zero❌

The Allowed RODC Password Replication Group has no control by default. The msDS-RevealOnDemandGroup attribute of RODC computer objects includes this group by default. This means that the RODCs can retrieve the credentials of members of the group unless the msDS-NeverRevealGroup attribute covers the members as well.

The msDS-NeverRevealGroup attribute of RODC computer objects should cover all Tier Zero principals to ensure an attacker cannot compromise Tier Zero principals from an RODC host after adding the Tier Zero principals to the Allowed RODC Password Replication Group. With this practice, the Allowed RODC Password Replication Group is not Tier Zero and non-Tier Zero admins can manage membership of the group.

Denied RODC Password Replication Group: Not Tier Zero❌

The Denied RODC Password Replication Group has no control by default. The msDS-NeverRevealGroup attribute of RODC computer objects includes this group by default. This means that the RODCs cannot retrieve the credentials of members of the group.

The msDS-NeverRevealGroup attribute of RODC computer objects should cover all Tier Zero principals to ensure an attacker cannot compromise Tier Zero principals from an RODC host after adding the Tier Zero principals to the Allowed RODC Password Replication Group. We recommend using dedicated Tier Zero groups for this rather than the Denied RODC Password Replication Group. With this practice, the Denied RODC Password Replication Group is not Tier Zero and non-Tier Zero admins can manage membership of the group.

Enterprise Read-only Domain Controllers (group): Not Tier Zero❌

The Enterprise Read-only Domain Controllers group has no Tier Zero privileges and is not a security dependency for Tier Zero. The Enterprise Read-only Domain Controllers group has the GetChanges privilege on all domains in the forest. This privilege is not enough to perform DCSync, where the GetChangesAll privilege is also required.

Updated TierZeroTable

We have updated the TierZeroTable on GitHub:
https://github.com/SpecterOps/TierZeroTable

The sharp ones will notice that a few things have changed. First of all, it has been moved from the BloodHoundAD GitHub organization to the SpecterOps organization, just like the BloodHound Community Edition. More importantly, we have changed some of the columns in the table to make it more clear when a publicly known attack is possible and if it enables a takeover of control or disruption of operations.

Any feedback and contributions are much appreciated!