Active Directory: Tactical Containment to Curb Domain Dominance

Author: Chris Tierney, chris.tierney@tr0n.info
Advisor: Russell Eubanks

Accepted: 04/08/2024

Abstract

More than two decades after Microsoft released Active Directory, the identity platform remains in use by organizations worldwide. Significant risks may exist in its implementation, administration, and configuration. With such widespread use, threat actors often aim to gain complete control over Active Directory, referred to as *domain dominance*. Once a threat actor has established complete control over the directory, this frequently results in the deployment of ransomware or the exfiltration of sensitive data. After a total domain compromise, containing and restoring control of Active Directory takes significant time, effort, and expertise. This research aims to provide a novel approach to curb domain dominance through a process referred to as *tactical containment.*

1. Introduction

Across industry verticals and the security community, misconfigurations may exist within Active Directory environments, leading to a total domain compromise, destruction of systems, and the likely exfiltration of sensitive data. Defenders often lack the literature, guidance, and expertise to tactically contain a threat actor that has established domain dominance in their environment. This often leads to a "whack-a-mole" approach to containment, and rarely does an organization regain confident control of its Active Directory. This research aims to provide concise and tactical mitigations to restore control of a compromised directory.

The study focuses on the tactical actions within Active Directory a defender can take to curb domain dominance by a threat actor. The study is not intended to include all actions that can be taken by a defender but focuses on key areas that can help in regaining control of Active Directory. These areas, or tests in this study, included reducing paths to domain dominance, establishing a Tier-0, and minimizing administrative privilege. The study does not focus on attacks by a threat actor to gain domain dominance, as ample research exists on this topic.

2. Research Method

The research was done in a controlled manner using a set of well-defined procedures. Two separate test environments were created to support testing: a controlled environment where no changes were made, and the environment itself is effectively an out-of-the-box Microsoft Active Directory domain, and a variable environment where all the tests discussed in this study were performed.

The control environment was an Active Directory domain consisting of two Windows Server 2019 Domain Controllers, two Member Servers (Windows Server 2016 and 2019, respectively), and two Windows 10 client endpoints. The variable environment was set up in the same manner. Using automation, both environments were created using a publicly available PowerShell tool, AutomatedLab, hosted on GitHub (AutomatedLab, 2024). Default Operating System images (ISOs) available from Microsoft were used.

Before running the tests within the variable environment, BadBlood was executed to simulate a domain in the real world, applying severe misconfigurations that would allow a threat actor to gain dominance of Active Directory (Prowe, 2024).

Before testing, data was collected using three publicly available tools on GitHub. Active Directory data was collected using SharpHound and executed on the control and variable environments. On a Windows 11 system separate from the testing environment, BloodHound Community Edition with Docker Compose was used to collate the data by SharpHound. Further, on the separate Windows 11 system, AD_Miner, an Active Directory audit tool, extracted the graph data from BloodHound. The AD_Miner data allowed for easy identification of misconfigurations within Active Directory. Lastly, a backup of the primary domain controller in the variable environment was taken and used later in testing.

The tests conducted in the study covered a wide range of containment actions within the variable environment. These tests broadly included actions to the Active Directory infrastructure, database, privileged accounts, and Group Policy. Tests were done using a combination of the tools native to Active Directory, PowerShell commands and scripts, and Group Policy templates publicly available on GitHub, discussed throughout Section 3. Colloquially referred to as *tactical containment*, these tests were selected as practical actions defenders can take during a domain dominance event. Each containment action was taken one at a time to minimize unintended changes and ensure that the containment actions were controlled.

3. Findings and Discussion

Tactical containment actions vary widely and are rarely a one-size-fits-all approach. However, a general approach was used during testing to conduct tactical containment actions within Active Directory. This approach included pre-containment data collection, actions on Active Directory, Group Policy configurations, and post-containment data collection. In addressing the problem in this study, this research aimed to highlight methods to tactically contain Active Directory by reducing the paths to

domain dominance, establishing a Tier-0, and minimizing administrative privilege. A comprehensive tactical containment checklist was used and can be found in the Appendix.

3.1. Pre-Containment Data Collection

Before testing, data was collected using three publicly available tools on GitHub. Active Directory data was collected using SharpHound and executed in the control and variable environments, respectively. On a Windows 11 system separate from the testing environment, BloodHound Community Edition with Docker Compose was used to collate the SharpHound data. Lastly, on the separate Windows 11 system, AD_Miner, an Active Directory audit tool, extracted the graph data from BloodHound. The AD_Miner data allowed for easy identification of misconfigurations within Active Directory, which was used to drive the tactical containment actions. During the data analysis, 2,300 paths leading to Domain Admin within the variable environment were noted.

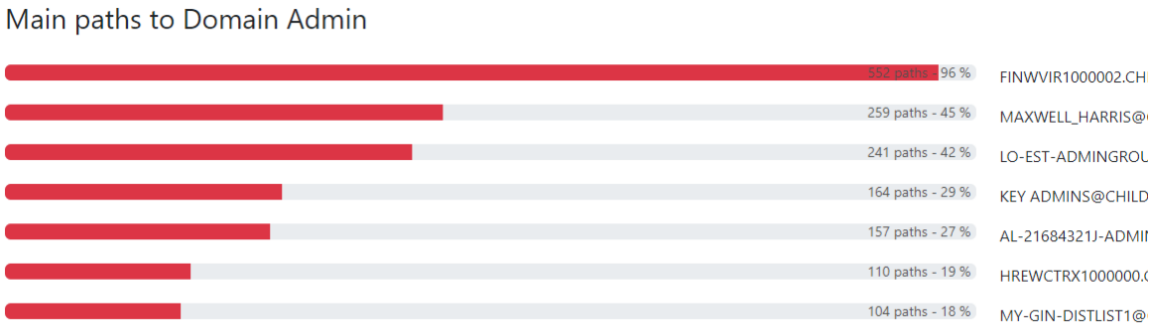

Figure 1: AD_Miner – Paths to Domain Admin

3.2. Active Directory Actions

The first phase of testing covered tactical containment actions taken in Active Directory. This included the restoration of a Domain Controller, implementing a Tier 0, and actions on accounts in the directory. All actions were taken only in the variable Forest and Child Domain. The control environment remained unchanged.

3.2.1. Infrastructure

In cases where a threat actor has compromised or may have compromised domain controllers, it is prudent to remove them from the environment. During testing, two

approaches were used to recover Domain Controllers to a trusted state. It is worth noting that this testing does not cover all methods to recover Domain Controllers, such as a nonauthoritative restore of Active Directory.

The first method of testing was to recover Domain Controllers #1 from a *known good backup*. A *known good backup* would be before the earliest evidence of threat actor activity identified from a forensic investigation. For testing, the system state backup of Active Directory was taken during the pre-containment data collection phase using the native Windows Server Backup feature native to the operating system. An entirely new Virtual Machine (VM) was created using the default Windows Server 2019 installation with the same resource specifications as the original system to reduce the likelihood of issues during restoration and recovery. Further, the VM remained in an isolated Hyper-V virtual switch while the system was restored. The system state data, including Active Directory data, was restored, and the VM was moved to the Hyper-V switch for the variable environment. Lastly, Active Directory and SYSVOL replication tests were performed to ensure that the restored Domain Controller was functioning normally.

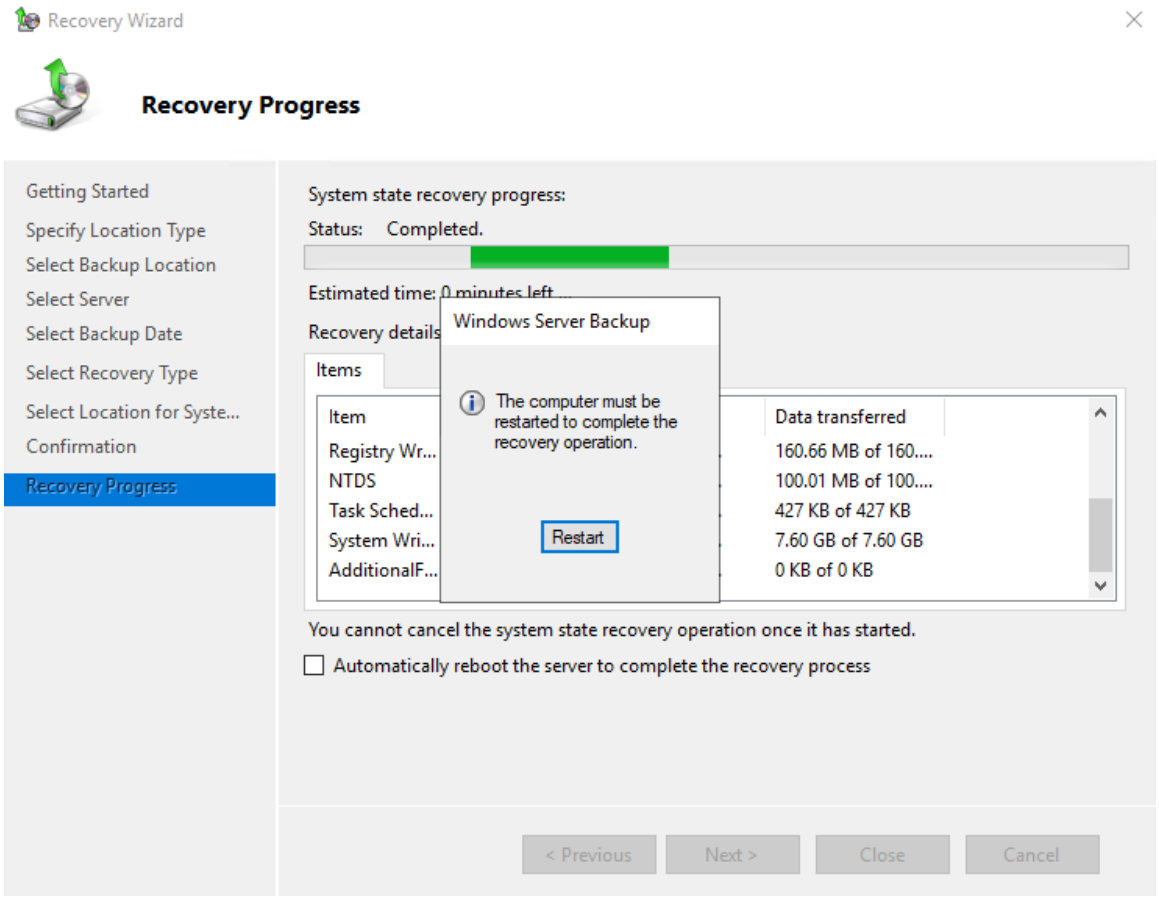

Figure 2: Recovery of Domain Controller #1

The second test was performed on Domain Controller #2. First, the system was moved into an isolated Hyper-V virtual switch to isolate it from the variable environment. Then, the approach used in this test was executing the Microsoft Safety Scanner (MSS), Microsoft Safety Scanner Download | Microsoft, to scan for malware, remove any identified malware, and undo any changes made by the malware. In cases where there is no evidence of a confirmed compromised Domain Controller, or the Domain Controller is suspected of being compromised, the latest version of Safety Scanner can be used instead of recovering or rebuilding the Domain Controller, as rebuilds can bring problems if done improperly. During testing, the scan was completed using Version 1.403.3619.0 of Microsoft Safety Scanner, and no signs of malware were found. Domain Controller #2 was then moved back into the Hyper-V switch for the variable environment.

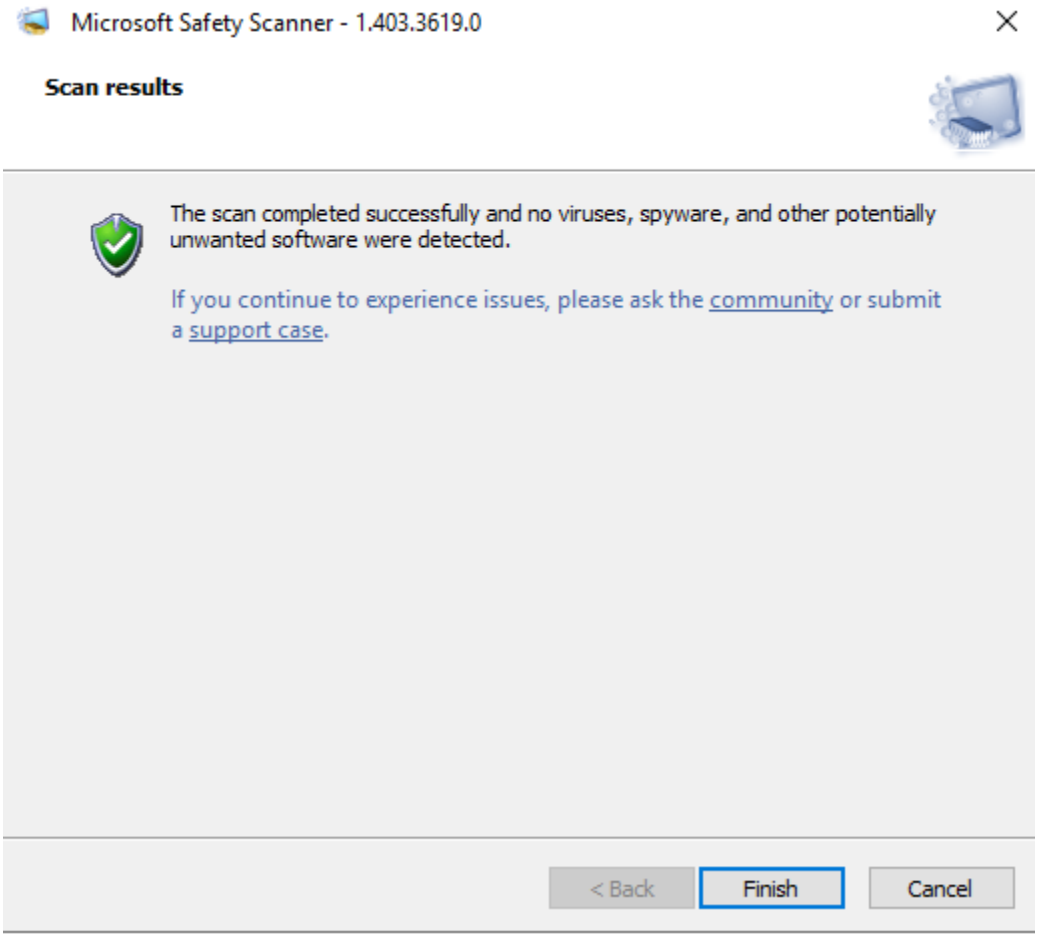

Figure 3: Microsoft Safety Scanner Results on Domain Controller #2

One additional test was conducted to reset the Directory Services Restore Mode (DSRM) password on all Domain Controllers in the environment using `ntdsutil`. The DSRM account is a local administrator account stored in the SAM database on Domain Controllers and is used to troubleshoot or restore Active Directory in an emergency (Pyle, 2023). These passwords should be unique on every Domain Controller to reduce the risk of lateral movement when DSRM passwords are the same (Metcalf, 2024).

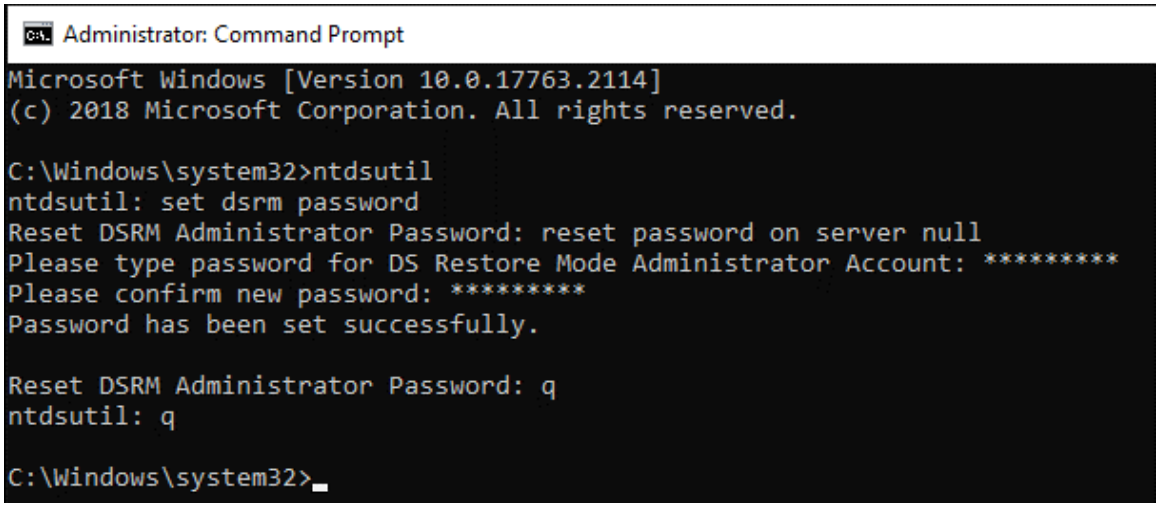

Figure 4: Ntdsutil to Reset the DSRM Password

3.2.2. Tiering

The Active Directory Tier model is intended to segregate highly privileged activities into distinct tiers to create security boundaries. Although there is debate that Active Directory was not designed with security boundaries in mind, the Tier model aims to overcome this. Implementing Tier 0 can prevent a threat actor from gaining and maintaining domain dominance if done properly. Tier 0 restricts what privileged identities can control and which systems they can log on to. Broadly, Tier 0 consists of any system that controls identity, including Domain Controllers, certificate authorities, hybrid cloud identity systems, federated identity systems, or any identity platform in a chain of trusted identity management. In terms of Active Directory, Tier 0 typically includes privileged accounts, service accounts, computer objects, and groups that have direct or indirect control over Active Directory and the aforementioned systems. Figure 5 below illustrates the Tier model (Heidecker, 2024).

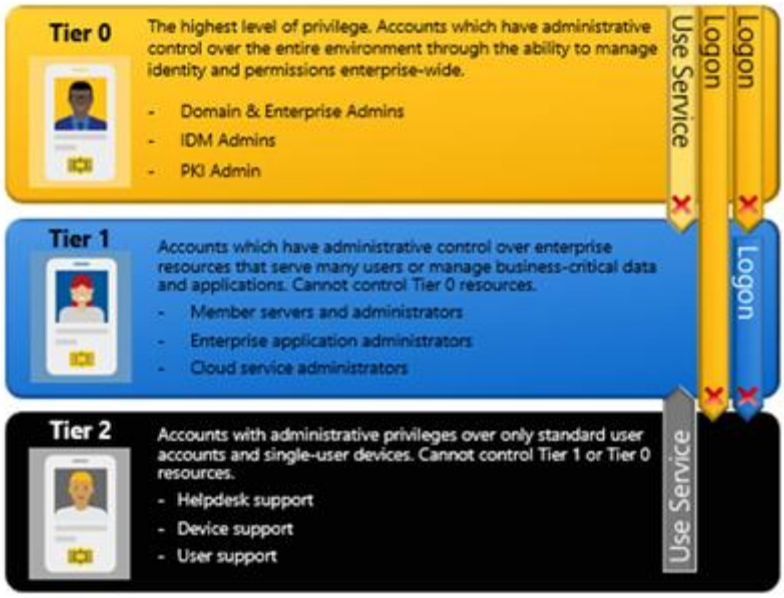

Figure 5: Active Directory Administrative Tier Model

In this scope of testing, a Tier 0 was established in Active Directory using a PowerShell script, [AD-Tier-Administration/Create-Structure.ps1 at master · SalutAToi/AD-Tier-Administration (github.com)](), to automate the creation of the Organizational Unit (OU) structure within Active Directory. On observation, the previous execution of BadBlood during the environment setup created a Tier 0 infrastructure, which was manually renamed to *Legacy-Tiering*, to prevent any confusion or issues with script execution and proper setup of the OU structure for Tier 0. Figure 6 below illustrates the result after the script was executed.

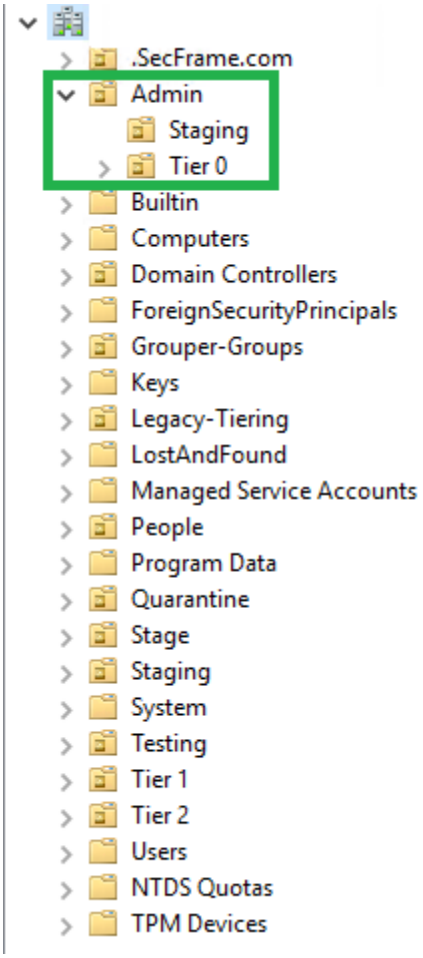

Figure 6: Result of Tier 0 Scripted Creation

3.2.3. Privileged Account & Sensitive/Built-In Group Cleanup

After the Tier 0 infrastructure was established, the next testing phase involved the cleanup of privileged accounts, sensitive groups, and the Administrator groups built-in to Active Directory.

First, two new Tier 0 accounts were created within the Tier 0 OU, which would be used to conduct the remainder of the tactical containment testing. Then, the newly created accounts were added as members of the *Domain Admins* group in Active Directory, pictured in Figure 7 below.

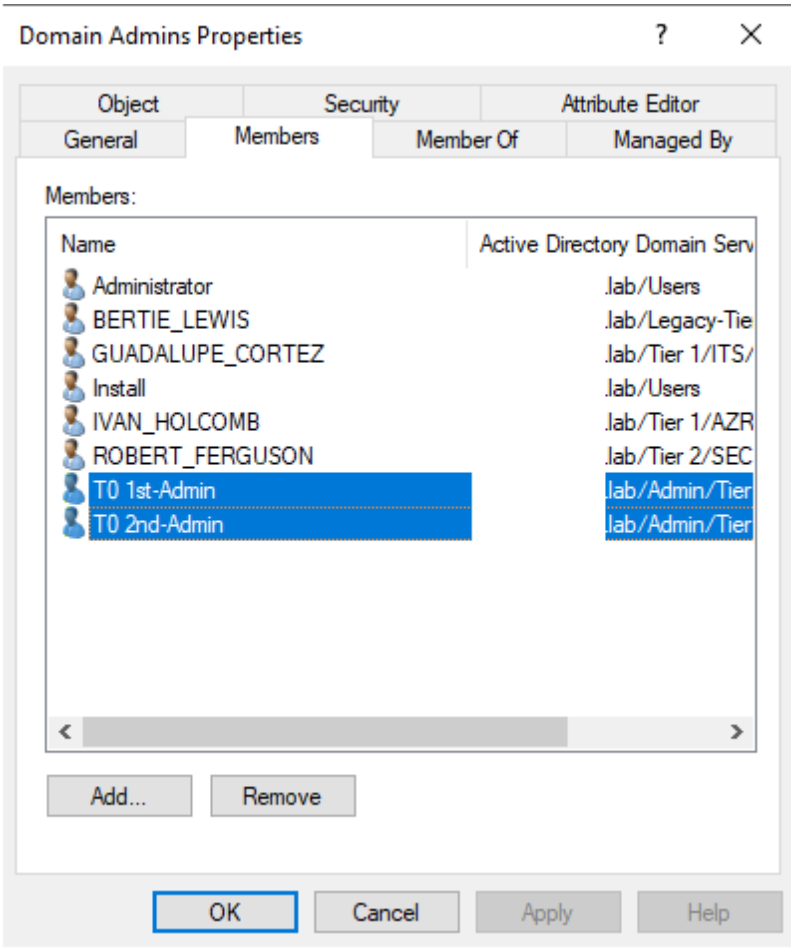

Figure 7: Add New Tier 0 Accounts to the Domain Admins Group

Then, except for the built-in *Administrator* account and the new Tier 0 accounts, all accounts were removed as members of the *Domain Admins* group.

The next containment action test performed was the manual cleanup of group membership across all sensitive Active Directory groups, including the built-in groups. The table in the Appendix details the default memberships in a new domain covering both the *Member* and *Member Of* tabs for each group. A total of 14 sensitive and 17 built-in groups were covered in this procedure. This manual procedure was time-consuming but was critical in minimizing domain privilege as much as possible in testing. Figure 8 below is one example that illustrates the intended outcome of the *Member Of* tab for the Domain Admins group.

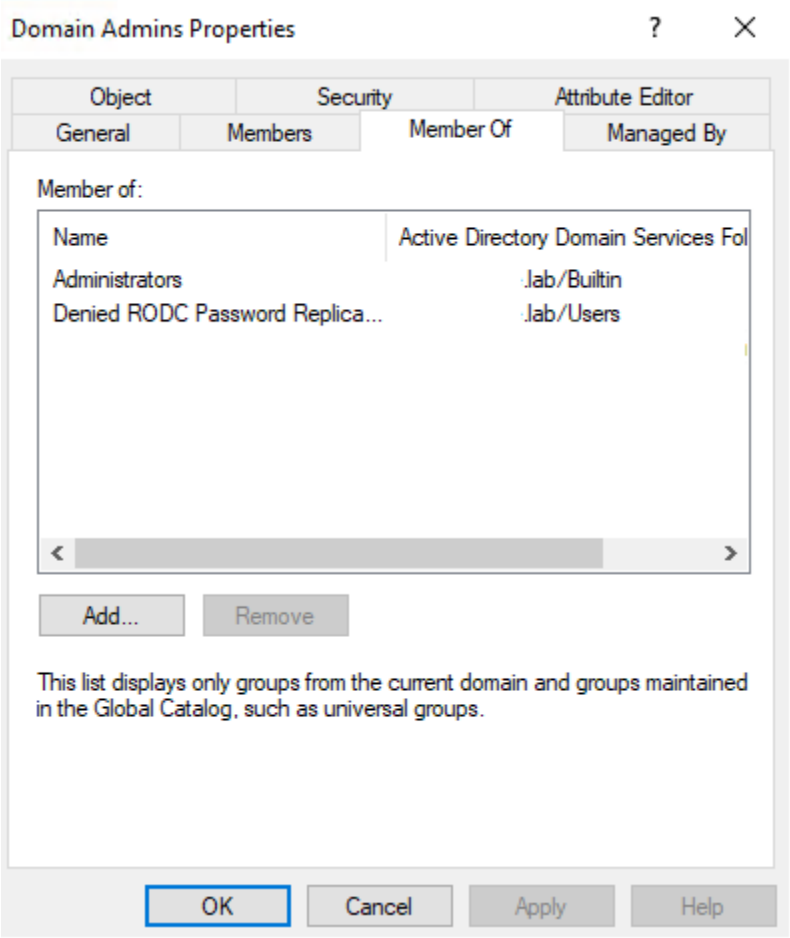

Figure 8: Domain Admins Properties - *Member Of* Tab

3.2.4. Cleanup Actions

In this testing phase, tactical containment actions were performed on the Active Directory. These actions included resetting the AdminSDHolder permissions, correcting orphaned AdminSDHolder rights, and raising (and invalidation) of the RID (relative identifier) pool.

In Active Directory, the SDProp process evaluates the protected groups and accounts. It then periodically resets the Access Control Lists (ACLs) on these objects to mirror the AdminSDHolder object (Microsoft Corporation, 2024). In many environments, an Access Control Entry (ACE) within the ACL for AdminSDHolder can be created, which delegates users' or groups' rights to protected objects. In some instances, users may have excessive rights to these protected objects, which threat actors can exploit.

The ADSI Edit tool native to Domain Controllers was used to reset the AdminSDHolder permissions during testing. This action was performed on the AdminSDHolder object in the *System* container under the default naming context. Specifically, the Restore Defaults button within the Advanced Security Settings was used to baseline the permissions. Figure 9 below illustrates this action.

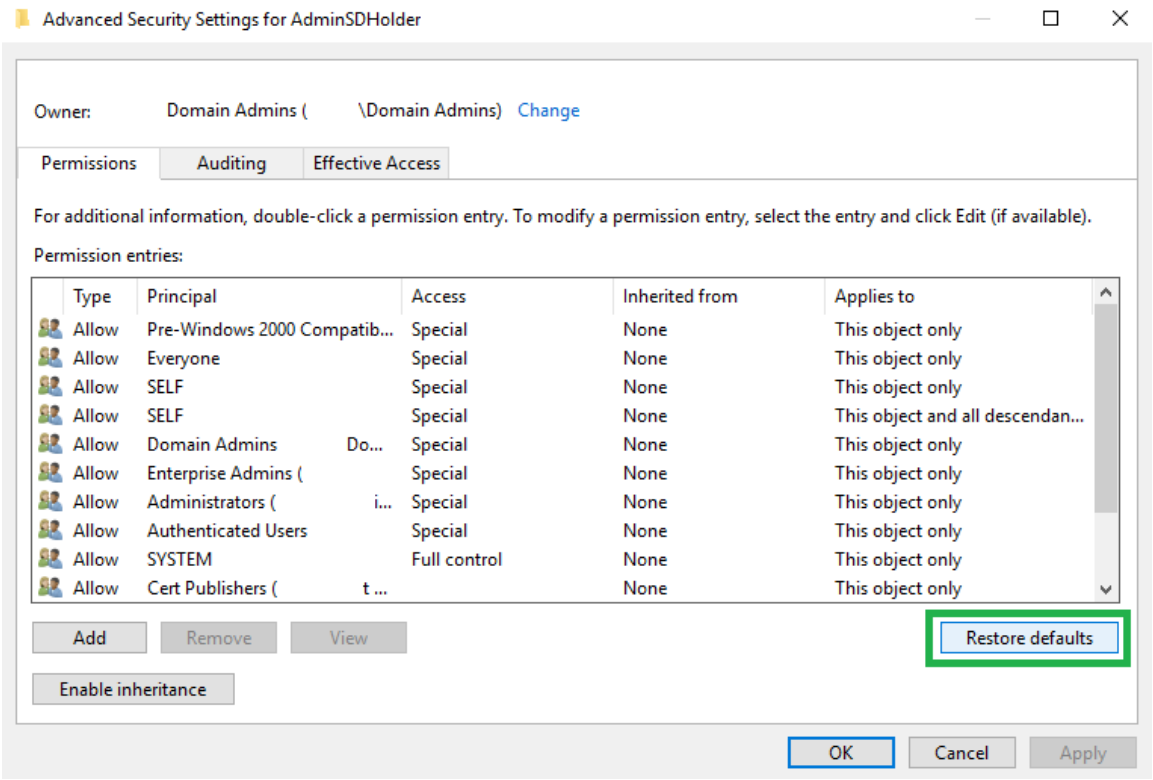

Figure 9: Restoring the Defaults for the AdminSDHolder Object

Next, orphaned accounts with an '`adminCount = 1`' were identified and removed. A PowerShell script available on GitHub, Detect and correct orphaned 'adminCount=1' users who are no longer in protected groups (github.com), created by Alan McBurney, was executed to find orphaned accounts, reset the admin count attribute to 0 and enable inheritable permissions. A total of 18 accounts were found and remediated. Figure 10 below illustrates several of the orphaned accounts before remediation occurred.

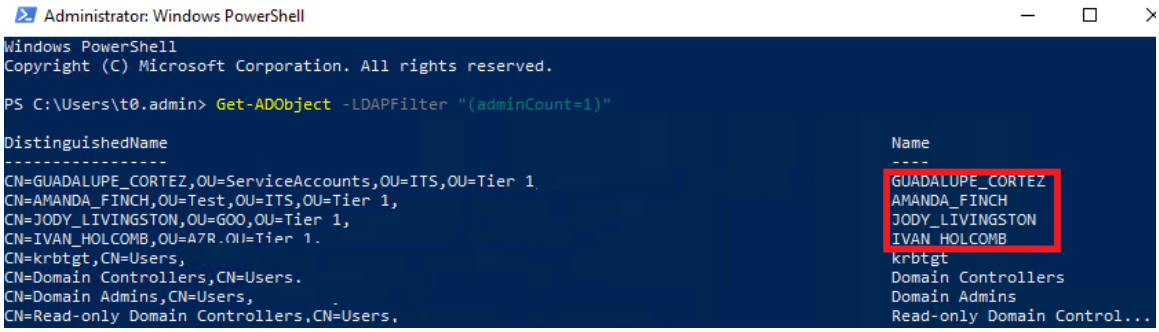

Figure 10: Orphaned AdminSDHolder Accounts Before Remediation

The last cleanup action performed was raising and invalidating the RID pool. If a domain controller has evidence of compromise or is suspected of compromise, the rIDAvaialblePool can be increased. This procedure ensures that the succeeding RID pools will not be used by any other Domain Controller in the directory. The Microsoft Learn documentation was referenced to raise and invalidate the RID pool. During testing, the rIDAvailablePool value was raised by a value of 100,000 per Microsoft's documentation (Microsoft Corporation, 2023).

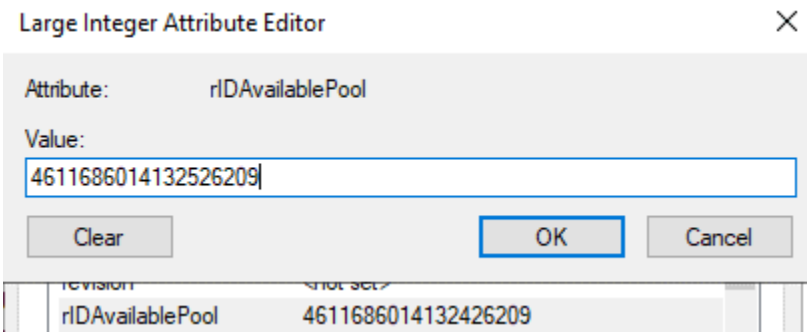

Figure 11: New rIDAvailablePool Value

3.2.5. Account Actions

The next phase of tactical containment actions performed during testing involved procedures on accounts in Active Directory. These tests included resetting the RID 500 account password twice, resetting the krbtgt account password twice, resetting the trust passwords between the Forest and Child domains, and enabling constrained delegation on privileged accounts. These actions aimed to invalidate any previously issued Kerberos tickets and reduce the risk of a threat actor's continued use of these tickets.

In Active Directory, the krbtgt account is responsible for the secure distribution of Kerberos tickets used by accounts, services, and computers. Should a threat actor obtain the hash of the krbtgt account, the actor could mint Kerberos tickets for virtually any user or service in Active Directory. This attack is described as a Golden Ticket attack (Microsoft Corporation, 2024).

Before resetting the krbtgt account twice in testing, Active Directory replication tests were performed to ensure that any password changes would replicate to the Domain Controllers in the environment and avoid unintended issues. Upon validation of healthy replication across the environment, the krbtgt account password was manually reset. After 12 hours, the krbtgt account was reset for a second time. Microsoft's documentation recommends at least a 10-hour waiting period between password resets to avoid any authentication issues (Microsoft Corporation, 2023). In an emergency, such as evidence of a Golden Ticket attack, the krbtgt password resets can be performed in quick succession; however, this may cause authentication issues (Fan, 2020).

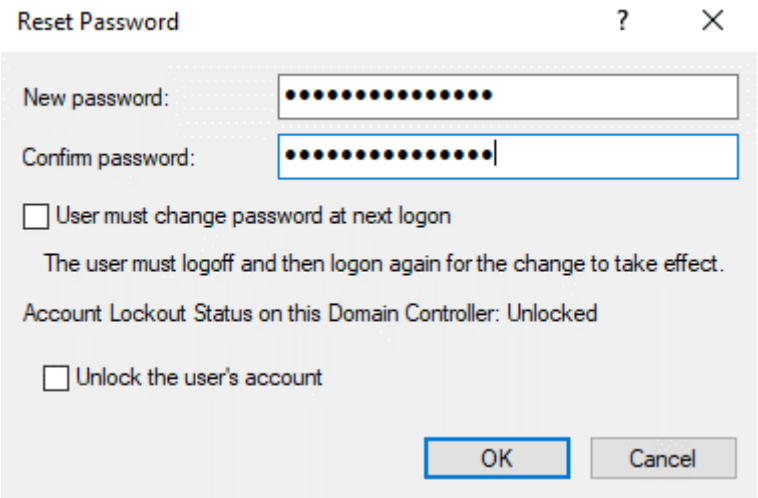

Figure 12: Krbtgt Password Reset

After validating replication and successful authentication of computers and accounts, the RID 500 account password was reset twice in quick succession. Colloquially, the RID 500 account is better known as the built-in Administrator account, named *Administrator* by default. In many environments, this account is renamed. As

such, PowerShell can be used to locate this account in the directory, as illustrated in Figure 13 below.

```
Get-ADUser -Filter * | Where-Object {$_.SID -like "*-500"}
```

Figure 13. PowerShell code to find the RID 500 account

The next step in testing involved resetting the trust passwords between the Forest and Child domains, colloquially known as the Interdomain Trust Account (ITA). Resetting the ITA password for all Active Directory trusts, whether internal or external trusts, ensures that the password hash of the ITA cannot be used to move between these trusts, thereby limiting the threat actor's ability to move laterally to other environments. Microsoft's documentation, AD Forest Recovery - Resetting a trust password, was followed to reset the ITA password on each side of the trusts, which in the testing environment was between the Forest trust and the (child) Domain trust.

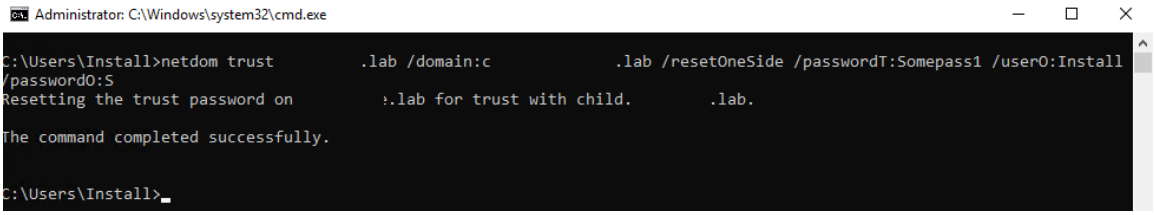

Figure 14: ITA Password Reset on the Forest Side of the Trust

The last account action taken in testing was configuring all Tier 0 accounts as "Account is sensitive and cannot be delegated." At this point in testing, three accounts had privileged permissions that required this setting to be configured to enable constrained delegation. For clarity, every Domain Admin, Enterprise Admin, Tier 0 Admin, or any account with delegated privilege to Tier 0 should be configured.

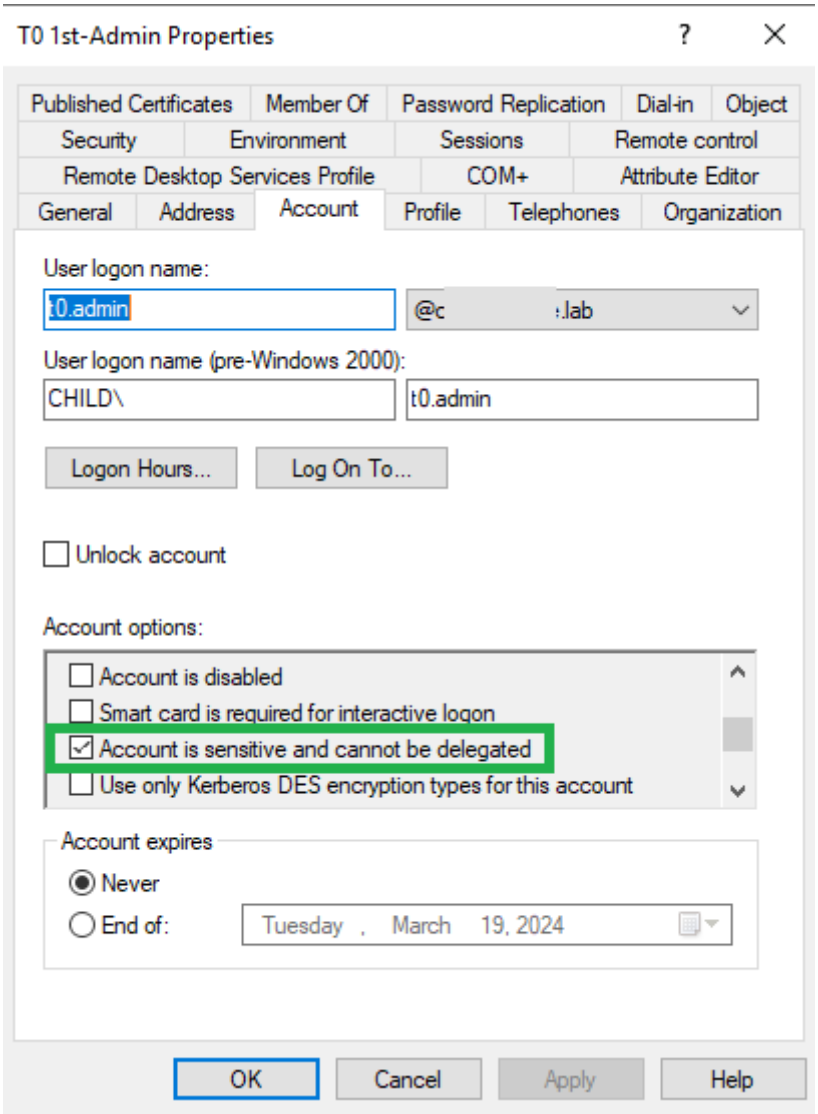

Figure 15: Constrained Delegation Checkbox

3.2.6. ACL Cleanup Actions

The most critical tactical containment action performed during testing was removing unnecessary and potentially overprivileged ACLs and delegations on OUs in Active Directory. These delegations, or Shadow Admins, are non-privileged Active Directory accounts with administrative rights to objects, potentially leading to an escalation of privilege. A subset of these delegations includes rights such as `GenericAll`, `GenericWrite`, and `Replicating Directory Changes All`

(Metcalf, 2017). As such, identifying and removing these unnecessary rights is tedious but essential to eliminating paths to domain dominance.

Using the Active Directory Users and Computers console, all OU's Advanced Security properties were reviewed, and unnecessary ACL permissions were removed. This included every single subordinate OU in the directory. Since BadBlood was executed during the environment setup, nearly 50 nested OUs were created, making this effort tedious. In reviewing and baselining these permissions, in most instances, the *Restore defaults* button was used with the understanding that BadBlood created these unnecessary permissions to create escalation paths to domain dominance. However, in production environments, due diligence and care should be used in reviewing and removing these permissions, as their removal could have unintended consequences. Figure 16 below shows one such example on the Domain Controllers OU.

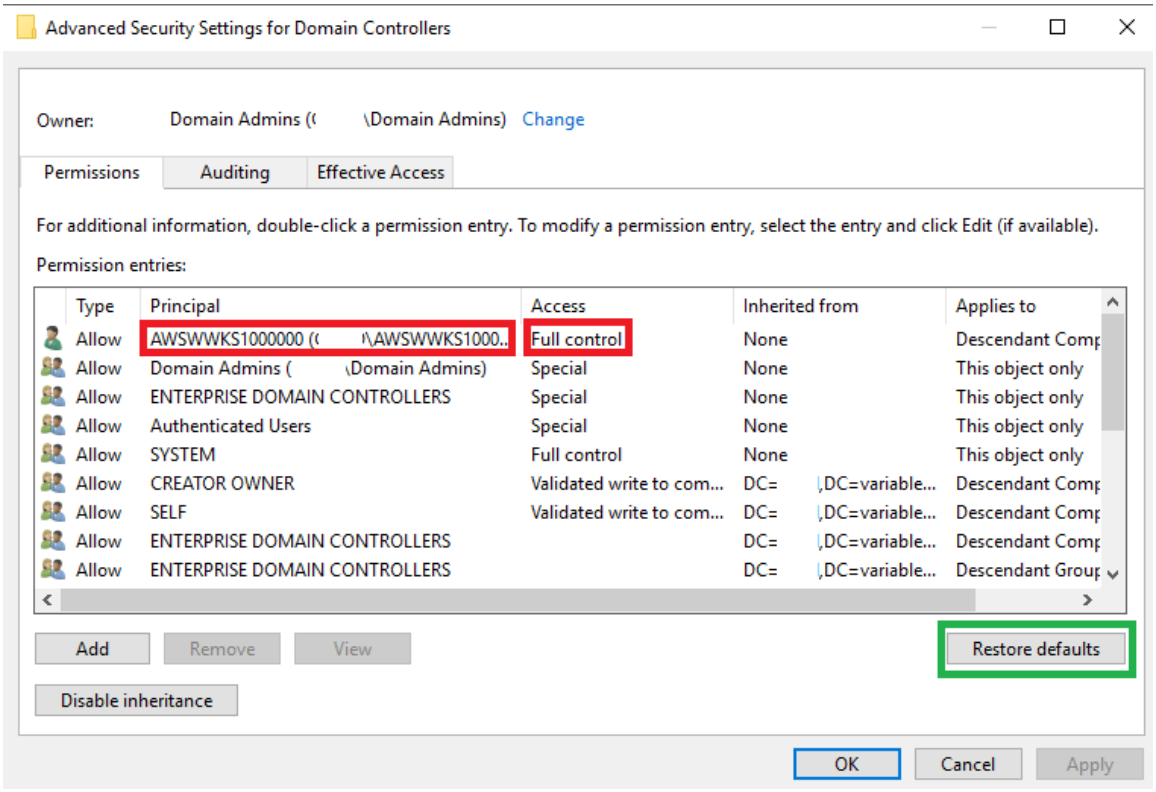

Figure 16: Restoring Security Defaults on the Domain Controllers OU

Further, ACL permissions on Group Policy Objects (GPO) were also reviewed using the Group Policy Management snap-in. However, in testing, no excessive ACL permissions were noted. In production environments, however, due diligence and care should be taken when reviewing and removing excessive permissions, as threat actors commonly leverage Group Policy to deploy ransomware and destructive wipers (M365Guy, 2022).

3.3. Group Policy Actions

Although threat actors can use Group Policy to deploy their malware, defenders can use Group Policy to enable advanced auditing to better spot nefarious activity, apply strict User Rights Assignments to limit the spread of privileged credentials across Tiers in Active Directory, and apply basic hardening policies to reduce the attack surface in the environment.

3.3.1. Auditing Policy

In production environments, enhanced auditing is often enabled to provide administrators and defenders with higher fidelity logs to understand what is happening in their environments (Bartolo, 2018). During the setup of the testing environment, enhanced auditing was not configured. In the spirit of providing opportunities for greater visibility in the testing environment, the pre-configured auditing policies available from Palantir on GitHub, AndyFriar/palantir-wef-gpo: Audit Policy GPO's for Palantir WEF (github.com), were imported into Group Policy including the policies for Domain Controllers, Member Servers, and client endpoints. The policy for Domain Controllers was linked to the Domain Controllers OU in the Group Policy Management console. Further, a Tier 0 policy was cloned and linked to the Tier 0 OU as well. Group policy was then forcibly updated on each Tier 0 system using `gpupdate /force` from an elevated administrative command prompt.

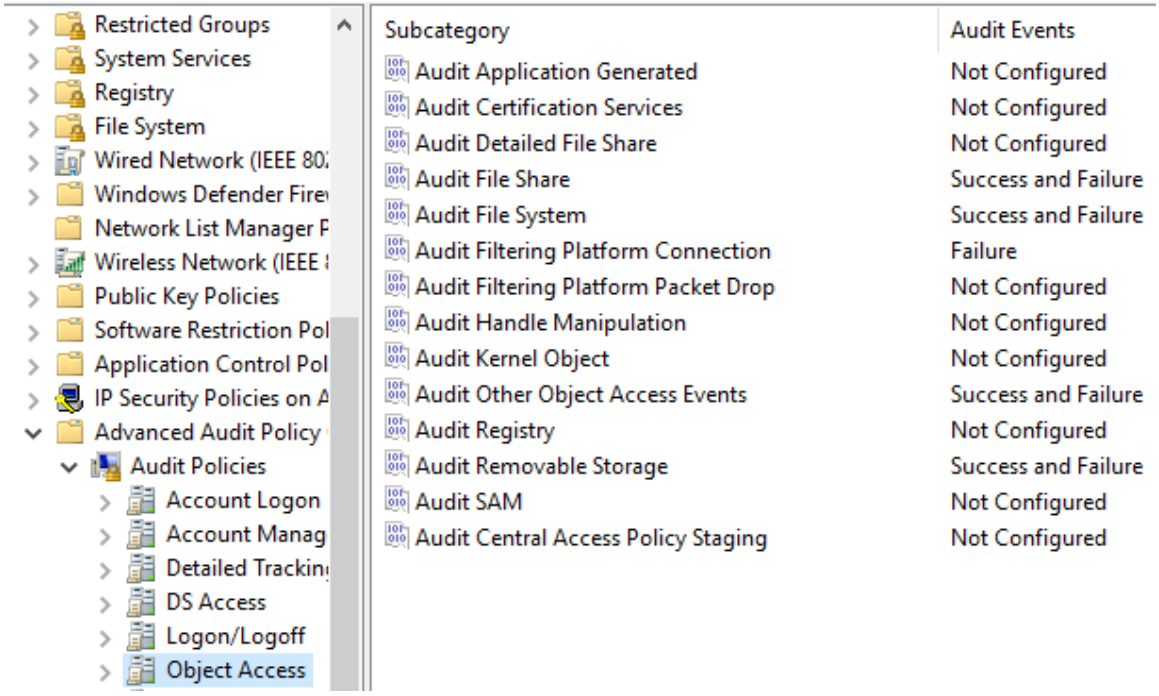

Figure 17: Tier 0 Auditing Policy Sample Settings

3.3.2. Default Domain Policy

As previously discussed, threat actors can leverage group policy to deploy malware throughout the environment. The Default Domain Policy is commonly used since the policy is linked at the root of the domain, which subsequently applies the policy to all objects in the directory. In the testing environment, `dcgpofix` was used to reset both the Default Domain Policy and the Default Domain Controllers Policy. After the policies were recreated, group policy was forcibly updated on each endpoint using `gpupdate /force` from an elevated administrative command prompt.

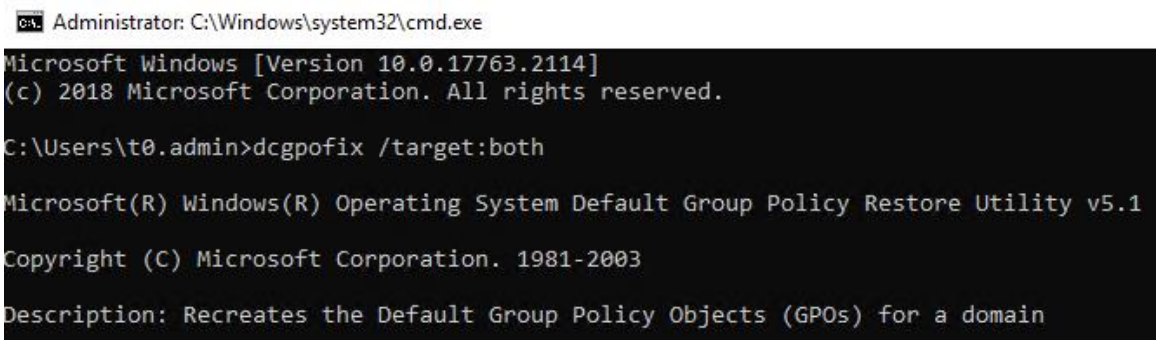

Figure 18: Dcgpofix to Recreate Both Default Policies

3.3.3. User Rights Assignments Policy

As a procedure for implementing the Tiering model in the Active Directory, Domain Group Policy can be used to limit the risk of credential exposure across Tiers. For example, a policy to prevent Tier 0 or Domain Admin account credentials from logons into lower Tier endpoints such as Tier 1 or Tier 2. This approach aims to limit exposing privileged Tier 0 credentials to systems where a threat actor can extract credentials from memory, laterally move through the environment, and ultimately achieve domain dominance.

In testing, the User Rights Assignment (URA) policy settings within Group Policy were used to prevent Tier 0 accounts from logging into a lower Tier system. Once again, the Microsoft documentation, Appendix G - Securing Administrators Groups in Active Directory, was used to configure the specific URAs configured, as shown in Figure 19 below. Also, an important step was to create a WMI filter to prevent the GPO from applying to Domain Controllers and avoid a scenario where the Domain Controllers would no longer be accessible. Then, the newly created Group Policy object was linked at the root of the domain to ensure the broadest possible coverage. Lastly, Group Policy was forcibly updated on each endpoint using `gpupdate /force` from an elevated administrative command prompt.

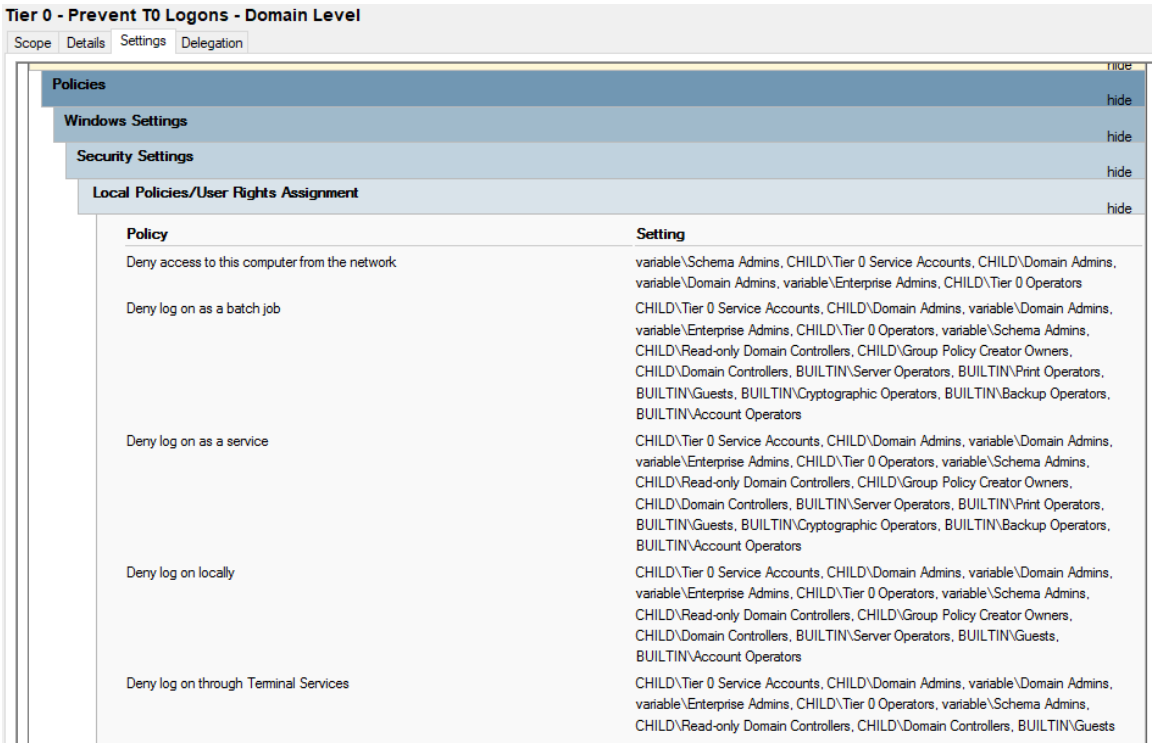

Figure 19: URAs to Prevent Tier 0 Logons at Lower Tiers

3.3.4. Basic Hardening Policy

The last set of tactical containment actions taken in the test environment was to reduce the attack surface of Active Directory. This set of actions included using Group Policy to enable NTLMv2 to harden authentication, enable SMB signing, and require LDAP signing. Separate Group Policy Objects were created for each hardening setting to allow flexibility in applying the settings to specific OUs or Tiers.

Microsoft's legacy authentication protocol, NTLM, has two versions, with v1 being more susceptible to brute force and NTLM relay attacks. As such, Group Policy was used in testing to enforce the strict usage of NTLMv2. Specifically, the "Send NTLMv2 response only. Refuse LM & NTLM" setting was configured, which meant that any device attempting authentication using LM or NTLM would be refused (Microsoft Corporation, 2023). After the setting was configured, the newly created Group Policy was linked at the root of the domain. In rare cases, production environments may have a small number of legacy devices and applications that may not support NTLMv2, which could lead to a service disruption.

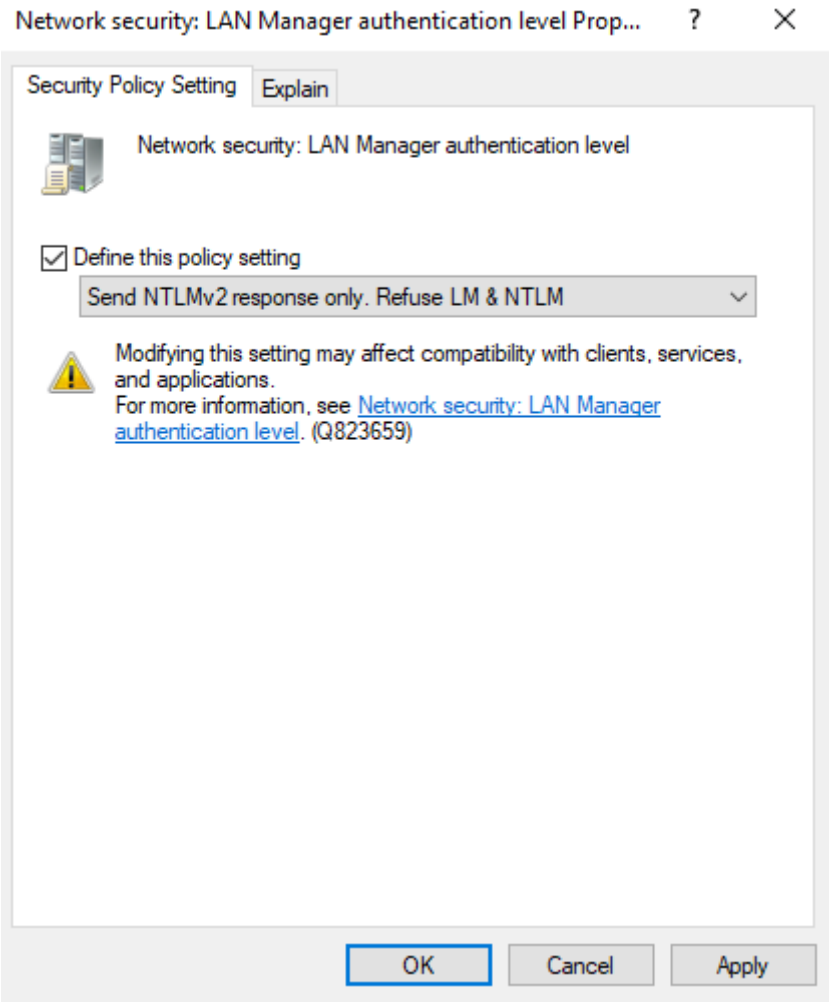

Figure 20: NTLMv2 Policy Setting

Next, SMB signing was configured as its implementation uses digital signatures to confirm the origin and the authenticity of the inbound packet. This aims to remove the possibility of tampering or attacker-in-the-middle attacks. This ensures that packets have not been altered in transit and come from the expected sender. The specific setting configured in testing was the “Microsoft network server: Digitally sign communication (if client) agrees," as shown in Figure 21 below. Again, a separate Group Policy was configured and linked at the root of the domain, allowing the broadest possible reach of the setting across the domain.

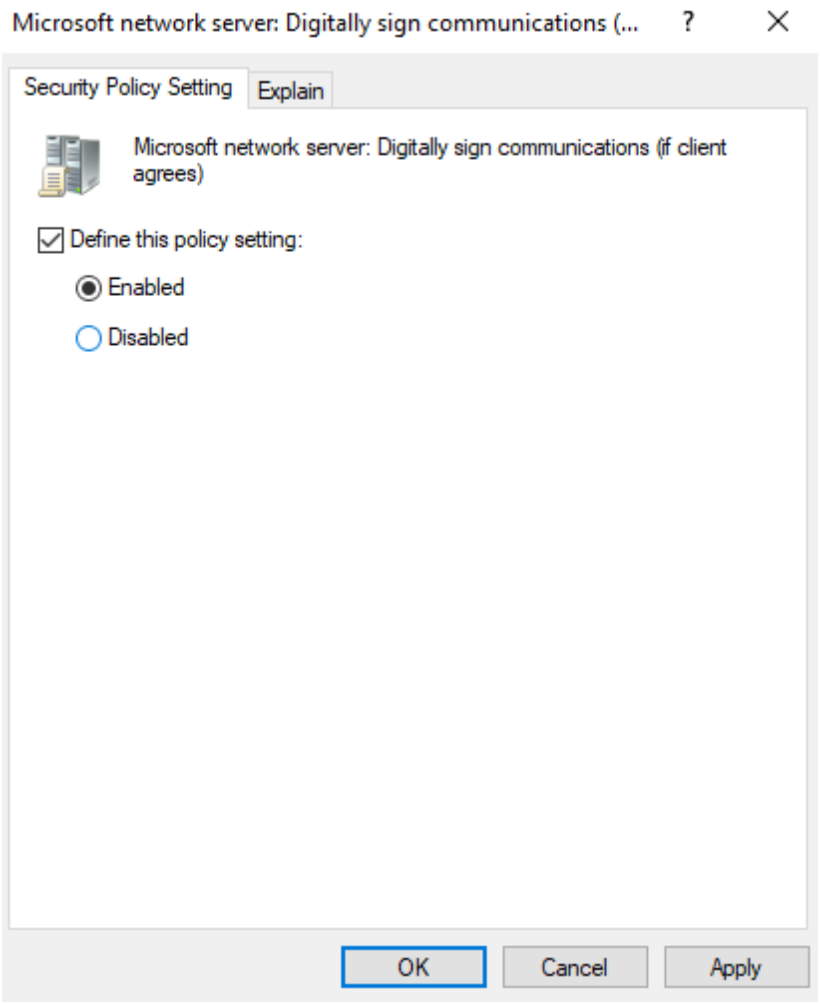

Figure 21: SMB Signing Policy Setting

The basic hardening performed during testing was configuring LDAP signing to increase security between clients and Domain Controllers. LDAP signing enablement mitigates the risk of replay attacks by adding a digital signature to each packet. The Microsoft Learn documentation, How to enable LDAP signing - Windows Server | Microsoft Learn, was used to configure LDAP signing using domain Group Policy. A third and separate Group Policy was configured and linked at the root of the domain, allowing the broadest possible reach of the setting across the domain. Lastly, Group Policy was forcibly updated on each endpoint using `gpupdate /force` from an elevated administrative command prompt.

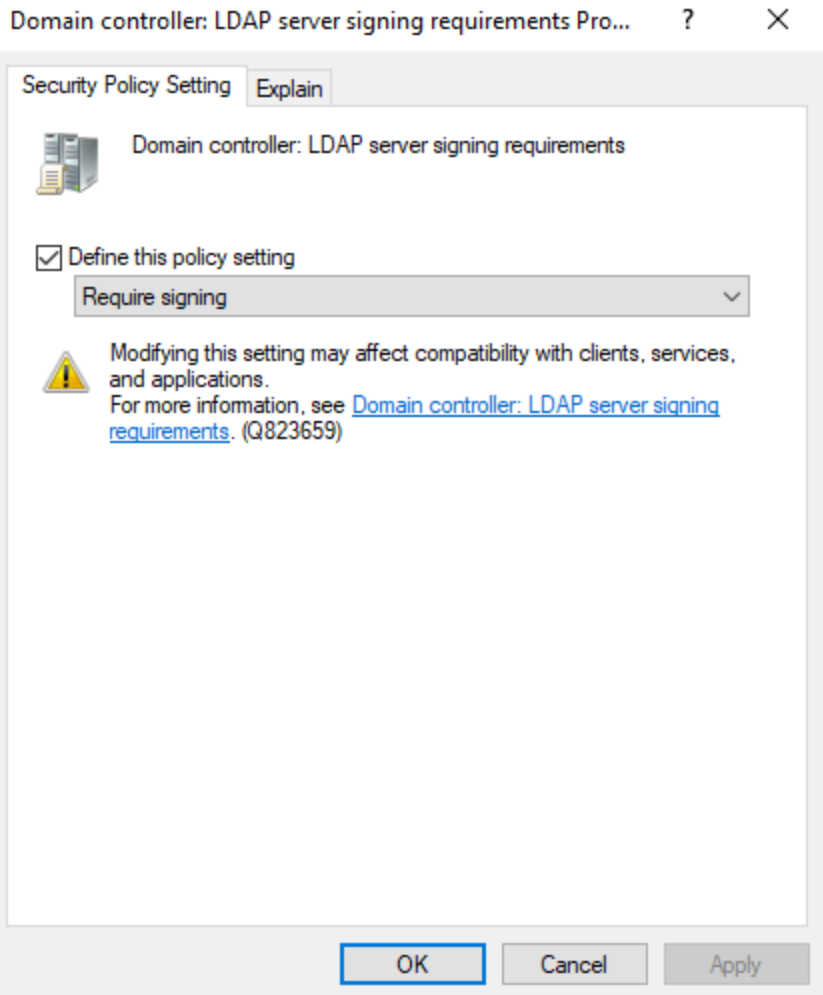

Figure 22: LDAP Signing Policy Setting

3.4. Post-Containment Data Collection

At the conclusion of the testing, data was collected again using the methodology described in Section 3.1. To summarize, Active Directory data was collected using SharpHound, executed in the control and variable environments. BloodHound Community Edition with Docker Compose was used to collate the SharpHound data. Lastly, AD_Miner extracted the graph data from BloodHound. The AD_Miner data allowed for easy comparison and analysis between the pre-containment and post-containment actions in the testing environment.

3.5. Discussion

Of note within the data collected was the total number of paths to a Domain Administrator within the variable environment – three total paths to a Domain Administrator. This data point represents a reduction of 2,297 paths, which can be attributed to the tactical containment actions performed during testing. Notably, the actions in sections 3.2.2 through 3.2.6 significantly reduced the paths to Domain Administrator by 99.87%. This reduction in paths displayed in AD_Miner is shown in Figure 23 below.

Figure 23: AD_Miner – Reduced Paths to Domain Admin

4. Recommendations and Implications

Active Directory containment to limit a threat actor's foothold in an enterprise environment is a vastly complex topic. Approaches to curb an actor's control over Active Directory may vary depending on factors such as the number of paths to domain dominance or how swiftly a threat actor wants to act on their objectives, like deploying ransomware. Furthermore, swift tactical containment actions by defenders and administrators could have unintended consequences that may be more disruptive than the threat actors themselves. Careful consideration should be factored into an organization's approach to tactical containment when responding to a domain dominance event. Lastly, in advance of a domain dominance event, defenders and administrators can proactively audit and implement these tactical containment actions in a controlled manner to reduce the likelihood of service disruptions.

4.1. Recommendations for Practice

Swift changes to Active Directory during a domain dominance event have the potential to impact the organization negatively. Challenges include de-provisioning service accounts that require elevated privileges leading to service disruptions, hardening

authentication mechanisms such as NTLM that cause unintended authentication issues with network appliances or other systems, and removing ACLs that disrupt the administration or operation of systems and applications. Understanding the potential impact on business processes and systems during tactical containment and a deep understanding of the environment is crucial to minimizing these risks.

4.2. Implications for Future Research

With such a vast topic to curb domain dominance by a determined threat actor, further research into this area remains uncharted. Several areas to further research around tactical containment to explore:

- Hybrid identity systems that extend administrative privilege into the cloud
- Tactical containment of multi-forest Active Directory environments
- A tactical approach to configuring split permissions within an on-premises Microsoft Exchange Server environment
- Mass password resets methodologies in response to NTDS.dit theft
- Tactically implementing Resource-Based Constrained Delegation (RBCD)
- Structured review of GPO changes
- Implementing Tactical PAWs

5. Conclusion

Active Directory remains widely used across organizations and is likely to see continued use for the foreseeable future. As such, considerable risk exists in the implementation and administration of Active Directory where controls such as using the Tiered model, limiting domain privileges to their fullest extent, and minimizing paths to domain dominance are not applied before or during domain dominance. Although these actions could have unintended side effects on the administration or operation of systems, having a deep understanding of Active Directory and the current state of the environment, including misconfigurations, is vital in minimizing operational disruption. Curbing risks

in Active Directory, paths to domain dominance, either before an attack or quickly implementing tactical containment actions during an attack, are essential to curb domain dominance by a determined threat actor.

References

Allen, R. (2023, November 28). *How to Restore Active Directory (Full Restore & System State)*. Retrieved from Active Directory Pro: https://activedirectorypro.com/restore-active-directory-from-backup/

AutomatedLab. (2024, February 12). *AutomatedLab*. Retrieved from GitHub: https://github.com/AutomatedLab/AutomatedLab

Bartolo, A. (2018, November 05). *Step-By-Step: Enabling Advanced Security Audit Policy via Directory Services Access*. Retrieved from Microsoft Tech Community: https://techcommunity.microsoft.com/t5/itops-talk-blog/step-by-step-enabling-advanced-security-audit-policy-via/ba-p/282452

Fan, F. (2020, September 7). *Reset krbtgt Password*. Retrieved from Microsoft Learn: https://learn.microsoft.com/en-us/answers/questions/87978/reset-krbtgt-password

Heidecker, D. (2024, February 19). *Protecting Tier 0 the Modern Way*. Retrieved from Microsoft Tech Community: https://techcommunity.microsoft.com/t5/core-infrastructure-and-security/protecting-tier-0-the-modern-way/ba-p/4052851

M365Guy. (2022, October 19). *Investigating Ransomware Deployments That Happened Via Group Policy*. Retrieved from Microsoft 365 Security: https://m365internals.com/2022/10/19/investigating-ransomware-deployments-that-happened-via-group-policy/

Metcalf, S. (2017, June 14). *Scanning for Active Directory Privileges & Privileged Accounts*. Retrieved from Active Directory Security: https://adsecurity.org/?p=3658

Metcalf, S. (2018, January 01). *Attacking Read-Only Domain Controllers (RODCs) to Own Active Directory*. Retrieved from Active Directory Security: https://adsecurity.org/?p=3592

Microsoft Corporation. (2023, July 11). *Active Directory Forest Recovery - Raise the value of available RID pools*. Retrieved from Microsoft Learn:

https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/forest-recovery-guide/ad-forest-recovery-raise-rid-pool#raise-the-value-of-available-rid-pools-using-adsiedit-and-the-calculator

Microsoft Corporation. (2023, July 11). *Active Directory Forest Recovery - Reset the krbtgt password*. Retrieved from Microsoft Learn: https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/forest-recovery-guide/ad-forest-recovery-reset-the-krbtgt-password#reset-the-krbtgt-password

Microsoft Corporation. (2023, February 17). *Network security: LAN Manager authentication level*. Retrieved from Microsoft Learn: https://learn.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/network-security-lan-manager-authentication-level#possible-values

Microsoft Corporation. (2024, February 01). *Appendix C: Protected Accounts and Groups in Active Directory*. Retrieved from Microsoft Learn: https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/appendix-c--protected-accounts-and-groups-in-active-directory#adminsdholder

Microsoft Corporation. (2024, January 07). *Persistence and privilege escalation alerts*. Retrieved from Microsoft Leatn: https://learn.microsoft.com/en-us/defender-for-identity/persistence-privilege-escalation-alerts

Prowe, D. (2024, February 12). *BadBlood*. Retrieved from GitHub: https://github.com/davidprowe/BadBlood

Pyle, N. (2019, April 04). *DS Restore Mode Password Maintenance*. Retrieved from Microsoft Tech Community: https://techcommunity.microsoft.com/t5/ask-the-directory-services-team/ds-restore-mode-password-maintenance/ba-p/396102

Zipper, E. (2009, August 21). *How to reset Default Domain Policy???*. Retrieved from Microsoft Learn: https://learn.microsoft.com/en-us/archive/msdn-technet-forums/e8a7c194-d3bf-4e1c-857c-7f779cc86705

Appendix

List of Figures

Figure 1: AD_Miner – Paths to Domain Admin 4
Figure 2: Recovery of Domain Controller #1 6
Figure 3: Microsoft Safety Scanner Results on Domain Controller #2 7
Figure 4: Ntdsutil to Reset the DSRM Password 8
Figure 5: Active Directory Administrative Tier Model 9
Figure 6: Result of Tier 0 Scripted Creation 10
Figure 7: Add New Tier 0 Accounts to the Domain Admins Group 11
Figure 8: Domain Admins Properties - *Member Of* Tab 12
Figure 9: Restoring the Defaults for the AdminSDHolder Object 13
Figure 10: Orphaned AdminSDHolder Accounts Before Remediation 14
Figure 11: New rIDAvailablePool Value 14
Figure 12: Krbtgt Password Reset 15
Figure 13. PowerShell code to find the RID 500 account 16
Figure 14: ITA Password Reset on the Forest Side of the Trust 16
Figure 15: Constrained Delegation Checkbox 17
Figure 16: Restoring Security Defaults on the Domain Controllers OU 18
Figure 17: Tier 0 Auditing Policy Sample Settings 20
Figure 18: Dcgpofix to Recreate Both Default Policies 20
Figure 19: URAs to Prevent Tier 0 Logons at Lower Tiers 22
Figure 20: NTLMv2 Policy Setting 23
Figure 21: SMB Signing Policy Setting 24
Figure 22: LDAP Signing Policy Setting 25
Figure 23: AD_Miner – Reduced Paths to Domain Admin 26

Appendix
Built-in Administrator & Sensitive Groups

Built-in Group	*Members* (default)	*Member of*
Enterprise Admins	Remove All – Empty	Administrators, Denied RODC Password Replication Group
Schema Admins	Remove All – Empty	Denied RODC Password Replication Group
Domain Admins	Built-in Administrator (RID 500) and new Tier-0 accounts	Administrator, Denied RODC Password Replication Group
DNS Admins	Remove All – Empty	Remove All – Empty
Key Admins	Remove All – Empty	Remove All – Empty
Enterprise Key Admins	Remove All – Empty	Remove All – Empty
Cert Publishers	Certificate Authority (CA) Server computer objects only	Denied RODC Password Replication Group
Denied RODC Password Replication Group	Cert Publishers, Domain Admins, Domain Controllers, Enterprise Admins, Group Policy Creator Owners, Group Policy Creator Owner, krbtgt, Read-only Domain Controllers, Schema Admins	Remove All – Empty

Domain Controllers	Legitimate Domain Controller (DC) computer objects only	Denied RODC Password Replication Group
Allowed RODC Password Replication Group	Remove All – Empty	Remove All – Empty
DnsUpdateProxy	Remove All – Empty	Remove All – Empty
Enterprise Read-only Domain Controllers	Remove All – Empty	Remove All – Empty
Group Policy Creator Owners	Built-in Administrator (RID 500)	Denied RODC Password Replication Group
Cloneable Domain Controllers	Remove All – Empty	Remove All – Empty
Sensitive Group	*Members* (default)	*Member of*
Access Control Assistance Operators	Remove All – Empty	Remove All – Empty
Account Operators	Remove All – Empty	Remove All – Empty
Administrators	Built-in Administrator (RID 500), Administrator, Domain Admins, Enterprise Admins	Remove All – Empty
Certificate Service DCOM Access	Remove All – Empty	Remove All – Empty
Cryptographic Operators	Remove All – Empty	Remove All – Empty
Distributed COM Users	Remove All – Empty	Remove All – Empty

Event Log Readers	Remove All – Empty	Remove All – Empty
Hyper-V Administrators	Remove All – Empty	Remove All – Empty
Incoming Forest Trust Builders	Remove All – Empty	Remove All – Empty
Network Configuration Operators	Remove All – Empty	Remove All – Empty
Print Operators	Remove All – Empty	Remove All – Empty
Remote Desktop Users	Built-in Administrator (RID 500) only	Remove All – Empty
Replicator	Remove All – Empty	Remove All – Empty
Server Operators	Remove All – Empty	Remove All – Empty
Storage Replica Administrators	Remove All – Empty	Remove All – Empty
Users	Authenticated Users, Domain Users	Remove All – Empty
Windows Authorization Access Group	Enterprise Domain Controllers only	Remove All – Empty

Appendix

Tactical Containment Checklist

Tactical Containment Actions	References	Percentage Complete	Notes and comments
Pre-Containment Data Collection			
Setup BloodHound	Install BloodHound Community Edition with Docker Compose – BloodHound (bloodhoundenterprise.io)	0%	
Setup AD_Miner	Mazars-Tech/AD_Miner (github.com)	0%	
Execute SharpHound (for every domain and forest)	Releases · BloodHoundAD/SharpHound (github.com)	0%	
Import data into BloodHound	Getting started with BloodHound Community Edition – BloodHound (bloodhoundenterprise.io)	0%	
Import data into AD_Miner	Mazars-Tech/AD_Miner (github.com)	0%	
Data analysis	BloodHound – Sniffing Out the Path Through Windows Domains | SANS Institute	0%	
Active Directory Actions			
Infrastructure – Restore DCs from clean source backup	How to Restore Active Directory (Full Restore & System State) - Active Directory Pro	0%	
Infrastructure – Execute Microsoft Safety Scanner on non-compromised DCs	Microsoft Safety Scanner Download | Microsoft Learn	0%	

Infrastructure – Validate AD & SYSVOL replication health	AD Forest Recovery - Verify Replication | Microsoft Learn	0%	
Infrastructure – Reset DSRM passwords (all DCs)	How to reset the Directory Services Restore Mode administrator account password - Windows Server | Microsoft Learn	0%	
Tiering – Setup Tier 0	AD-Tier-Administration/Create-Structure.ps1 at master · SalutAToi/AD-Tier-Administration (github.com)	0%	
Cleanup – Create two new privileged accounts	Active Directory security groups | Microsoft Learn	0%	
Cleanup – Built-in group cleanup	Active Directory security groups | Microsoft Learn	0%	
Cleanup – Sensitive group cleanup	Active Directory security groups | Microsoft Learn	0%	
Cleanup – AdminSDHolder permissions	Detect and correct orphaned 'adminCount=1' users who are no longer in protected groups (github.com)	0%	
Cleanup – AdminSDHolder orphans	Detect and correct orphaned 'adminCount=1' users who are no longer in protected groups (github.com)	0%	
Cleanup – Raise RID Pool	AD Forest Recovery - Raising RID pools | Microsoft Learn	0%	
Cleanup – Invalidate RID pool	AD Forest Recovery - Invalidating the RID Pool | Microsoft Learn	0%	

Accounts – Reset krbtgt password	AD Forest Recovery - Resetting the krbtgt password | Microsoft Learn	0%	1st reset
Accounts – Reset krbtgt password	KRBTGT Account Password Reset Scripts now available for customers | Microsoft Security Blog	0%	2nd reset, at least 10-hours apart
Accounts – Reset RID 500 password	What's special about the builtin Administrator account? | Morgan Simonsen's Blog	0%	1st reset
Accounts – Reset RID 500 password	What's special about the builtin Administrator account? | Morgan Simonsen's Blog	0%	2nd reset
Accounts – Reset ITA (trust) passwords	AD Forest Recovery - Resetting a trust password | Microsoft Learn	0%	Each side of EVERY trust
ACL Cleanup – OUs	canix1/ADACLScanner: Repo for ADACLScan.ps1 - Your number one script for ACL's in Active Directory (github.com)	0%	Review EVERY OU (including sub-OUs) ACLs and cleanup
ACL Cleanup – GPOs	Active Directory Group Policy objects must have proper access control permissions. (stigviewer.com)	0%	Review ACLs for all GPOs and cleanup
Group Policy Actions			
GPO – Auditing Policy	AndyFriar/palantir-wef-gpo: Audit Policy GPO's for Palantir WEF (github.com)	0%	
GPO – Default Domain Policy reset	How to reset Default Domain Policy??? | Microsoft Learn	0%	

GPO – Default Domain Controllers Policy reset	How to reset Default Domain Policy??? | Microsoft Learn	0%	
GPO – User Rights Assignments (URA) policy	Appendix G - Securing Administrators Groups in Active Directory | Microsoft Learn	0%	
GPO – Basic Hardening – NTLMv2 only	Network security LAN Manager authentication level - Windows 10 | Microsoft Learn	0%	
GPO – Basic Hardening – SMB Signing	Configure SMB Signing with Confidence - Microsoft Community Hub	0%	
GPO – Basic Hardening – LDAP Signing	How to enable LDAP signing - Windows Server | Microsoft Learn	0%	
Post-Containment Data Collection			
Execute SharpHound (for every domain and forest)	Releases · BloodHoundAD/SharpHound (github.com)	0%	
Import data into BloodHound	Getting started with BloodHound Community Edition – BloodHound (bloodhoundenterprise.io)	0%	
Import data into AD_Miner	Mazars-Tech/AD_Miner (github.com)	0%	

Data analysis	BloodHound – Sniffing Out the Path Through Windows Domains | SANS Institute	0%	