Sneaky Active Directory Persistence #17: Group Policy

By Sean Metcalf

Published: 2016-03-14 · Archived: 2026-04-05 18:41:20 UTC

The content in this post describes a method through which an attacker could persist administrative access to Active Directory after having Domain Admin level rights for about 5 minutes.

Complete list of Sneaky Active Directory Persistence Tricks posts

This post explores how an attacker could leverage the built-in Active Directory management capability called Group Policy and how to mitigate potential security issues.

Group Policy Overview

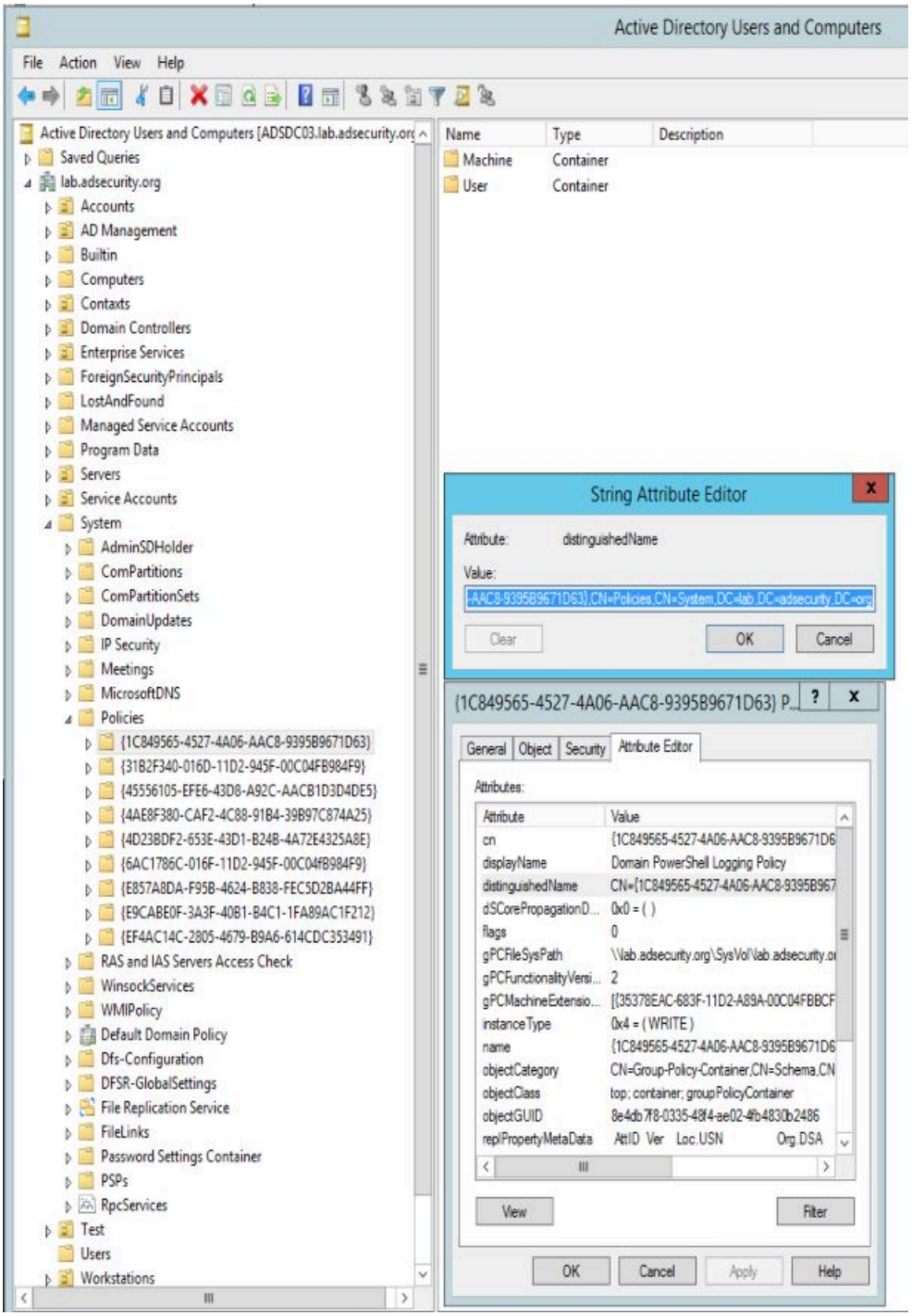
One of the key benefits to Active Directory is its management capability and core to this capability is Group Policy. Group Policy has several parts to it and can be challenging to manage in a large enterprise without third-party tools.

Group Policy enables administrators to manage computers and users in Active Directory. Group Policies are saved as Group Policy Objects (GPOs) which are then associated with Active Directory objects such as sites, domains, or organizational units (OUs). Group Policies can include security options, registry keys, software installation, and scripts for startup and shutdown and domain members refresh group policy settings every 90 minutes by default (5 minutes for Domain Controllers). This means that Group Policy enforces configured settings on the targeted computer.

In most Active Directory implementations, there is at least one GPO configured on the domain defining mandated password, Kerberos, and domain-wide policies; at least one GPO configured for the Domain Controllers OU; and at least one GPO configured for a servers and workstations OU. These GPOs define security settings specific to the environment and often configure administrative groups, include startup/shutdown scripts, etc.. GPOs can be configured to set organization-defined security requirements at each level, and can be used for installing software and setting file and registry permissions. GPOs only apply to users and computers and can be filtered with groups or more specifically targeted using the Preferences component. The “No Override” option ensures that the settings in a Group Policy are applied even if a GPO closer to the resource has contradicting settings.

There are two Group Policy components:

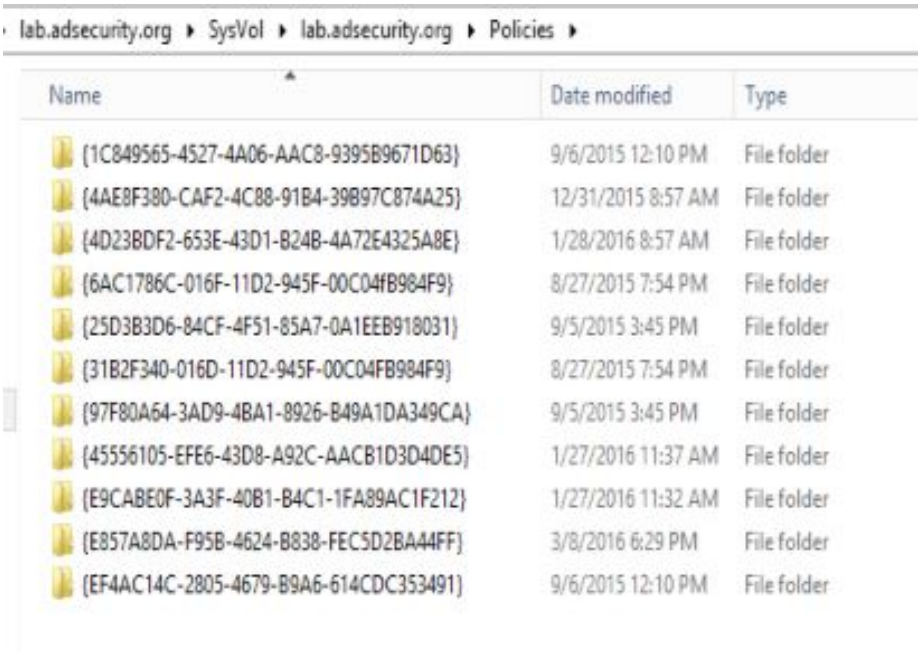
1. The “Group Policy Container” is stored in Active Directory (<DOMAIN>, System, Policies)



2. The files that actually contain the policy settings (collectively referred to as the "Group Policy Template") are stored in SYSVOL.

All domain Group Policies are stored in the following domain share: \\<DOMAIN>\SYSVOL\

<DOMAIN>\Policies\



Each Group Policy Object in Active Directory has the following attributes (on the policy object in AD):

- displayName: This is the name given to the GPO by the creator.
- gPCFileSysPath: This points to the location in SYSVOL where the associated GPO files (aka “Group Policy Template”) are located.
- gPCMachineExtensionNames: This attribute lists the GPO client side extensions (CSEs) required to by the client process the machine specific Group Policy settings.
- gPCUserExtensionNames: This attribute lists the GPO client side extensions (CSEs) required to by the client process the user specific Group Policy settings.

Using the PowerShell Active Directory module cmdlet “Get-ADObject“, we can retrieve key GPO specific attributes for the GPO.

```
PS C:\> get-adobject 'CN={1C849565-4527-4A06-AAC8-9395B9671D63},CN=Policies,CN=System,DC=lab,DC=adsecurity,DC=org'
 -Properties displayname,gpcfilesyspath,gpcmachineextensionnames,gpcuserextensionnames


DisplayName                : Domain PowerShell Logging Policy
DistinguishedName          : CN={1C849565-4527-4A06-AAC8-9395B9671D63},CN=Policies,CN=System,DC=lab,DC=adsecurity,DC=org
gpcfilesyspath             : \\lab.adsecurity.org\SysVol\lab.adsecurity.org\Policies\{1C849565-4527-4A06-AAC8-9395B9671D63}
gpcmachineextensionnames : [{35378EAC-683F-11D2-A89A-00C04FBBCFA2}{D02B1F72-3407-48AE-BA88-E8213C6761F1}]
Name                       : {1C849565-4527-4A06-AAC8-9395B9671D63}
ObjectClass                : groupPolicyContainer
ObjectGUID                 : 8e4db7f8-0335-48f4-ae02-4fb4830b2486
```

The PowerShell Active Directory module can be easily installed on Windows Server 2008 R2 (and newer) by running the following command in an Administrator PowerShell console:

> *Import-module servermanager ; add-windowsfeature rsat-ad-PowerShell*

Additionally, every Group Policy has a GPO GUID used to connect GPO components:

- The GPO policy files are stored in a GPO object with the GPO GUID as the name.

- The Group Policy Template files in SYSVOL are stored in a folder with the GPO GUID as the name.
- The GPO policy object Distinguished Name is added to the attribute “gPLink” on the Organizational Unit (OU) the GPO is linked to.

When a new GPO is created, it can be created in AD and not linked (in which case, it does nothing), or linked to an OU, domain, or site. Upon creation, a new Group Policy Object is created in the Group Policy Container (<DOMAIN>, System, Policies) and the associated files are created in SYSVOL structure (based on GPO GUID name). When linking a Group Policy to an OU, for example, the OU’s “gPLink” attribute is updated with the GPO’s Distinguished Name. This provides a method for the computer to identify what group policies apply to itself as well as any that apply to logging on user(s).

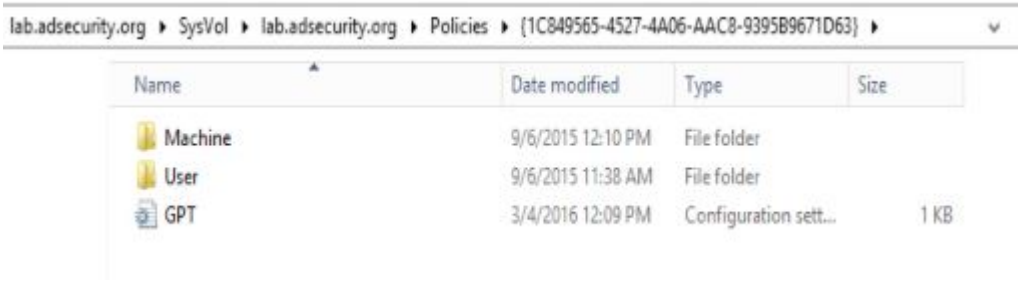
Using the PowerShell Active Directory module cmdlet “Get-ADOrganizationalUnit“, we can retrieve the Group Policies linked to the “Servers” OU.

```
PS C:\> Get-ADOrganizationalUnit 'OU=Servers,DC=lab,DC=adsecurity,DC=org' -prop GPLink


City                    :
Country                 :
DistinguishedName       : OU=Servers,DC=lab,DC=adsecurity,DC=org
GPLink                  : [LDAP://cn={E857A8DA-F95B-4624-B838-FEC5D2BA44FF},cn=policies,cn=system,DC=lab,DC=adsecurity,DC=org;
                          -3A3F-40B1-B4C1-1FA89AC1F212},cn=policies,cn=system,DC=lab,DC=adsecurity,DC=org;0]
LinkedGroupPolicyObjects : {cn={E857A8DA-F95B-4624-B838-FEC5D2BA44FF},cn=policies,cn=system,DC=lab,DC=adsecurity,DC=org,
                          cn={E9CABE0F-3A3F-40B1-B4C1-1FA89AC1F212},cn=policies,cn=system,DC=lab,DC=adsecurity,DC=org}
ManagedBy               :
Name                    : Servers
ObjectClass             : organizationalUnit
ObjectGUID              : a5b91aee-f168-451d-bd90-07d622a6caec
PostalCode              :
State                   :
StreetAddress           :
```

SYSVOL is the domain-wide share in Active Directory to which all authenticated users have read access. SYSVOL contains logon scripts, group policy data, and other domain-wide data which needs to be available anywhere there is a Domain Controller. The SYSVOL share is automatically synchronized and shared among all Domain Controllers.

Within this Policies folder are folders for each GPO with the folder name the same as that GPO’s GUID. Each GPO folder in SYSVOL has the following:

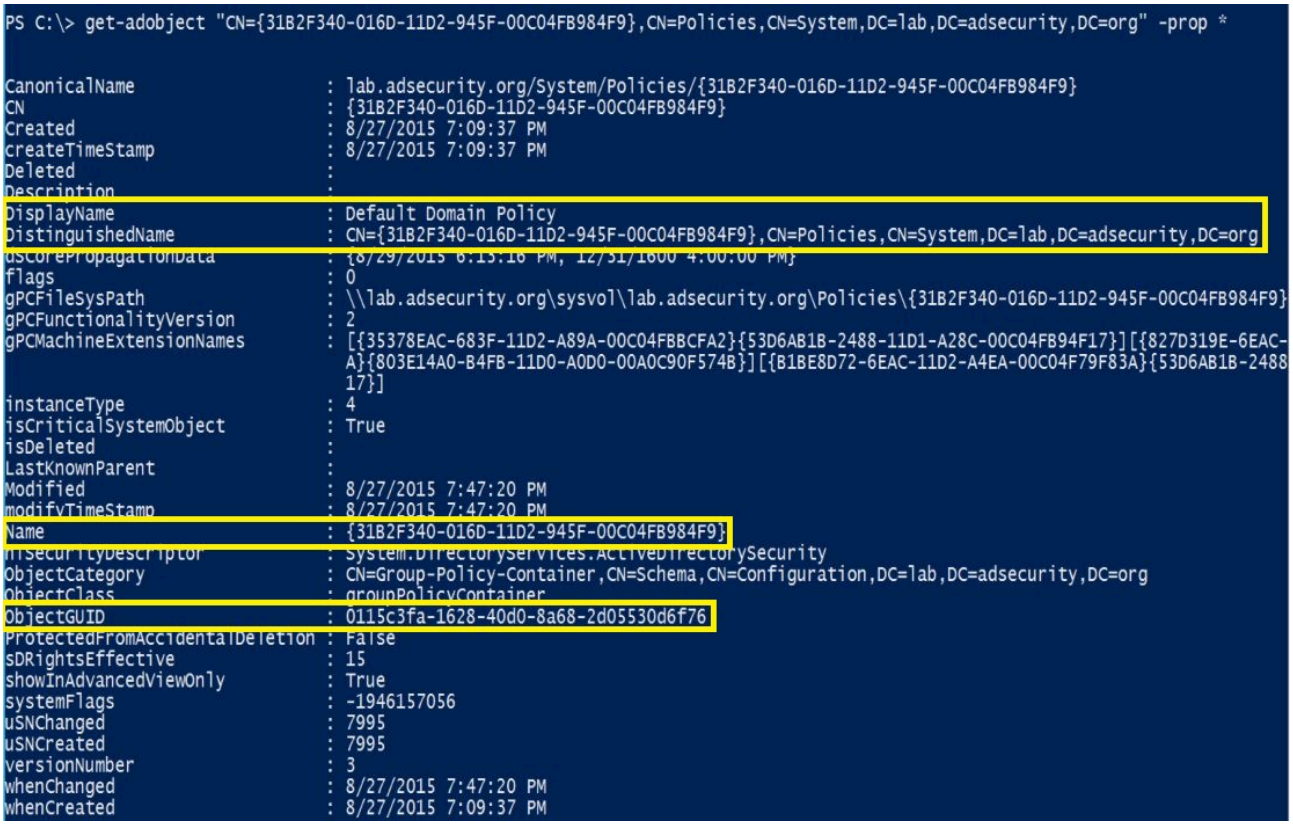
- Machine – this folder contains the machine specific settings for the GPO.
- User – this folder contains the user specific settings for the GPO.
- GPT.INI – this file contains the configuration settings for the GPO.



Note that the GPO is tracked in AD via the GPO GUID which has a separate AD object GUID for the AD object. There are a few different reasons for this, and one of the key reasons is to ensure there are predictable GUIDs for

specific Group Policy Objects regardless of the Active Directory instance. The “Default Domain Policy” GPO’s GUID is “31B2F340-016D-11D2-945F-00C04FB984F9” and the “Default Domain Controller Policy ” GPO’s GUID is “6AC1786C-016F-11D2-945F-00C04FB984F9” by default.

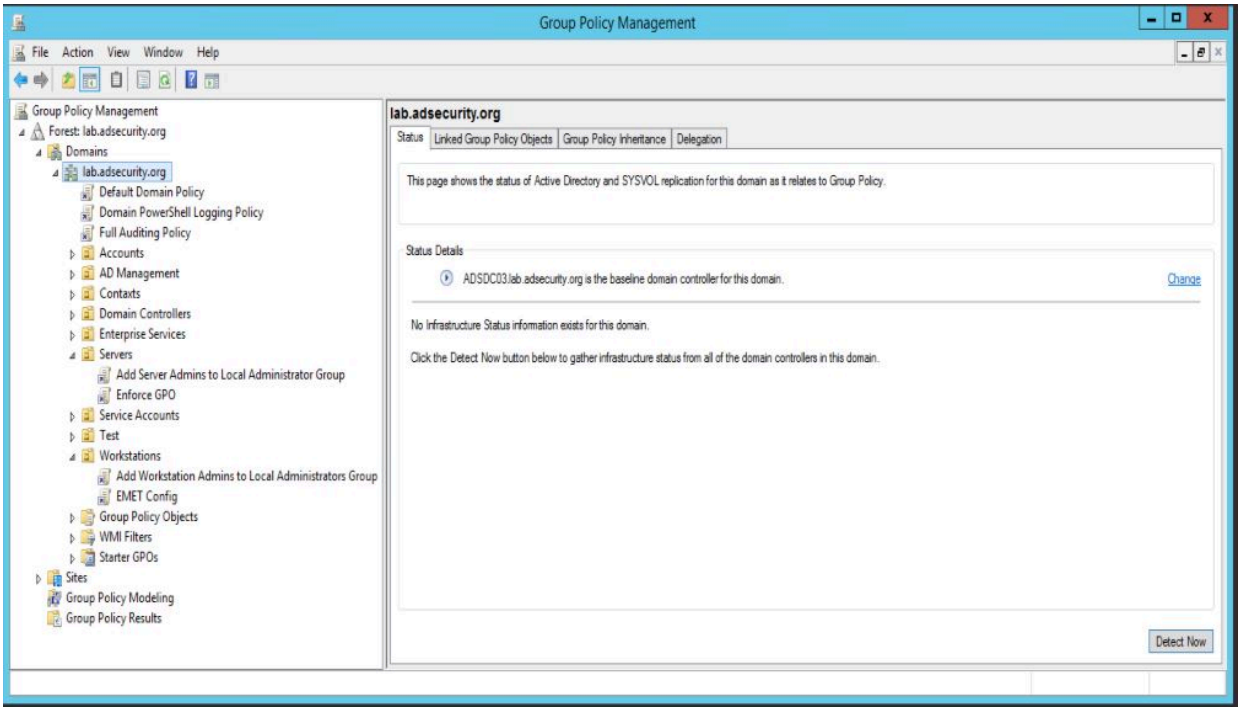
In this graphic the ObjectGUID attribute is “0115c3fa-1628-40d0-8a68-2d05530d6f76” which is obviously not the same as the GPO GUID “31B2F340-016D-11D2-945F-00C04FB984F9”.

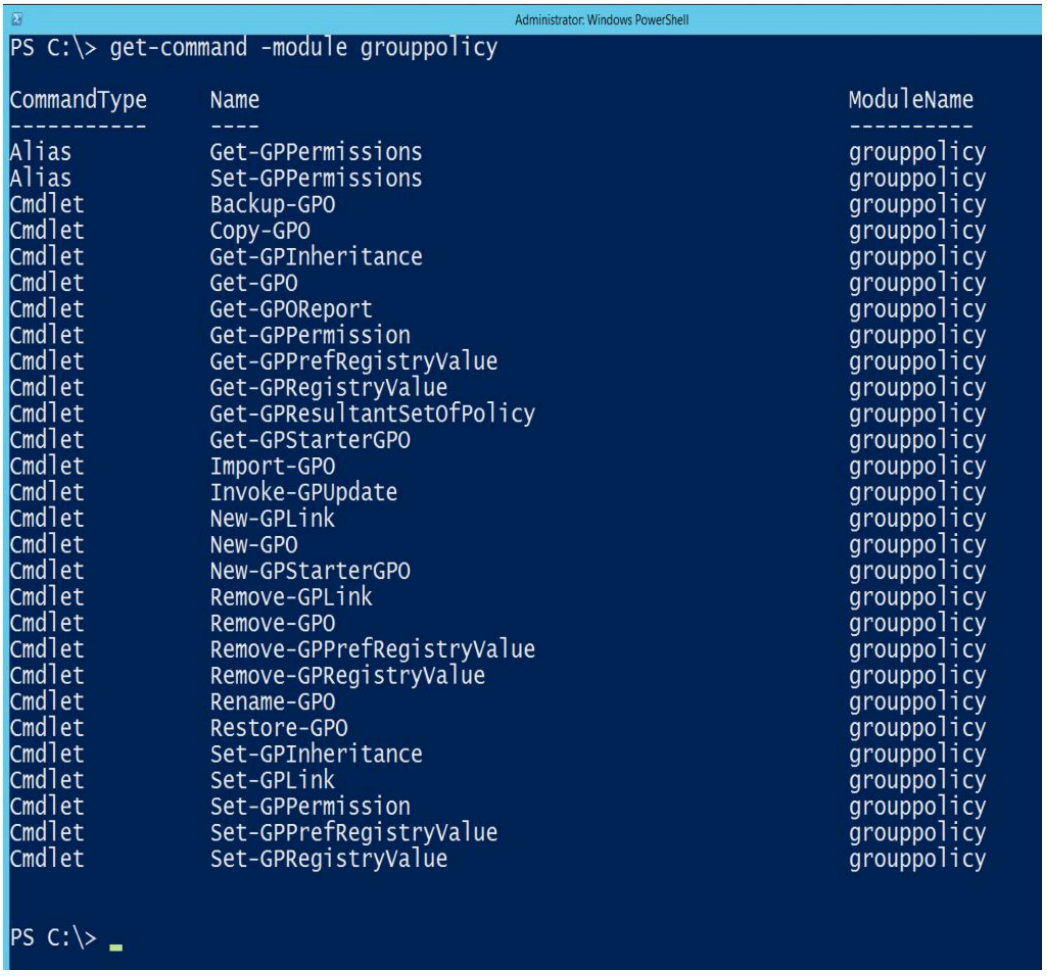


Group Policy Management

Group Policy management is often delegated in large enterprises so several different organizations are able to create, modify, and delete Group Policies. This issue with this is that Group Policy quickly gets unruly and difficult to manage since many more than the originally designed (and selected) admins have GPO admin rights. These rights are often delegated at the domain level, so edit (or full) rights apply to all domain GPOs, even those that apply to the Domain (everything) and/or Domain Controllers.

The Group Policy Management Console (GPMC) is the primary tool for Group Policy administration and there’s a PowerShell module (GroupPolicy) which is extremely useful for reporting on and backing up GPOs (please backup the domain GPOs regularly) using Backup-GPO.





Group Policy Persistence Capability

Group Policy was designed to provide simplified management of resources in a domain, though its capability can also be co-opted by an attacker to push out malware, create/modify scheduled tasks, downgrade credential protections, add a new local account to all computers that are added to the local Administrators group. and even change existing security policies enabling clear-text password extraction.

Some possibilities:

- Configure a PowerShell or VBS script to set group membership at the domain or server level
- Perform one of the other "Sneaky Persistence Tricks" I outlined previously.
- Running Invoke-Mimikatz on all Domain Controllers as SYSTEM every week.
- Pull the KRBTGT account and then schedule a task that runs DCSync on certain computers throughout the forest using forged Kerberos tickets.
- Install & Re-install malware on every computer in the Domain/Forest.
- Dump all Microsoft LAPS passwords for all computer local Administrator accounts by running a PowerShell script automatically on one or all Domain Controllers. There are plenty of options for an attacker once Group Policy is part of their toolkit.

In fact, the Mandiant M-Trends 2016 report covering activity in 2015 includes information about how attackers are leveraging Group Policy to deploy malware:

> An attacker with domain administrator-level access to a victim's Active Directory environment attempted to distribute ransomware through scheduled tasks and Group Policy objects (GPOs). The attacker created a scheduled task and pushed it onto the target systems via GPOs. The scheduled task loaded a malicious script from the domain controller (DC). The script then copied over an executable from the DC to the target systems and executed it. The executable was designed to encrypt user files (documents, photos, emails, backups, etc.) on the file system and instruct the victim to visit a website that contained instructions to obtain the decryption key.

Note that the person who hacked Haking Team leveraged Group Policy as part of the hack: http://pastebin.com/raw/0SNSvyjJ

Red Team Note on Group Policy:
The default Group Policy application behavior is to "refresh the group policy" on the client, though this doesn't actually mean the GPO settings are re-applied. By default, the GPO's settings are only reapplied if the GPO was modified prior to the refresh. This means that one could reverse a GPO enforced setting via the computer's registry (typically with admin rights) and the unauthorized setting remains until the GPO is modified, after which the GPO settings are re-applied.

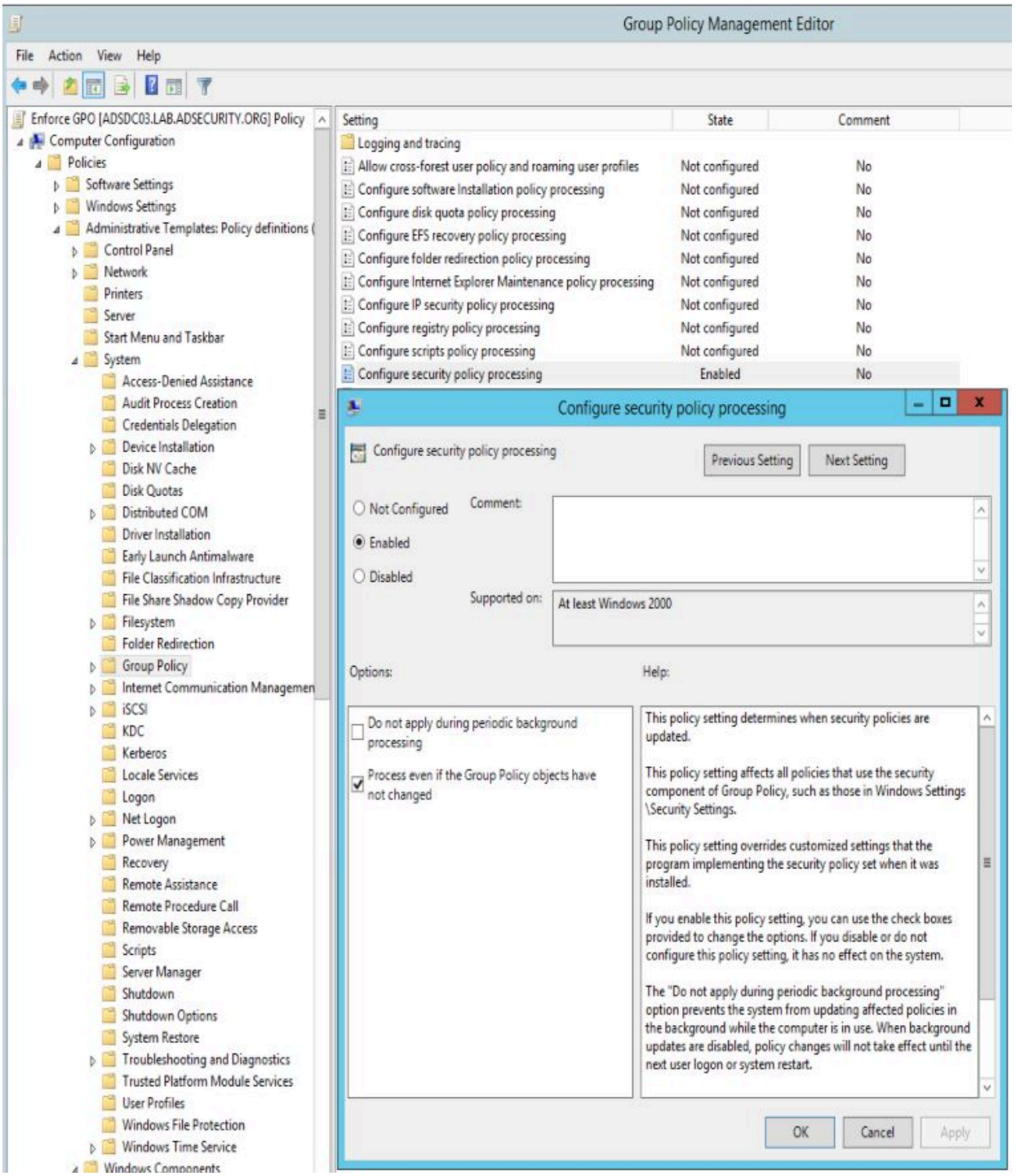
Blue Team Defenses:
After testing, change the Group Policy default setting to re-apply GPO settings at every refresh (Process even if

the Group Policy objects have not changed). This does have a potential performance hit on the client, but will ensure all GPO enforced settings are re-applied.

Computer Configuration, Policies, Administrative Templates, System, Group Policy, Configure security policy processing: Set to Enabled.
Also check the box for "*Process even if the Group Policy objects have not changed*"

It's also recommended to configure the same settings for each of the following:

- *Computer Configuration, Policies, Administrative Templates, System, Group Policy, Configure registry policy processing*
- *Computer Configuration, Policies, Administrative Templates, System, Group Policy, Configure scripts policy processing*
- As well as any other policy settings as needed.



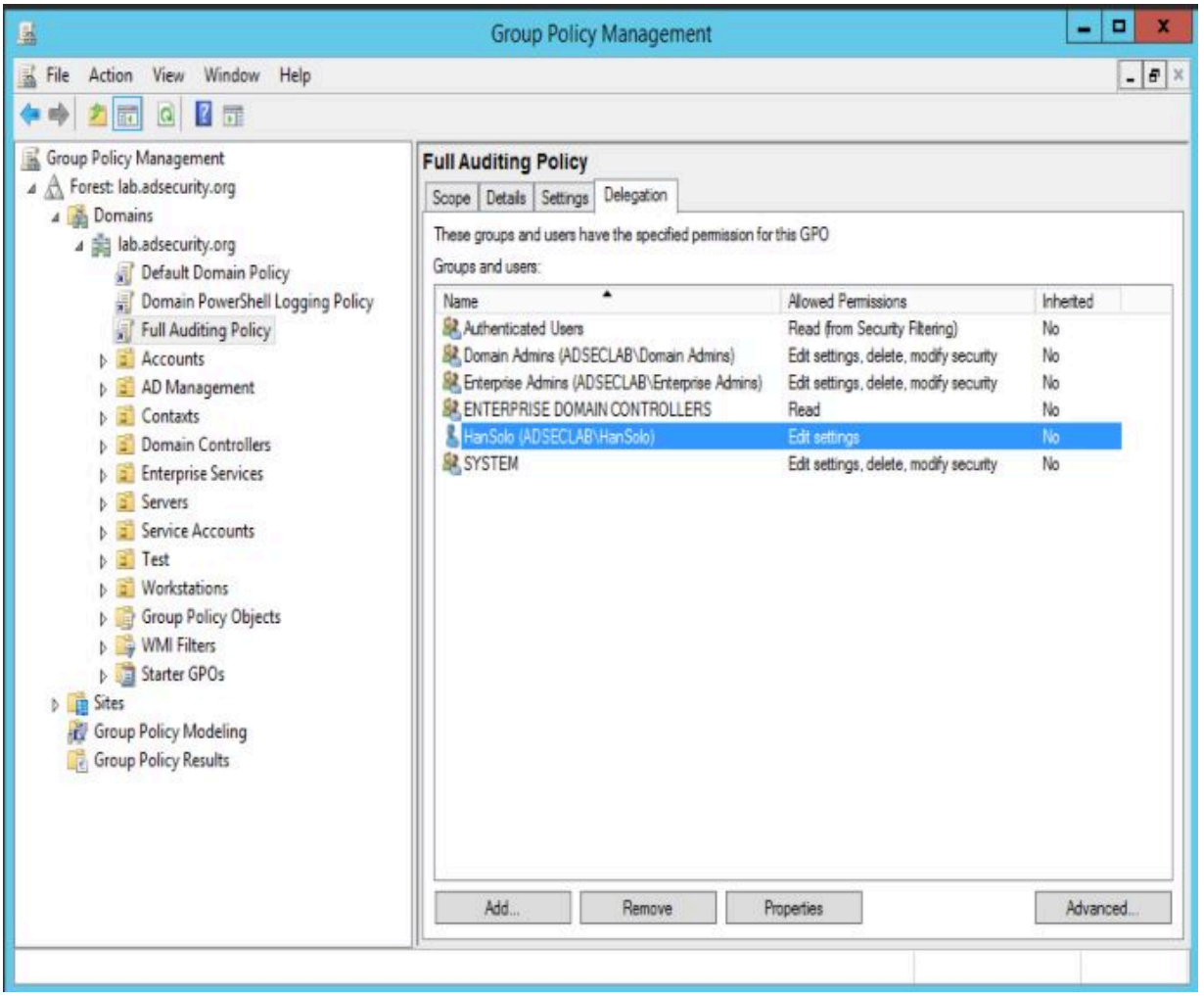
Group Policy Exploit Capability

Though this post focuses on retaining domain-level privileged access ("Domain Admin" rights), there are several ways in which an existing organization's Group Policy configuration could be used to escalate access. The obvious method is to exploit existing Group Policy Preference credentials in the environment which enables an attacker to escalate access from domain user to server/application/OU admin, or even Domain Admin. The less obvious method involves finding GPOs linked at either the domain or a top level OU with custom security settings.

Based on AD security assessments I have performed, I've found that organizations frequently have GPOs linked at a high level with custom security settings providing edit rights to accounts that are not Active Directory Administrators. This provides an avenue for privilege escalation since the GPO can be reconfigured to run a script or change security.

PowerView, now integrated into PowerSploit, includes some interesting Group Policy enumeration capability via PowerShell.

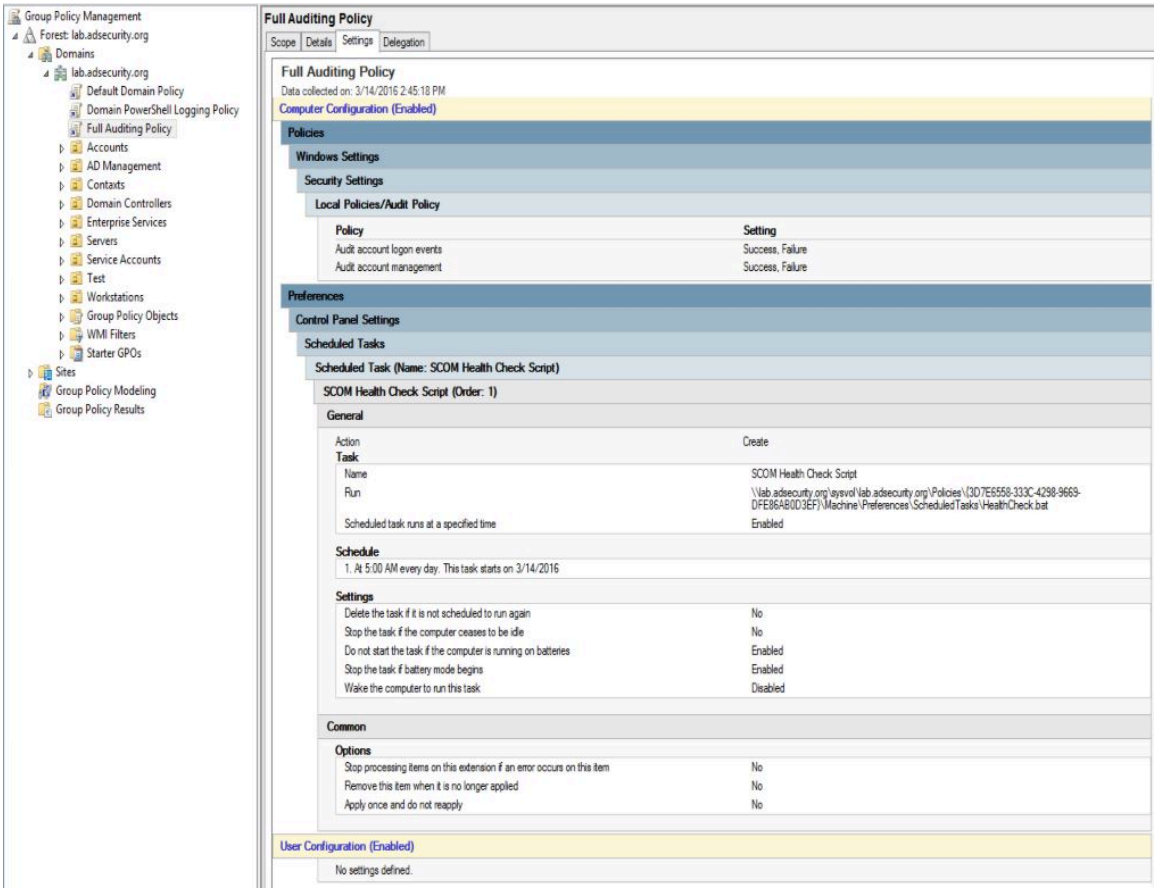
The following example shows a Group Policy called "Full Auditing Policy" linked at the Domain level which has "Edit settings" rights delegated to the "Han Solo" (Server Admin) account.



Han Solo is a member of the "Server Admins" group which is not a domain admin group.

```
PS C:\> get-aduser hansolo -prop memberof


DistinguishedName : CN=HanSolo,OU=AD Management,DC=lab,DC=adsecurity,DC=org
Enabled           : True
GivenName         :
MemberOf          : {CN=Server Admins,OU=AD Management,DC=lab,DC=adsecurity,DC=org}
Name              : HanSolo
ObjectClass       : user
ObjectGUID        : dddf808a-484f-44c3-bbd6-b1261681dd6f
SamAccountName    : HanSolo
SID               : S-1-5-21-1581655573-3923512380-696647894-2631
Surname           :
UserPrincipalName :



PS C:\> get-adgroup 'CN=Server Admins,OU=AD Management,DC=lab,DC=adsecurity,DC=org'  -prop memberof


DistinguishedName : CN=Server Admins,OU=AD Management,DC=lab,DC=adsecurity,DC=org
GroupCategory     : Security
GroupScope        : Global
MemberOf          : {}
Name              : Server Admins
ObjectClass       : group
ObjectGUID        : 3877c311-9321-41c0-a6b5-c0d88684b335
SamAccountName    : ServerAdmins
SID               : S-1-5-21-1581655573-3923512380-696647894-2628
```

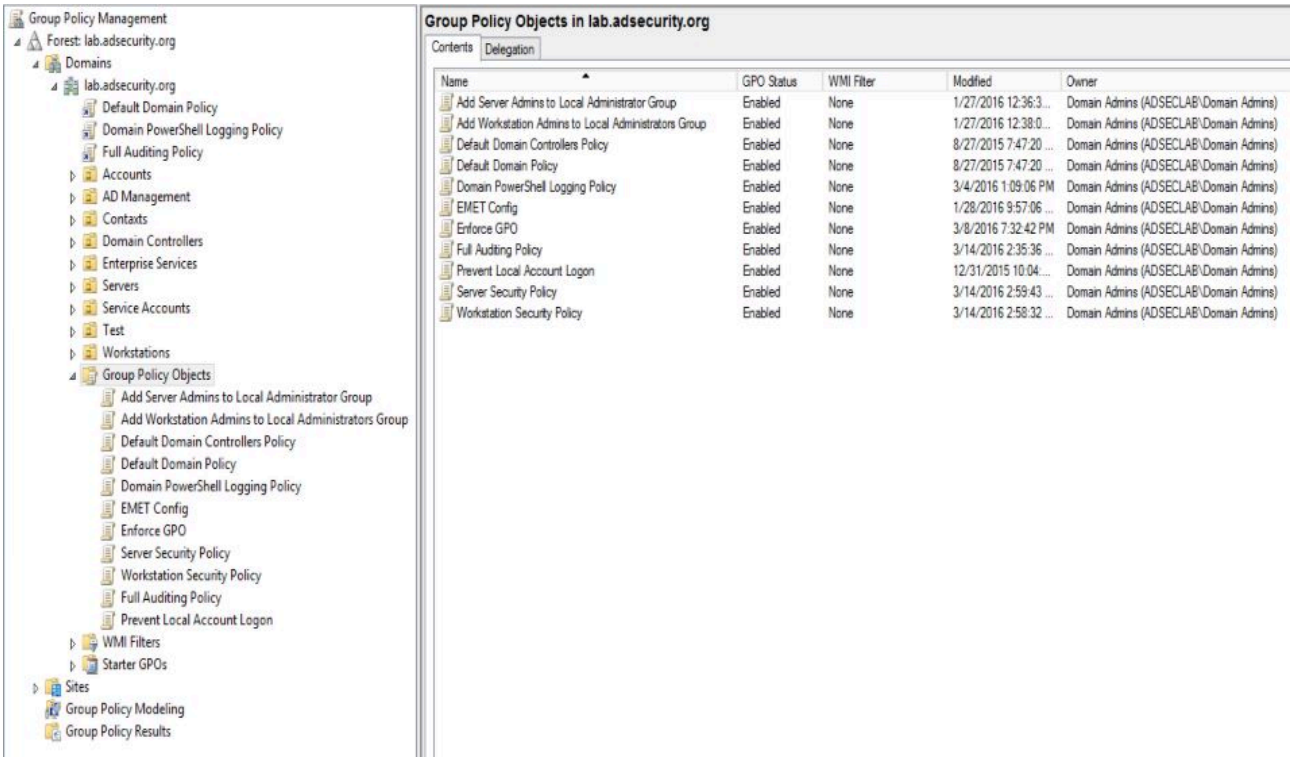
Since Han Solo has the rights to edit this domain linked GPO, let's modify it.



After editing the Group Policy, this GPO will now add a scheduled task on every computer in the domain, enabling any type of activity the attacker wishes.

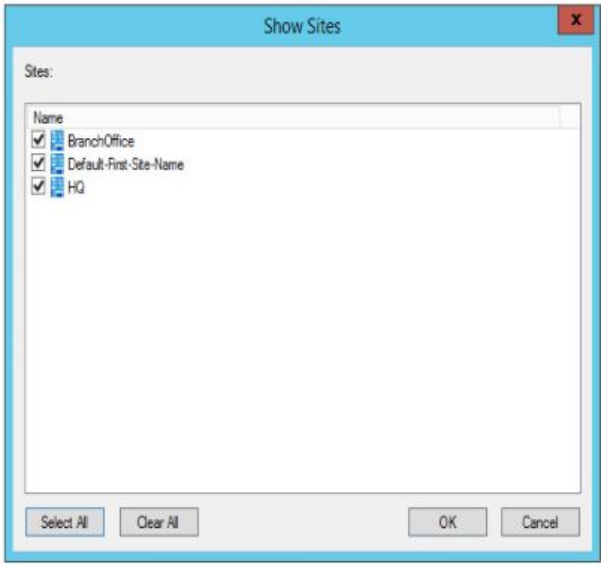
“Hidden Group Policy” – Group Policies Applied to Sites

We know that Group Policy is typically applied to Organizational Units and we can easily view them in GPMC.

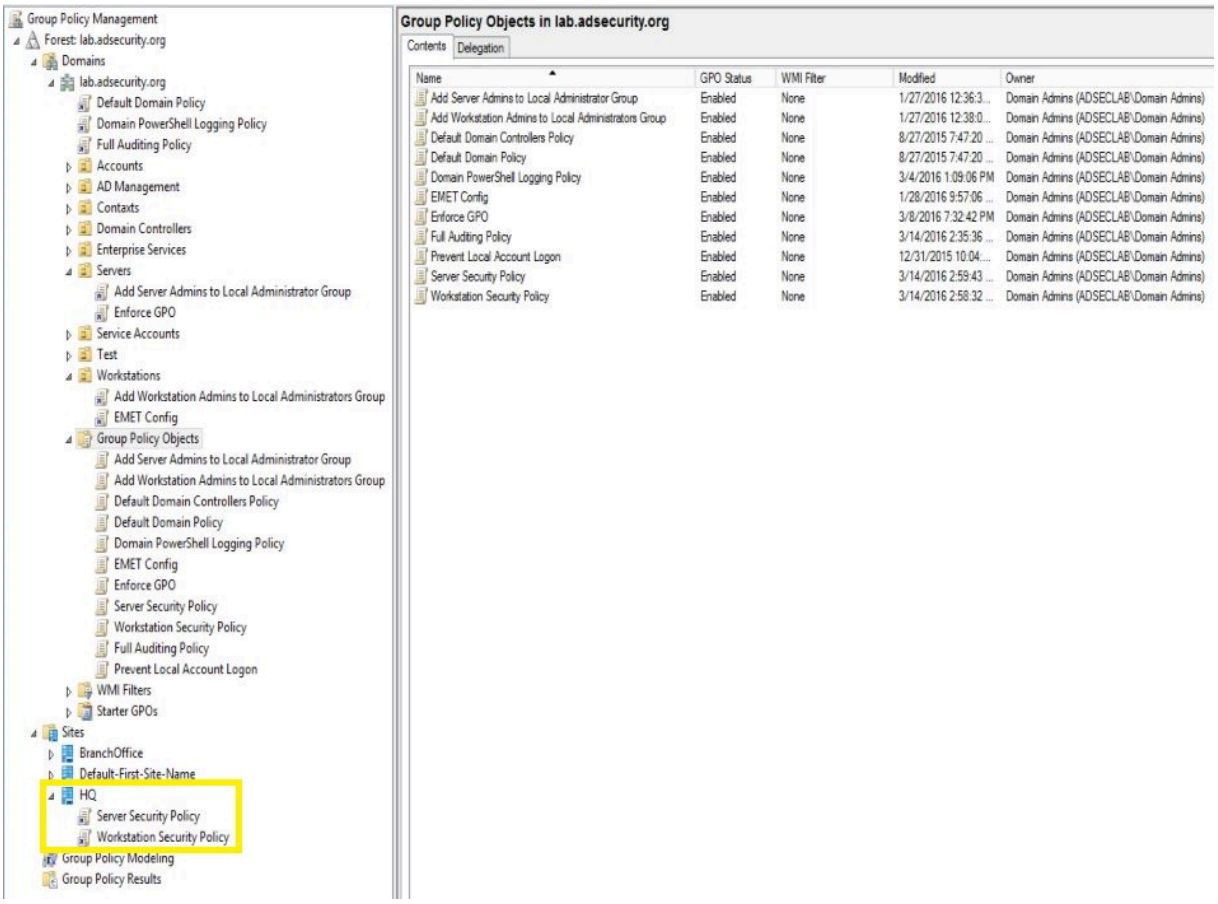


However, many admins don’t realize that while it is not best practice to link Group Policies to sites, there’s nothing preventing a Domain Admin (Enterprise Admin) from doing so.

All that’s needed is to select the AD sites that should be shown in the Group Policy Management Console and link the new/updated Group Policies to the site(s)



Looking at GPMC, we can see there are two Group Policies linked to the HQ Site and both of them were last modified recently (hint, hint).



Would your current monitoring system notify on this change?

Mitigation

All Group Policies in the AD environment should be configured for a single purpose and monitored for unauthorized modification, especially GPOs linked to the domain, Domain Controllers OU, and/or a top level OU such as workstations, servers, admins, etc.

Delegated permissions to Group Policy should be reviewed on a regular basis, especially those linked to top-level OUs. Only Active Directory administrators should have modify rights to GPOs applied to the domain, top-level OUs, and any GPOs linked to critical assets (Domain Controllers, servers, admin computers, etc).

Additionally, all sites should be reviewed for linked Group Policies since these GPOs can cross domain boundaries enabling privilege escalation across domains in the same AD forest.

SYSVOL permissions are critical and must remain the same as the default settings since SYSYOL contains the actual Group Policy settings in files that are applied by GPO clients. If a GPO configuration file has permissions that enables some on who is not an Active Directory admin to change the file, and thus change what action the GPO client actually performs, an attacker could quickly escalate permissions up to Domain Admin level.

Resources

- Group Policy Basics – Part 1: Understanding the Structure of a Group Policy Object

- Local Group Enumeration using PowerView (includes Group Policy features)
- Configure a Scheduled Task
- Group Policy Preferences
- Exploit existing Group Policy Preference credentials
- Complete list of Sneaky Active Directory Persistence Tricks posts

---