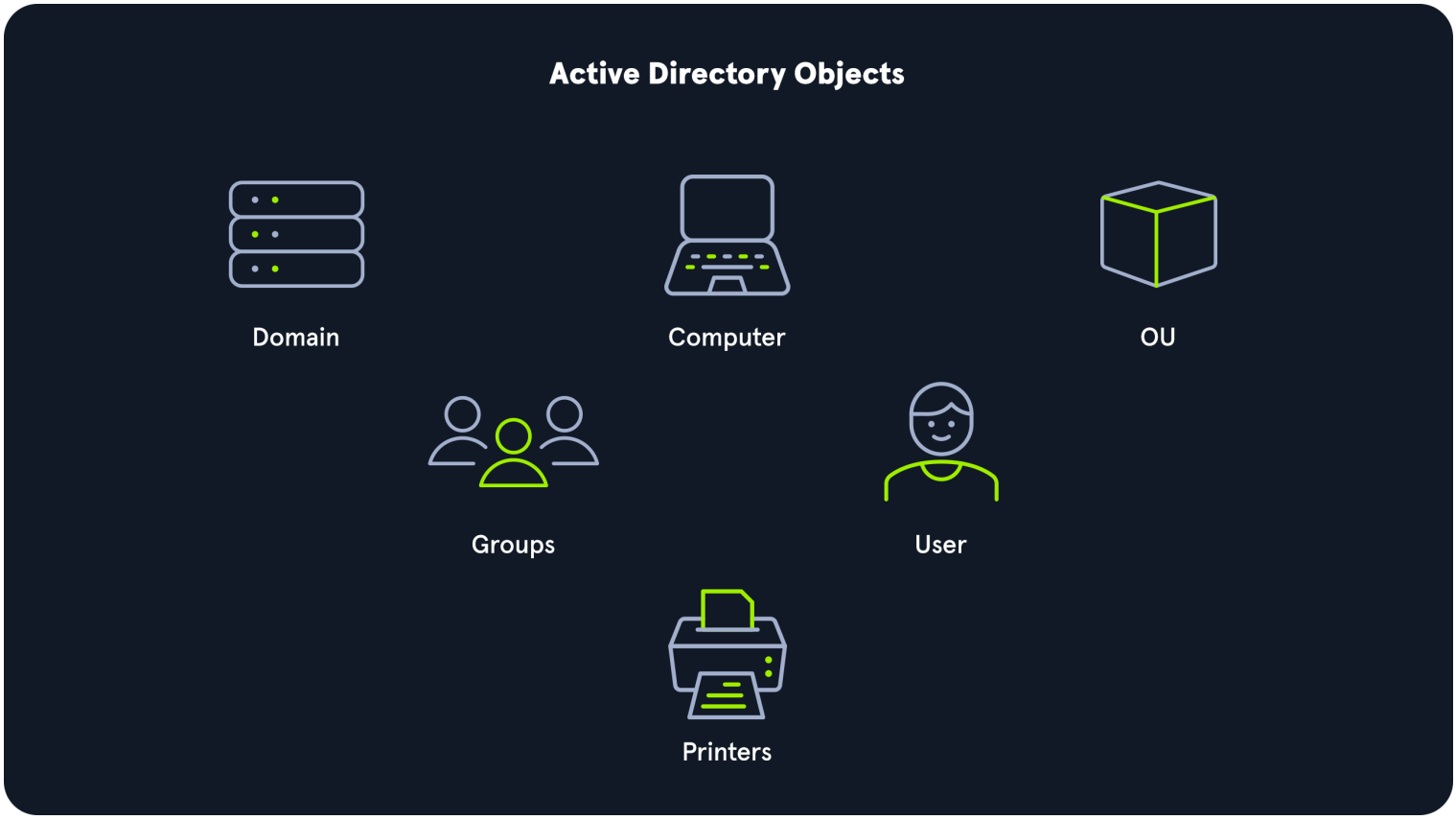
Introduction to Active Directory

Active Directory Objects

We will often see the term "objects" when referring to AD. What is an object? An object can be defined as ANY resource present within an Active Directory environment such as OUs, printers, users, domain controllers.

AD Objects



Users

These are the users within the organization's AD environment. Users are considered `leaf objects`, which means that they cannot contain any other objects within them. Another example of a leaf object is a mailbox in Microsoft Exchange. A user object is considered a security principal and has a security identifier (SID) and a global unique identifier (GUID). User objects have many possible attributes, such as their display name, last login time, date of last password change, email address, account description, manager, address, and more. Depending on how a particular Active Directory environment is set up, there can be over 800 possible user attributes when accounting for ALL possible attributes as detailed here. This example goes far beyond what is typically populated for a standard user in most environments but shows Active Directory's sheer size and complexity. They are a crucial target for attackers since gaining access to even a low privileged user can grant access to many objects and resources and allow for detailed enumeration of the entire domain (or forest).

Contacts

A contact object is usually used to represent an external user and contains informational attributes such as first name, last name, email address, telephone number, etc. They are `leaf objects` and are NOT security principals (securable objects), so they don't have a SID, only a GUID. An example would be a contact card for a third-party vendor or a customer.

Printers

A printer object points to a printer accessible within the AD network. Like a contact, a printer is a `leaf object` and not a security principal, so it only has a GUID. Printers have attributes such as the printer's name, driver information, port number, etc.

Computers

A computer object is any computer joined to the AD network (workstation or server). Computers are `leaf objects` because they do not contain other objects. However, they are considered security principals and have a SID and a GUID. Like users, they are prime targets for attackers since full administrative access to a computer (as the all-powerful `NT AUTHORITY\SYSTEM` account) grants similar rights to a standard domain user and can be used to perform the majority of the enumeration tasks that a user account can (save for a few exceptions across domain trusts.)

Shared Folders

A shared folder object points to a shared folder on the specific computer where the folder resides. Shared folders can have stringent access control applied to them and can be either accessible to everyone (even those without a valid AD account), open to only authenticated users (which means anyone with even the lowest privileged user account OR a computer account (`NT AUTHORITY\SYSTEM`) could access it), or be locked down to only allow certain users/groups access. Anyone not explicitly allowed access will be denied from listing or reading its contents. Shared folders are NOT security principals and only have a GUID. A shared folder's attributes can include the name, location on the system, security access rights.

Groups

A group is considered a `container object` because it can contain other objects, including users, computers, and even other groups. A group IS regarded as a security principal and has a SID and a GUID. In AD, groups are a way to manage user permissions and access to other securable objects (both users and computers). Let's say we want to give 20 help desk users access to the Remote Management Users group on a jump host. Instead of adding the users one by one, we could add the group, and the users would inherit the intended permissions via their membership in the group. In Active Directory, we commonly see what are called " nested groups" (a group added as a member of another group), which can lead to a user(s) obtaining unintended rights. Nested group membership is something we see and often leverage during penetration tests. The tool BloodHound helps to discover attack paths within a network and

illustrate them in a graphical interface. It is excellent for auditing group membership and uncovering/seeing the sometimes unintended impacts of nested group membership. Groups in AD can have many attributes, the most common being the name, description, membership, and other groups that the group belongs to. Many other attributes can be set, which we will discuss more in-depth later in this module.

Organizational Units (OUs)

An organizational unit, or OU from here on out, is a container that systems administrators can use to store similar objects for ease of administration. OUs are often used for administrative delegation of tasks without granting a user account full administrative rights. For example, we may have a top-level OU called Employees and then child OUs under it for the various departments such as Marketing, HR, Finance, Help Desk, etc. If an account were given the right to reset passwords over the top-level OU, this user would have the right to reset passwords for all users in the company. However, if the OU structure were such that specific departments were child OUs of the Help Desk OU, then any user placed in the Help Desk OU would have this right delegated to them if granted. Other tasks that may be delegated at the OU level include creating/deleting users, modifying group membership, managing Group Policy links, and performing password resets. OUs are very useful for managing Group Policy (which we will study later in this module) settings across a subset of users and groups within a domain. For example, we may want to set a specific password policy for privileged service accounts so these accounts could be placed in a particular OU and then have a Group Policy object assigned to it, which would enforce this password policy on all accounts placed inside of it. A few OU attributes include its name, members, security settings, and more.

Domain

A domain is the structure of an AD network. Domains contain objects such as users and computers, which are organized into container objects: groups and OUs. Every domain has its own separate database and sets of policies that can be applied to any and all objects within the domain. Some policies are set by default (and can be tweaked), such as the domain password policy. In contrast, others are created and applied based on the organization's need, such as blocking access to cmd.exe for all non-administrative users or mapping shared drives at log in.

Domain Controllers

Domain Controllers are essentially the brains of an AD network. They handle authentication requests, verify users on the network, and control who can access the various resources in the domain. All access requests are validated via the domain controller and privileged access requests are based on predetermined roles assigned to users. It also enforces security policies and stores information about every other object in the domain.

Sites

A site in AD is a set of computers across one or more subnets connected using high-speed links. They are used to make replication across domain controllers run efficiently.

Built-in

In AD, built-in is a container that holds default groups in an AD domain. They are predefined when an AD domain is created.

Foreign Security Principals

A foreign security principal (FSP) is an object created in AD to represent a security principal that belongs to a trusted external forest. They are created when an object such as a user, group, or computer from an external (outside of the current) forest is added to a group in the current domain. They are created automatically after adding a security principal to a group. Every foreign security principal is a placeholder object that holds the SID of the foreign object (an object that belongs to another forest.) Windows uses this SID to resolve the object's name via the trust relationship. FSPs are created in a specific container named ForeignSecurityPrincipals with a distinguished name like `cn=ForeignSecurityPrincipals,dc=inlanefreight,dc=local`.

Questions

Answer the question(s) below
to complete this Section and earn cubes!

Cheat Sheet

+ 0 True or False; Computers are considered leaf objects.

Submit

Hint

+ 1 <___> are objects that are used to store similar objects for ease of administration. (Fill in the blank)

Submit

Hint

+ 1 What AD object handles all authentication requests for a domain?

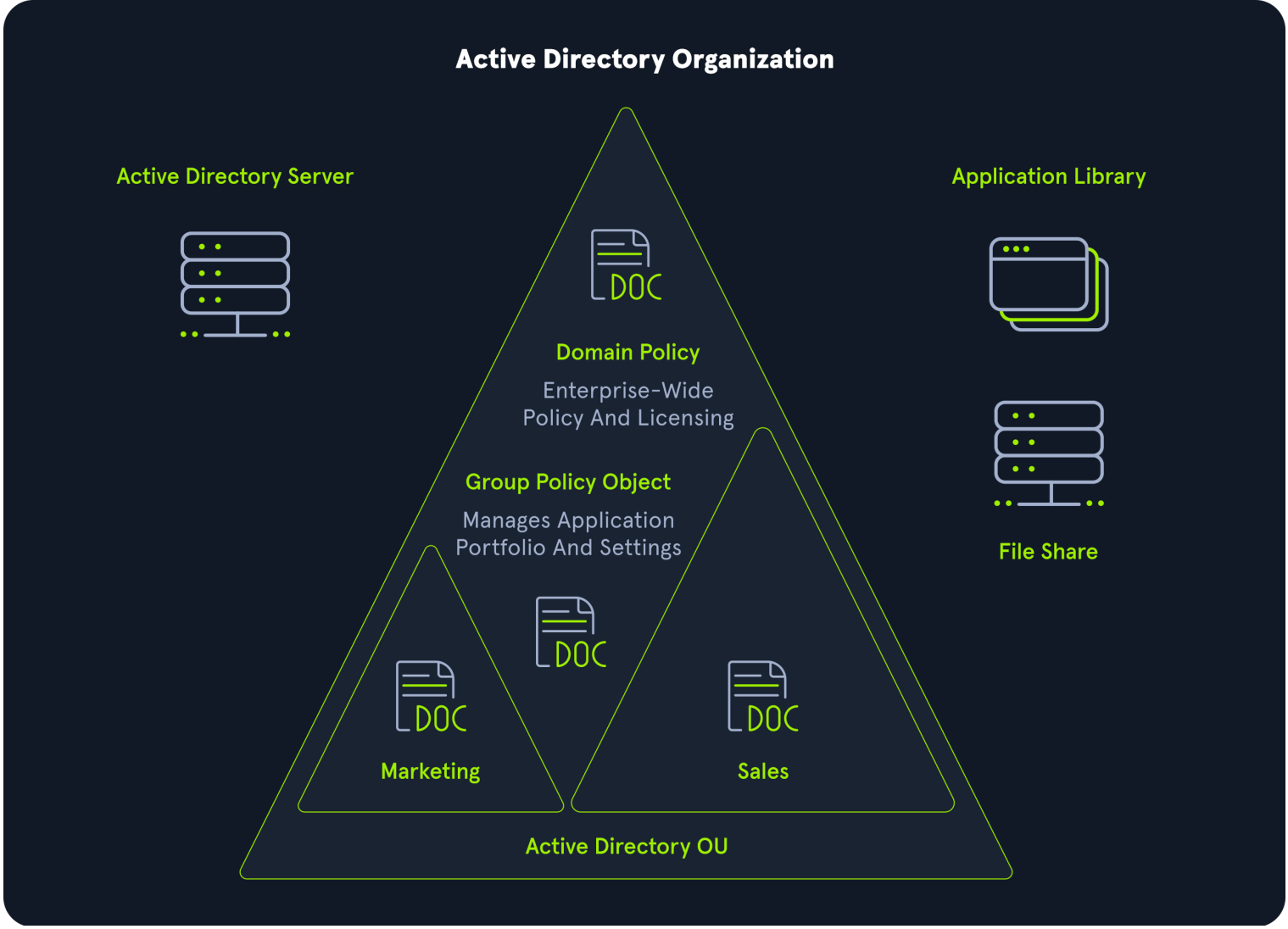
Submit

Hint

Why Active Directory?

Active Directory (AD) is a directory service for Windows network environments. It is a distributed, hierarchical structure that allows for centralized management of an organization's resources, including users, computers, groups, network devices, file shares, group policies, devices, and trusts. AD provides authentication and authorization functions within a Windows domain environment. It has come under increasing attack in recent years. It is designed to be backward-compatible, and many features are arguably not "secure by default," and it can be easily misconfigured. This weakness can be leveraged to move laterally and vertically within a network and gain unauthorized access. AD is essentially a sizeable read-only database accessible to all users within the domain, regardless of their privilege level. A basic AD user account with no added privileges can enumerate most objects within AD. This fact makes it extremely important to properly secure an AD implementation because ANY user account, regardless of their privilege level, can be used to enumerate the domain and hunt for misconfigurations and flaws thoroughly. Also, multiple attacks can be performed with only a standard domain user account, showing the importance of a defense-in-depth strategy and careful planning focusing on security and hardening AD, network segmentation, and least privilege. One example is the noPac attack that was first released in December of 2021.

Active Directory makes information easy to find and use for administrators and users. AD is highly scalable, supports millions of objects per domain, and allows the creation of additional domains as an organization grows.



Source

It is estimated that around 95% of Fortune 500 companies run Active Directory, making AD a key focus for attackers. A successful attack such as a phish that lands an attacker within the AD environment as a standard domain user would give them enough access to begin mapping out the domain and looking for avenues of attack. As security professionals, we will encounter AD environments of all sizes throughout our careers. It is essential to understand the structure and function of AD to become better informed as both an attacker and a defender.

Ransomware operators have been increasingly targeting Active Directory as a key part of their attack paths. The Conti Ransomware which has been used in more than 400 attacks around the world has been shown to leverage recent critical Active Directory flaws such as PrintNightmare (CVE-2021-34527) and Zerologon (CVE-2020-1472) to escalate privileges and move laterally in a target network. Understanding the structure and function of Active Directory is the first step in a career path to find and prevent these types of flaws before attackers do. Researchers are continually finding new, extremely high-risk attacks that affect Active Directory environments that often require no more than a standard domain user to obtain complete administrative control over the entire domain. There are many great open-source tools for penetration testers to enumerate and attack Active Directory. Still, to use these most effectively, we must understand how Active Directory works to identify obvious and nuanced flaws. Tools are only as effective as their operator is knowledgeable. So let's take the time to understand the structure and function of Active Directory before moving into later modules that will focus on in-depth manual and tool-based enumeration, attacks, lateral movement, post-exploitation, and persistence.

This module will lay the foundations for starting down the path of enumerating and attacking Active Directory. We will cover, in-depth, the structure and function of AD, discuss the various AD objects, discuss user rights and privileges, tools, and processes for managing AD, and even walk through examples of setting up a small AD environment.

History of Active Directory

LDAP, the foundation of Active Directory, was first introduced in RFCs as early as 1971. Active Directory was predated by the X.500 organizational unit concept, which was the earliest version of all directory systems created by Novell and Lotus and released in 1993 as Novell Directory Services.

Active Directory was first introduced in the mid-'90s but did not become part of the Windows operating system until the release of Windows Server 2000. Microsoft first attempted to provide directory services in 1990 with the release of Windows NT 3.0. This operating system combined features of the LAN Manager protocol and the OS/2 operating systems, which Microsoft created initially along with IBM lead by Ed Iacobucci who also led the design of IBM DOS and later co-founded Citrix Systems. The NT operating system evolved throughout the 90s, adapting protocols such as LDAP and Kerberos with Microsoft's proprietary elements. The first beta release of Active Directory was in 1997.

The release of Windows Server 2003 saw extended functionality and improved administration and added the `Forest` feature, which allows sysadmins to create "containers" of separate domains, users, computers, and other objects all under the same umbrella. Active Directory Federation Services (ADFS) was introduced in Server 2008 to provide Single Sign-On (SSO) to systems and applications for users on Windows Server operating systems. ADFS made it simpler and more streamlined for users to sign into applications and systems, not on their same LAN.

ADFS enables users to access applications across organizational boundaries using a single set of credentials. ADFS uses the claims-based Access Control Authorization model, which attempts to ensure security across applications by identifying users by a set of claims related to their identity, which are packaged into a security token by the identity provider.

The release of Server 2016 brought even more changes to Active Directory, such as the ability to migrate AD environments to the cloud and additional security enhancements such as user access monitoring and Group Managed Service Accounts (gMSA). gMSA offers a more secure way to run specific automated tasks, applications, and services and is often a recommended mitigation against the infamous Kerberoasting attack.

2016 saw a more significant push towards the cloud with the release of Azure AD Connect, which was designed as a single sign-on method for users being migrated to the Microsoft Office 365 environment.

Active Directory has suffered from various misconfigurations from 2000 to the present day. New vulnerabilities are discovered regularly that affect Active Directory and other technologies that interface with AD, such as Microsoft Exchange. As security researchers continue to uncover new flaws, organizations that run Active Directory need to remain on top of patching and

implementing fixes. As penetration testers, we are tasked with finding these flaws for our clients before attackers.

For this reason, we must have a solid foundation in Active Directory fundamentals and understand its structure, function, the various protocols that it uses to operate, how user rights and privileges are managed, how sysadmins administer AD and the multitude of vulnerabilities and misconfigurations that can be present in an AD environment. Managing AD is no easy task. One change/fix can introduce additional issues elsewhere. Before beginning to enumerate and then attack Active Directory, let's cover foundational concepts that will follow us throughout our infosec careers.

As said before, 95% of Fortune 500 companies run Active Directory, and Microsoft has a near-complete monopoly in the directory services space. Even though many companies are transitioning to cloud and hybrid environments, on-prem AD is not going away for many companies. If you are performing network penetration testing engagements, you can be nearly sure to encounter AD in some way on almost all of them.

This fundamental knowledge will make us better attackers and give us insight into AD that will be extremely useful when providing remediation advice to our clients. A deep understanding of AD will make peeling back the layers less daunting, and we will have the same confidence when approaching an environment with 10,000 hosts as we do with one with 20.

Active Directory Research Over the Years

Active Directory has been a major area of focus for security researchers for the past decade or so. Starting in 2014, we began to see many of the tools and much of the research emerge, leading to the discovery of common attacks and techniques that are still used today. Below is a timeline of events highlighting the discovery of some of the most impactful attacks and flaws by incredible researchers, as well as the release of some of the most widely used tools by penetration testers until this day. While many techniques have been discovered over the years, AD (and now Azure AD) presents a vast attack surface, and new attacks are still being discovered. New tools emerge that both penetration testers and defenders must have a firm grasp of to assist organizations with the difficult but critical task of securing AD environments.

As we can see from the timeline below, critical flaws are continuously being discovered. The noPac attack was discovered in December of 2021 and is the most recent critical AD attack that has been discovered at the time of writing (January of 2022). As we continue into 2022, we will surely see new tools and attacks released and new methods for exploiting and chaining together known vulnerabilities. We must stay on top of the latest and greatest Active Directory research to best help customers secure it or secure it ourselves.

This is by no means a comprehensive list of all the excellent research and tools that have been released over the years, but this is a snapshot of many of the most consequential ones seen over the past decade. The hard work of the researchers listed in this timeline (and many others) has led to remarkable discoveries and the creation of tools that can help both penetration testers and defenders dig deep into Active Directory environments. With them, it's easier to find both obvious and obscure, high-risk flaws before attackers do.

AD Attacks & Tools Timeline

2021

The PrintNightmare vulnerability was released. This was a remote code execution flaw in the Windows Print Spooler that could be used to take over hosts in an AD environment. The Shadow Credentials attack was released which allows for low privileged users to impersonate other user and computer accounts if conditions are right, and can be used to escalate privileges in a domain. The noPac attack was released in mid-December of 2021 when much of the security world was focused on the Log4j vulnerabilities. This attack allows an attacker to gain full control over a domain from a standard domain user account if the right conditions exist.

2020

The ZeroLogon attack debuted late in 2020. This was a critical flaw that allowed an attacker to impersonate any unpatched domain controller in a network.

2019

harmj0y delivered the talk "Kerberoasting Revisited" at DerbyCon which laid out new approaches to Kerberoasting. Elad Shamir released a blog post outlining techniques for abusing resource-based constrained delegation (RBCD) in Active Directory. The company BC Security released Empire 3.0 (now version 4) which was a re-release of the PowerShell Empire framework written in Python3 with many additions and changes.

2018

The "Printer Bug" bug was discovered by Lee Christensen and the SpoolSample PoC tool was released which leverages this bug to coerce Windows hosts to authenticate to other machines via the MS-RPRN RPC interface. harmj0y released the Rubeus toolkit for attacking Kerberos. Late in 2018 harmj0y also released the blog "Not A Security Boundary: Breaking Forest Trusts" which presented key research on performing attacks across forest trusts. The DCShadow attack technique was also released by Vincent LE TOUX and Benjamin Delpy at the Bluehat IL 2018 conference. The Ping Castle tool was released by Vincent LE TOUX for performing security audits of Active Directory by looking for misconfigurations and other flaws that can raise the risk level of a domain and producing a report that can be used to identify ways to further harden the environment.

2017

The ASREPRoast technique was introduced for attacking user accounts that don't require Kerberos preauthentication. _wald0 and harmj0y delivered the pivotal talk on Active Directory ACL attacks "ACE Up the Sleeve" at Black Hat and DEF CON. harmj0y released his "A Guide to Attacking Domain Trusts" blog post on enumerating and attacking domain trusts.

2016

BloodHound was released as a game changing tool for visualizing attack paths in AD at DEF CON 24.

2015

2015 saw the release of some of the most impactful Active Directory tools of all time. The PowerShell Empire framework was released. PowerView 2.0 released as part of the (now deprecated) PowerTools repository, which was a part of the PowerShellEmpire GitHub account. The DCSync attack was first released by Benjamin Delpy and Vincent Le Toux as part of the mimikatz tool. It has since been included in other tools. The first stable release of CrackMapExec ((v1.0.0) was introduced. Sean Metcalf gave a talk at Black Hat USA about the dangers of Kerberos Unconstrained Delegation and released an excellent blog post on the topic. The Impacket toolkit was also released in 2015. This is a collection of Python tools, many of which can be used to perform Active Directory attacks. It is still actively maintained as of January 2022 and is a key part of most every penetration tester's toolkit.

2014

Veil-PowerView first released. This project later became part of the PowerSploit framework as the (no longer supported) PowerView.ps1 AD recon tool. The Kerberoasting attack was first presented at a conference by Tim Medin at SANS Hackfest 2014.

2013

The Responder tool was released by Laurent Gaffie. Responder is a tool used for poisoning LLMNR, NBT-NS, and MDNS on an Active Directory network. It can be used to obtain password hashes and also perform SMB Relay attacks (when combined with other tools) to move laterally and vertically in an AD environment. It has evolved considerably over the years and is still actively supported (with new features added) as of January 2022.

Active Directory Structure

Active Directory (AD) is a directory service for Windows network environments. It is a distributed, hierarchical structure that allows for centralized management of an organization's resources, including users, computers, groups, network devices and file shares, group policies, servers and workstations, and trusts. AD provides authentication and authorization functions within a Windows domain environment. A directory service, such as Active Directory Domain Services (AD DS) gives an organization ways to store directory data and make it available to both standard users and administrators on the same network. AD DS stores information such as usernames and passwords and manages the rights needed for authorized users to access this information. It was first shipped with Windows Server 2000; it has come under increasing attack in recent years. It is designed to be backward-compatible, and many features are arguably not "secure by default." It is difficult to manage properly, especially in large environments where it can be easily misconfigured.

Active Directory flaws and misconfigurations can often be used to obtain a foothold (internal access), move laterally and vertically within a network, and gain unauthorized access to protected resources such as databases, file shares, source code, and more. AD is essentially a large database accessible to all users within the domain, regardless of their privilege level. A basic AD user account with no added privileges can be used to enumerate the majority of objects contained within AD, including but not limited to:

Domain Computers	Domain Users
Domain Group Information	Organizational Units (OUs)
Default Domain Policy	Functional Domain Levels
Password Policy	Group Policy Objects (GPOs)
Domain Trusts	Access Control Lists (ACLs)

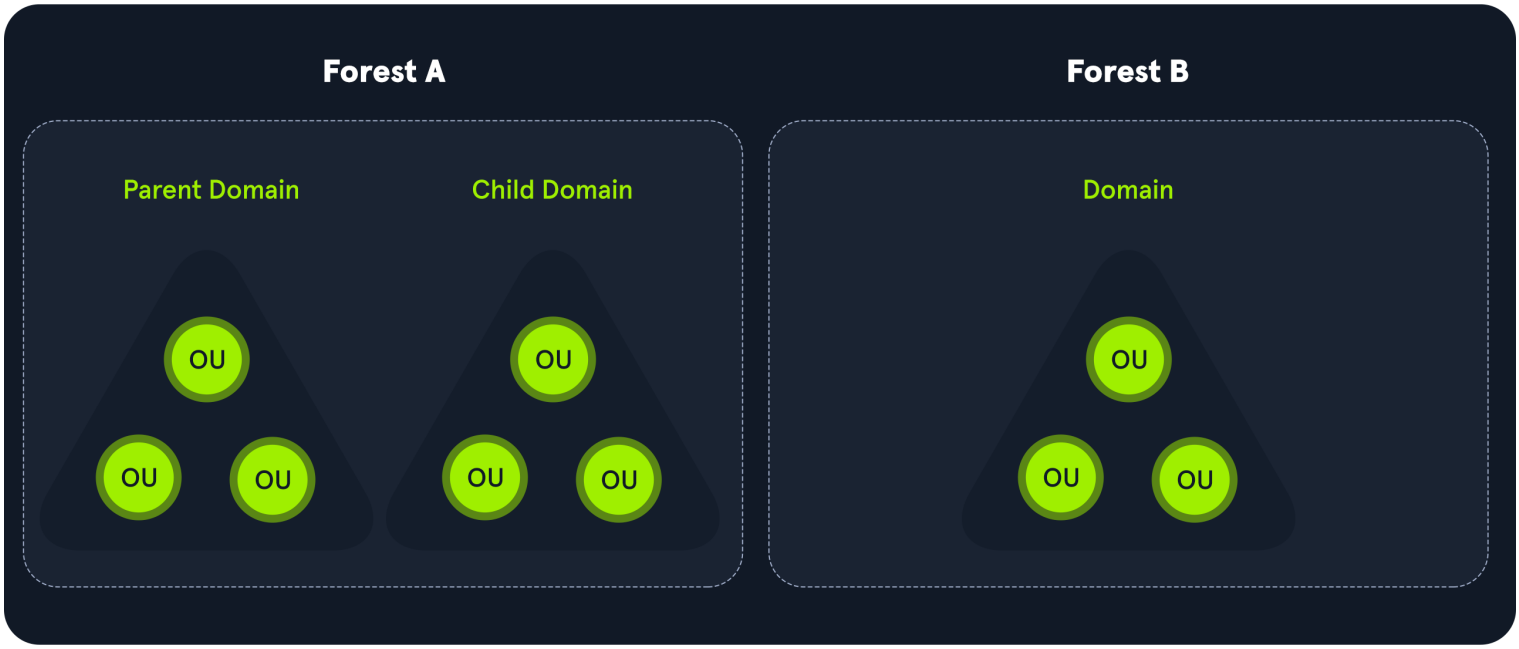
For this reason, we must understand how Active Directory is set up and the basics of administration before attempting to attack it. It's always easier to "break" things if we already know how to build them.

Active Directory is arranged in a hierarchical tree structure, with a forest at the top containing one or more domains, which can themselves have nested subdomains. A forest is the security boundary within which all objects are under administrative control. A forest may contain multiple domains, and a domain may include further child or sub-domains. A domain is a structure within which contained objects (users, computers, and groups) are accessible. It has many built-in Organizational Units (OUs), such as `Domain Controllers`, `Users`, `Computers`, and new OUs can be created as required. OUs may contain objects and sub-OUs, allowing for the assignment of different group policies.

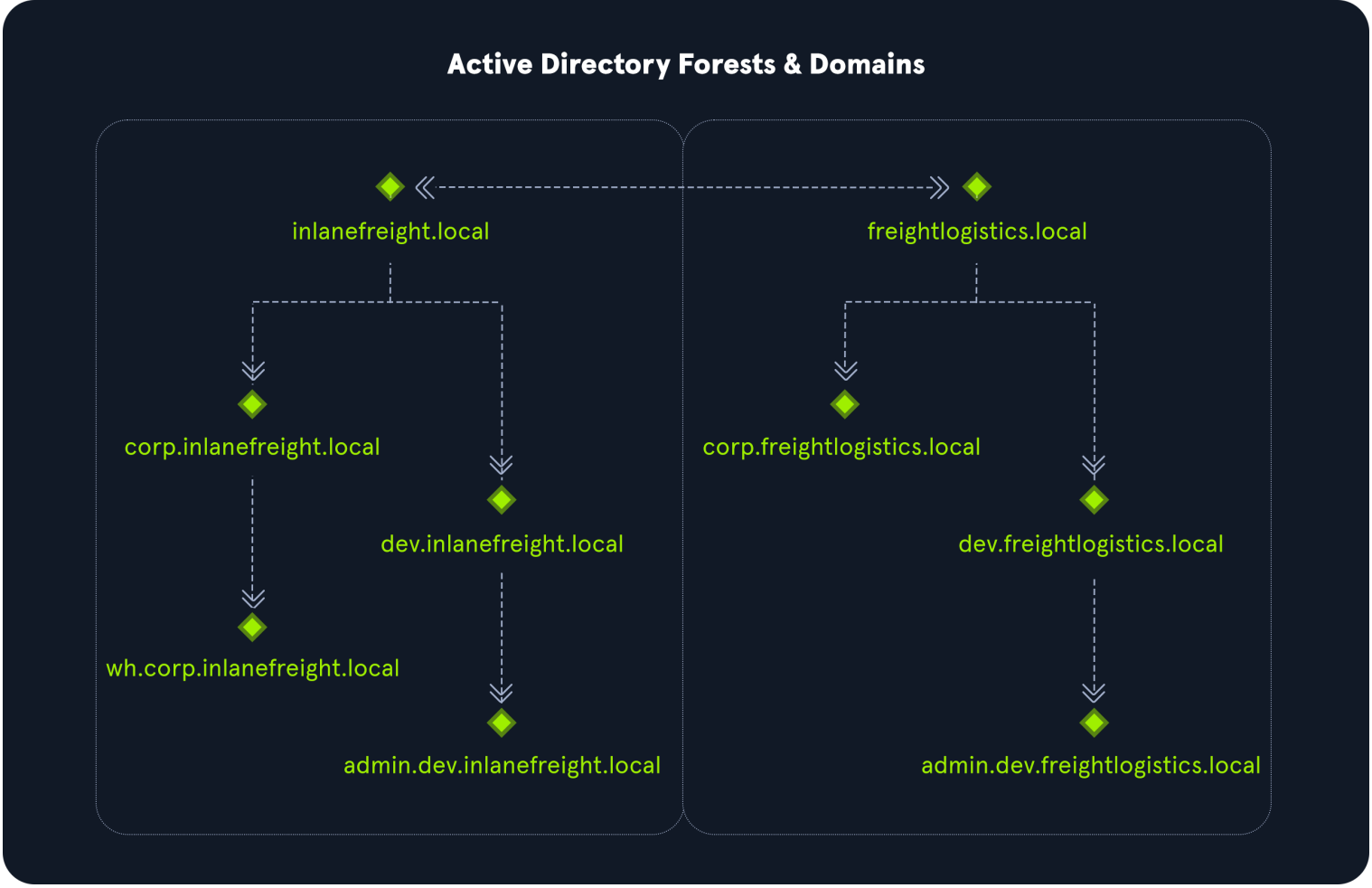
At a very (simplistic) high level, an AD structure may look as follows:

```
INLANEFREIGHT.LOCAL/
├── ADMIN.INLANEFREIGHT.LOCAL
│   ├── GPOs
│   └── OU
│       └── EMPLOYEES
│           ├── COMPUTERS
│           │   └── FILE01
│           ├── GROUPS
│           │   └── HQ Staff
│           └── USERS
│               └── barbara.jones
├── CORP.INLANEFREIGHT.LOCAL
└── DEV.INLANEFREIGHT.LOCAL
```

Here we could say that `INLANEFREIGHT.LOCAL` is the root domain and contains the subdomains (either child or tree root domains) `ADMIN.INLANEFREIGHT.LOCAL`, `CORP.INLANEFREIGHT.LOCAL`, and `DEV.INLANEFREIGHT.LOCAL` as well as the other objects that make up a domain such as users, groups, computers, and more as we will see in detail below. It is common to see multiple domains (or forests) linked together via trust relationships in organizations that perform a lot of acquisitions. It is often quicker and easier to create a trust relationship with another domain/forest than recreate all new users in the current domain. As we will see in later modules, domain trusts can introduce a slew of security issues if not appropriately administered.



The graphic below shows two forests, `INLANEFREIGHT.LOCAL` and `FREIGHTLOGISTICS.LOCAL`. The two-way arrow represents a bidirectional trust between the two forests, meaning that users in `INLANEFREIGHT.LOCAL` can access resources in `FREIGHTLOGISTICS.LOCAL` and vice versa. We can also see multiple child domains under each root domain. In this example, we can see that the root domain trusts each of the child domains, but the child domains in forest A do not necessarily have trusts established with the child domains in forest B. This means that a user that is part of `admin.dev.freightlogistics.local` would NOT be able to authenticate to machines in the `wh.corp.inlanefreight.local` domain by default even though a bidirectional trust exists between the top-level `inlanefreight.local` and `freightlogistics.local` domains. To allow direct communication from `admin.dev.freightlogistics.local` and `wh.corp.inlanefreight.local`, another trust would need to be set up.



Questions

Answer the question(s) below
to complete this Section and earn cubes!

Cheat Sheet

+ 1 What Active Directory structure can contain one or more domains?

Submit

Hint

+ 0 True or False; It can be common to see multiple domains linked together by trust relationships?

Submit

Hint

+ 1 Active Directory provides authentication and <____> within a Windows domain environment.

Submit

Hint

Active Directory Terminology

Before we go any further, let's take a step back and define some key terminology that will be used throughout this module and in general when dealing with Active Directory in any capacity.

Object

An object can be defined as ANY resource present within an Active Directory environment such as OUs, printers, users, domain controllers, etc.

Attributes

Every object in Active Directory has an associated set of attributes used to define characteristics of the given object. A computer object contains attributes such as the hostname and DNS name. All attributes in AD have an associated LDAP name that can be used when performing LDAP queries, such as `displayName` for `Full Name` and `given name` for `First Name`.

Schema

The Active Directory schema is essentially the blueprint of any enterprise environment. It defines what types of objects can exist in the AD database and their associated attributes. It lists definitions corresponding to AD objects and holds information about each object. For example, users in AD belong to the class "user," and computer objects to "computer," and so on. Each object has its own information (some required to be set and others optional) that are stored in Attributes. When an object is created from a class, this is called instantiation, and an object created from a specific class is called an instance of that class. For example, if we take the computer RDS01. This computer object is an instance of the "computer" class in Active Directory.

Domain

A domain is a logical group of objects such as computers, users, OUs, groups, etc. We can think of each domain as a different city within a state or country. Domains can operate entirely independently of one another or be connected via trust relationships.

Forest

A forest is a collection of Active Directory domains. It is the topmost container and contains all of the AD objects introduced below, including but not limited to domains, users, groups, computers, and Group Policy objects. A forest can contain one or multiple domains and be thought of as a state in the US or a country within the EU. Each forest operates independently but may have various trust relationships with other forests.

Tree

A tree is a collection of Active Directory domains that begins at a single root domain. A forest is a collection of AD trees. Each domain in a tree shares a boundary with the other domains. A parent-child trust relationship is formed when a domain is added under another domain in a tree. Two trees in the same forest cannot share a name (namespace). Let's say we have two trees in an AD forest: `inlanefreight.local` and `ilfreight.local`. A child domain of the first would be `corp.inlanefreight.local` while a child domain of the second could be `corp.ilfreight.local`. All domains in a tree share a standard Global Catalog which contains all information about objects that belong to the tree.

Container

Container objects hold other objects and have a defined place in the directory subtree hierarchy.

Leaf

Leaf objects do not contain other objects and are found at the end of the subtree hierarchy.

Global Unique Identifier (GUID)

A GUID is a unique 128-bit value assigned when a domain user or group is created. This GUID value is unique across the enterprise, similar to a MAC address. Every single object created by Active Directory is assigned a GUID, not only user and group objects. The GUID is stored in the `ObjectGUID` attribute. When querying for an AD object (such as a user, group, computer, domain, domain controller, etc.), we can query for its `objectGUID` value using PowerShell or search for it by specifying its distinguished name, GUID, SID, or SAM account name. GUIDs are used by AD to identify objects internally. Searching in Active Directory by GUID value is probably the most accurate and reliable way to find the exact object you are looking for, especially if the global catalog may contain similar matches for an object name. Specifying the `ObjectGUID` value when performing AD enumeration will ensure that we get the most accurate results pertaining to the object we are searching for information about. The `ObjectGUID` property `never` changes and is associated with the object for as long as that object exists in the domain.

Security principals

Security principals are anything that the operating system can authenticate, including users, computer accounts, or even threads/processes that run in the context of a user or computer account (i.e., an application such as Tomcat running in the context of a service account within the domain). In AD, security principles are domain objects that can manage access to other resources within the domain. We can also have local user accounts and security groups used to control access to resources on only that specific computer. These are not managed by AD but rather by the Security Accounts Manager (SAM).

Security Identifier (SID)

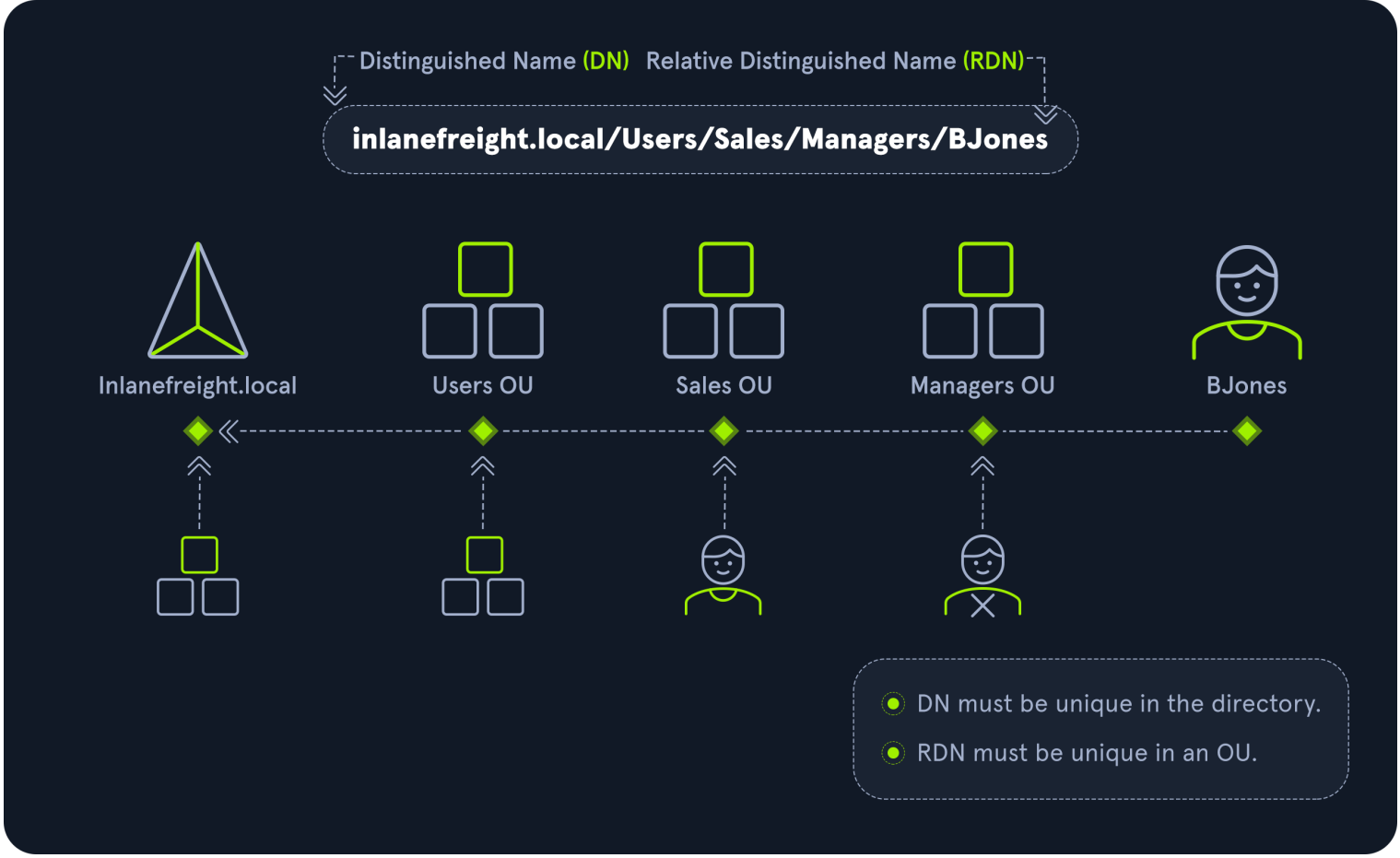
A security identifier, or SID is used as a unique identifier for a security principal or security group. Every account, group, or process has its own unique SID, which, in an AD environment, is issued by the domain controller and stored in a secure database. A SID can only be used once. Even if the security principle is deleted, it can never be used again in that environment to identify another user or group. When a user logs in, the system creates an access token for them which contains the user's SID, the rights they have been granted, and the SIDs for any groups that the user is a member of. This token is used to check rights whenever the user performs an action on the computer. There are also well-known SIDs that are used to identify generic users and groups. These are the same across all operating systems. An example is the `Everyone` group.

Distinguished Name (DN)

A Distinguished Name (DN) describes the full path to an object in AD (such as `cn=bjones, ou=IT, ou=Employees, dc=inlanefreight, dc=local`). In this example, the user `bjones` works in the IT department of the company Inlanefreight, and his account is created in an Organizational Unit (OU) that holds accounts for company employees. The Common Name (CN) `bjones` is just one way the user object could be searched for or accessed within the domain.

Relative Distinguished Name (RDN)

A Relative Distinguished Name (RDN) is a single component of the Distinguished Name that identifies the object as unique from other objects at the current level in the naming hierarchy. In our example, `bjones` is the Relative Distinguished Name of the object. AD does not allow two objects with the same name under the same parent container, but there can be two objects with the same RDNs that are still unique in the domain because they have different DNs. For example, the object `cn=bjones,dc=dev,dc=inlanefreight,dc=local` would be recognized as different from `cn=bjones,dc=inlanefreight,dc=local`.



sAMAccountName

The sAMAccountName is the user's logon name. Here it would just be `bjones`. It must be a unique value and 20 or fewer characters.

userPrincipalName

The userPrincipalName attribute is another way to identify users in AD. This attribute consists of a prefix (the user account name) and a suffix (the domain name) in the format of `[email protected]`. This attribute is not mandatory.

FSMO Roles

In the early days of AD, if you had multiple DCs in an environment, they would fight over which DC gets to make changes, and sometimes changes would not be made properly. Microsoft then implemented "last writer wins," which could introduce its own problems if the last change breaks things. They then introduced a model in which a single "master" DC could apply changes to the domain while the others merely fulfilled authentication requests. This was a flawed design because if the master DC went down, no changes could be made to the environment until it was restored. To resolve this single point of failure model, Microsoft separated the various responsibilities that a DC can have into Flexible Single Master Operation (FSMO) roles. These give Domain Controllers (DC) the ability to continue authenticating users and granting permissions without interruption (authorization and authentication). There are five FMSO roles: `Schema Master` and `Domain Naming Master` (one of each per forest), `Relative ID (RID) Master` (one per domain), `Primary Domain Controller (PDC) Emulator` (one per domain), and `Infrastructure Master` (one per domain). All five roles are assigned to the first DC in the forest root domain in a new AD forest. Each time a new domain is added to a forest, only the RID Master, PDC Emulator, and Infrastructure Master roles are assigned to the new domain. FSMO roles are typically set when domain controllers are created, but sysadmins can transfer these roles if needed. These roles help replication in AD to run smoothly and ensure that critical services are operating correctly. We will walk through each of these roles in detail later in this section.

Global Catalog

A global catalog (GC) is a domain controller that stores copies of ALL objects in an Active Directory forest. The GC stores a full copy of all objects in the current domain and a partial copy of objects that belong to other domains in the forest. Standard domain controllers hold a complete replica of objects belonging to its domain but not those of different domains in the forest. The GC allows both users and applications to find information about any objects in ANY domain in the forest. GC is a feature that is enabled on a domain controller and performs the following functions:

- Authentication (provided authorization for all groups that a user account belongs to, which is included when an access token is generated)
- Object search (making the directory structure within a forest transparent, allowing a search to be carried out across all domains in a forest by providing just one attribute about an object.)

Read-Only Domain Controller (RODC)

A Read-Only Domain Controller (RODC) has a read-only Active Directory database. No AD account passwords are cached on an RODC (other than the RODC computer account & RODC KRBTGT passwords.) No changes are pushed out via an RODC's AD database, SYSVOL, or DNS. RODCs also include a read-only DNS server, allow for administrator role separation, reduce replication traffic in the environment, and prevent SYSVOL modifications from being replicated to other DCs.

Replication

Replication happens in AD when AD objects are updated and transferred from one Domain Controller to another. Whenever a DC is added, connection objects are created to manage replication between them. These connections are made by the Knowledge Consistency Checker (KCC) service, which is present on all DCs. Replication ensures that changes are synchronized with all other DCs in a forest, helping to create a backup in case one domain controller fails.

Service Principal Name (SPN)

A Service Principal Name (SPN) uniquely identifies a service instance. They are used by Kerberos authentication to associate an instance of a service with a logon account, allowing a client application to request the service to authenticate an account without needing to know the account name.

Group Policy Object (GPO)

Group Policy Objects (GPOs) are virtual collections of policy settings. Each GPO has a unique GUID. A GPO can contain local file system settings or Active Directory settings. GPO settings can be applied to both user and computer objects. They can be applied to all users and computers within the domain or defined more granularly at the OU level.

Access Control List (ACL)

An Access Control List (ACL) is the ordered collection of Access Control Entries (ACEs) that apply to an object.

Access Control Entries (ACEs)

Each Access Control Entry (ACE) in an ACL identifies a trustee (user account, group account, or logon session) and lists the access rights that are allowed, denied, or audited for the given trustee.

Discretionary Access Control List (DACL)

DACLs define which security principles are granted or denied access to an object; it contains a list of ACEs. When a process tries to access a securable object, the system checks the ACEs in the object's DACL to determine whether or not to grant access. If an object does NOT have a DACL, then the system will grant full access to everyone, but if the DACL has no ACE entries, the system will deny all access attempts. ACEs in the DACL are checked in sequence until a match is found that allows the requested rights or until access is denied.

System Access Control Lists (SACL)

Allows for administrators to log access attempts that are made to secured objects. ACEs specify the types of access attempts that cause the system to generate a record in the security event log.

Fully Qualified Domain Name (FQDN)

An FQDN is the complete name for a specific computer or host. It is written with the hostname and domain name in the format [host name].[domain name].[tld]. This is used to specify an object's location in the tree hierarchy of DNS. The FQDN can be used to locate hosts in an Active Directory without knowing the IP address, much like when browsing to a website such as google.com instead of typing in the associated IP address. An example would be the host `DC01` in the domain `INLANEFREIGHT.LOCAL`. The FQDN here would be `DC01.INLANEFREIGHT.LOCAL`.

Tombstone

A tombstone is a container object in AD that holds deleted AD objects. When an object is deleted from AD, the object remains for a set period of time known as the `Tombstone Lifetime,` and the `isDeleted` attribute is set to `TRUE`. Once an object exceeds the `Tombstone Lifetime`, it will be entirely removed. Microsoft recommends a tombstone lifetime of 180 days to increase the usefulness of backups, but this value may differ across environments. Depending on the DC operating system version, this value will default to 60 or 180 days. If an object is deleted in a domain that does not have an AD Recycle Bin, it will become a tombstone object. When this happens, the object is stripped of most of its attributes and placed in the `Deleted Objects` container for the duration of the `tombstoneLifetime`. It can be recovered, but any attributes that were lost can no longer be recovered.

AD Recycle Bin

The AD Recycle Bin was first introduced in Windows Server 2008 R2 to facilitate the recovery of deleted AD objects. This made it easier for sysadmins to restore objects, avoiding the need to restore from backups, restarting Active Directory Domain Services (AD DS), or rebooting a Domain Controller. When the AD Recycle Bin is enabled, any deleted objects are preserved for a period of time, facilitating restoration if needed. Sysadmins can set how long an object remains in a deleted, recoverable state. If this is not specified, the object will be restorable for a default value of 60 days. The biggest advantage of using the AD Recycle Bin is that most of a deleted object's attributes are preserved, which makes it far easier to fully restore a deleted object to its previous state.

SYSVOL

The SYSVOL folder, or share, stores copies of public files in the domain such as system policies, Group Policy settings, logon/logoff scripts, and often contains other types of scripts that are executed to perform various tasks in the AD environment. The contents of the SYSVOL folder are replicated to all DCs within the environment using File Replication Services (FRS). You can read more about the SYSVOL structure here.

AdminSDHolder

The AdminSDHolder object is used to manage ACLs for members of built-in groups in AD marked as privileged. It acts as a container that holds the Security Descriptor applied to members of protected groups. The SDProp (SD Propagator) process runs on a schedule on the PDC Emulator Domain Controller. When this process runs, it checks members of protected groups to ensure that the correct ACL is applied to them. It runs every hour by default. For example, suppose an attacker is able to create a malicious ACL entry to grant a user certain rights over a member of the Domain Admins group. In that case, unless they modify other settings in AD, these rights will be removed (and they will lose any persistence they were hoping to achieve) when the SDProp process runs on the set interval.

dsHeuristics

The dsHeuristics attribute is a string value set on the Directory Service object used to define multiple forest-wide configuration settings. One of these settings is to exclude built-in groups from the Protected Groups list. Groups in this list are protected from modification via the `AdminSDHolder` object. If a group is excluded via the `dsHeuristics` attribute, then any changes that affect it will not be reverted when the SDProp process runs.

adminCount

The adminCount attribute determines whether or not the SDProp process protects a user. If the value is set to `0` or not specified, the user is not protected. If the attribute value is set to `value`, the user is protected. Attackers will often look for accounts with the `adminCount` attribute set to `1` to target in an internal environment. These are often privileged accounts and may lead to further access or full domain compromise.

Active Directory Users and Computers (ADUC)

ADUC is a GUI console commonly used for managing users, groups, computers, and contacts in AD. Changes made in ADUC can be done via PowerShell as well.

ADSI Edit

ADSI Edit is a GUI tool used to manage objects in AD. It provides access to far more than is available in ADUC and can be used to set or delete any attribute available on an object, add, remove, and move objects as well. It is a powerful tool that allows a user to access AD at a much deeper level. Great care should be taken when using this tool, as changes here could cause major problems in AD.

sIDHistory

This attribute holds any SIDs that an object was assigned previously. It is usually used in migrations so a user can maintain the same level of access when migrated from one domain to another. This attribute can potentially be abused if set insecurely, allowing an attacker to gain prior elevated access that an account had before a migration if SID Filtering (or removing SIDs from another domain from a user's access token that could be used for elevated access) is not enabled.

NTDS.DIT

The NTDS.DIT file can be considered the heart of Active Directory. It is stored on a Domain Controller at `C:\Windows\NTDS\` and is a database that stores AD data such as information about user and group objects, group membership, and, most important to attackers and penetration testers, the password hashes for all users in the domain. Once full domain compromise is reached, an attacker can retrieve this file, extract the hashes, and either use them to perform a pass-the-hash attack or crack them offline using a tool such as Hashcat to access additional resources in the domain. If the setting Store password with reversible encryption is enabled, then the NTDS.DIT will also store the cleartext passwords for all users created or who changed their password

after this policy was set. While rare, some organizations may enable this setting if they use applications or protocols that need to use a user's existing password (and not Kerberos) for authentication.

MSBROWSE

MSBROWSE is a Microsoft networking protocol that was used in early versions of Windows-based local area networks (LANs) to provide browsing services. It was used to maintain a list of resources, such as shared printers and files, that were available on the network, and to allow users to easily browse and access these resources.

In older version of Windows we could use `nbtstat -A ip-address` to search for the Master Browser. If we see MSBROWSE it means that's the Master Browser. Aditionally we could use `nltest` utility to query a Windows Master Browser for the names of the Domain Controllers.

Today, MSBROWSE is largely obsolete and is no longer in widespread use. Modern Windows-based LANs use the Server Message Block (SMB) protocol for file and printer sharing, and the Common Internet File System (CIFS) protocol for browsing services.

Questions

Answer the question(s) below
to complete this Section and earn cubes!

Cheat Sheet

+ 1 What is known as the "Blueprint" of an Active Directory environment?

Submit

Hint

+ 0 What uniquely identifies a Service instance? (full name, space-separated, not abbreviated)

Submit

Hint

+ 0 True or False; Group Policy objects can be applied to user and computer objects.

Submit

Hint

+ 0 What container in AD holds deleted objects?

Submit

Hint

+ 1 What file contains the hashes of passwords for all users in a domain?

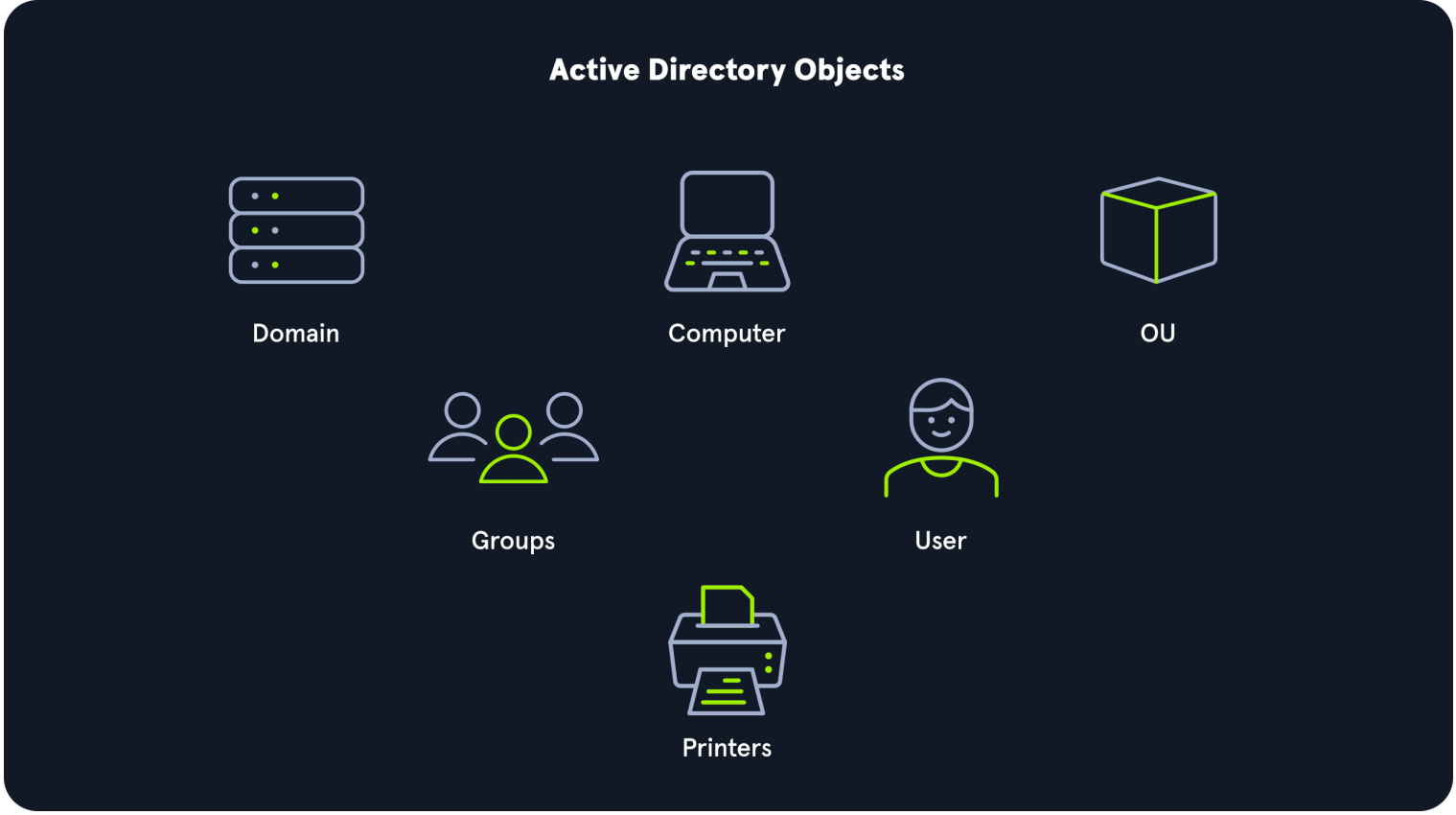
Submit

Hint

Active Directory Objects

We will often see the term "objects" when referring to AD. What is an object? An object can be defined as ANY resource present within an Active Directory environment such as OUs, printers, users, domain controllers.

AD Objects



Users

These are the users within the organization's AD environment. Users are considered `leaf objects`, which means that they cannot contain any other objects within them. Another example of a leaf object is a mailbox in Microsoft Exchange. A user object is considered a security principal and has a security identifier (SID) and a global unique identifier (GUID). User objects have many possible attributes, such as their display name, last login time, date of last password change, email address, account description, manager, address, and more. Depending on how a particular Active Directory environment is set up, there can be over 800 possible user attributes when accounting for ALL possible attributes as detailed here. This example goes far beyond what is typically populated for a standard user in most environments but shows Active Directory's sheer size and complexity. They are a crucial target for attackers since gaining access to even a low privileged user can grant access to many objects and resources and allow for detailed enumeration of the entire domain (or forest).

Contacts

A contact object is usually used to represent an external user and contains informational attributes such as first name, last name, email address, telephone number, etc. They are `leaf objects` and are NOT security principals (securable objects), so they don't have a SID, only a GUID. An example would be a contact card for a third-party vendor or a customer.

Printers

A printer object points to a printer accessible within the AD network. Like a contact, a printer is a `leaf object` and not a security principal, so it only has a GUID. Printers have attributes such as the printer's name, driver information, port number, etc.

Computers

A computer object is any computer joined to the AD network (workstation or server). Computers are `leaf objects` because they do not contain other objects. However, they are considered security principals and have a SID and a GUID. Like users, they are prime targets for attackers since full administrative access to a computer (as the all-powerful `NT AUTHORITY\SYSTEM` account) grants similar rights to a standard domain user and can be used to perform the majority of the enumeration tasks that a user account can (save for a few exceptions across domain trusts.)

Shared Folders

A shared folder object points to a shared folder on the specific computer where the folder resides. Shared folders can have stringent access control applied to them and can be either accessible to everyone (even those without a valid AD account), open to only authenticated users (which means anyone with even the lowest privileged user account OR a computer account (`NT AUTHORITY\SYSTEM`) could access it), or be locked down to only allow certain users/groups access. Anyone not explicitly allowed access will be denied from listing or reading its contents. Shared folders are NOT security principals and only have a GUID. A shared folder's attributes can include the name, location on the system, security access rights.

Groups

A group is considered a `container object` because it can contain other objects, including users, computers, and even other groups. A group IS regarded as a security principal and has a SID and a GUID. In AD, groups are a way to manage user permissions and access to other securable objects (both users and computers). Let's say we want to give 20 help desk users access to the Remote Management Users group on a jump host. Instead of adding the users one by one, we could add the group, and the users would inherit the intended permissions via their membership in the group. In Active Directory, we commonly see what are called " nested groups" (a group added as a member of another group), which can lead to a user(s) obtaining unintended rights. Nested group membership is something we see and often leverage during penetration tests. The tool BloodHound helps to discover attack paths within a network and illustrate them in a graphical interface. It is excellent for auditing group membership and uncovering/seeing the sometimes unintended impacts of nested group membership. Groups in AD can have many attributes, the most common being the name, description, membership, and other groups that the group belongs to. Many other attributes can be set, which we will discuss more in-depth later in this module.

Organizational Units (OUs)

An organizational unit, or OU from here on out, is a container that systems administrators can use to store similar objects for ease of administration. OUs are often used for administrative delegation of tasks without granting a user account full administrative rights. For example, we may have a top-level OU called Employees and then child OUs under it for the various departments such as Marketing, HR, Finance, Help Desk, etc. If an account were given the right to reset passwords over the top-level OU, this user would have the right to reset passwords for all users in the company. However, if the OU structure were such that specific departments were child OUs of the Help Desk OU, then any user placed in the Help Desk OU would have this right delegated to them if granted. Other tasks that may be delegated at the OU level include creating/deleting users, modifying group membership, managing Group Policy links, and performing password resets. OUs are very useful for managing Group Policy (which we will study later in this module) settings across a subset of users and groups within a domain. For example, we may want to set a specific password policy for privileged service accounts so these accounts could be placed in a particular OU and then have a Group Policy object assigned to it, which would enforce this password policy on all accounts placed inside of it. A few OU attributes include its name, members, security settings, and more.

Domain

A domain is the structure of an AD network. Domains contain objects such as users and computers, which are organized into container objects: groups and OUs. Every domain has its own separate database and sets of policies that can be applied to any and all objects within the domain. Some policies are set by default (and can be tweaked), such as the domain password policy. In contrast, others are created and applied based on the organization's need, such as blocking access to cmd.exe for all non-administrative users or mapping shared drives at log in.

Domain Controllers

Domain Controllers are essentially the brains of an AD network. They handle authentication requests, verify users on the network, and control who can access the various resources in the domain. All access requests are validated via the domain controller and privileged access requests are based on predetermined roles assigned to users. It also enforces security policies and stores information about every other object in the domain.

Sites

A site in AD is a set of computers across one or more subnets connected using high-speed links. They are used to make replication across domain controllers run efficiently.

Built-in

In AD, built-in is a container that holds default groups in an AD domain. They are predefined when an AD domain is created.

Foreign Security Principals

A foreign security principal (FSP) is an object created in AD to represent a security principal that belongs to a trusted external forest. They are created when an object such as a user, group, or computer from an external (outside of the current) forest is added to a group in the current domain. They are created automatically after adding a security principal to a group. Every foreign security principal is a placeholder object that holds the SID of the foreign object (an object that belongs to another forest.) Windows uses this SID to resolve the object's name via the trust relationship. FSPs are created in a specific container named ForeignSecurityPrincipals with a distinguished name like `cn=ForeignSecurityPrincipals,dc=inlanefreight,dc=local`.

Questions

Answer the question(s) below
to complete this Section and earn cubes!

Cheat Sheet

+ 0 True or False; Computers are considered leaf objects.

Submit

Hint

+ 1 <___> are objects that are used to store similar objects for ease of administration. (Fill in the blank)

Submit

Hint

+ 1 What AD object handles all authentication requests for a domain?

Submit

Hint

Active Directory Functionality

As mentioned before, there are five Flexible Single Master Operation (FSMO) roles. These roles can be defined as follows:

Roles	Description
`Schema Master`	This role manages the read/write copy of the AD schema, which defines all attributes that can apply to an object in AD.
`Domain Naming Master`	Manages domain names and ensures that two domains of the same name are not created in the same forest.
`Relative ID (RID) Master`	The RID Master assigns blocks of RIDs to other DCs within the domain that can be used for new objects. The RID Master helps ensure that multiple objects are not assigned the same SID. Domain object SIDs are the domain SID combined with the RID number assigned to the object to make the unique SID.
`PDC Emulator`	The host with this role would be the authoritative DC in the domain and respond to authentication requests, password changes, and manage Group Policy Objects (GPOs). The PDC Emulator also maintains time within the domain.
`Infrastructure Master`	This role translates GUIDs, SIDs, and DNs between domains. This role is used in organizations with multiple domains in a single forest. The Infrastructure Master helps them to communicate. If this role is not functioning properly, Access Control Lists (ACLs) will show SIDs instead of fully resolved names.

Depending on the organization, these roles may be assigned to specific DCs or as defaults each time a new DC is added. Issues with FSMO roles will lead to authentication and authorization difficulties within a domain.

Domain and Forest Functional Levels

Microsoft introduced functional levels to determine the various features and capabilities available in Active Directory Domain Services (AD DS) at the domain and forest level. They are also used to specify which Windows Server operating systems can run a Domain Controller in a domain or forest. This and this article describe both the domain and forest functional levels from Windows 2000 native to Windows Server 2012 R2. Below is a quick overview of the differences in `domain functional levels` from Windows 2000 native up to Windows Server 2016, aside from all default Active Directory Directory Services features from the level just below it (or just the default AD DS features in the case of Windows 2000 native.)

Domain Functional Level	Features Available	Supported Domain Controller Operating Systems
Windows 2000 native	Universal groups for distribution and security groups, group nesting, group conversion (between security and distribution and security groups), SID history.	Windows Server 2008 R2, Windows Server 2008, Windows Server 2003, Windows 2000
Windows Server 2003	Netdom.exe domain management tool, lastLogonTimestamp attribute introduced, well-known users and computers containers, constrained delegation, selective authentication.	Windows Server 2012 R2, Windows Server 2012, Windows Server 2008 R2, Windows Server 2008, Windows Server 2003

Domain Functional Level	Features Available	Supported Domain Controller Operating Systems
Windows Server 2008	Distributed File System (DFS) replication support, Advanced Encryption Standard (AES 128 and AES 256) support for the Kerberos protocol, Fine-grained password policies	Windows Server 2012 R2, Windows Server 2012, Windows Server 2008 R2, Windows Server 2008
Windows Server 2008 R2	Authentication mechanism assurance, Managed Service Accounts	Windows Server 2012 R2, Windows Server 2012, Windows Server 2008 R2
Windows Server 2012	KDC support for claims, compound authentication, and Kerberos armoring	Windows Server 2012 R2, Windows Server 2012
Windows Server 2012 R2	Extra protections for members of the Protected Users group, Authentication Policies, Authentication Policy Silos	Windows Server 2012 R2
Windows Server 2016	Smart card required for interactive logon new Kerberos features and new credential protection features	Windows Server 2019 and Windows Server 2016

A new functional level was not added with the release of Windows Server 2019. However, Windows Server 2008 functional level is the minimum requirement for adding Server 2019 Domain Controllers to an environment. Also, the target domain has to use DFS-R for SYSVOL replication.

Forest functional levels have introduced a few key capabilities over the years:

Version	Capabilities
`Windows Server 2003`	saw the introduction of the forest trust, domain renaming, read-only domain controllers (RODC), and more.
`Windows Server 2008`	All new domains added to the forest default to the Server 2008 domain functional level. No additional new features.
`Windows Server 2008 R2`	Active Directory Recycle Bin provides the ability to restore deleted objects when AD DS is running.
`Windows Server 2012`	All new domains added to the forest default to the Server 2012 domain functional level. No additional new features.
`Windows Server 2012 R2`	All new domains added to the forest default to the Server 2012 R2 domain functional level. No additional new features.
`Windows Server 2016`	Privileged access management (PAM) using Microsoft Identity Manager (MIM).

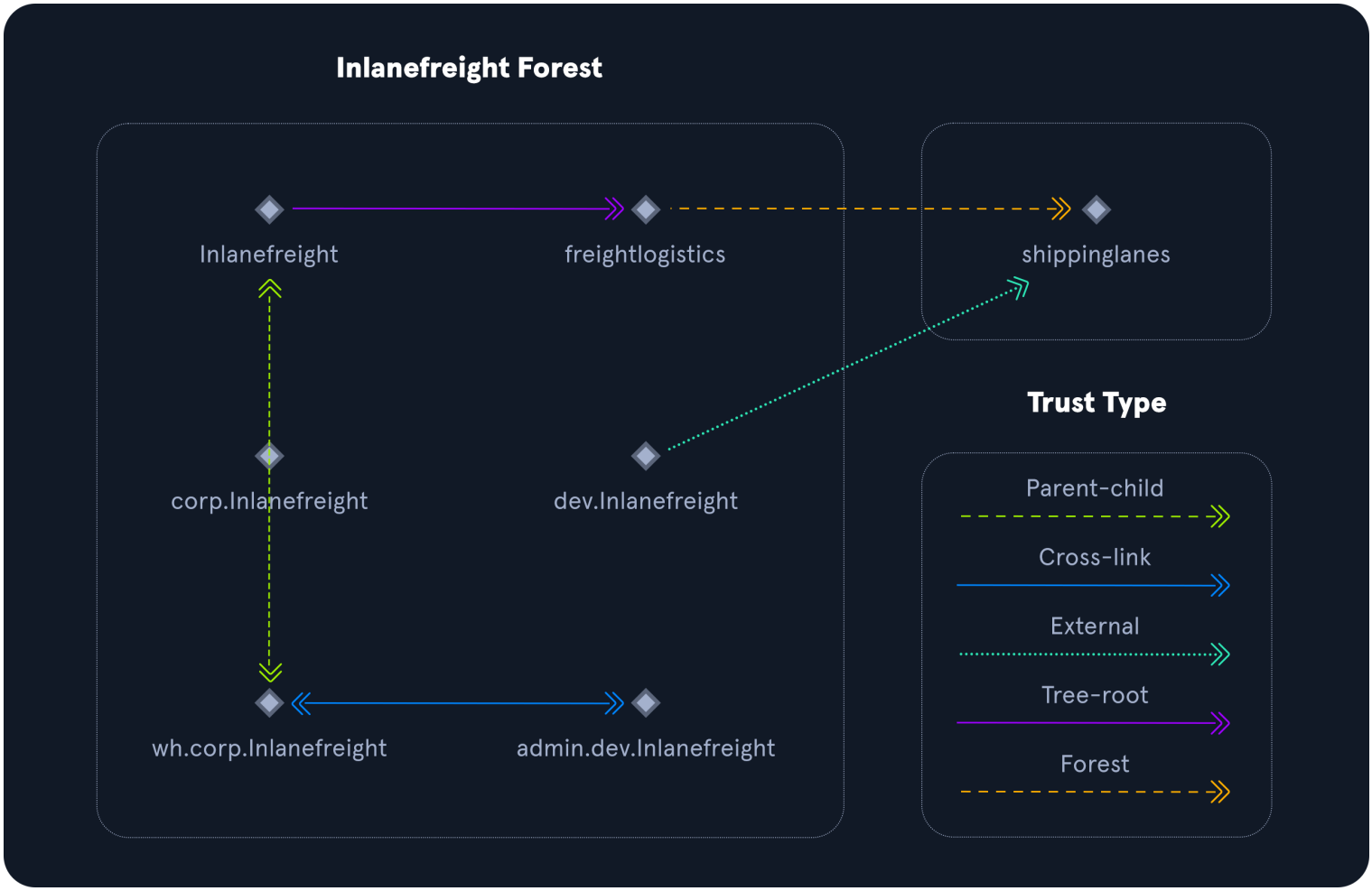
Trusts

A trust is used to establish `forest-forest` or `domain-domain` authentication, allowing users to access resources in (or administer) another domain outside of the domain their account resides in. A trust creates a link between the authentication systems of two domains.

There are several trust types.

Trust Type	Description
`Parent-child`	Domains within the same forest. The child domain has a two-way transitive trust with the parent domain.
`Cross-link`	a trust between child domains to speed up authentication.
`External`	A non-transitive trust between two separate domains in separate forests which are not already joined by a forest trust. This type of trust utilizes SID filtering.
`Tree-root`	a two-way transitive trust between a forest root domain and a new tree root domain. They are created by design when you set up a new tree root domain within a forest.
`Forest`	a transitive trust between two forest root domains.

Trust Example



Trusts can be transitive or non-transitive.

- A transitive trust means that trust is extended to objects that the child domain trusts.
- In a non-transitive trust, only the child domain itself is trusted.

Trusts can be set up to be one-way or two-way (bidirectional).

- In bidirectional trusts, users from both trusting domains can access resources.
- In a one-way trust, only users in a trusted domain can access resources in a trusting domain, not vice-versa. The direction of trust is opposite to the direction of access.

Often, domain trusts are set up improperly and provide unintended attack paths. Also, trusts set up for ease of use may not be reviewed later for potential security implications. Mergers and acquisitions can result in bidirectional trusts with acquired companies, unknowingly introducing risk into the acquiring company's environment. It is not uncommon to be able to perform an attack such as Kerberoasting against a domain outside the principal domain and obtain a user that has administrative access within the principal domain.

Questions

Answer the question(s) below
to complete this Section and earn cubes!

Cheat Sheet

+ 0 What role maintains time for a domain?

Submit

+ 1 What domain functional level introduced Managed Service Accounts?

Submit

+ 0 What type of trust is a link between two child domains in a forest?

Submit

Hint

+ 0 What role ensures that objects in a domain are not assigned the same SID? (full name)

Submit

Hint

Kerberos, DNS, LDAP, MSRPC

While Windows operating systems use a variety of protocols to communicate, Active Directory specifically requires Lightweight Directory Access Protocol (LDAP), Microsoft's version of Kerberos, DNS for authentication and communication, and MSRPC which is the Microsoft implementation of Remote Procedure Call (RPC), an interprocess communication technique used for client-server model-based applications.

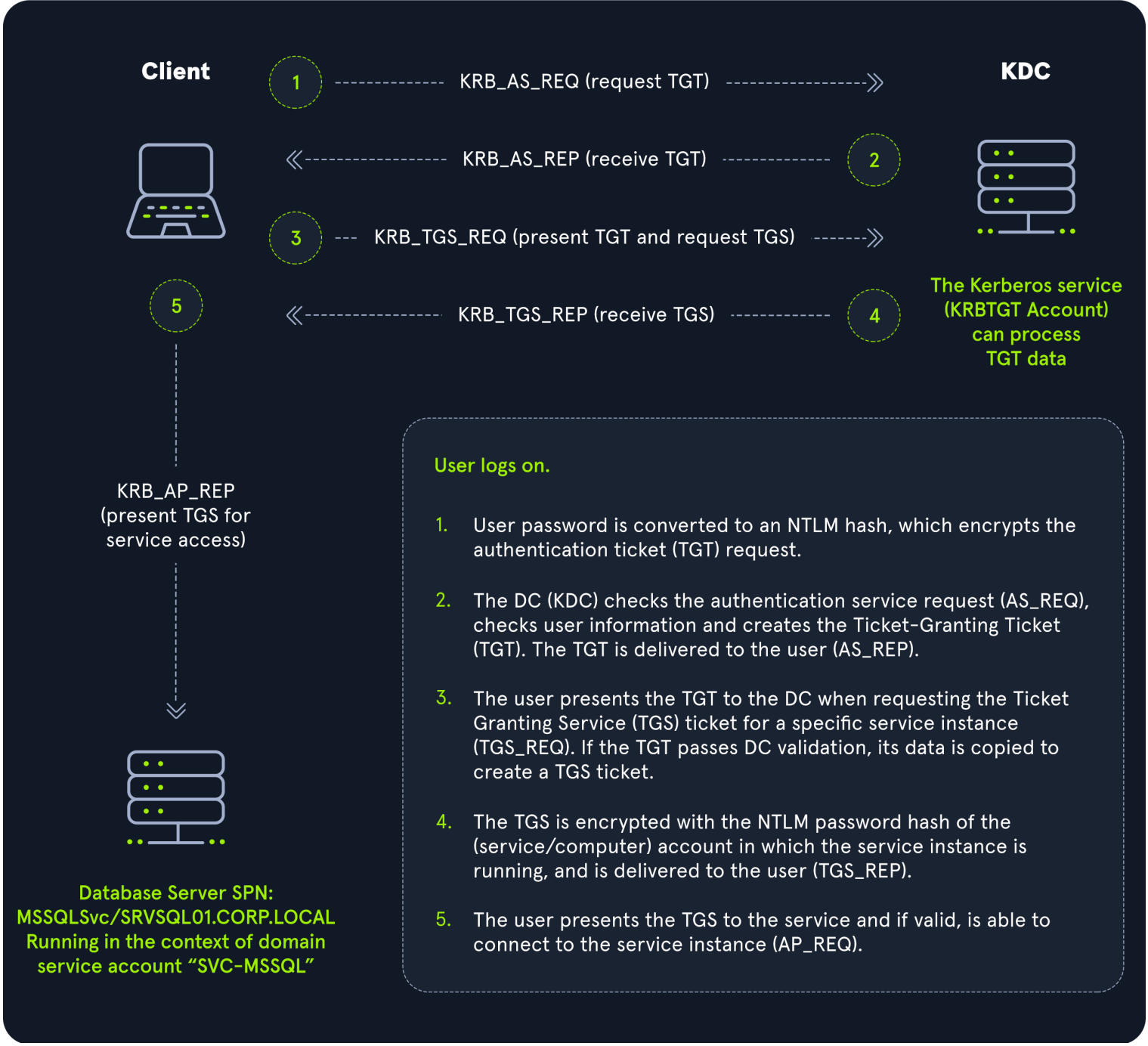
Kerberos

Kerberos has been the default authentication protocol for domain accounts since Windows 2000. Kerberos is an open standard and allows for interoperability with other systems using the same standard. When a user logs into their PC, Kerberos is used to authenticate them via mutual authentication, or both the user and the server verify their identity. Kerberos is a stateless authentication protocol based on tickets instead of transmitting user passwords over the network. As part of Active Directory Domain Services (AD DS), Domain Controllers have a Kerberos Key Distribution Center (KDC) that issues tickets. When a user initiates a login request to a system, the client they are using to authenticate requests a ticket from the KDC, encrypting the request with the user's password. If the KDC can decrypt the request (AS-REQ) using their password, it will create a Ticket Granting Ticket (TGT) and transmit it to the user. The user then presents its TGT to a Domain Controller to request a Ticket Granting Service (TGS) ticket, encrypted with the associated service's NTLM password hash. Finally, the client requests access to the required service by presenting the TGS to the application or service, which decrypts it with its password hash. If the entire process completes appropriately, the user will be permitted to access the requested service or application.

Kerberos authentication effectively decouples users' credentials from their requests to consumable resources, ensuring that their password isn't transmitted over the network (i.e., accessing an internal SharePoint intranet site). The Kerberos Key Distribution Centre (KDC) does not record previous transactions. Instead, the Kerberos Ticket Granting Service ticket (TGS) relies on a valid Ticket Granting Ticket (TGT). It assumes that if the user has a valid TGT, they must have proven their identity. The following diagram walks through this process at a high level.

Kerberos Authentication Process

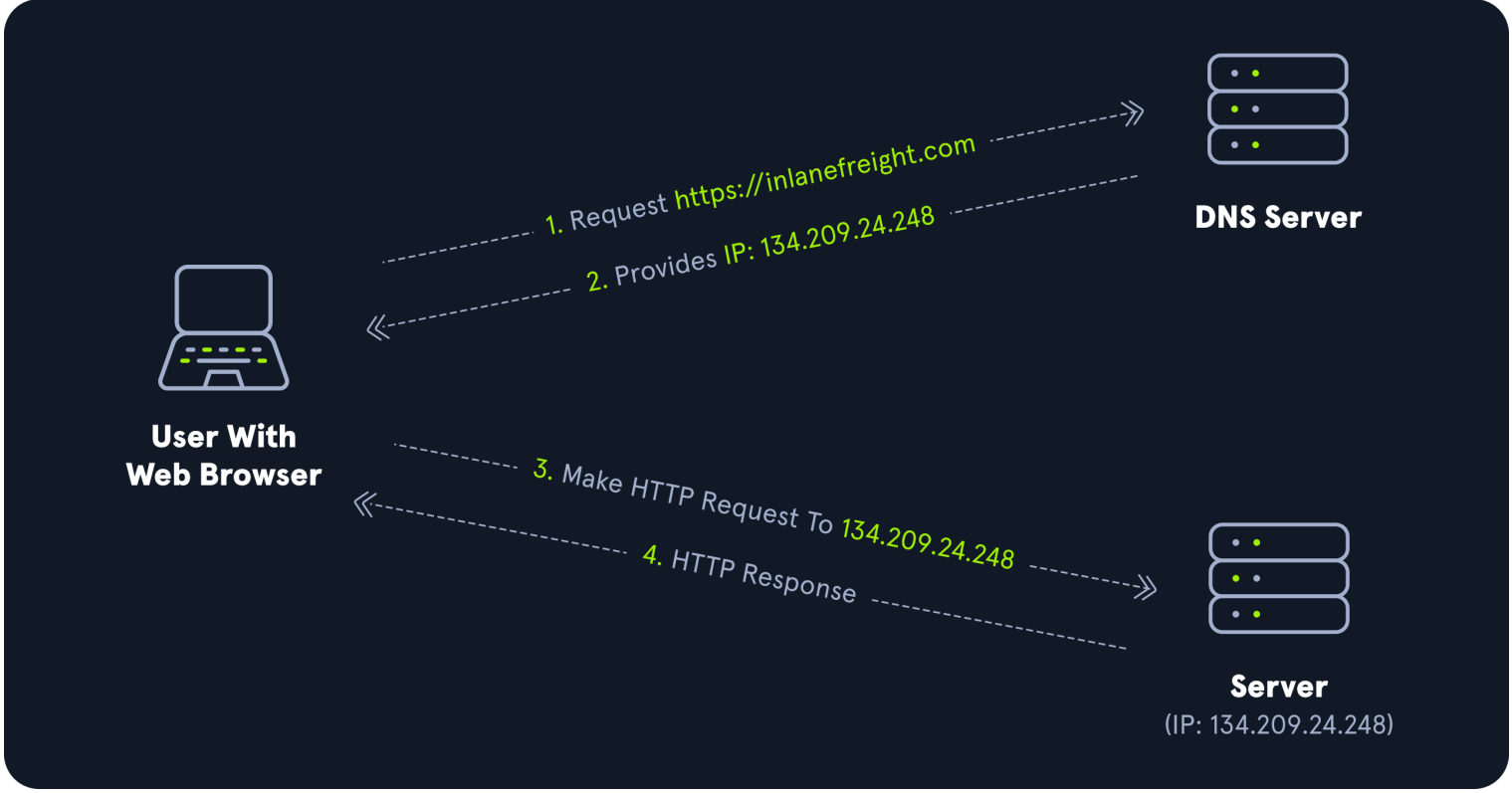
1. The user logs on, and their password is converted to an NTLM hash, which is used to encrypt the TGT ticket. This decouples the user's credentials from requests to resources.
2. The KDC service on the DC checks the authentication service request (AS-REQ), verifies the user information, and creates a Ticket Granting Ticket (TGT), which is delivered to the user.
3. The user presents the TGT to the DC, requesting a Ticket Granting Service (TGS) ticket for a specific service. This is the TGS-REQ. If the TGT is successfully validated, its data is copied to create a TGS ticket.
4. The TGS is encrypted with the NTLM password hash of the service or computer account in whose context the service instance is running and is delivered to the user in the TGS_REP.
5. The user presents the TGS to the service, and if it is valid, the user is permitted to connect to the resource (AP_REQ).



The Kerberos protocol uses port 88 (both TCP and UDP). When enumerating an Active Directory environment, we can often locate Domain Controllers by performing port scans looking for open port 88 using a tool such as Nmap.

DNS

Active Directory Domain Services (AD DS) uses DNS to allow clients (workstations, servers, and other systems that communicate with the domain) to locate Domain Controllers and for Domain Controllers that host the directory service to communicate amongst themselves. DNS is used to resolve hostnames to IP addresses and is broadly used across internal networks and the internet. Private internal networks use Active Directory DNS namespaces to facilitate communications between servers, clients, and peers. AD maintains a database of services running on the network in the form of service records (SRV). These service records allow clients in an AD environment to locate services that they need, such as a file server, printer, or Domain Controller. Dynamic DNS is used to make changes in the DNS database automatically should a system's IP address change. Making these entries manually would be very time-consuming and leave room for error. If the DNS database does not have the correct IP address for a host, clients will not be able to locate and communicate with it on the network. When a client joins the network, it locates the Domain Controller by sending a query to the DNS service, retrieving an SRV record from the DNS database, and transmitting the Domain Controller's hostname to the client. The client then uses this hostname to obtain the IP address of the Domain Controller. DNS uses TCP and UDP port 53. UDP port 53 is the default, but it falls back to TCP when no longer able to communicate and DNS messages are larger than 512 bytes.



Forward DNS Lookup

Let's look at an example. We can perform a `nslookup` for the domain name and retrieve all Domain Controllers' IP addresses in a domain.

```
PS C:\htb> nslookup INLANEFREIGHT.LOCAL

Server:  172.16.6.5
Address:  172.16.6.5

Name:    INLANEFREIGHT.LOCAL
Address:  172.16.6.5
```

Reverse DNS Lookup

If we would like to obtain the DNS name of a single host using the IP address, we can do this as follows:

```
PS C:\htb> nslookup 172.16.6.5

Server:  172.16.6.5
Address:  172.16.6.5

Name:    ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL
Address:  172.16.6.5
```

Finding IP Address of a Host

If we would like to find the IP address of a single host, we can do this in reverse. We can do this with or without specifying the FQDN.

```
PS C:\htb> nslookup ACADEMY-EA-DC01

Server:   172.16.6.5
Address:  172.16.6.5

Name:    ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL
Address:  172.16.6.5
```

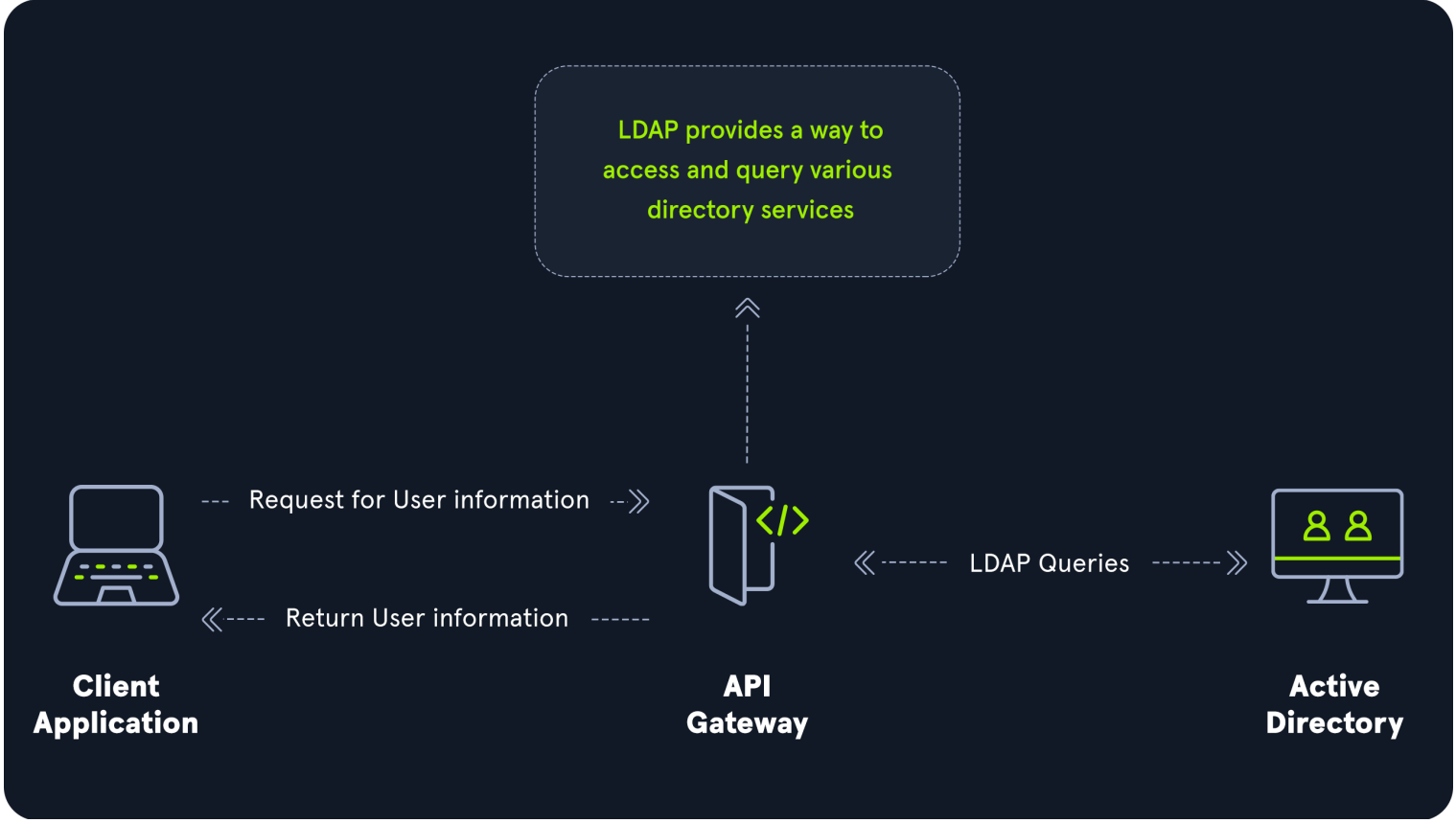
For deeper dives into DNS, check out the DNS Enumeration Using Python module and the DNS section of the Information Gathering - Web Edition module.

LDAP

Active Directory supports Lightweight Directory Access Protocol (LDAP) for directory lookups. LDAP is an open-source and cross-platform protocol used for authentication against various directory services (such as AD). The latest LDAP specification is Version 3, published as RFC 4511. A firm understanding of how LDAP works in an AD environment is crucial for attackers and defenders. LDAP uses port 389, and LDAP over SSL (LDAPS) communicates over port 636.

AD stores user account information and security information such as passwords and facilitates sharing this information with other devices on the network. LDAP is the language that applications use to communicate with other servers that provide directory services. In other words, LDAP is how systems in the network environment can "speak" to AD.

An LDAP session begins by first connecting to an LDAP server, also known as a Directory System Agent. The Domain Controller in AD actively listens for LDAP requests, such as security authentication requests.



The relationship between AD and LDAP can be compared to Apache and HTTP. The same way Apache is a web server that uses the HTTP protocol, Active Directory is a directory server that uses the LDAP protocol.

While uncommon, you may come across organization while performing an assessment that do not have AD but are using LDAP, meaning that they most likely use another type of LDAP server such as OpenLDAP.

AD LDAP Authentication

LDAP is set up to authenticate credentials against AD using a "BIND" operation to set the authentication state for an LDAP session. There are two types of LDAP authentication.

1. `Simple Authentication`: This includes anonymous authentication, unauthenticated authentication, and username/password authentication. Simple authentication means that a `username` and `password` create a BIND request to authenticate to the LDAP server.
2. `SASL Authentication`: The Simple Authentication and Security Layer (SASL) framework uses other authentication services, such as Kerberos, to bind to the LDAP server and then uses this authentication service (Kerberos in this example) to authenticate to LDAP. The LDAP server uses the LDAP protocol to send an LDAP message to the authorization service, which initiates a series of challenge/response messages resulting in either successful or unsuccessful authentication. SASL can provide additional security due to the separation of authentication methods from application protocols.

LDAP authentication messages are sent in cleartext by default so anyone can sniff out LDAP messages on the internal network. It is recommended to use TLS encryption or similar to safeguard this information in transit.

MSRPC

As mentioned above, MSRPC is Microsoft's implementation of Remote Procedure Call (RPC), an interprocess communication technique used for client-server model-based applications. Windows systems use MSRPC to access systems in Active Directory using four key RPC interfaces.

Interface Name	Description
`lsarpc`	A set of RPC calls to the Local Security Authority (LSA) system which manages the local security policy on a computer, controls the audit policy, and provides interactive authentication services. LSARPC is used to perform management on domain security policies.
`netlogon`	Netlogon is a Windows process used to authenticate users and other services in the domain environment. It is a service that continuously runs in the background.
`samr`	Remote SAM (samr) provides management functionality for the domain account database, storing information about users and groups. IT administrators use the protocol to manage users, groups, and computers by enabling admins to create, read, update, and delete information about security principles. Attackers (and pentesters) can use the samr protocol to perform reconnaissance about the internal domain using tools such as BloodHound to visually map out the AD network and create "attack paths" to illustrate visually how administrative access or full domain compromise could be achieved. Organizations can protect against this type of reconnaissance by changing a Windows registry key to only allow administrators to perform remote SAM queries since, by default, all authenticated domain users can make these queries to gather a considerable amount of information about the AD domain.
`drsuapi`	drsuapi is the Microsoft API that implements the Directory Replication Service (DRS) Remote Protocol which is used to perform replication-related tasks across Domain Controllers in a multi-DC environment. Attackers can utilize drsuapi to create a copy of the Active Directory domain database (NTDS.dit) file to retrieve password hashes for all accounts in the domain, which can then be used to perform Pass-the-Hash attacks to access more systems or cracked offline using a tool such as Hashcat to obtain the cleartext password to log in to systems using remote management protocols such as Remote Desktop (RDP) and WinRM.

Questions

Answer the question(s) below
to complete this Section and earn cubes!

Cheat Sheet

+ 1 What networking port does Kerberos use?

Submit

Hint

+ 0 What protocol is utilized to translate names into IP addresses? (acronym)

Submit

Hint

+ 0 What protocol does RFC 4511 specify? (acronym)

Submit

Hint

NTLM Authentication

Aside from Kerberos and LDAP, Active Directory uses several other authentication methods which can be used (and abused) by applications and services in AD. These include LM, NTLM, NTLMv1, and NTLMv2. LM and NTLM here are the hash names, and NTLMv1 and NTLMv2 are authentication protocols that utilize the LM or NT hash. Below is a quick comparison between these hashes and protocols, which shows us that, while not perfect by any means, Kerberos is often the authentication protocol of choice wherever possible. It is essential to understand the difference between the hash types and the protocols that use them.

Hash Protocol Comparison

Hash/Protocol	Cryptographic technique	Mutual Authentication	Message Type	Trusted Third Party
`NTLM`	Symmetric key cryptography	No	Random number	Domain Controller
`NTLMv1`	Symmetric key cryptography	No	MD4 hash, random number	Domain Controller
`NTLMv2`	Symmetric key cryptography	No	MD4 hash, random number	Domain Controller
`Kerberos`	Symmetric key cryptography & asymmetric cryptography	Yes	Encrypted ticket using DES, MD5	Domain Controller/Key Distribution Center (KDC)

LM

`LAN Manager` (LM or LANMAN) hashes are the oldest password storage mechanism used by the Windows operating system. LM debuted in 1987 on the OS/2 operating system. If in use, they are stored in the SAM database on a Windows host and the NTDS.DIT database on a Domain Controller. Due to significant security weaknesses in the hashing algorithm used for LM hashes, it has been turned off by default since Windows Vista/Server 2008. However, it is still common to encounter, especially in large environments where older systems are still used. Passwords using LM are limited to a maximum of `14` characters. Passwords are not case sensitive and are converted to uppercase before generating the hashed value, limiting the keyspace to a total of 69 characters making it relatively easy to crack these hashes using a tool such as Hashcat.

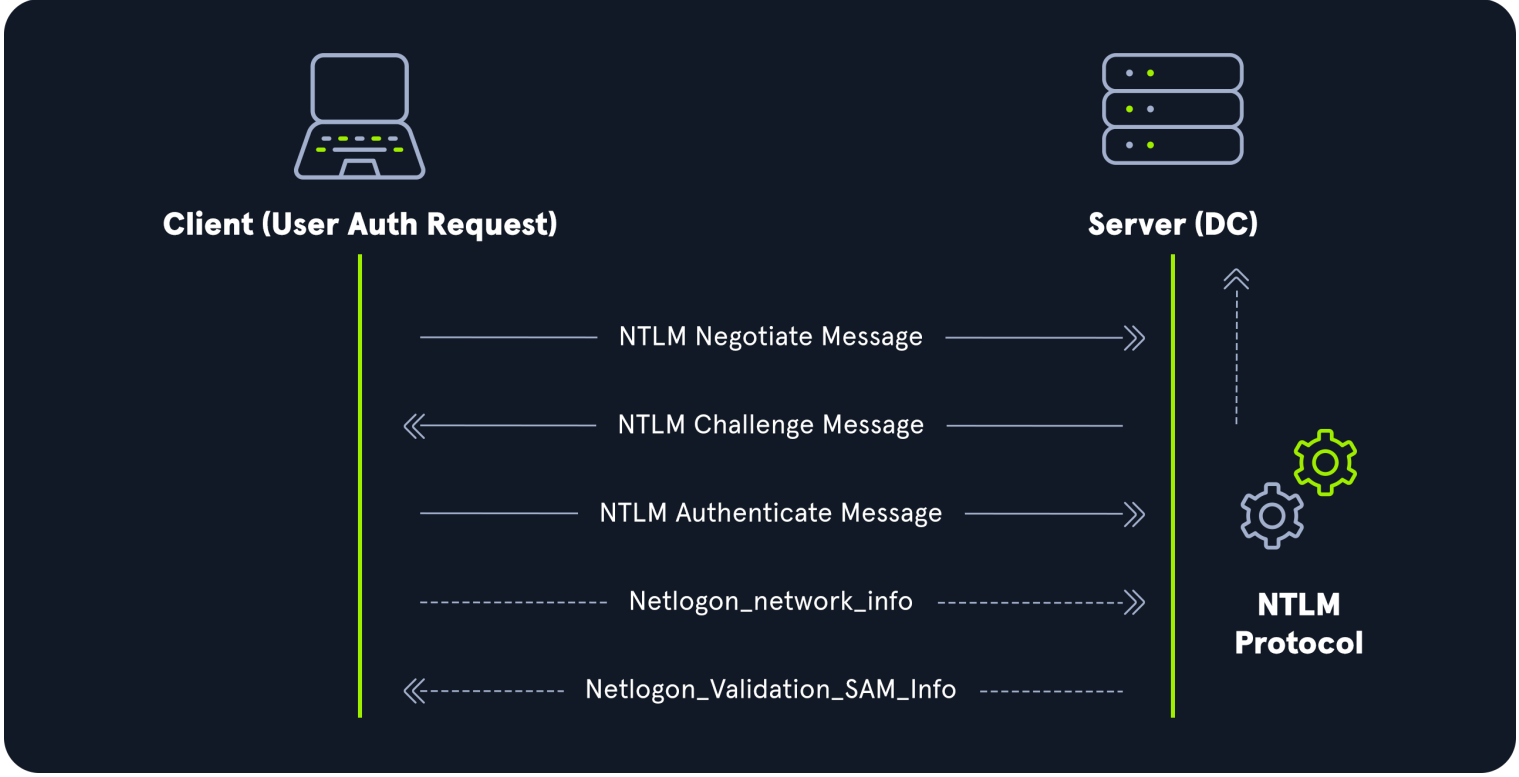
Before hashing, a 14 character password is first split into two seven-character chunks. If the password is less than fourteen characters, it will be padded with NULL characters to reach the correct value. Two DES keys are created from each chunk. These chunks are then encrypted using the string `KGS!@#$%`, creating two 8-byte ciphertext values. These two values are then concatenated together, resulting in an LM hash. This hashing algorithm means that an attacker only needs to brute force seven characters twice instead of the entire fourteen characters, making it fast to crack LM hashes on a system with one or more GPUs. If a password is seven characters or less, the second half of the LM hash will always be the same value and could even be determined visually without even needed tools such as Hashcat. The use of LM hashes can be disallowed using Group Policy. An LM hash takes the form of `299bd128c1101fd6`.

Note: Windows operating systems prior to Windows Vista and Windows Server 2008 (Windows NT4, Windows 2000, Windows 2003, Windows XP) stored both the LM hash and the NTLM hash of a user's password by default.

NTHash (NTLM)

`NT LAN Manager` (NTLM) hashes are used on modern Windows systems. It is a challenge-response authentication protocol and uses three messages to authenticate: a client first sends a `NEGOTIATE_MESSAGE` to the server, whose response is a `CHALLENGE_MESSAGE` to verify the client's identity. Lastly, the client responds with an `AUTHENTICATE_MESSAGE`. These hashes are stored locally in the SAM database or the NTDS.DIT database file on a Domain Controller. The protocol has two hashed password values to choose from to perform authentication: the LM hash (as discussed above) and the NT hash, which is the MD4 hash of the little-endian UTF-16 value of the password. The algorithm can be visualized as: `MD4(UTF-16-LE(password))`.

NTLM Authentication Request



Even though they are considerably stronger than LM hashes (supporting the entire Unicode character set of 65,536 characters), they can still be brute-forced offline relatively quickly using a tool such as Hashcat. GPU attacks have shown that the entire NTLM 8 character keyspace can be brute-forced in under `3 hours`. Longer NTLM hashes can be more challenging to crack depending on the password chosen, and even long passwords (15+ characters) can be cracked using an offline dictionary attack combined with rules. NTLM is also vulnerable to the pass-the-hash attack, which means an attacker can use just the NTLM hash (after obtaining via another successful attack) to authenticate to target systems where the user is a local admin without needing to know the cleartext value of the password.

An NT hash takes the form of `b4b9b02e6f09a9bd760f388b67351e2b`, which is the second half of the full NTLM hash. An NTLM hash looks like this:

```
Rachel:500:aad3c435b514a4eeaad3b935b51304fe:e46b9e548fa0d122de7f59fb6d48eaa2:::
```

Looking at the hash above, we can break the NTLM hash down into its individual parts:

- `Rachel` is the username
- `500` is the Relative Identifier (RID). 500 is the known RID for the `administrator` account
- `aad3c435b514a4eeaad3b935b51304fe` is the LM hash and, if LM hashes are disabled on the system, can not be used for anything
- `e46b9e548fa0d122de7f59fb6d48eaa2` is the NT hash. This hash can either be cracked offline to reveal the cleartext value (depending on the length/strength of the password) or used for a pass-the-hash attack. Below is an example of a successful pass-the-hash attack using the CrackMapExec tool:

```
crackmapexec smb 10.129.41.19 -u rachel -H e46b9e548fa0d122de7f59fb6d48eaa2

SMB         10.129.43.9     445    DC01      [*] Windows 10.0 Build 17763 (name:DC01) (domain:INLANEFREIGHT.LOCAL) (signing:True) (SMBv1:False)
SMB         10.129.43.9     445    DC01      [+] INLANEFREIGHT.LOCAL\rachel:e46b9e548fa0d122de7f59fb6d48eaa2 (Pwn3d!)
```

Now that we understand the capabilities and structure of NTLM let's examine the progression of the protocol through NTLMv1 and NTLMv2.

Note: Neither LANMAN nor NTLM uses a salt.

NTLMv1 (Net-NTLMv1)

The NTLM protocol performs a challenge/response between a server and client using the NT hash. NTLMv1 uses both the NT and the LM hash, which can make it easier to "crack" offline after capturing a hash using a tool such as Responder or via an NTLM relay attack (both of which are out of scope for this module and will be covered in later modules on Lateral Movement). The protocol is used for network authentication, and the Net-NTLMv1 hash itself is created from a challenge/response algorithm. The server sends the client an 8-byte random number (challenge), and the client returns a 24-byte response. These hashes can NOT be used for pass-the-hash attacks. The algorithm looks as follows:

V1 Challenge & Response Algorithm

```
C = 8-byte server challenge, random
K1 | K2 | K3 = LM/NT-hash | 5-bytes-0
response = DES(K1,C) | DES(K2,C) | DES(K3,C)
```

An example of a full NTLMv1 hash looks like:

NTLMv1 Hash Example

```
u4-netntlm::kNS:338d08f8e26de93300000000000000000000000000000000:9526fb8c23a90751cdd619b6cea564742e1e4bf33006ba41:cb8086049ec4736c
```

NTLMv1 was the building block for modern NTLM authentication. Like any protocol, it has flaws and is susceptible to cracking and other attacks. Now let us move on and take a look at NTLMv2 and see how it improves on the foundation that version one set.

NTLMv2 (Net-NTLMv2)

The NTLMv2 protocol was first introduced in Windows NT 4.0 SP4 and was created as a stronger alternative to NTLMv1. It has been the default in Windows since Server 2000. It is hardened against certain spoofing attacks that NTLMv1 is susceptible to. NTLMv2 sends two responses to the 8-byte challenge received by the server. These responses contain a 16-byte HMAC-MD5 hash of the challenge, a randomly generated challenge from the client, and an HMAC-MD5 hash of the user's credentials. A second response is sent, using a variable-length client challenge including the current time, an 8-byte random value, and the domain name. The algorithm is as follows:

V2 Challenge & Response Algorithm

```
SC = 8-byte server challenge, random
CC = 8-byte client challenge, random
CC* = (X, time, CC2, domain name)
v2-Hash = HMAC-MD5(NT-Hash, user name, domain name)
LMv2 = HMAC-MD5(v2-Hash, SC, CC)
NTv2 = HMAC-MD5(v2-Hash, SC, CC*)
response = LMv2 | CC | NTv2 | CC*
```

An example of an NTLMv2 hash is:

NTLMv2 Hash Example

```
admin::N46iSNekpT:08ca45b7d7ea58ee:88dcbe4446168966a153a0064958dac6:5c7830315c7830310000000000000b45c67103d07d7b95acd12ffa11230e0000000052920b85f78d013c31cdb3
b92f5d765c783030
```

We can see that developers improved upon v1 by making NTLMv2 harder to crack and giving it a more robust algorithm made up of multiple stages. We have one more authentication mechanism to discuss before moving on. This method is of note to us because it does not require a persistent network connection to work.

Domain Cached Credentials (MSCache2)

In an AD environment, the authentication methods mentioned in this section and the previous require the host we are trying to access to communicate with the "brains" of the network, the Domain Controller. Microsoft developed the MS Cache v1 and v2 algorithm (also known as `Domain Cached Credentials` (DCC) to solve the potential issue of a domain-joined host being unable to communicate with a domain controller (i.e., due to a network outage or other technical issue) and, hence, NTLM/Kerberos authentication not working to access the host in question. Hosts save the last `ten` hashes for any domain users that successfully log into the machine in the `HKEY_LOCAL_MACHINE\SECURITY\Cache` registry key. These hashes cannot be used in pass-the-hash attacks. Furthermore, the hash is very slow to crack with a tool such as Hashcat, even when using an extremely powerful GPU cracking rig, so attempts to crack these hashes typically need to be extremely targeted or rely on a very weak password in use. These hashes can be obtained by an attacker or pentester after gaining local admin access to a host and have the following format: `$DCC2$10240#bjones#e4e938d12fe5974dc42a90120bd9c90f`. It is vital as penetration testers that we understand the varying types of hashes that we may encounter while assessing an AD environment, their strengths, weaknesses, how they can be abused (cracking to cleartext, pass-the-hash, or relayed), and when an attack may be futile (i.e., spending days attempting to crack a set of Domain Cached Credentials).

Moving On

Now that we have covered authentication protocols and associated password hashes let's look at users and groups in Active Directory, which are typically the most important target for penetration testers and attackers alike. They can have varying privileges and be used to move laterally in an environment or gain access to protected resources.

Questions

Answer the question(s) below
to complete this Section and earn cubes!

Cheat Sheet

+ 0 What Hashing protocol is capable of symmetric and asymmetric cryptography?

Submit

Hint

+ 0 NTLM uses three messages to authenticate; Negotiate, Challenge, and <__>. What is the missing message? (fill in the blank)

Submit

Hint

+ 0 How many hashes does the Domain Cached Credentials mechanism save to a host by default?

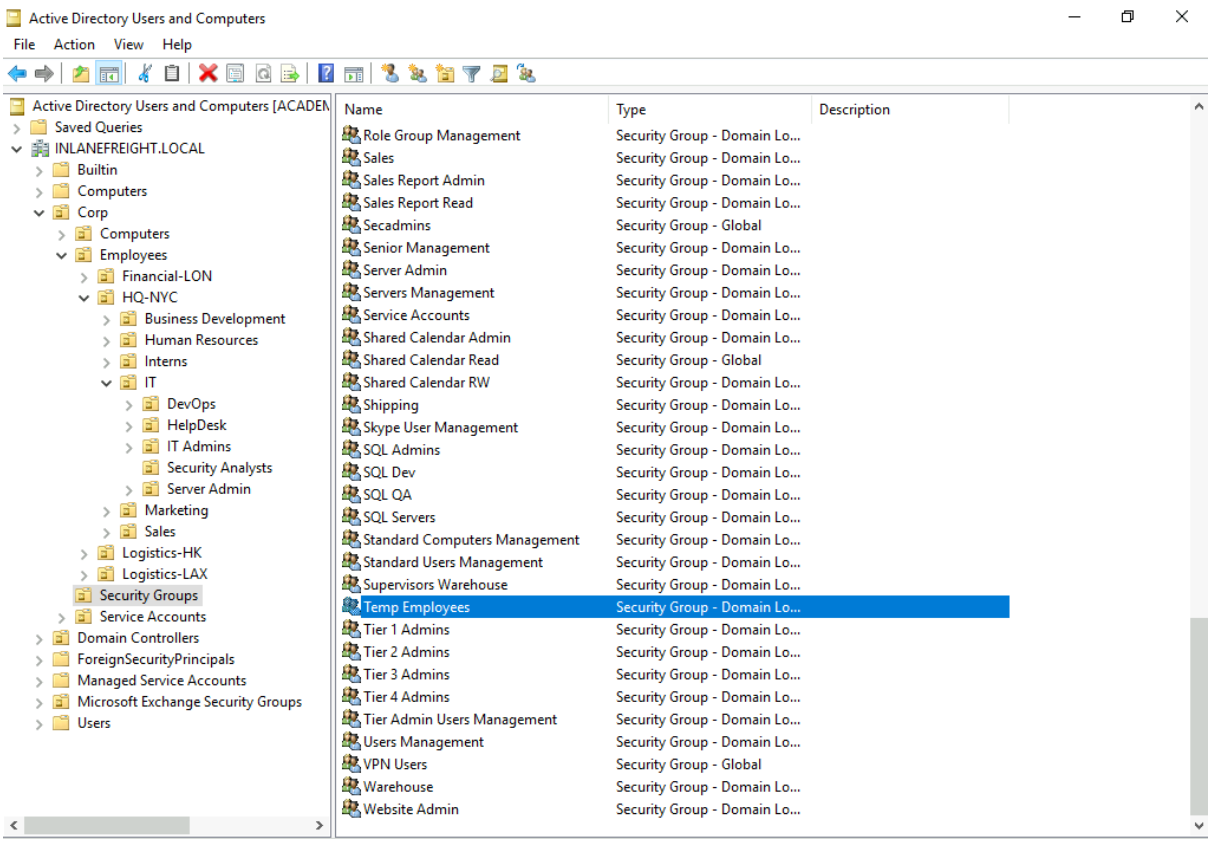
Submit

Hint

User and Machine Accounts

User accounts are created on both local systems (not joined to AD) and in Active Directory to give a person or a program (such as a system service) the ability to log on to a computer and access resources based on their rights. When a user logs in, the system verifies their password and creates an access token. This token describes the security content of a process or thread and includes the user's security identity and group membership. Whenever a user interacts with a process, this token is presented. User accounts are used to allow employees/contractors to log in to a computer and access resources, to run programs or services under a specific security context (i.e., running as a highly privileged user instead of a network service account), and to manage access to objects and their properties such as network file shares, files, applications, etc. Users can be assigned to groups that can contain one or more members. These groups can also be used to control access to resources. It can be easier for an administrator to assign privileges once to a group (which all group members inherit) instead of many times to each individual user. This helps simplify administration and makes it easier to grant and revoke user rights.

The ability to provision and manage user accounts is one of the core elements of Active Directory. Typically, every company we encounter will have at least one AD user account provisioned per user. Some users may have two or more accounts provisioned based on their job role (i.e., an IT admin or Help Desk member). Aside from standard user and admin accounts tied back to a specific user, we will often see many service accounts used to run a particular application or service in the background or perform other vital functions within the domain environment. An organization with 1,000 employees could have 1,200 active user accounts or more! We may also see organizations with hundreds of disabled accounts from former employees, temporary/seasonal employees, interns, etc. Some companies must retain records of these accounts for audit purposes, so they will deactivate them (and hopefully remove all privileges) once the employee is terminated, but they will not delete them. It is common to see an OU such as `FORMER EMPLOYEES` that will contain many deactivated accounts.



As we will see later in this module, user accounts can be provisioned many rights in Active Directory. They can be configured as basically read-only users who have read access to most of the environment (which are the permissions a standard Domain User receives) up to Enterprise Admin (with complete control of every object in the domain) and countless combinations in between. Because users can have so many rights assigned to them, they can also be misconfigured relatively easily and granted unintended rights that an attacker or a penetration tester can leverage. User accounts present an immense attack surface and are usually a key focus for gaining a foothold during a penetration test. Users are often the weakest link in any organization. It is difficult to manage human behavior and account for every user choosing weak or shared passwords, installing unauthorized software, or admins making careless mistakes or being overly permissive with account management. To combat this, an organization needs to have policies and procedures to combat issues that can arise around user accounts and must have defense in depth to mitigate the inherent risk that users bring to the domain.

Specifics on user-related misconfigurations and attacks are outside the scope of this module. Still, it is important to understand the sheer impact users can have within any Active Directory network and to understand the nuances between the different types of users/accounts we may encounter.

Local Accounts

Local accounts are stored locally on a particular server or workstation. These accounts can be assigned rights on that host either individually or via group membership. Any rights assigned can only be granted to that specific host and will not work across the domain. Local user accounts are considered security principals but can only manage access to and secure resources on a standalone host. There are several default local user accounts that are created on a Windows system:

- `Administrator`: this account has the SID `S-1-5-domain-500` and is the first account created with a new Windows installation. It has full control over almost every resource on the system. It cannot be deleted or locked, but it can be disabled or renamed. Windows 10 and Server 2016 hosts disable the built-in administrator account by default and create another local account in the local administrator's group during setup.
- `Guest`: this account is disabled by default. The purpose of this account is to allow users without an account on the computer to log in temporarily with limited access rights. By default, it has a blank password and is generally recommended to be left disabled because of the security risk of allowing anonymous access to a host.
- `SYSTEM`: The SYSTEM (or `NT AUTHORITY\SYSTEM`) account on a Windows host is the default account installed and used by the operating system to perform many of its internal functions. Unlike the Root account on Linux, `SYSTEM` is a service account and does not run entirely in the same context as a regular user. Many of the processes and services running on a host are run under the SYSTEM context. One thing to note with this account is that a profile for it does not exist, but it will have permissions over almost everything on the host. It does not appear in User Manager and cannot be added to any groups. A `SYSTEM` account is the highest permission level one can achieve on a Windows host and, by default, is granted Full Control permissions to all files on a Windows system.
- `Network Service`: This is a predefined local account used by the Service Control Manager (SCM) for running Windows services. When a service runs in the context of this particular account, it will present credentials to remote services.
- `Local Service`: This is another predefined local account used by the Service Control Manager (SCM) for running Windows services. It is configured with minimal privileges on the computer and presents anonymous credentials to the network.

It is worth studying Microsoft's documentation on local default accounts in-depth to gain a better understanding of how the various accounts work together on an individual Windows system and across a domain network. Take some time to look them over and understand the nuances between them.

Domain Users

Domain users differ from local users in that they are granted rights from the domain to access resources such as file servers, printers, intranet hosts, and other objects based on the permissions granted to their user account or the group that account is a member of. Domain user accounts can log in to any host in the domain, unlike local users. For more information on the many different Active Directory account types, check out this link. One account to keep in mind is the `KRBTGT` account, however. This is a type of local account built into the AD infrastructure. This account acts as a service account for the Key Distribution service providing authentication and access for domain resources. This account is a common target of many attackers since gaining control or

access will enable an attacker to have unconstrained access to the domain. It can be leveraged for privilege escalation and persistence in a domain through attacks such as the Golden Ticket attack.

User Naming Attributes

Security in Active Directory can be improved using a set of user naming attributes to help identify user objects like logon name or ID. The following are a few important Naming Attributes in AD:

`UserPrincipalName` (UPN)	This is the primary logon name for the user. By convention, the UPN uses the email address of the user.
`ObjectGUID`	This is a unique identifier of the user. In AD, the ObjectGUID attribute name never changes and remains unique even if the user is removed.
`SAMAccountName`	This is a logon name that supports the previous version of Windows clients and servers.
`objectSID`	The user's Security Identifier (SID). This attribute identifies a user and its group memberships during security interactions with the server.
`sIDHistory`	This contains previous SIDs for the user object if moved from another domain and is typically seen in migration scenarios from domain to domain. After a migration occurs, the last SID will be added to the `sIDHistory` property, and the new SID will become its `objectSID`.

Common User Attributes

```
PS C:\htb Get-ADUser -Identity htb-student

DistinguishedName : CN=htb student,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
Enabled           : True
GivenName         : htb
Name              : htb student
ObjectClass       : user
ObjectGUID        : aa799587-c641-4c23-a2f7-75850b4dd7e3
SamAccountName    : htb-student
SID               : S-1-5-21-3842939050-3880317879-2865463114-1111
Surname           : student
UserPrincipalName : [email protected]
```

For a deeper look at user object attributes, check out this page. Many attributes can be set for any object in AD. Many objects will never be used or are not relevant to us as security professionals. Still, it is essential to familiarize ourselves with the most common and more obscure ones that may contain sensitive data or help mount an attack.

Domain-joined vs. Non-Domain-joined Machines

When it comes to computer resources, there are several ways they are typically managed. Below we will discuss the differences between a host joined to a domain versus a host that is only in a workgroup.

`Domain joined`

Hosts joined to a domain have greater ease of information sharing within the enterprise and a central management point (the DC) to gather resources, policies, and updates from. A host joined to a domain will acquire any configurations or changes necessary through the domain's Group Policy. The benefit here is that a user in the domain can log in and access resources from any host joined to the domain, not just the one they work on. This is the typical setup you will see in enterprise environments.

`Non-domain joined`

Non-domain joined computers or computers in a `workgroup` are not managed by domain policy. With that in mind, sharing resources outside your local network is much more complicated than it would be on a domain. This is fine for computers meant for home use or small business clusters on the same LAN. The advantage of this setup is that the individual users are in charge of any changes they wish to make to their host. Any user accounts on a workgroup computer only exist on that host, and profiles are not migrated to other hosts within the workgroup.

It is important to note that a machine account (`NT AUTHORITY\SYSTEM` level access) in an AD environment will have most of the same rights as a standard domain user account. This is important because we do not always need to obtain a set of valid credentials for an individual user's account to begin enumerating and attacking a domain (as we will see in later modules). We may obtain `SYSTEM` level access to a domain-joined Windows host through a successful remote code execution exploit or by escalating privileges on a host. This access is often overlooked as only useful for pillaging sensitive data (i.e., passwords, SSH keys, sensitive files, etc.) on a particular host. In reality, access in the context of the `SYSTEM` account will allow us read access to much of the data within the domain and is a great launching point for gathering as much information about the domain as possible before proceeding with applicable AD-related attacks.

Questions

Answer the question(s) below
to complete this Section and earn cubes!

Cheat Sheet

+ 0 True or False; A local user account can be used to login to any domain connected host.

Submit

Hint

+ 0 What default user account has the SID "S-1-5-domain-500" ?

Submit

Hint

+ 1 What account has the highest permission level possible on a Windows host

Submit

Hint

+ 0 What user naming attribute is unique to the user and will remain so even if the account is deleted?

Submit

Hint

Active Directory Groups

After users, groups are another significant object in Active Directory. They can place similar users together and mass assign rights and access. Groups are another key target for attackers and penetration testers, as the rights that they confer on their members may not be readily apparent but may grant excessive (and even unintended) privileges that can be abused if not set up correctly. There are many built-in groups in Active Directory, and most organizations also create their own groups to define rights and privileges, further managing access within the domain. The number of groups in an AD environment can snowball and become unwieldy, potentially leading to unintended access if left unchecked. It is essential to understand the impact of using different group types and for any organization to periodically `audit` which groups exist within their domain, the privileges that these groups grant their members, and check for excessive group membership beyond what is required for a user to perform their day-to-day work. Moving on, we will discuss the different types of groups that exist and the scopes they can be assigned to.

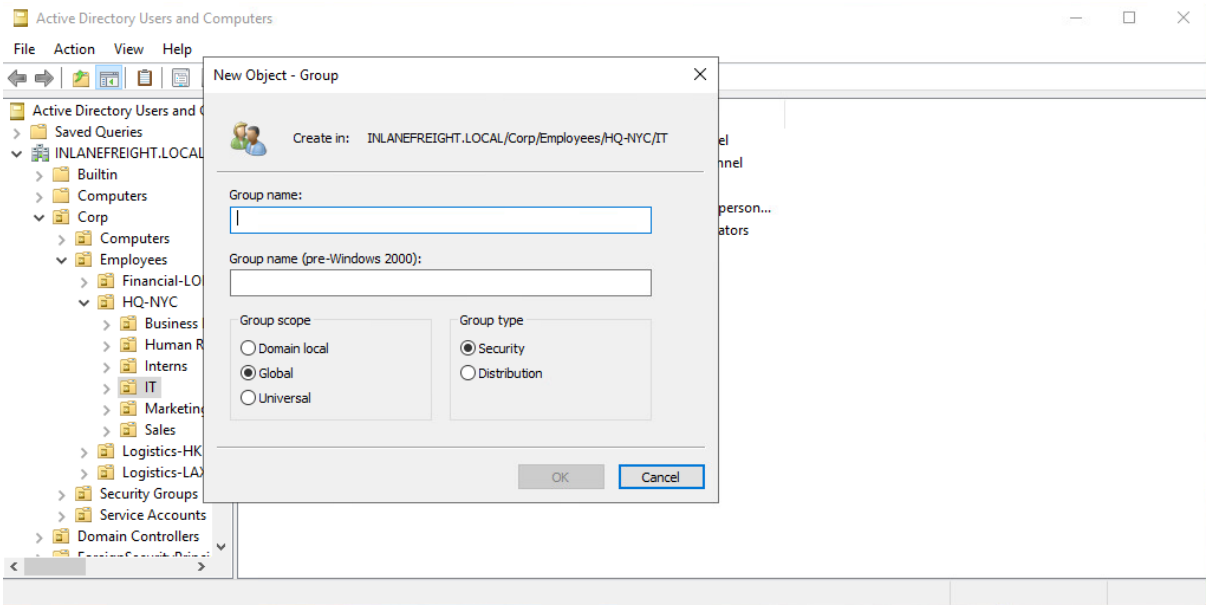
One question that comes up often is the difference between Groups and Organizational Units (OUs). As discussed earlier in the module, OUs are useful for grouping users, groups, and computers to ease management and deploying Group Policy settings to specific objects in the domain. Groups are primarily used to assign permissions to access resources. OUs can also be used to delegate administrative tasks to a user, such as resetting passwords or unlocking user accounts without giving them additional admin rights that they may inherit through group membership.

Types of Groups

In simpler terms, groups are used to place users, computers, and contact objects into management units that provide ease of administration over permissions and facilitate the assignment of resources such as printers and file share access. For example, if an admin needs to assign 50 members of a department access to a new share drive, it would be time-consuming to add each user's account individually. Granting permissions this way would also make it more difficult to audit who has access to resources and difficult to clean up/revoke permissions. Instead, a sysadmin can either use an existing group or create a new group and grant that specific group permissions over the resource. From here, every user in the group will inherit the permissions based on their membership in the group. If the permissions need to be modified or revoked for one or more users, they could merely be removed from the group, leaving the other users unaffected and their permissions intact.

Groups in Active Directory have two fundamental characteristics: `type` and `scope`. The `group type` defines the group's purpose, while the `group scope` shows how the group can be used within the domain or forest. When creating a new group, we must select a group type. There are two main types: `security` and `distribution` groups.

Group Type And Scope



The `Security groups` type is primarily for ease of assigning permissions and rights to a collection of users instead of one at a time. They simplify management and reduce overhead when assigning permissions and rights for a given resource. All users added to a security group will inherit any permissions assigned to the group, making it easier to move users in and out of groups while leaving the group's permissions unchanged.

The `Distribution groups` type is used by email applications such as Microsoft Exchange to distribute messages to group members. They function much like mailing lists and allow for auto-adding emails in the "To" field when creating an email in Microsoft Outlook. This type of group cannot be used to assign permissions to resources in a domain environment.

Group Scopes

There are three different `group scopes` that can be assigned when creating a new group.

1. Domain Local Group
2. Global Group
3. Universal Group

Domain Local Group

Domain local groups can only be used to manage permissions to domain resources in the domain where it was created. Local groups cannot be used in other domains but `CAN` contain users from `OTHER` domains. Local groups can be nested into (contained within) other local groups but `NOT` within global groups.

Global Group

Global groups can be used to grant access to resources in `another domain`. A global group can only contain accounts from the domain where it was created. Global groups can be added to both other global groups and local groups.

Universal Group

The universal group scope can be used to manage resources distributed across multiple domains and can be given permissions to any object within the same `forest`. They are available to all domains within an organization and can contain users from any domain. Unlike domain local and global groups, universal groups are stored in the Global Catalog (GC), and adding or removing objects from a universal group triggers forest-wide replication. It is recommended that administrators maintain other groups (such as global groups) as members of universal groups because global group membership within universal groups is less likely to change than individual user membership in global groups. Replication is only triggered at the individual domain level when a user is removed from a global group. If individual users and computers (instead of global groups) are maintained within universal groups, it will trigger forest-wide replication each time a change is made. This can create a lot of network overhead and potential for issues. Below is an example of the groups in AD and their scope settings. Please pay attention to some of the critical groups and their scope. (Enterprise and Schema admins compared to Domain admins, for example.)

AD Group Scope Examples

```
PS C:\htb> Get-ADGroup  -Filter * |select samaccountname,groupscope

samaccountname                      groupscope
--------------                      ----------
Administrators                     DomainLocal
Users                              DomainLocal
Guests                             DomainLocal
Print Operators                    DomainLocal
Backup Operators                   DomainLocal
Replicator                         DomainLocal
Remote Desktop Users               DomainLocal
Network Configuration Operators    DomainLocal
Distributed COM Users              DomainLocal
IIS_IUSRS                          DomainLocal
Cryptographic Operators            DomainLocal
Event Log Readers                  DomainLocal
Certificate Service DCOM Access    DomainLocal
RDS Remote Access Servers          DomainLocal
RDS Endpoint Servers               DomainLocal
RDS Management Servers             DomainLocal
Hyper-V Administrators             DomainLocal
Access Control Assistance Operators DomainLocal
Remote Management Users            DomainLocal
Storage Replica Administrators     DomainLocal
Domain Computers                        Global
Domain Controllers                      Global
Schema Admins                         Universal
Enterprise Admins                     Universal
Cert Publishers                    DomainLocal
Domain Admins                           Global
Domain Users                            Global
Domain Guests                           Global

<SNIP>
```

Group scopes can be changed, but there are a few caveats:

- A Global Group can only be converted to a Universal Group if it is NOT part of another Global Group.
- A Domain Local Group can only be converted to a Universal Group if the Domain Local Group does NOT contain any other Domain Local Groups as members.
- A Universal Group can be converted to a Domain Local Group without any restrictions.
- A Universal Group can only be converted to a Global Group if it does NOT contain any other Universal Groups as members.

Built-in vs. Custom Groups

Several built-in security groups are created with a Domain Local Group scope when a domain is created. These groups are used for specific administrative purposes and are discussed more in the next section. It is important to note that only user accounts can be added to these built-in groups as they do not allow for group nesting (groups within groups). Some examples of built-in groups included `Domain Admins`, which is a `Global` security group and can only contain accounts from its own domain. If an organization wants to allow an account from domain B to perform administrative functions on a domain controller in domain A, the account would have to be added to the built-in Administrators group, which is a `Domain Local` group. Though Active Directory comes prepopulated with many groups, it is common for most organizations to create additional groups (both security and distribution) for their own purposes. Changes/additions to an AD environment can also trigger the creation of additional groups. For example, when Microsoft Exchange is added to a domain, it adds various different security groups to the domain, some of which are highly privileged and, if not managed properly, can be used to gain privileged access within the domain.

Nested Group Membership

Nested group membership is an important concept in AD. As mentioned previously, a Domain Local Group can be a member of another Domain Local Group in the same domain. Through this membership, a user may inherit privileges not assigned directly to their account or even the group they are directly a member of, but rather the group that their group is a member of. This can sometimes lead to unintended privileges granted to a user that are difficult to uncover without an in-depth assessment of the domain. Tools such as BloodHound are particularly useful in uncovering privileges that a user may inherit through one or more nestings of groups. This is a key tool for penetration testers for uncovering nuanced misconfigurations and is also extremely powerful for sysadmins and the like to gain deep insights (visually) into the security posture of their domain(s).

Below is an example of privileges inherited through nested group membership. Though `DCorner` is not a direct member of `Helpdesk Level 1`, their membership in `Help Desk` grants them the same privileges that any member of `Helpdesk Level 1` has. In this case, the privilege would allow them to add a member to the `Tier 1 Admins` group (`GenericWrite`). If this group confers any elevated privileges in the domain, it would likely be a key target for a penetration tester. Here, we could add our user to the group and obtain privileges that members of the `Tier 1 Admins` group are granted, such as local administrator access to one or more hosts that could be used to further access.

Examining Nested Groups via BloodHound

Important Group Attributes

Like users, groups have many attributes. Some of the most important group attributes include:

- `cn`: The `cn` or Common-Name is the name of the group in Active Directory Domain Services.
- `member`: Which user, group, and contact objects are members of the group.
- `groupType`: An integer that specifies the group type and scope.
- `memberOf`: A listing of any groups that contain the group as a member (nested group membership).
- `objectSid`: This is the security identifier or SID of the group, which is the unique value used to identify the group as a security principal.

Groups are fundamental objects in AD that can be used to group other objects together and facilitate the management of rights and access. Take the time to study the differences between group types and scopes. This knowledge is useful for administering AD as well as understanding the relationships between groups in the same and different domains and what information can be enumerated during the recon phase of a penetration test. Understanding how different group types can be utilized to perform attacks in a single domain and across trust boundaries is an excellent bit of knowledge to have. We deep-dived into Groups in this section, now let's examine the differences between `Rights` and `Privileges`.

Questions

Answer the question(s) below
to complete this Section and earn cubes!

Cheat Sheet

+ 1 What group type is best utilized for assigning permissions and right to users?

Submit

Hint

+ 0 True or False; A "Global Group" can only contain accounts from the domain where it was created.

Submit

+ 0 Can a Universal group be converted to a Domain Local group? (yes or no)

Submit

Hint

Active Directory Rights and Privileges

Rights and privileges are the cornerstones of AD management and, if mismanaged, can easily lead to abuse by attackers or penetration testers. Access rights and privileges are two important topics in AD (and infosec in general), and we must understand the difference. `Rights` are typically assigned to users or groups and deal with permissions to `access` an object such as a file, while `privileges` grant a user permission to `perform an action` such as run a program, shut down a system, reset passwords, etc. Privileges can be assigned individually to users or conferred upon them via built-in or custom group membership. Windows computers have a concept called `User Rights Assignment`, which, while referred to as rights, are actually types of privileges granted to a user. We will discuss these later in this section. We must have a firm grasp of the differences between rights and privileges in a broader sense and precisely how they apply to an AD environment.

Built-in AD Groups

AD contains many default or built-in security groups, some of which grant their members powerful rights and privileges which can be abused to escalate privileges within a domain and ultimately gain Domain Admin or SYSTEM privileges on a Domain Controller (DC). Membership in many of these groups should be tightly managed as excessive group membership/privileges is a common flaw in many AD networks that attackers look to abuse. Some of the most common built-in groups are listed below.

Group Name	Description
`Account Operators`	Members can create and modify most types of accounts, including those of users, local groups, and global groups, and members can log in locally to domain controllers. They cannot manage the Administrator account, administrative user accounts, or members of the Administrators, Server Operators, Account Operators, Backup Operators, or Print Operators groups.
`Administrators`	Members have full and unrestricted access to a computer or an entire domain if they are in this group on a Domain Controller.

Group Name	Description
Backup Operators	Members can back up and restore all files on a computer, regardless of the permissions set on the files. Backup Operators can also log on to and shut down the computer. Members can log onto DCs locally and should be considered Domain Admins. They can make shadow copies of the SAM/NTDS database, which, if taken, can be used to extract credentials and other juicy info.
DnsAdmins	Members have access to network DNS information. The group will only be created if the DNS server role is or was at one time installed on a domain controller in the domain.
Domain Admins	Members have full access to administer the domain and are members of the local administrator's group on all domain-joined machines.
Domain Computers	Any computers created in the domain (aside from domain controllers) are added to this group.
Domain Controllers	Contains all DCs within a domain. New DCs are added to this group automatically.
Domain Guests	This group includes the domain's built-in Guest account. Members of this group have a domain profile created when signing onto a domain-joined computer as a local guest.
Domain Users	This group contains all user accounts in a domain. A new user account created in the domain is automatically added to this group.
Enterprise Admins	Membership in this group provides complete configuration access within the domain. The group only exists in the root domain of an AD forest. Members in this group are granted the ability to make forest-wide changes such as adding a child domain or creating a trust. The Administrator account for the forest root domain is the only member of this group by default.
Event Log Readers	Members can read event logs on local computers. The group is only created when a host is promoted to a domain controller.
Group Policy Creator Owners	Members create, edit, or delete Group Policy Objects in the domain.
Hyper-V Administrators	Members have complete and unrestricted access to all the features in Hyper-V. If there are virtual DCs in the domain, any virtualization admins, such as members of Hyper-V Administrators, should be considered Domain Admins.
IIS_IUSRS	This is a built-in group used by Internet Information Services (IIS), beginning with IIS 7.0.
Pre-Windows 2000 Compatible Access	This group exists for backward compatibility for computers running Windows NT 4.0 and earlier. Membership in this group is often a leftover legacy configuration. It can lead to flaws where anyone on the network can read information from AD without requiring a valid AD username and password.
Print Operators	Members can manage, create, share, and delete printers that are connected to domain controllers in the domain along with any printer objects in AD. Members are allowed to log on to DCs locally and may be used to load a malicious printer driver and escalate privileges within the domain.
Protected Users	Members of this group are provided additional protections against credential theft and tactics such as Kerberos abuse.
Read-only Domain Controllers	Contains all Read-only domain controllers in the domain.
Remote Desktop Users	This group is used to grant users and groups permission to connect to a host via Remote Desktop (RDP). This group cannot be renamed, deleted, or moved.
Remote Management Users	This group can be used to grant users remote access to computers via Windows Remote Management (WinRM)
Schema Admins	Members can modify the Active Directory schema, which is the way all objects with AD are defined. This group only exists in the root domain of an AD forest. The Administrator account for the forest root domain is the only member of this group by default.
Server Operators	This group only exists on domain controllers. Members can modify services, access SMB shares, and backup files on domain controllers. By default, this group has no members.

Below we have provided some output regarding domain admins and server operators.

Server Operators Group Details

```
PS C:\htb>  Get-ADGroup -Identity "Server Operators" -Properties *

adminCount                      : 1
CanonicalName                   : INLANEFREIGHT.LOCAL/Builtin/Server Operators
CN                              : Server Operators
Created                         : 10/27/2021 8:14:34 AM
createTimeStamp                 : 10/27/2021 8:14:34 AM
Deleted                         :
Description                     : Members can administer domain servers
DisplayName                     :
DistinguishedName               : CN=Server Operators,CN=Builtin,DC=INLANEFREIGHT,DC=LOCAL
dSCorePropagationData           : {10/28/2021 1:47:52 PM, 10/28/2021 1:44:12 PM, 10/28/2021 1:44:11 PM, 10/27/2021
                                  8:50:25 AM...}
GroupCategory                   : Security
GroupScope                      : DomainLocal
groupType                       : -2147483643
HomePage                        :
instanceType                    : 4
isCriticalSystemObject          : True
isDeleted                       :
LastKnownParent                 :
ManagedBy                       :
MemberOf                        : {}
Members                         : {}
Modified                        : 10/28/2021 1:47:52 PM
modifyTimeStamp                 : 10/28/2021 1:47:52 PM
Name                            : Server Operators
nTSecurityDescriptor            : System.DirectoryServices.ActiveDirectorySecurity
ObjectCategory                  : CN=Group,CN=Schema,CN=Configuration,DC=INLANEFREIGHT,DC=LOCAL
ObjectClass                     : group
ObjectGUID                      : 0887487b-7b07-4d85-82aa-40d25526ec17
objectSid                       : S-1-5-32-549
ProtectedFromAccidentalDeletion : False
SamAccountName                  : Server Operators
sAMAccountType                  : 536870912
sDRightsEffective               : 0
SID                             : S-1-5-32-549
SIDHistory                      : {}
systemFlags                     : -1946157056
uSNChanged                      : 228556
uSNCreated                      : 12360
whenChanged                     : 10/28/2021 1:47:52 PM
whenCreated                     : 10/27/2021 8:14:34 AM
```

As we can see above, the default state of the `Server Operators` group is to have no members and is a domain local group by default. In contrast, the `Domain Admins` group seen below has several members and service accounts assigned to it. Domain Admins are also Global groups instead of domain local. More on group membership can be found later in this module. Be wary of who, if anyone, you give access to these groups. An attacker could easily gain the keys to the enterprise if they gain access to a user assigned to these groups.

Domain Admins Group Membership

```
PS C:\htb>  Get-ADGroup -Identity "Domain Admins" -Properties * | select DistinguishedName,GroupCategory,GroupScope,Name,Members

DistinguishedName : CN=Domain Admins,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
GroupCategory     : Security
GroupScope        : Global
Name              : Domain Admins
Members           : {CN=htb-student_adm,CN=Users,DC=INLANEFREIGHT,DC=LOCAL, CN=sharepoint
                    admin,CN=Users,DC=INLANEFREIGHT,DC=LOCAL, CN=FREIGHTLOGISTICSUSER,OU=Service
                    Accounts,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL, CN=PROXYAGENT,OU=Service
                    Accounts,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL...}
```

User Rights Assignment

Depending on their current group membership, and other factors such as privileges that administrators can assign via Group Policy (GPO), users can have various rights assigned to their account. This Microsoft article on User Rights Assignment provides a detailed explanation of each of the user rights that can be set in Windows. Not every right listed here is important to us from a security standpoint as penetration testers or defenders, but some rights granted to an account can lead to unintended consequences such as privilege escalation or access to sensitive files. For example, let's say we can gain write access over a Group Policy Object (GPO) applied to an OU containing one or more users that we control. In this example, we could potentially leverage a tool such as SharpGPOAbuse to assign targeted rights to a user. We may perform many actions in the domain to further our access with these new rights. A few examples include:

Privilege	Description
`SeRemoteInteractiveLogonRight`	This privilege could give our target user the right to log onto a host via Remote Desktop (RDP), which could potentially be used to obtain sensitive data or escalate privileges.
`SeBackupPrivilege`	This grants a user the ability to create system backups and could be used to obtain copies of sensitive system files that can be used to retrieve passwords such as the SAM and SYSTEM Registry hives and the NTDS.dit Active Directory database file.
`SeDebugPrivilege`	This allows a user to debug and adjust the memory of a process. With this privilege, attackers could utilize a tool such as Mimikatz to read the memory space of the Local System Authority (LSASS) process and obtain any credentials stored in memory.
`SeImpersonatePrivilege`	This privilege allows us to impersonate a token of a privileged account such as `NT AUTHORITY\SYSTEM`. This could be leveraged with a tool such as JuicyPotato, RogueWinRM, PrintSpoofer, etc., to escalate privileges on a target system.
`SeLoadDriverPrivilege`	A user with this privilege can load and unload device drivers that could potentially be used to escalate privileges or compromise a system.
`SeTakeOwnershipPrivilege`	This allows a process to take ownership of an object. At its most basic level, we could use this privilege to gain access to a file share or a file on a share that was otherwise not accessible to us.

There are many techniques available to abuse user rights detailed here and here. Though outside the scope of this module, it is essential to understand the impact that assigning the wrong privilege to an account can have within Active Directory. A small admin mistake can lead to a complete system or enterprise compromise.

Viewing a User's Privileges

After logging into a host, typing the command `whoami /priv` will give us a listing of all user rights assigned to the current user. Some rights are only available to administrative users and can only be listed/leveraged when running an elevated CMD or PowerShell session. These concepts of elevated rights and User Account Control (UAC) are security features introduced with Windows Vista that default to restricting applications from running with full permissions unless absolutely necessary. If we compare and contrast the rights available to us as an admin in a non-elevated console vs. an elevated console, we will see that they differ drastically. First, let's look at the rights available to a standard Active Directory user.

Standard Domain User's Rights

```
PS C:\htb> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                    State
============================= ============================== ========
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Disabled
```

We can see that the rights are very `limited`, and none of the "dangerous" rights outlined above are present. Next, let's take a look at a privileged user. Below are the rights available to a Domain Admin user.

Domain Admin Rights Non-Elevated

We can see the following in a `non-elevated` console which does not appear to be anything more than available to the standard domain user. This is because, by default, Windows systems do not enable all rights to us unless we run the CMD or PowerShell console in an elevated context. This is to prevent every application from running with the highest possible privileges. This is controlled by something called User Account Control (UAC) which is covered in-depth in the Windows Privilege Escalation module.

```
PS C:\htb> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                          State
============================= ==================================== ========
SeShutdownPrivilege           Shut down the system                 Disabled
SeChangeNotifyPrivilege       Bypass traverse checking             Enabled
SeUndockPrivilege             Remove computer from docking station Disabled
SeIncreaseWorkingSetPrivilege Increase a process working set       Disabled
```

```
SeTimeZonePrivilege           Change the time zone                Disabled
```

Domain Admin Rights Elevated

If we enter the same command from an elevated PowerShell console, we can see the complete listing of rights available to us:

```
PS C:\htb> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                            Description                                                        State
========================================= ================================================================== ========
SeIncreaseQuotaPrivilege                  Adjust memory quotas for a process                                 Disabled
SeMachineAccountPrivilege                 Add workstations to domain                                         Disabled
SeSecurityPrivilege                       Manage auditing and security log                                   Disabled
SeTakeOwnershipPrivilege                  Take ownership of files or other objects                           Disabled
SeLoadDriverPrivilege                     Load and unload device drivers                                     Disabled
SeSystemProfilePrivilege                  Profile system performance                                         Disabled
SeSystemtimePrivilege                     Change the system time                                             Disabled
SeProfileSingleProcessPrivilege           Profile single process                                             Disabled
SeIncreaseBasePriorityPrivilege           Increase scheduling priority                                       Disabled
SeCreatePagefilePrivilege                 Create a pagefile                                                  Disabled
SeBackupPrivilege                         Back up files and directories                                      Disabled
SeRestorePrivilege                        Restore files and directories                                      Disabled
SeShutdownPrivilege                       Shut down the system                                               Disabled
SeDebugPrivilege                          Debug programs                                                     Enabled
SeSystemEnvironmentPrivilege              Modify firmware environment values                                 Disabled
SeChangeNotifyPrivilege                   Bypass traverse checking                                           Enabled
SeRemoteShutdownPrivilege                 Force shutdown from a remote system                                Disabled
SeUndockPrivilege                         Remove computer from docking station                               Disabled
SeEnableDelegationPrivilege               Enable computer and user accounts to be trusted for delegation     Disabled
SeManageVolumePrivilege                   Perform volume maintenance tasks                                   Disabled
SeImpersonatePrivilege                    Impersonate a client after authentication                          Enabled
SeCreateGlobalPrivilege                   Create global objects                                              Enabled
SeIncreaseWorkingSetPrivilege             Increase a process working set                                     Disabled
SeTimeZonePrivilege                       Change the time zone                                               Disabled
SeCreateSymbolicLinkPrivilege             Create symbolic links                                              Disabled
SeDelegateSessionUserImpersonatePrivilege Obtain an impersonation token for another user in the same session Disabled
```

User rights increase based on the groups they are placed in or their assigned privileges. Below is an example of the rights granted to a `Backup Operators` group member. Users in this group have other rights currently restricted by UAC (additional rights such as the powerful `SeBackupPrivilege` are not enabled by default in a standard console session). Still, we can see from this command that they have the `SeShutdownPrivilege`, which means they can shut down a domain controller. This privilege on its own could not be used to gain access to sensitive data but could cause a massive service interruption should they log onto a domain controller locally (not remotely via RDP or WinRM).

Backup Operator Rights

```
PS C:\htb> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                    State
============================= ============================== ========
SeShutdownPrivilege           Shut down the system           Disabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Disabled
```

As attackers and defenders, we need to understand the rights that are granted to users via membership from built-in security groups in Active Directory. It's not uncommon to find seemingly low privileged users added to one or more of these groups, which can be used to further access or compromise the domain. Access to these groups should be strictly controlled. It is typically best practice to leave most of these groups empty and only add an account to a group if a one-off action needs to be performed or a repetitive task needs to be set up. Any accounts added to one of the groups discussed in this section or granted extra privileges should be strictly controlled and monitored, assigned a very strong password or passphrase, and should be separate from an account used by a sysadmin to perform their day-to-day duties.

Now that we've begun to touch on some security considerations in AD related to user privileges and built-in group membership let's walk through some critical points for securing an Active Directory installation.

Questions

Answer the question(s) below
to complete this Section and earn cubes!

Cheat Sheet

+ 0 What built-in group will grant a user full and unrestricted access to a computer?

Submit

Hint

+ 0 What user right grants a user the ability to make backups of a system?

Submit

+ 0 What Windows command can show us all user rights assigned to the current user?

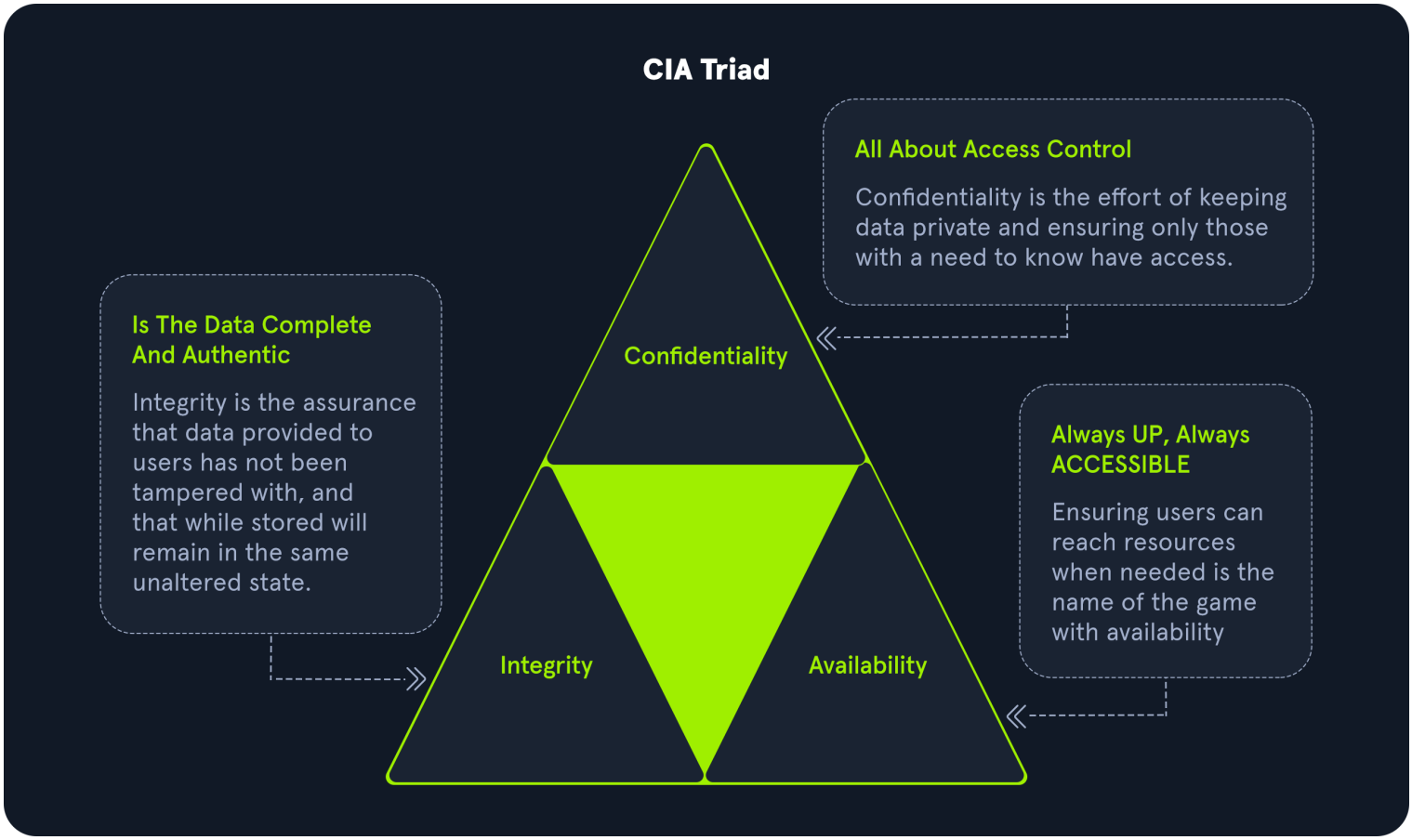
Submit

Hint

Security in Active Directory

As we have progressed through this module, we have looked at the many features and functionalities built into Active Directory. All of which are built around the premise of central management and the ability to share information quickly, at will, to a large userbase. Active Directory can be considered insecure by design because of this. A default Active Directory installation will be missing many hardening measures, settings, and tools that can be used to secure an AD implementation. When we think about cybersecurity, one of the first things that come up is the balance between Confidentiality, Integrity, and Availability, also known as the CIA Triad. Finding this balance is hard, and AD leans heavily toward `Availability` and `Confidentiality` at its core.

CIA Triad



We can help balance the scales by utilizing Microsoft's built-in features that can be enabled/tweaked to harden AD against common attacks. The list below is not exhaustive. Many other general security hardening principles must be in place within an organization to ensure a proper `defense-in-depth` approach (having an accurate asset inventory, vulnerability patches, configuration management, endpoint protection, security awareness training, network segmentation, etc.). This section can be considered the bare minimum general AD security best practices that any organization will benefit from. We will deep dive into Active Directory Defense in a later module. Let's dive in and start with a few general hardening measures for AD.

General Active Directory Hardening Measures

The Microsoft Local Administrator Password Solution (LAPS) is used to randomize and rotate local administrator passwords on Windows hosts and prevent lateral movement.

LAPS

Accounts can be set up to have their password rotated on a fixed interval (i.e., 12 hours, 24 hours, etc.). This free tool can be beneficial in reducing the impact of an individual compromised host in an AD environment. Organizations should not rely on tools like this alone. Still, when combined with other hardening measures and security best practices, it can be a very effective tool for local administrator account password management.

Audit Policy Settings (Logging and Monitoring)

Every organization needs to have logging and monitoring setup to detect and react to unexpected changes or activities that may indicate an attack. Effective logging and monitoring can be used to detect an attacker or unauthorized employee adding a user or computer, modifying an object in AD, changing an account password, accessing a system in an unauthorized or non-standard manner, performing an attack such as password spraying, or more advanced attacks such as modern Kerberos attacks.

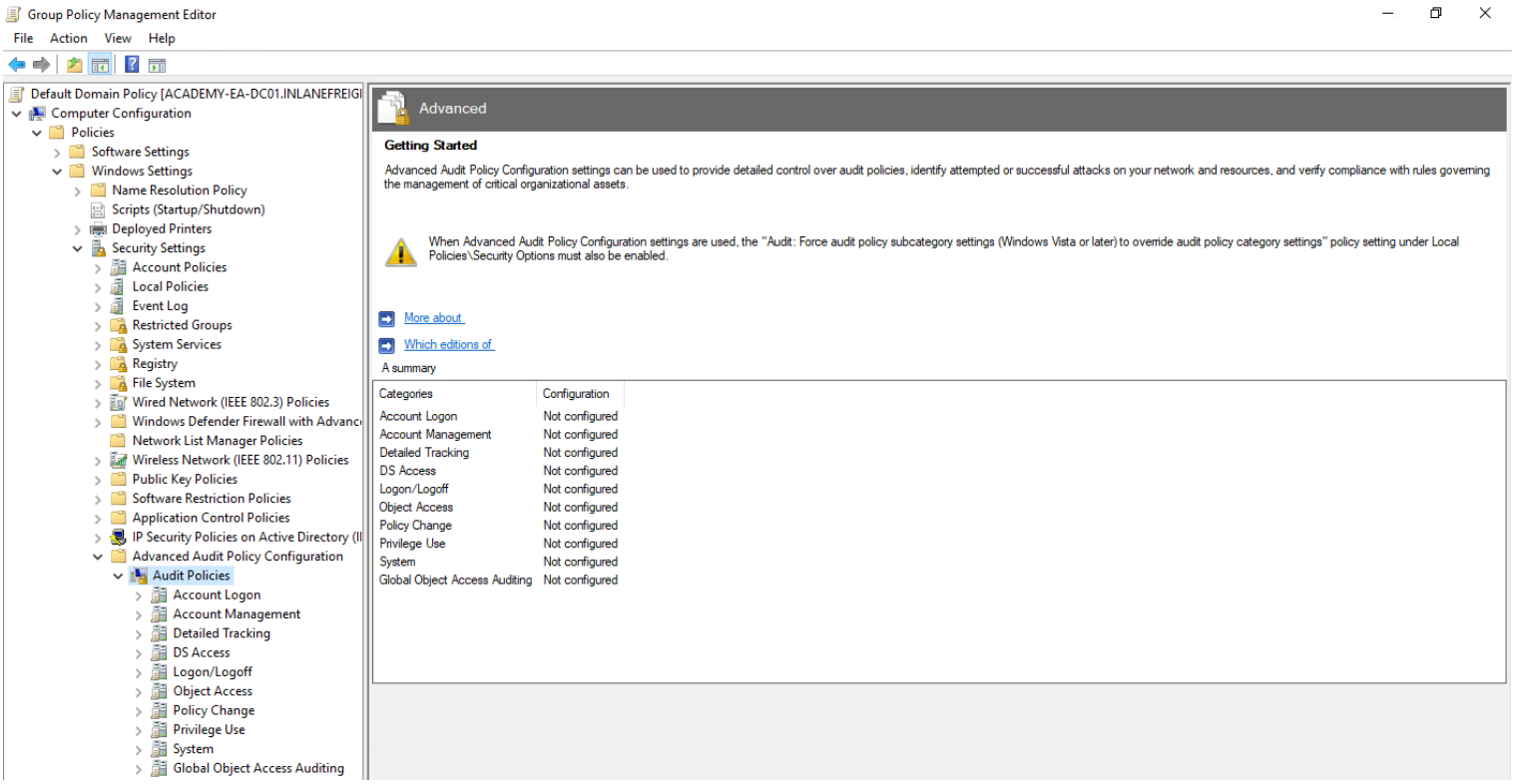
Group Policy Security Settings

As mentioned earlier in the module, Group Policy Objects (GPOs) are virtual collections of policy settings that can be applied to specific users, groups, and computers at the OU level. These can be used to apply a wide variety of security policies to help harden Active Directory. The following is a non-exhaustive list of the types of security policies that can be applied:

- Account Policies - Manage how user accounts interact with the domain. These include the password policy, account lockout policy, and Kerberos-related settings such as the lifetime of Kerberos tickets
- Local Policies - These apply to a specific computer and include the security event audit policy, user rights assignments (user privileges on a host), and specific security settings such as the ability to install drivers, whether the administrator and guest accounts are enabled, renaming the guest and administrator accounts, preventing users from installing printers or using removable media, and a variety of network access and network security controls.
- Software Restriction Policies - Settings to control what software can be run on a host.
- Application Control Policies - Settings to control which applications can be run by certain users/groups. This may include blocking certain users from running all executables, Windows Installer files, scripts, etc. Administrators use AppLocker to restrict access to certain types of applications and files. It is not uncommon to see organizations block access to CMD and PowerShell (among other executables) for users that do not require them for their day-to-day job. These policies are imperfect and can often be bypassed but necessary for a defense-in-depth strategy.

- Advanced Audit Policy Configuration - A variety of settings that can be adjusted to audit activities such as file access or modification, account logon/logoff, policy changes, privilege usage, and more.

Advanced Audit Policy



Update Management (SCCM/WSUS)

Proper patch management is critical for any organization, especially those running Windows/Active Directory systems. The Windows Server Update Service (WSUS) can be installed as a role on a Windows Server and can be used to minimize the manual task of patching Windows systems. `System Center Configuration Manager` (SCCM) is a paid solution that relies on the WSUS Windows Server role being installed and offers more features than WSUS on its own. A patch management solution can help ensure timely deployment of patches and maximize coverage, making sure that no hosts miss critical security patches. If an organization relies on a manual method for applying patches, it could take a very long time depending on the size of the environment and also could result in systems being missed and left vulnerable.

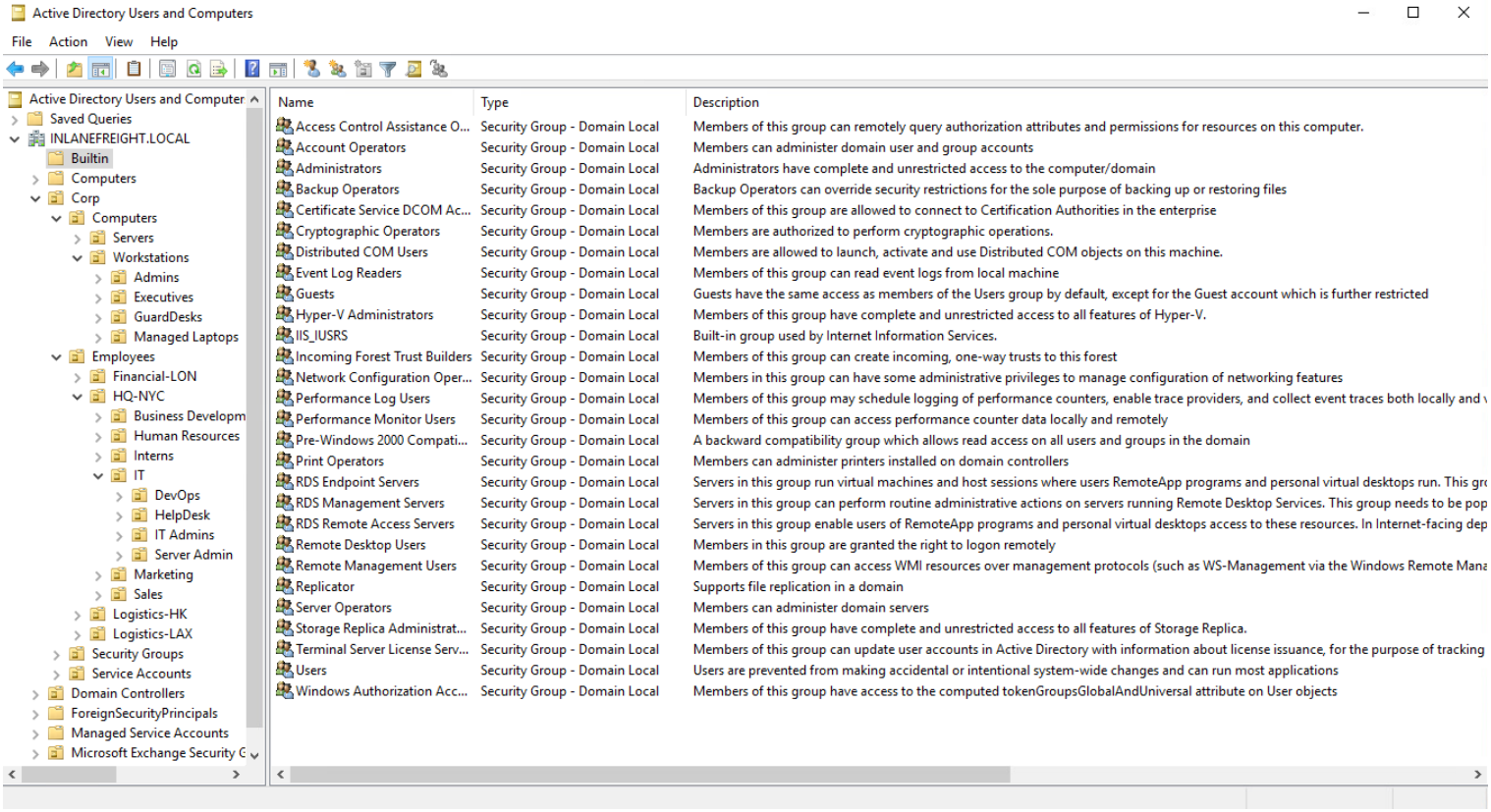
Group Managed Service Accounts (gMSA)

A gMSA is an account managed by the domain that offers a higher level of security than other types of service accounts for use with non-interactive applications, services, processes, and tasks that are run automatically but require credentials to run. They provide automatic password management with a 120 character password generated by the domain controller. The password is changed at a regular interval and does not need to be known by any user. It allows for credentials to be used across multiple hosts.

Security Groups

Security groups offer an easy way to assign access to network resources. They can be used to assign specific rights to the group (instead of directly to the user) to determine what members of the group can do within the AD environment. Active Directory automatically creates some default security groups during installation. Some examples are Account Operators, Administrators, Backup Operators, Domain Admins, and Domain Users. These groups can also be used to assign permission to access resources (i.e., a file share, folder, printer, or a document). Security groups help ensure you can assign granular permissions to users en masse instead of individually managing each user.

Built-in AD Security Groups



Account Separation

Administrators must have two separate accounts. One for their day-to-day work and a second for any administrative tasks they must perform. For example, a user could log into their machine using their `sjones` account to send/receive an email, create documents, etc. They should have a separate account, such as `sjones_adm`, to access a secure administrative host used to perform administrative tasks. This can help ensure that if a user's host is compromised (through a phishing attack, for example), the attacker would be limited to that host and would not obtain credentials for a highly privileged user with considerable access within the domain. It is also essential for the individual to use different passwords for each account to mitigate the risk of password reuse attacks if their non-admin account is compromised.

Password Complexity Policies + Passphrases + 2FA

Ideally, an organization should be using passphrases or large randomly generated passwords using an enterprise password manager. The standard 7-8 character passwords can be cracked offline using a tool such as Hashcat very quickly with a GPU password cracking rig. Shorter, less complex passwords may also be guessed through a password spraying attack, giving an attacker a foothold in the domain. Password complexity rules alone in AD are not enough to ensure strong passwords. For example, the password `Welcome1` would meet the standard complexity rules (3 out of 4 of uppercase, lowercase, number, and special character) but would be one of the first passwords I would try in a password spraying attack. An organization should also consider implementing a password filter to disallow passwords containing the months or seasons of the year, the company name, and common words such as `password` and `welcome`. The minimum password length for standard users should be at least 12 characters and ideally longer for administrators/service accounts. Another important security measure is the implementation of multi-factor authentication (MFA) for Remote Desktop Access to any host. This can help to limit lateral movement attempts that may rely on GUI access to a host.

Limiting Domain Admin Account Usage

All-powerful Domain Admin accounts should only be used to log in to Domain Controllers, not personal workstations, jump hosts, web servers, etc. This can significantly reduce the impact of an attack and cut down potential attack paths should a host be compromised. This would ensure that Domain Admin account passwords are not left in memory on hosts throughout the environment.

Periodically Auditing and Removing Stale Users and Objects

It is important for an organization to periodically audit Active Directory and remove or disable any unused accounts. For example, there may be a privileged service account that was created eight years ago with a very weak password that was never changed, and the account is no longer in use. Even if the password policy had since been changed to be more resistant to attacks such as password spraying, an account such as this may be a quick and easy foothold or method for lateral movement or privilege escalation within the domain.

Auditing Permissions and Access

Organizations should also periodically perform access control audits to ensure that users only have the level of access required for their day-to-day work. It is important to audit local admin rights, the number of Domain Admins (do we really need 30 of them?), and Enterprise Admins to limit the attack surface, file share access, user rights (i.e., membership in certain privileged security groups), and more.

Audit Policies & Logging

Visibility into the domain is a must. An organization can achieve this through robust logging and then using rules to detect anomalous activity (such as many failed login attempts that could be indicative of a password spraying attack) or indicators that a Kerberoasting attack is being attempted. These can also be used to detect Active Directory enumeration. It is worth familiarizing ourselves with Microsoft's Audit Policy Recommendations to help detect compromise.

Using Restricted Groups

Restricted Groups allow for administrators to configure group membership via Group Policy. They can be used for a number of reasons, such as controlling membership in the local administrator's group on all hosts in the domain by restricting it to just the local Administrator account and Domain Admins and controlling membership in the highly privileged Enterprise Admins and Schema Admins groups and other key administrative groups.

Limiting Server Roles

It is important not to install additional roles on sensitive hosts, such as installing the `Internet Information Server` (IIS) role on a Domain Controller. This would increase the attack surface of the Domain Controller, and this type of role should be installed on a separate standalone web server. Some other examples would be not hosting web applications on an Exchange mail server and separating web servers and database servers out to different hosts. This type of role separation can help to reduce the impact of a successful attack.

Limiting Local Admin and RDP Rights

Organizations should tightly control which users have local admin rights on which computers. As stated above, this can be achieved using Restricted Groups. I have seen too many organizations with the entire Domain Users group with local admin rights on one or more hosts. This would allow an attacker that compromises `ANY` account (even a very low privileged one) to access that host as a local admin and potentially obtain sensitive data or steal high privileged domain account credentials from memory if another user is logged in. The same goes for Remote Desktop (RDP) rights. If many users can RDP to one or many machines, this increases the risk of sensitive data exposure or potential privilege escalation attacks, leading to further compromise.

This link provides further reading on Microsoft's Best Practices for Securing Active Directory.

Questions

Answer the question(s) below
to complete this Section and earn cubes!

Cheat Sheet

+ 0 Confidentiality, <___>, and Availability are the pillars of the CIA Triad. What term is missing? (fill in the blank)

Submit

Hint

+ 0 What security policies can block certain users from running all executables?

Submit

Examining Group Policy

Group Policy is a Windows feature that provides administrators with a wide array of advanced settings that can apply to both user and computer accounts in a Windows environment. Every Windows host has a Local Group Policy editor to manage local settings. For our purposes, we will focus on Group Policy in a domain context for managing users and computers in Active Directory. Group Policy is a powerful tool for managing and configuring user settings, operating systems, and applications. Group Policy is also a potent tool for managing security in a domain environment. From a security context, leveraging Group Policy is one of the best ways to widely affect your enterprise's security posture. Active Directory is by no means secure "out of the box," and Group Policy, when used properly, is a crucial part of a defense-in-depth strategy.

While Group Policy is an excellent tool for managing the security of a domain, it can also be abused by attackers. Gaining rights over a Group Policy Object could lead to lateral movement, privilege escalation, and even full domain compromise if the attacker can leverage them in a way to take over a high-value user or computer. They can also be used as a way for an attacker to maintain persistence within a network. Understanding how Group Policy works will give us a leg up against attackers and can help us greatly on penetration tests, sometimes finding nuanced misconfigurations that other penetration testers may miss.

Group Policy Objects (GPOs)

A Group Policy Object (GPO) is a virtual collection of policy settings that can be applied to `user(s)` or `computer(s)`. GPOs include policies such as screen lock timeout, disabling USB ports, enforcing a custom domain password policy, installing software, managing applications, customizing remote access settings, and much more. Every GPO has a unique name and is assigned a unique identifier (a GUID). They can be linked to a specific OU, domain, or site. A single GPO can be linked to multiple containers, and any container can have multiple GPOs applied to it. They can be applied to individual users, hosts, or groups by being applied directly to an OU. Every GPO contains one or more Group Policy settings that may apply at the local machine level or within the Active Directory context.

Example GPOs

Some examples of things we can do with GPOs may include:

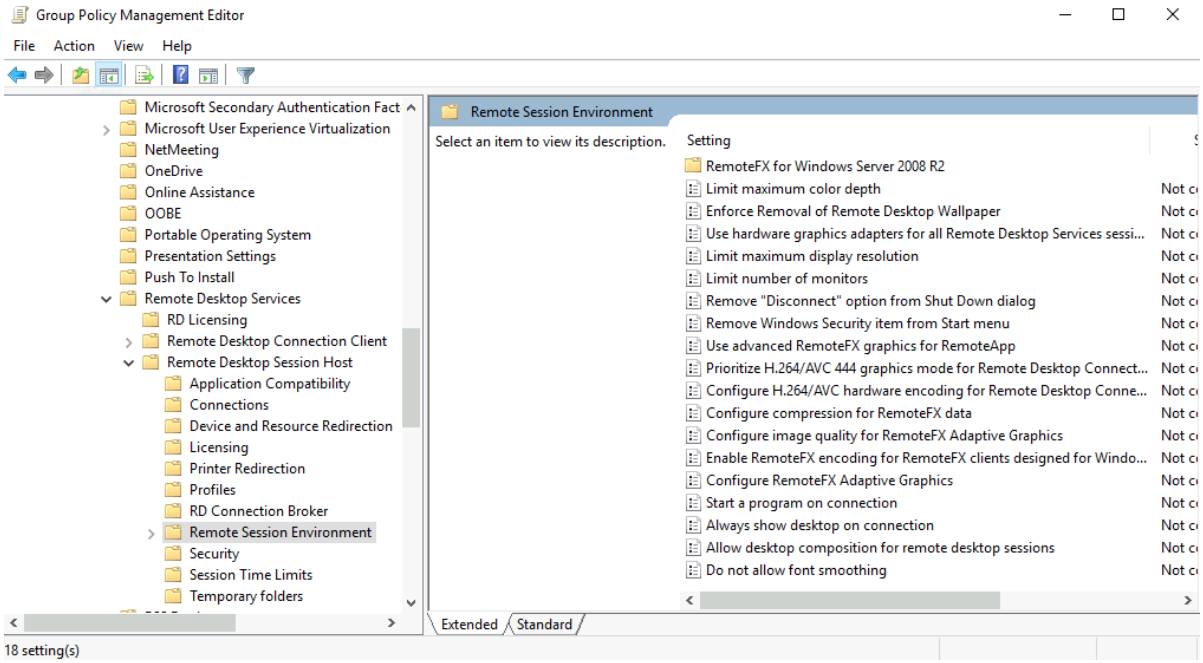
- Establishing different password policies for service accounts, admin accounts, and standard user accounts using separate GPOs
- Preventing the use of removable media devices (such as USB devices)
- Enforcing a screensaver with a password
- Restricting access to applications that a standard user may not need, such as cmd.exe and PowerShell
- Enforcing audit and logging policies
- Blocking users from running certain types of programs and scripts
- Deploying software across a domain
- Blocking users from installing unapproved software
- Displaying a logon banner whenever a user logs into a system
- Disallowing LM hash usage in the domain
- Running scripts when computers start/shutdown or when a user logs in/out of their machine

Let's use as example a default Windows Server 2008 Active Directory implementation, password complexity is enforced by default. The password complexity requirements are as follows:

- Passwords must be at least 7 characters long.
- Passwords must contain characters from at least three of the following four categories:
  - Uppercase characters (A-Z)
  - Lowercase characters (a-z)
  - Numbers (0-9)
  - Special characters (e.g. !@#$%^&*()_+|~-=`{}[]:";'<>?,./)

These are just a few examples of what can be done with Group Policy. There are hundreds of settings that can be applied within a GPO, which can get extremely granular. For example, below are some options that we can set for Remote Desktop sessions.

RDP GPO Settings



GPO settings are processed using the hierarchical structure of AD and are applied using the `Order of Precedence` rule as seen in the table below:

Order of Precedence

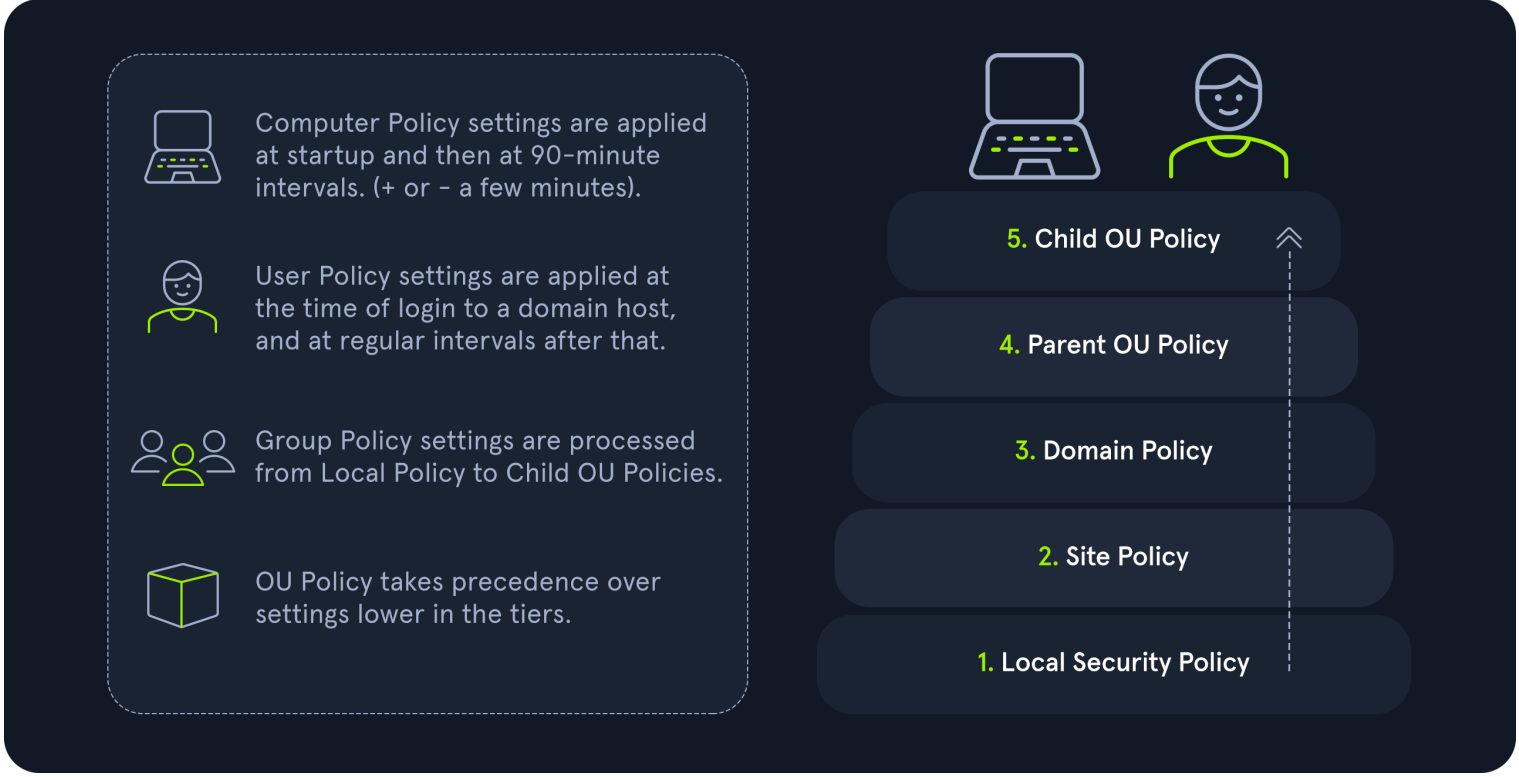
Level	Description
`Local Group Policy`	The policies are defined directly to the host locally outside the domain. Any setting here will be overwritten if a similar setting is defined at a higher level.
`Site Policy`	Any policies specific to the Enterprise Site that the host resides in. Remember that enterprise environments can span large campuses and even across countries. So it stands to reason that a site might have its own policies to follow that could differentiate it from the rest of the organization. Access Control policies are a great example of this. Say a specific building or `site` performs secret or restricted research and requires a higher level of authentication for access to resources. You could specify those settings at the site level and ensure they are linked so as not to be overwritten by domain policy. This is also a great way to perform actions like printer and share mapping for users in specific sites.
`Domain-wide Policy`	Any settings you wish to have applied across the domain as a whole. For example, setting the password policy complexity level, configuring a Desktop background for all users, and setting a Notice of Use and Consent to Monitor banner at the login screen.
`Organizational Unit` (OU)	These settings would affect users and computers who belong to specific OUs. You would want to place any unique settings here that are role-specific. For example, the mapping of a particular share drive that can only be accessed by HR, access to specific resources like printers, or the ability for IT admins to utilize PowerShell and command-prompt.
`Any OU Policies nested within other OU's`	Settings at this level would reflect special permissions for objects within nested OUs. For example, providing Security Analysts a specific set of Applocker policy settings that differ from the standard IT Applocker settings.

We can manage Group Policy from the Group Policy Management Console (found under Administrative Tools in the Start Menu on a domain controller), custom applications, or using the PowerShell GroupPolicy module via command line. The Default Domain Policy is the default GPO that is automatically created and linked to the domain. It has the highest precedence of all GPOs and is applied by default to all users and computers. Generally, it is best practice to use this default GPO to manage default settings that will apply domain-wide. The Default Domain Controllers policy is also created automatically with a domain and sets baseline security and auditing settings for all domain controllers in a given domain. It can be customized as needed, like any GPO.

GPO Order of Precedence

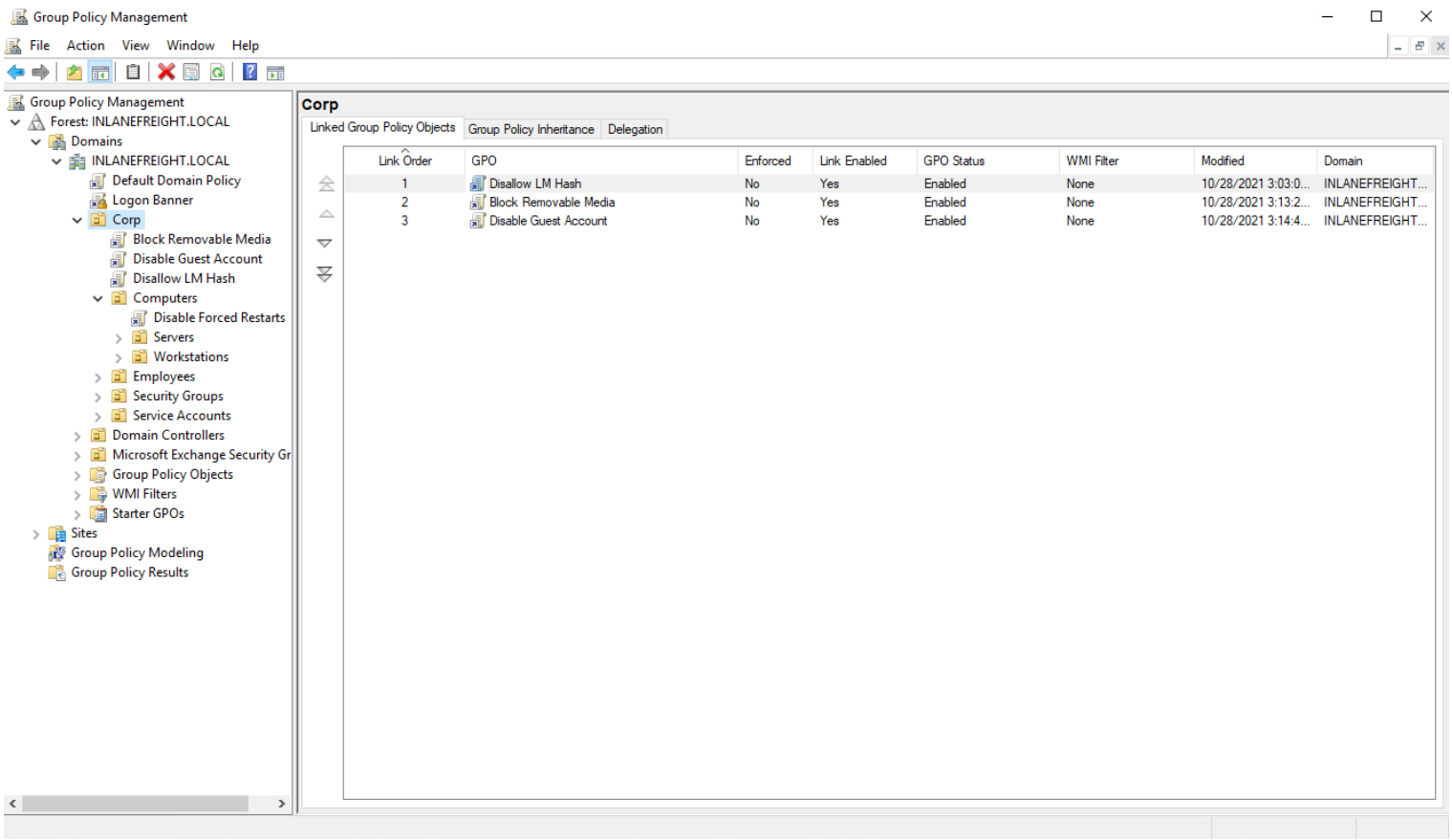
GPOs are processed from the top down when viewing them from a domain organizational standpoint. A GPO linked to an OU at the highest level in an Active Directory network (at the domain level, for example) would be processed first, followed by those linked to a child OU, etc. This means that a GPO linked directly to an OU containing user or computer objects is processed last. In other words, a GPO attached to a specific OU would have precedence over a GPO attached at the domain level because it will be processed last and could run the risk of overriding settings in a GPO higher up in the domain hierarchy. One more thing to keep track of with precedence is that a setting configured in Computer policy will always have a higher priority of the same setting applied to a user. The following graphic illustrates precedence and how it is applied.

GPO Precedence Order



Let's look at another example using the Group Policy Management Console on a Domain Controller. In this image, we see several GPOs. The `Disabled Forced Restarts` GPO will have precedence over the `Logon Banner` GPO since it would be processed last. Any settings configured in the `Disabled Forced Restarts` GPO could potentially override settings in any GPOs higher up in the hierarchy (including those linked to the `Corp` OU).

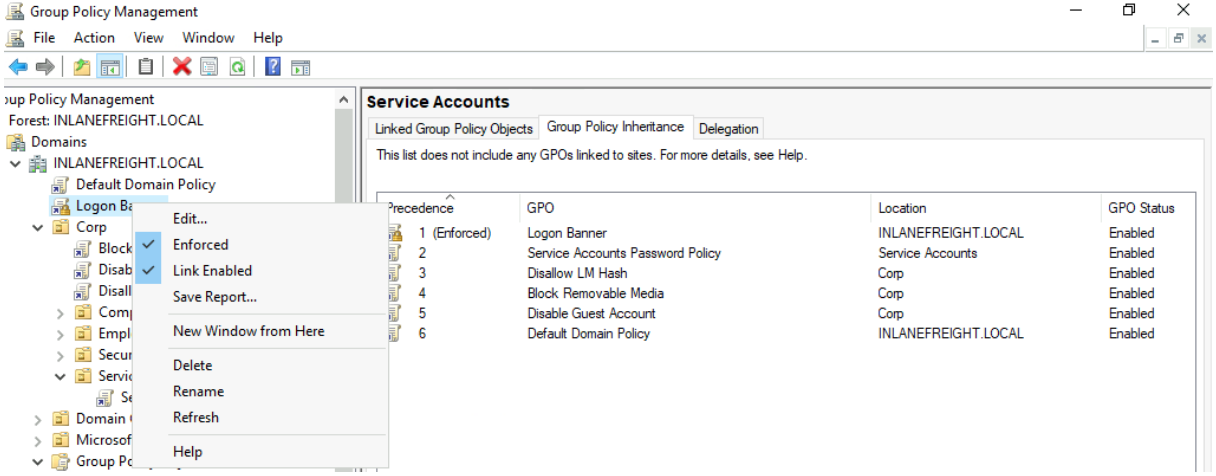
GPMC Hive Example



This image also shows an example of several GPOs being linked to the `Corp` OU. When more than one GPO is linked to an OU, they are processed based on the `Link Order`. The GPO with the lowest Link Order is processed last, or the GPO with link order 1 has the highest precedence, then 2, and 3, and so on. So in our example above, the `Disallow LM Hash` GPO will have precedence over the `Block Removable Media` and `Disable Guest Account` GPOs, meaning it will be processed first.

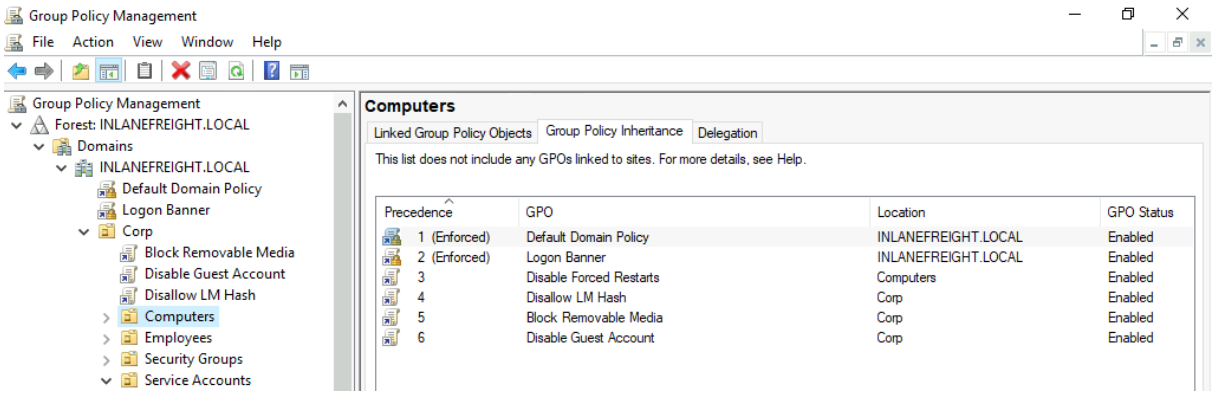
It is possible to specify the `Enforced` option to enforce settings in a specific GPO. If this option is set, policy settings in GPOs linked to lower OUs `CANNOT` override the settings. If a GPO is set at the domain level with the `Enforced` option selected, the settings contained in that GPO will be applied to all OUs in the domain and cannot be overridden by lower-level OU policies. In the past, this setting was called `No Override` and was set on the container in question under Active Directory Users and Computers. Below we can see an example of an `Enforced` GPO, where the `Logon Banner` GPO is taking precedence over GPOs linked to lower OUs and therefore will not be overridden.

Enforced GPO Policy Precedence

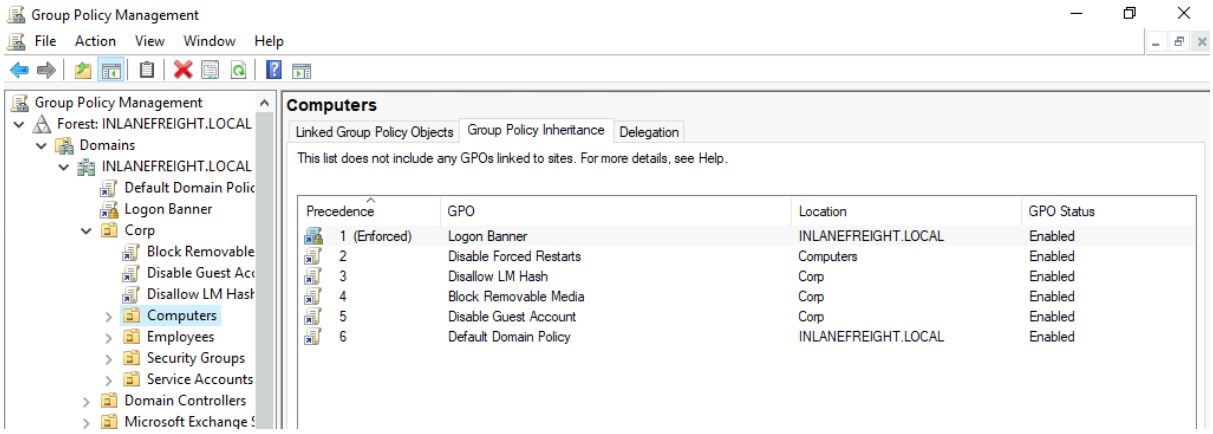


Regardless of which GPO is set to enforced, if the `Default Domain Policy` GPO is enforced, it will take precedence over all GPOs at all levels.

Default Domain Policy Override

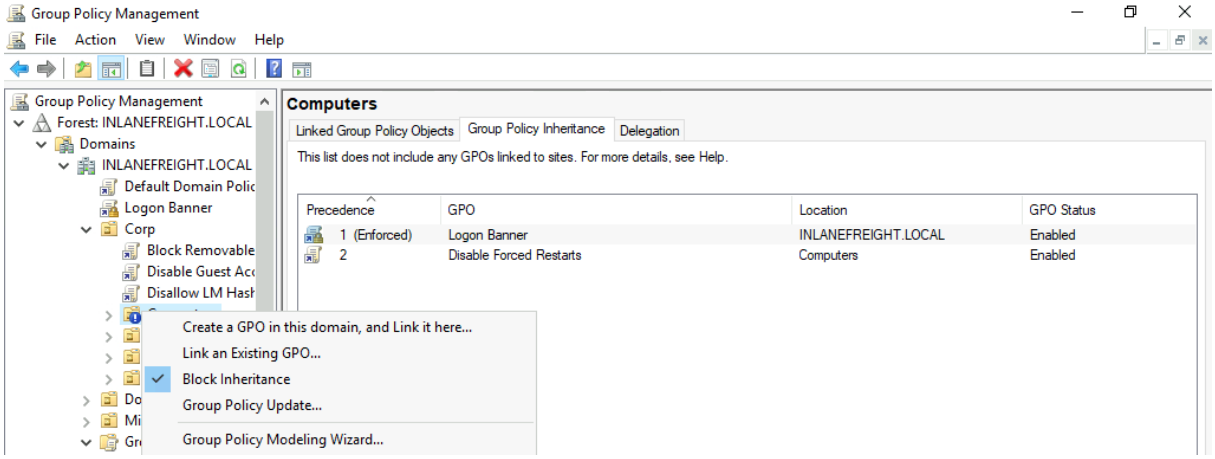


It is also possible to set the `Block inheritance` option on an OU. If this is specified for a particular OU, then policies higher up (such as at the domain level) will NOT be applied to this OU. If both options are set, the `No Override` option has precedence over the `Block inheritance` option. Here is a quick example. The `Computers` OU is inheriting GPOs set on the `Corp` OU in the below image.



If the `Block Inheritance` option is chosen, we can see that the 3 GPOs applied higher up to the `Corp` OU are no longer enforced on the `Computers` OU.

Block Inheritance

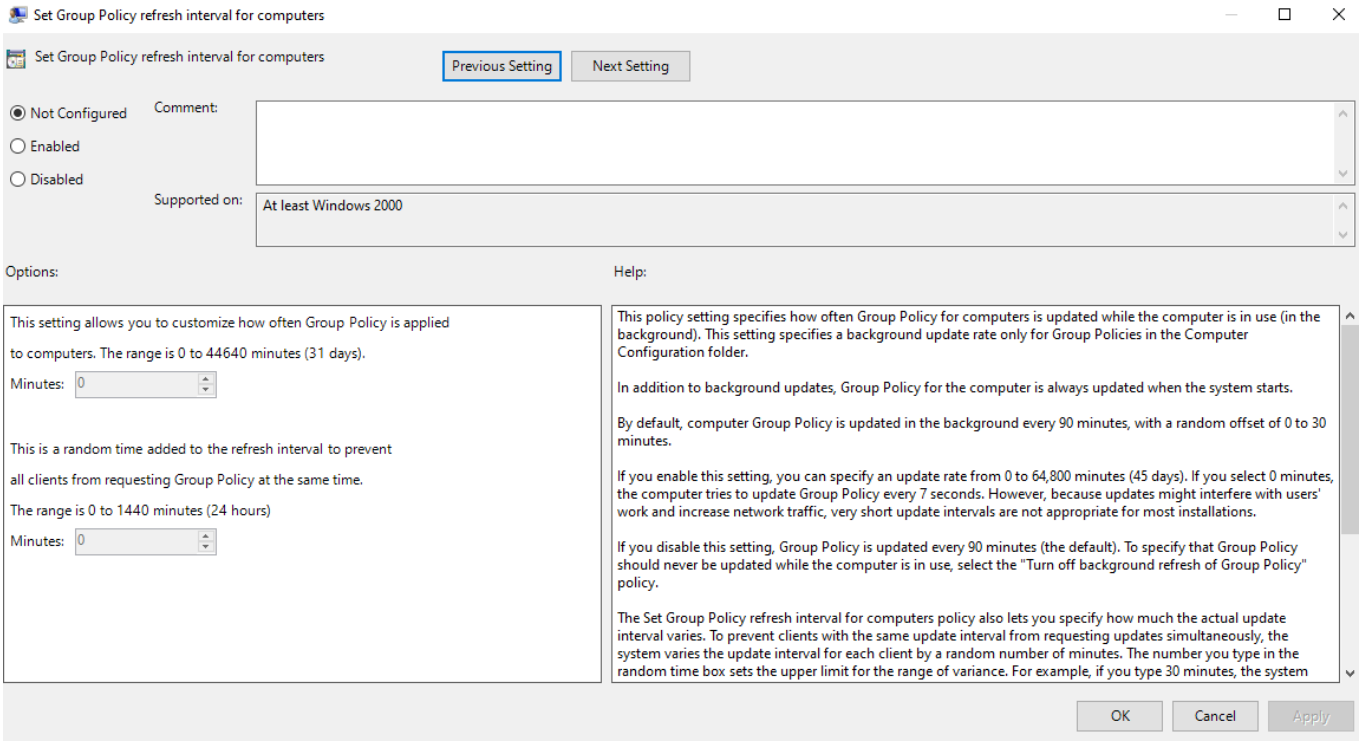


Group Policy Refresh Frequency

When a new GPO is created, the settings are not automatically applied right away. Windows performs periodic Group Policy updates, which by default is done every 90 minutes with a randomized offset of +/- 30 minutes for users and computers. The period is only 5 minutes for domain controllers to update by default. When a new GPO is created and linked, it could take up to 2 hours (120 minutes) until the settings take effect. This random offset of +/- 30 minutes is set to avoid overwhelming domain controllers by having all clients request Group Policy from the domain controller simultaneously.

It is possible to change the default refresh interval within Group Policy itself. Furthermore, we can issue the command `gpupdate /force` to kick off the update process. This command will compare the GPOs currently applied on the machine against the domain controller and either modify or skip them depending on if they have changed since the last automatic update.

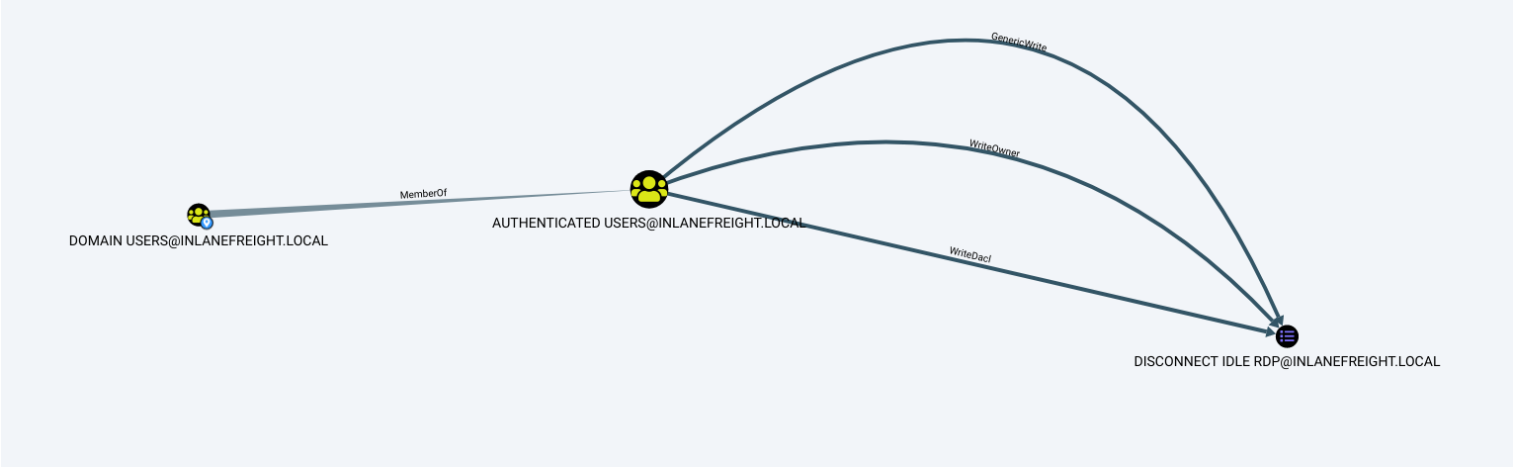
We can modify the refresh interval via Group Policy by clicking on `Computer Configuration --> Policies --> Administrative Templates --> System --> Group Policy` and selecting `Set Group Policy refresh interval for computers`. While it can be changed, it should not be set to occur too often, or it could cause network congestion leading to replication issues.



Security Considerations of GPOs

As mentioned earlier, GPOs can be used to carry out attacks. These attacks may include adding additional rights to a user account that we control, adding a local administrator to a host, or creating an immediate scheduled task to run a malicious command such as modifying group membership, adding a new admin account, establishing a reverse shell connection, or even installing targeted malware throughout a domain. These attacks typically happen when a user has the rights required to modify a GPO that applies to an OU that contains either a user account that we control or a computer.

Below is an example of a GPO attack path identified using the BloodHound tool. This example shows that the `Domain Users` group can modify the `Disconnect Idle RDP` GPO due to nested group membership. In this case, we would next look to see which OUs this GPO applies to and if we can leverage these rights to gain control over a high-value user (administrator or Domain Admin) or computer (server, DC, or critical host) and move laterally to escalate privileges within the domain.



We have covered a lot of information up to this point. `Active Directory` is a vast topic, and we have just scratched the surface. We have covered the foundational theory now; let's get our hands dirty and play around with Active Directory objects, Group Policy, and more in the next section.

Questions

Answer the question(s) below
to complete this Section and earn cubes!

Cheat Sheet

+ 0 Computer settings for Group Policies are gathered and applied at a <___> minute interval? (answer is a number, fill in the blank)

Submit

Hint

+ 0 True or False: A policy applied to a user at the domain level would be overwritten by a policy at the site level.

Submit

Hint

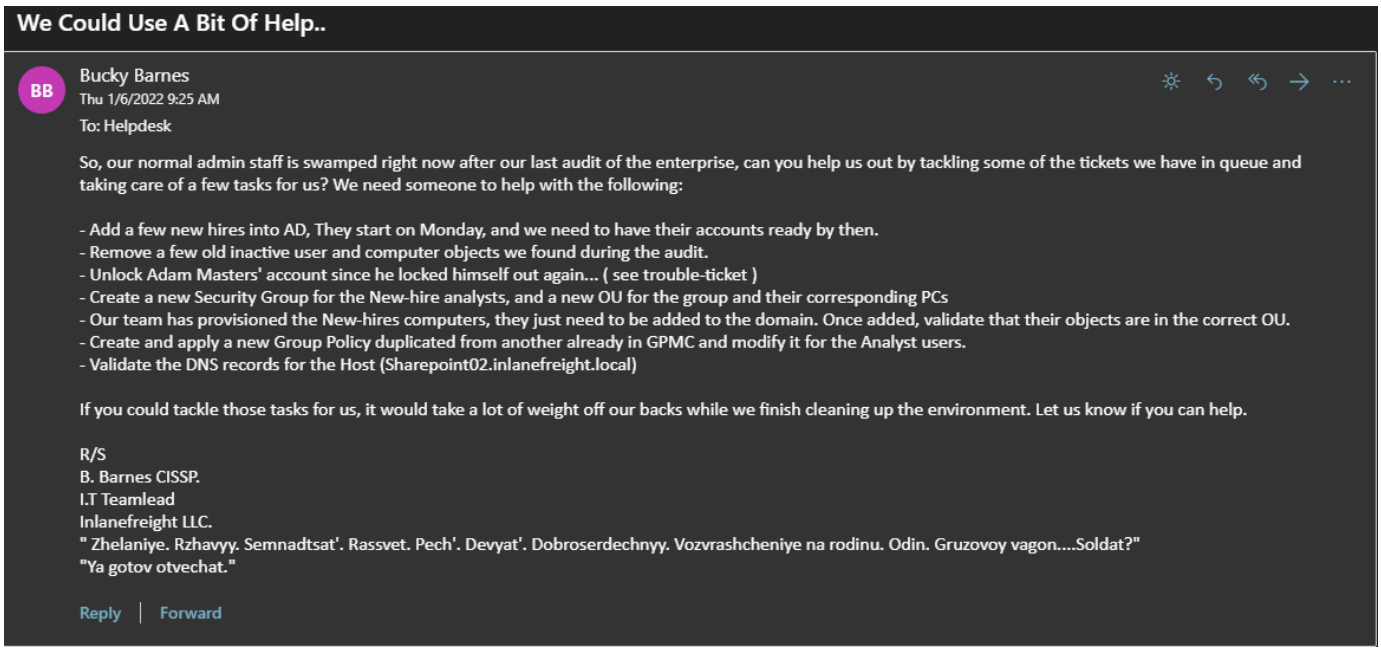
+ 0 What Group Policy Object is created when the domain is created?

Submit

Hint

AD Administration: Guided Lab Part I

Let us consider the following case:



In this section, we will serve as domain administrators to Inlanefreight for a day. We have been tasked to help the IT department close some work orders, so we will be performing actions such as adding and removing users and groups, managing group policy, and more. Successful completion of the tasks can lead to us gaining a promotion to the Tier II IT team from the helpdesk.

Connection Instructions

For this lab, you will have access to a domain-joined Windows server from which you can perform any actions needed to complete the lab. The environment will require you to RDP from Pwnbox or your own VM over VPN to the Windows server. Follow the steps below to utilize `RDP` and connect to the lab's Windows host.

- Click below in the `Questions` section to spawn the target host and obtain an IP address. The image below shows where to spawn the target and acquire a VPN key for the lab if needed.
  - IP ==
  - Username == `htb-student_adm`
  - Password == `Academy_student_DA!`
- We will use `xfreerdp` to connect with the target.
- Open a terminal in pwnbox or from your lab vm over vpn and enter the following command:
  - xfreerdp /v: /u: `htb-student_adm` /p: `Academy_student_DA!`

Once connected, open an MMC console, PowerShell, or the ADDS tools to begin.

Tasks:

Attempt to complete the challenges on your own. If you get stuck, the `Solutions` dropdown below each task can help you. This reference on the Active Directory PowerShell module will be extremely helpful. As an introductory course on AD, we do not expect you to know everything about the topic and how to administer it. The Solutions below each task offer a step-by-step of how to complete the task. This section is provided to give you a taste of the daily tasks that AD administrators perform. Instead of providing the information to you in a static format, we have opted to provide it in a more hands-on manner.

Task 1: Manage Users

Our first task of the day includes adding a few new-hire users into AD. We are just going to create them under the `"inlanefreight.local"` scope, drilling down into the `"Corp > Employees > HQ-NYC > IT "` folder structure for now. Once we create our other groups, we will move them into the new folders. You can utilize the Active Directory PowerShell module (New-ADUser), the Active Directory Users and Computers snap-in, or MMC to perform these actions.

Users to Add:

User
`Andromeda Cepheus`

User
Orion Starchaser
Artemis Callisto

Each user should have the following attributes set, along with their name:

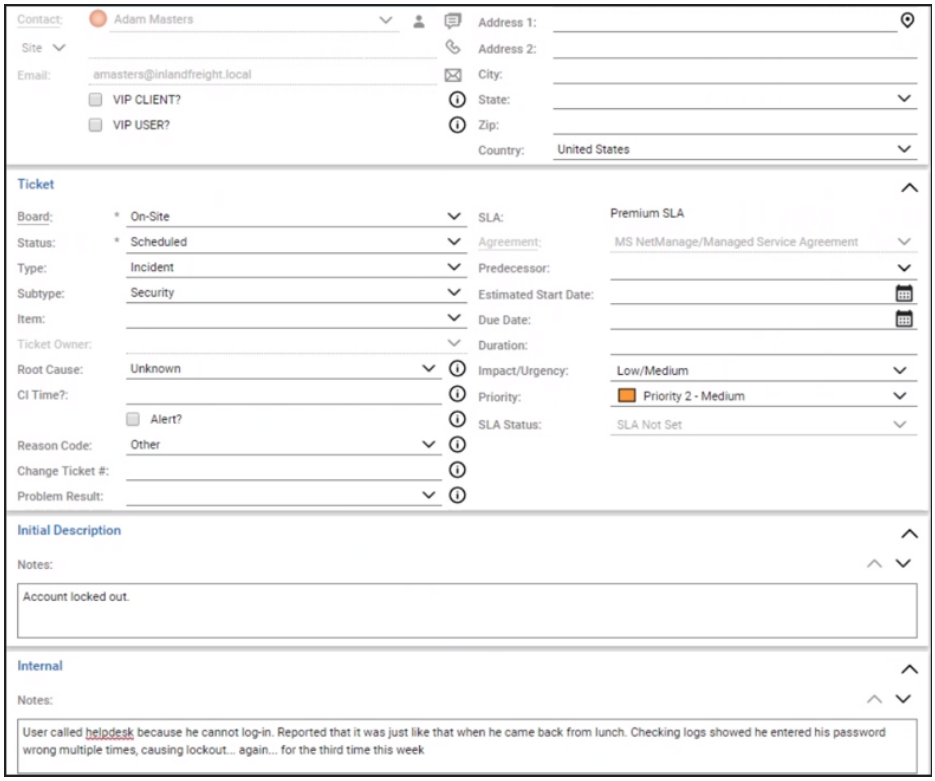
Attribute
full name
email () (ex. )
display name
User must change password at next logon

Once we have added our new hires, take a quick second and remove a few old user accounts found in an audit that are no longer required.

Users to Remove

User
Mike O'Hare
Paul Valencia

Lastly `Adam Masters` has submitted a trouble ticket over the phone saying his account is locked because he typed his password wrong too many times. The helpdesk has verified his identity and that his Cyber awareness training is up to date. The ticket requests that you unlock his user account and force him to change his password at the next login.



`Solution: Task 1`

Open PowerShell as an administrator. To ADD a user into Active Directory, we First need to load the module with the "Import-Module -Name ActiveDirectory" cmdlet. The AD module can be installed via the RSAT feature pack, but for now, it's already installed on the host used in this lab.

PowerShell Terminal Output for Adding a User

```
PS C:\htb> New-ADUser -Name "Orion Starchaser" -Accountpassword (ConvertTo-SecureString -AsPlainText (Read-Host "Enter a secure password") -Force ) -Enabled $true -OtherAttributes @{'title'="Analyst";'mail'="[email protected]"}
```

After you hit enter, a prompt will appear, enter a secure password for the user.

Adding a User from the MMC Snap-in

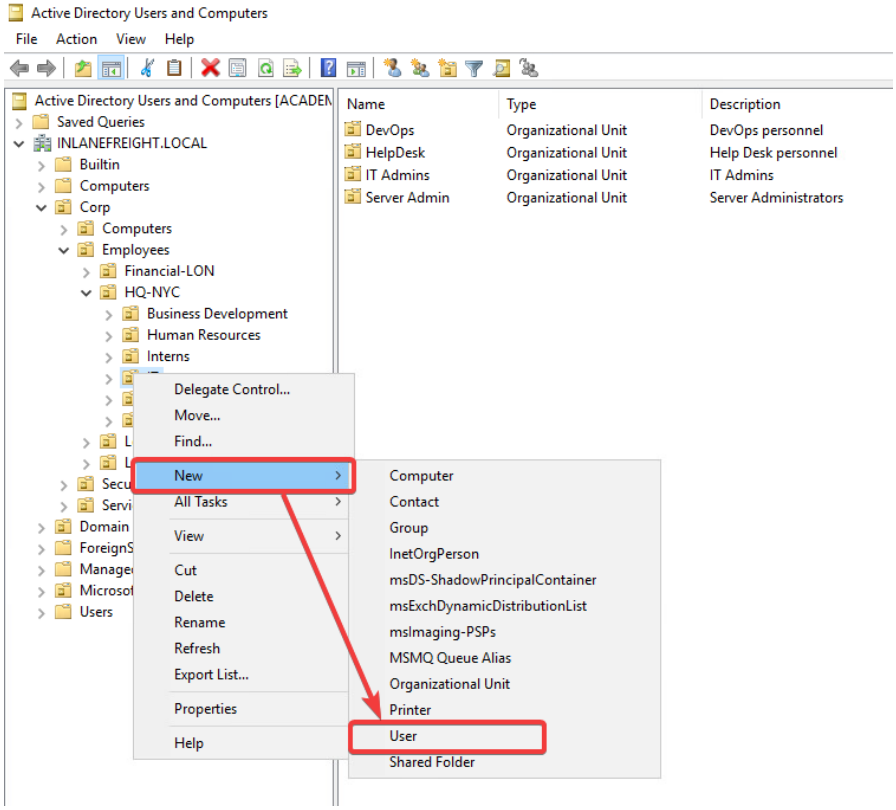
Before adding a user from the GUI we need to open the Active Directory Users and Computers (ADUC) MMC tool. As a standard user, we may have access to view the ADUC objects, but we will not be able to modify or add. We need to log in as our administrator account (credentials above) to complete these actions. Once logged in, open the ADUC snap-in by performing the following actions:

- From the Server Manager window, select Tools > then ADUC
- Expand the scope "inlanefreight.local" and drill down to "Corp > Employees > HQ-NYC > IT ". This is where we will be creating our new users, OU's, and Groups.

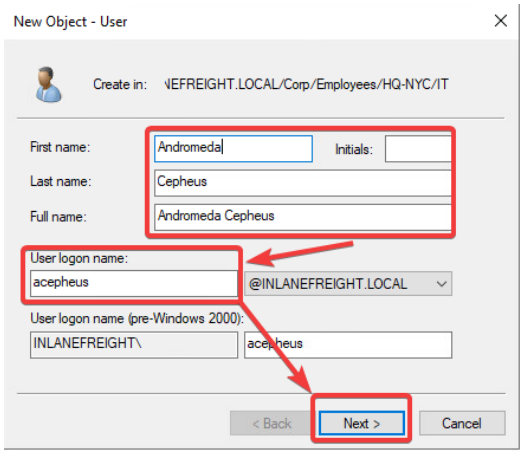
W3.CSS

Adding an AD User via the GUI

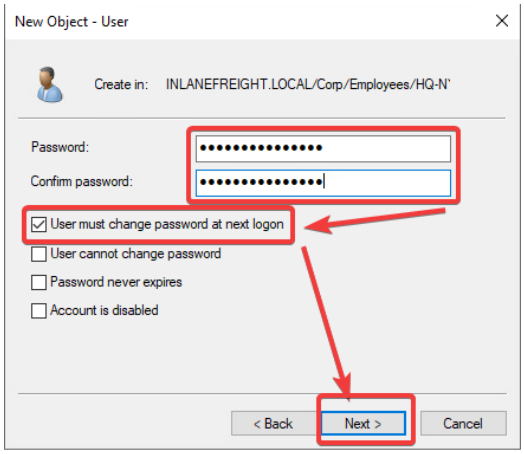
To add an AD user via the GUI we first need to open Active Directory Users and Computers via the Start Menu folder Administrative Tools.



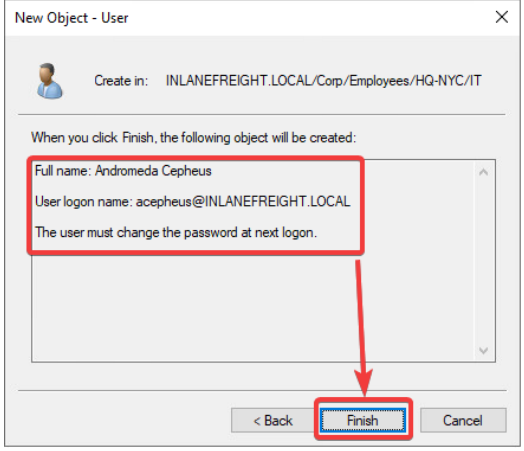
1. Right click on "IT", Select "New" > "User".



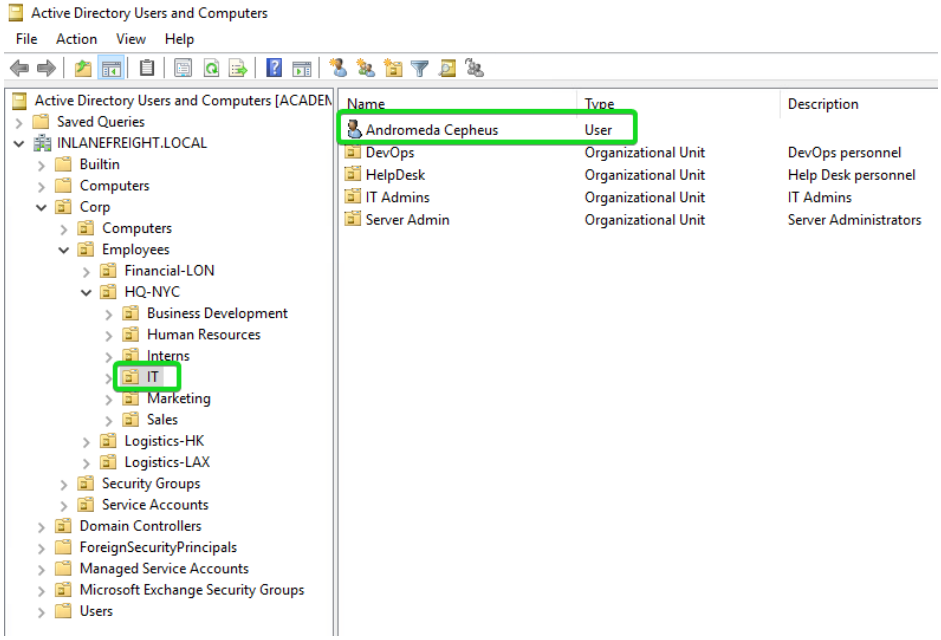
2. Add the users First and Last name, set the "User Logon Name:" as `acepheus` and then hit Next.



3. Set a password of `NewP@ssw0rd123!`and check the box for "User must change password at next login".



4. If all attributes look correct, select "Finish" in the last window



5. Our new User exists now in the OU.

❮❯

Add A User

We will add the new user `Andromeda Cepheus` to our domain. We can do so by:

- Right-click on "IT" > Select "New" > "User". A popup window will appear with a field for you to fill in.
- Add the user's First and Last name, set the "User Logon Name:" as `acepheus`, and then hit Next.
- Now supply the new user with a password of `NewP@ssw0rd123!`, confirm the password again, and check the box for " User must change password at next login", then hit next. Select "Finish" in the last window if all attributes look correct.

To `REMOVE` a user account from Active Directory, we can:

PowerShell to Remove a User

```
PS C:\htb> Remove-ADUser -Identity pvalencia
```

The `Remove-ADUser` cmdlet above targets the user by its user logon name. Ensure you are targeting the right user before executing it. If we are unsure of the value needed, we can use the `Get-ADUser` command to validate first.

Remove a User from the MMC Snap-in

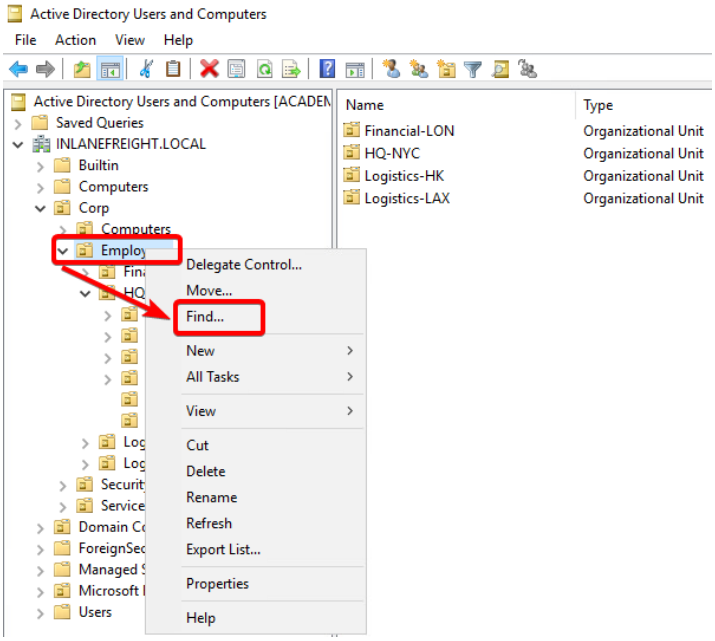
Now we will remove a user `Paul Valencia` from our domain.
We can do so by:

- The most straightforward method from the ADUC snap-in will be to use the `find` functionality. Inlanefreight has many users across several OU's. To use find:
  - Right-click on `Employees` and select "find".
  - Type in the username you wish to search for, in this case, "Paul Valencia" and hit "Find Now." If a user has that name, the search results will appear lower in the find window.
- Now, right-click on the user and select delete. A popup window will appear to confirm the deletion of the user. Hit yes.
- To validate the user is deleted, you can use the `Find` feature again to search for the user.

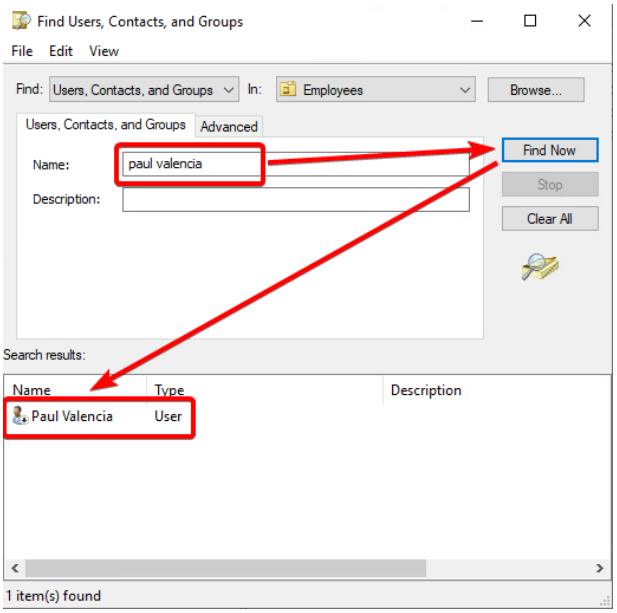
W3.CSS

Deleting a User via the GUI

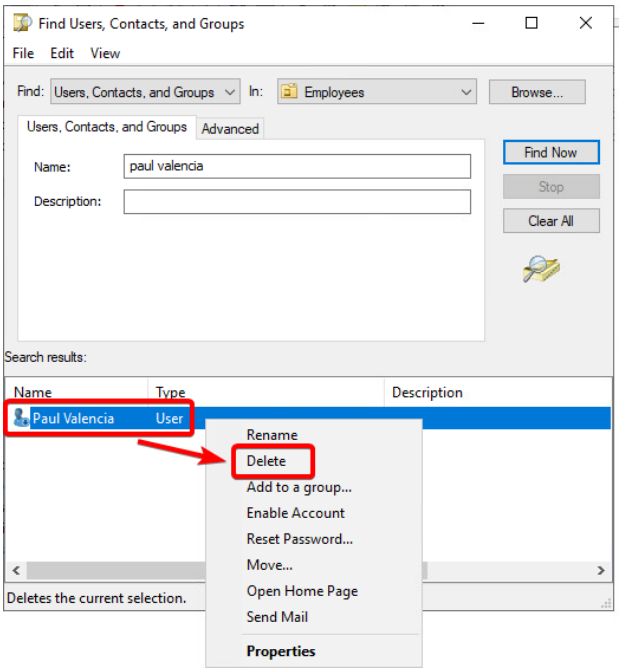
To delete a user via the GUI, we will use the ADUC snap-in just like when we added a user to the domain above.



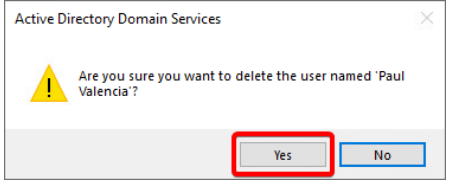
1. Right click on the "Employees OU" and select "find".



2. Type in the username you wish to search for, in this case "Paul Valencia" and hit "Find Now".



3. Right click on the user and select delete.



4. Confirm the deletion in the pop-up window. Find can be utilized again to determine if the user is gone.

❮❯

Now we need to help `Adam Masters` out and unlock his account again.

To `UNLOCK` a user account we can:

PowerShell To Unlock a User

```
PS C:\htb> Unlock-ADAccount -Identity amasters
```

We also need to set a new password for the user and force them to change the password at the next logon. We will do this with the `SetADAccountPassword` and `Set-ADUser` cmdlets.

Reset User Password (Set-ADAccountPassword)

```
PS C:\htb> Set-ADAccountPassword -Identity 'amasters' -Reset -NewPassword (ConvertTo-SecureString -AsPlainText "NewP@ssw0rdReset!" -Force)
```

Force Password Change (Set-ADUser)

```
PS C:\htb> Set-ADUser -Identity amasters -ChangePasswordAtLogon $true
```

Unlock from Snap-in

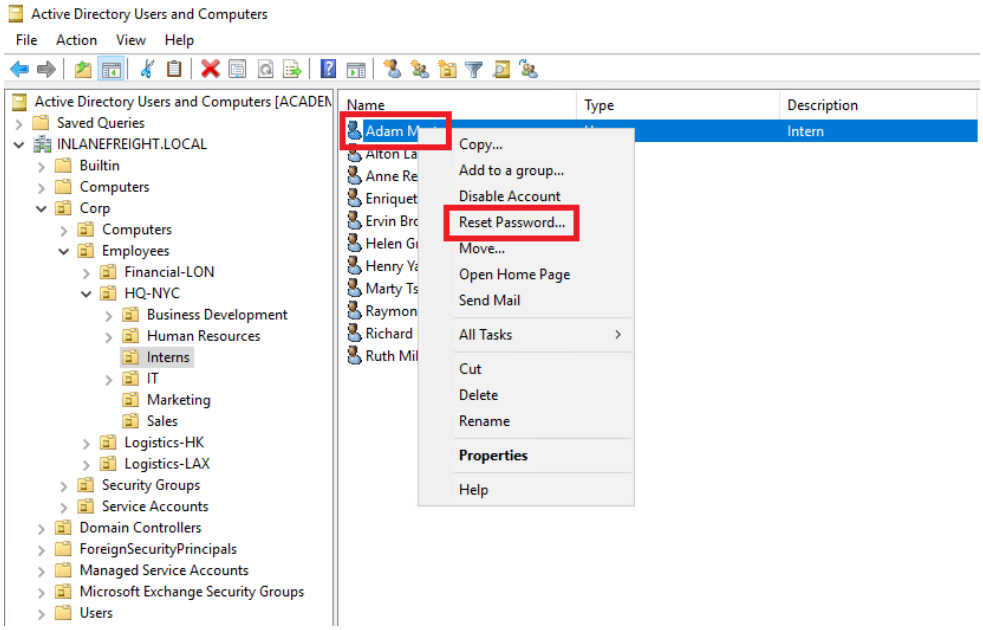
Unlocking this user account will take several steps. The first is to unlock the account, then we set it so that the user must change his password at the next login, and then we reset his password to a temporary one so that he can log in and reset it himself. We can do so by:

- right-click on the user and select `Reset Password`.
- In the next window, type in the temporary password, confirm it, and check the boxes for "User must change password at next logon" and "Unlock the user's account."
- Once done, hit OK to apply changes. If no error occurs, you will get a prompt informing you that the user's password was changed.

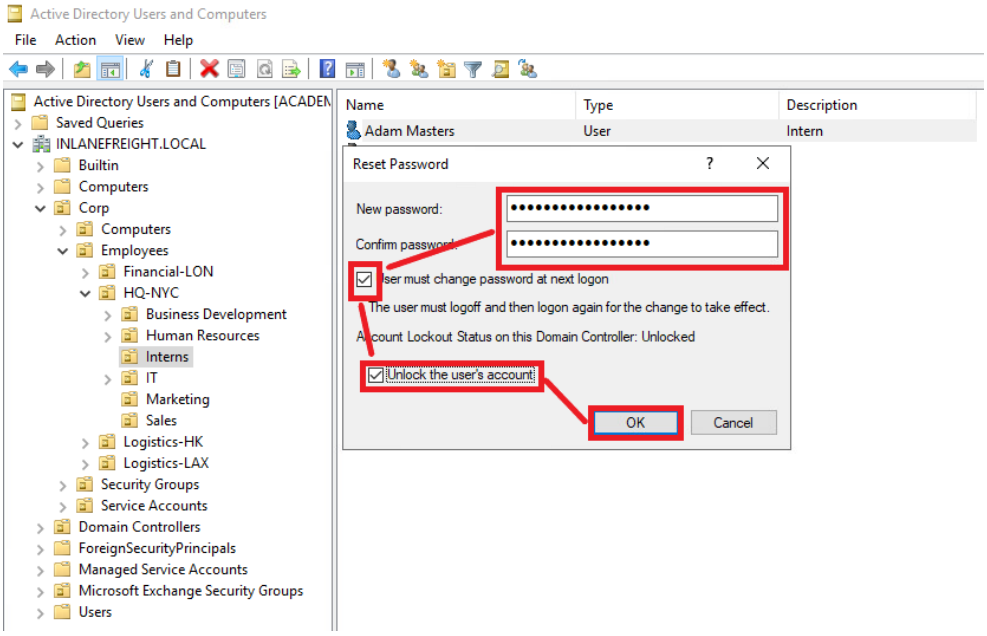
W3.CSS

Unlock Users Account From GUI

To unlock Adam Masters' account, we will use the ADUC snap-in just like when we added a user to the domain above.



1. Right click on Adam Master's account and select "Reset Password".



2. Set a new temporary password and select the "Unlock" and "User must change password" dialog boxes.

❮❯

Task 2: Manage Groups and Other Organizational Units

Next up for us is to create a new Security Group called `Analysts` and then add our new hires into the group. This group should also be nested in an OU named the same under the `IT` hive. The `New-ADOrganizationalUnit` PowerShell command should enable you to quickly add a new security group. We can also utilize the AD Users and Computers snap-in like in Task-1 to complete this task.

`Solution: Task 2`

Create a New AD OU and Security Group from PowerShell

To create a new OU and Group, we can perform the following actions:

```
PS C:\htb> New-ADOrganizationalUnit -Name "Security Analysts" -Path "OU=IT,OU=HQ-NYC,OU=Employees,OU=CORP,DC=INLANEFREIGHT,DC=LOCAL"
```

First, we created the new OU to hold our Analysts and their resources. Next, we need to create a security group for these users.

```
PS C:\htb> New-ADGroup -Name "Security Analysts" -SamAccountName analysts -GroupCategory Security -GroupScope Global -DisplayName "Security Analysts" -Path "OU=Security Analysts,OU=IT,OU=HQ-NYC,OU=Employees,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL" -Description "Members of this group are Security Analysts under the IT OU"
```

From MMC Snap-in

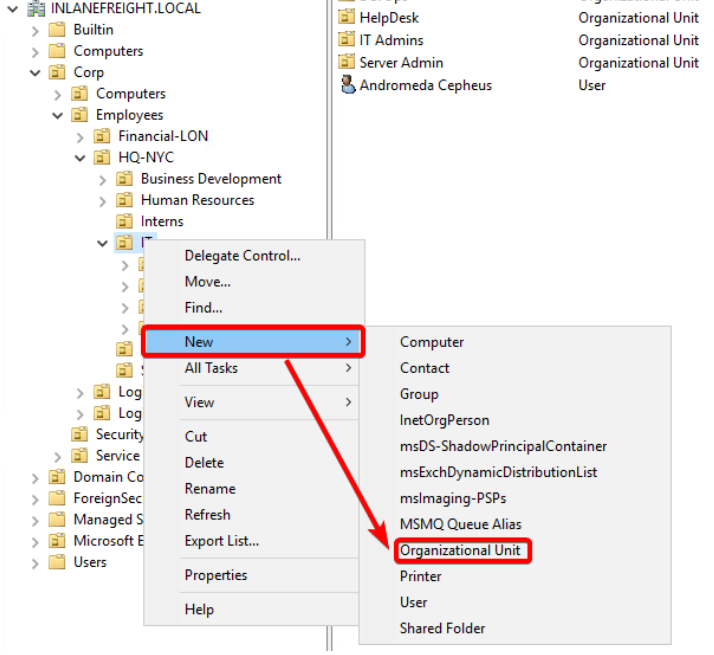
This will be a quick two-step process for us. We first need to create a new OU to host our Security Analysts. To do so, we will :

- navigate to the "Corp > Employees > HQ-NYC > IT "OU. We are going to build out a new container within `IT`.
- Right-click on `IT` and select "New > Organizational Unit". A new window should appear.
  - input the name `Security Analysts` into the Name field and leave the default option set for the Protect checkbox. Hit OK, and the OU should be created.

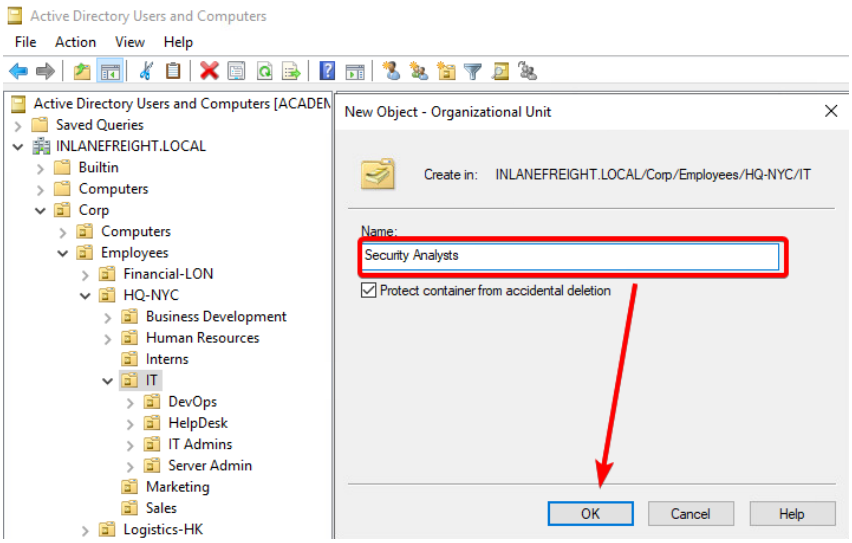
W3.CSS

Create A New OU Under I.T.

Our new OU "Security Analysts" should exist in the IT hive.



1. From within the IT OU, Right click and select "New" > "Organizational Unit"



2. Type in the name for the OU, "Security Analysts" in this instance. Hit OK when done.

❮❯

Now that we have our OU, let's create the `Security Group` for our Analysts.

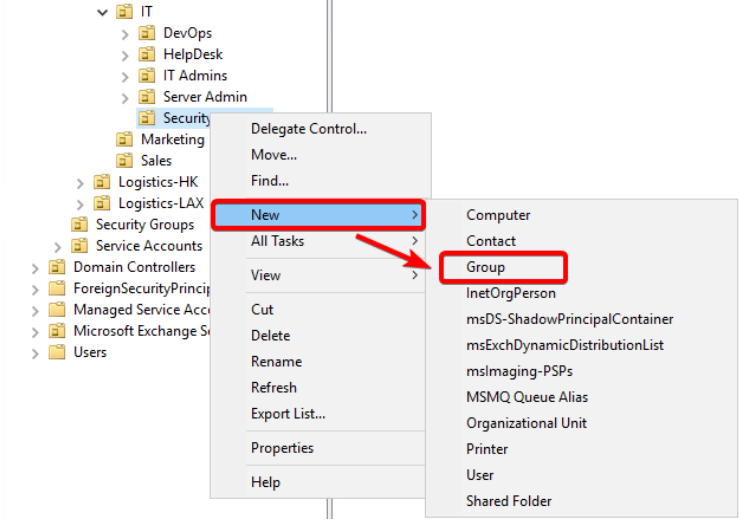
Right-click on our new OU `Security Analysts` and select "New > Group" and a popup window should appear.

- Input the name of the group `Security Analysts`
- Select the Group scope `Domain local`
- ensure group type says `Security` not "Distribution".
- Once you check the options, hit OK.

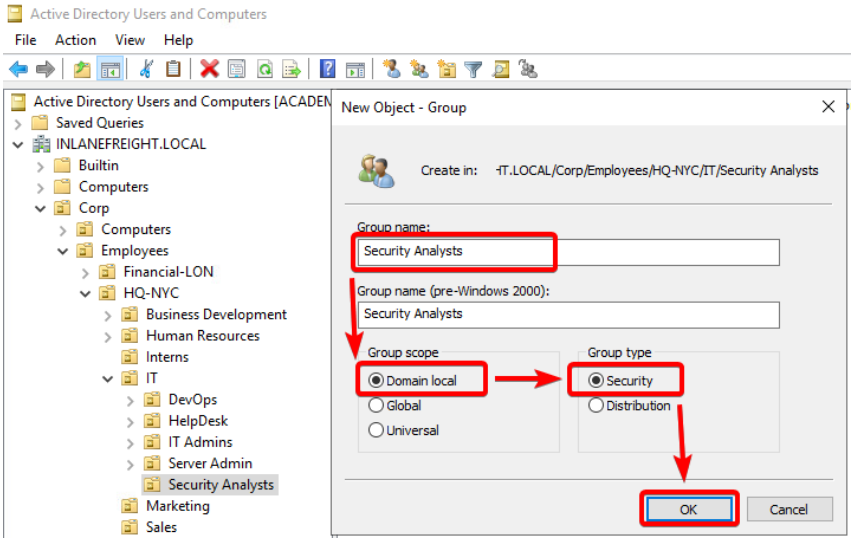
W3.CSS

Creating A Security Group

Our Security Group will go in the OU we just created.



1. right click on our new OU `Security Analysts` and select "New > Group". A popup window should appear.



2. Enter a Name, scope, and type, then hit OK.

❮❯

When done, a new Security Group should exist in our OU. We need to move our new users into the OU and add them to the security group. Keep in mind the purpose of this is to logically organize our AD objects for easy location and administration. Utilizing the security groups, we can quickly assign permissions and resources to specific users instead of managing each user individually.

To ADD a user to a `group`, we can:

Add User to Group via PowerShell

```
PS C:\htb> Add-ADGroupMember -Identity analysts -Members ACepheus,OStarchaser,ACallisto
```

Here we use the `SAMAccountName` of the users to add them to the Analysts group via the `Add-ADGroup Member` Cmdlet. Ensure your list is comma separated without spaces between each.

From MMC Snap-in

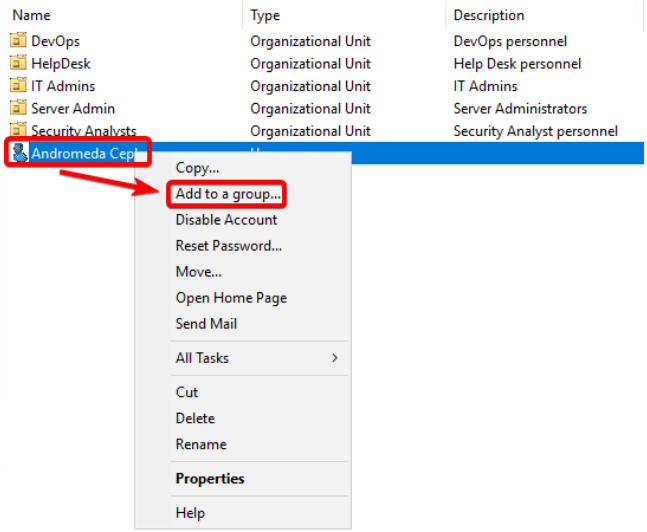
To add the users to the security group, we can:

- Find the user you wish to add
- Right-click on the user and select "Add to a group". A new window will appear for you to specify the group name.
- type in part or all of the group you wish to add the user to. In this case, we are adding Andromeda to the Security Analysts group. If our query matches one or more groups, another dialog box will appear, providing us with a list of groups to choose from. Pick the group you need and hit "OK".
- The choice you selected will now be highlighted in the previous window. More than one group can be selected at a time if necessary. Once done, hit "OK."
- If no issues arise, you will get a new popup informing you that the operation is completed. To validate, we can view the group or user properties.

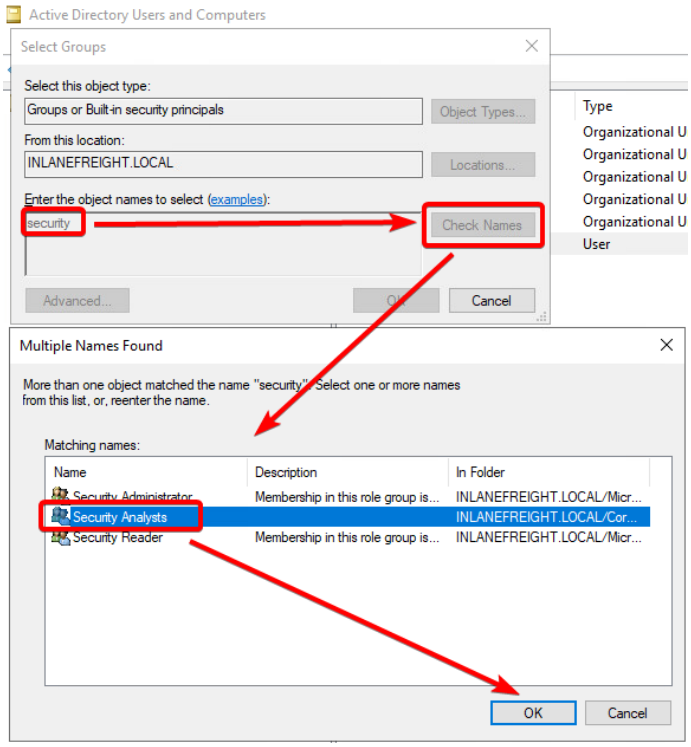
W3.CSS

Add A User To A Security Group

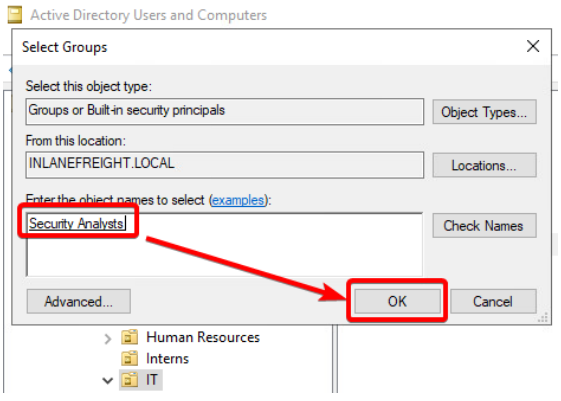
In this example we are adding Andromeda to the Security Analysts group, then moving her into the correct OU.



1. Right click on the user and select "Add to a group"



2. Enter a full or partial group name in the search box and hit "Check Names".



3. Once you have selected the correct group (it will be underlined if correct) hit OK.

❮❯

Thats `two` of our major tasks for the day done. Now let's move on to managing some Group Policy Objects.

Task 3: Manage Group Policy Objects

Next, we have been asked to duplicate the group policy `Logon Banner`, rename it `Security Analysts Control`, and modify it to work for the new Analysts OU. We will need to make the following changes to the Policy Object:

- we will be modifying the Password policy settings for users in this group and expressly allowing users to access PowerShell and CMD since their daily duties require it.
- For computer settings, we need to ensure the Logon Banner is applied and that removable media is blocked from access.

Once done, make sure the Group Policy is applied to the `Security Analysts` OU. This will require the use of the Group Policy Management snap-in found under `Tools` in the Server Manager window. For more of a challenge, the `Copy-GPO` cmdlet in PowerShell can also be utilized.

`Solution: Task 3`

To Duplicate a Group Policy Object we can use the `Copy-GPO` cmdlet or do it from the Group Policy Management Console.

Duplicate the Object via PowerShell

```
PS C:\htb> Copy-GPO -SourceName "Logon Banner" -TargetName "Security Analysts Control"
```

The command above will take `Logon Banner` GPO and copy it to a new object named `Security Analyst Control`. This object will have all the old attributes of the Logon Banner GPO, but it will not be applied to anything until we link it.

Link the New GPO to an OU

```
PS C:\htb> New-GPLink -Name "Security Analysts Control" -Target "ou=Security Analysts,ou=IT,OU=HQ-NYC,OU=Employees,OU=Corp,dc=INLANEFREIGHT,dc=LOCAL" -
LinkEnabled Yes
```

The command above will take the new GPO we created, link it to the OU `Security Analysts`, and enable it. For now, that's all we are going to do from PowerShell. We still need to make a few modifications to the policy, but we will perform these actions from Group Policy Management Console. Editing GPO preferences from PowerShell can be a bit daunting and way beyond the scope of this module.

Modify a GPO via GPMC

To modify our new policy object:

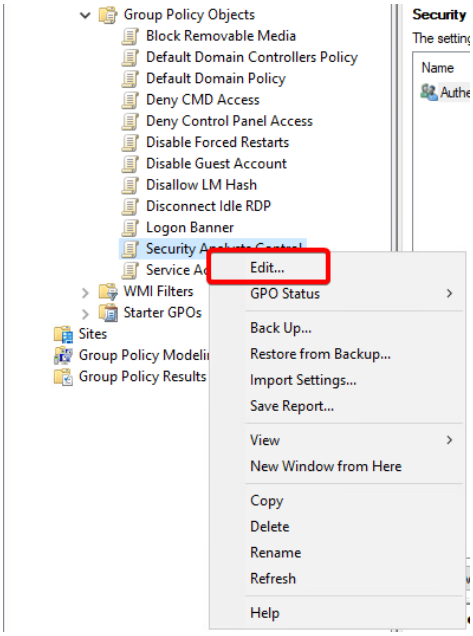
- We need to open GPMC and expand the Group Policy Objects hive so we can see what GPOs exist.
- Right-click on the policy object we wish to modify and select "Edit. The Group Policy Management Editor should pop up in a new window.
- From here, we have several options to enable or disable.
- We need to modify the removable media settings and ensure they are set to block any removable media from access. We will expressly allow security analysts to access PowerShell and CMD since their daily duties require it.
  - location of removable media policy settings = `User Configuration > Policies > Administrative Templates > System > Removable Storage Access.`
  - Location of Command Prompt settings = `User Configuration > Policies > Administrative Templates > System.`
- For `Computer settings`, we need to ensure the Logon Banner is applied and that the password policy settings for this group are strengthened.
  - Location of Logon Banner settings = `Computer Configuration > Policies > Windows Settings > Security Settings > Local Policies > Security Options.`
  - For reference, this setting should already be enabled since the GPO we copied was for a Logon Banner. We are validating the settings and ensuring it is enabled and applied.
  - Location of Password Policy settings = `Computer Configuration > Policies > Windows Settings > Security Settings > Account Policies > Password Policy.`

Let's get started.

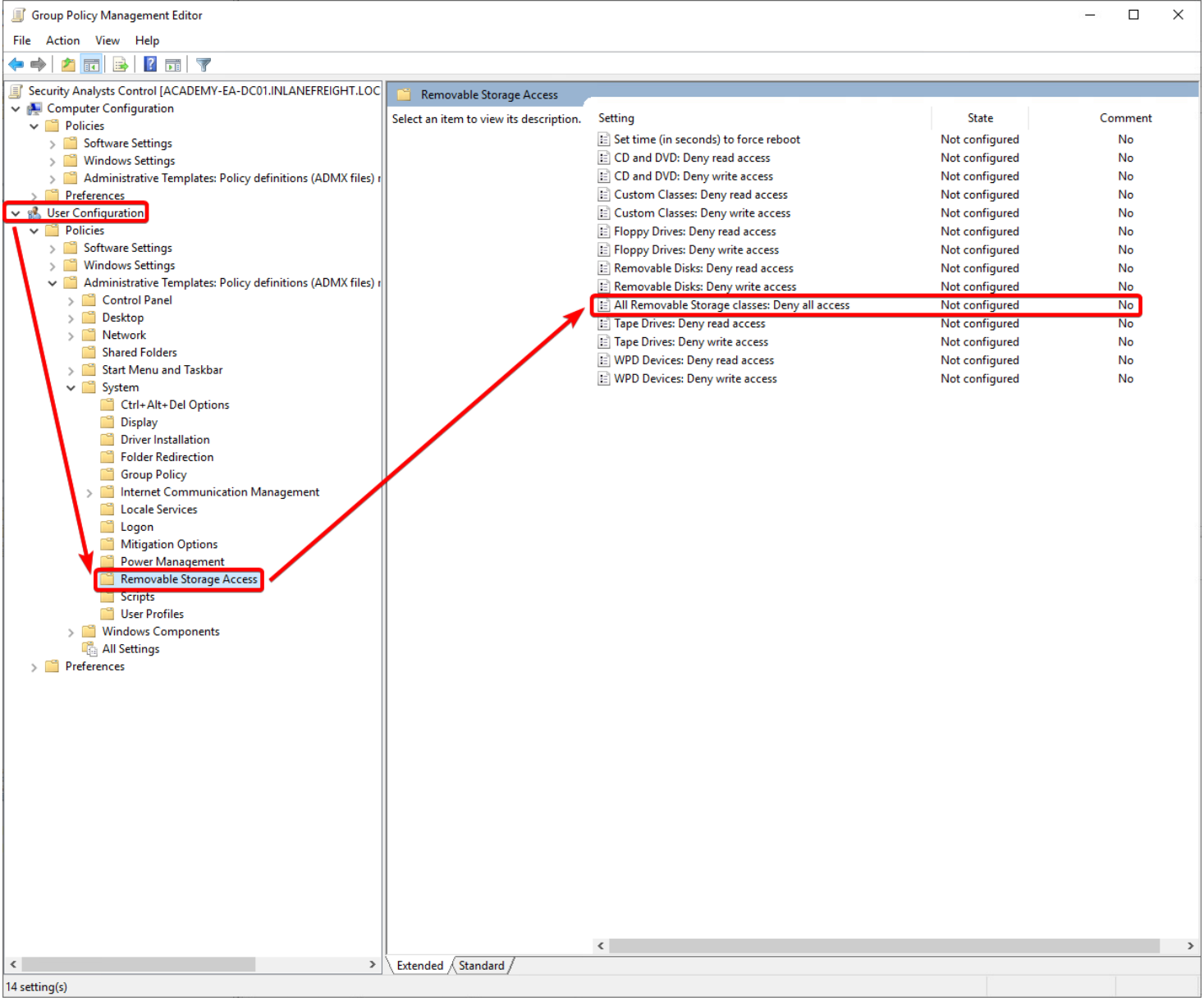
W3.CSS

User Configuration Group Policies

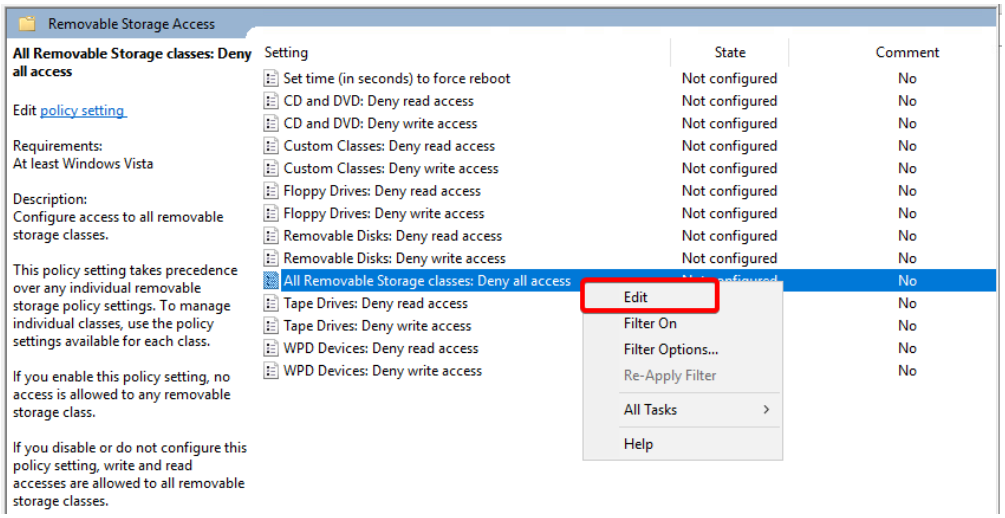
This slideshow will walk us through modifying group policies that affect Users directly. We will be modifying the policies affecting users access to the command prompt as well as their ability to use removeable media.



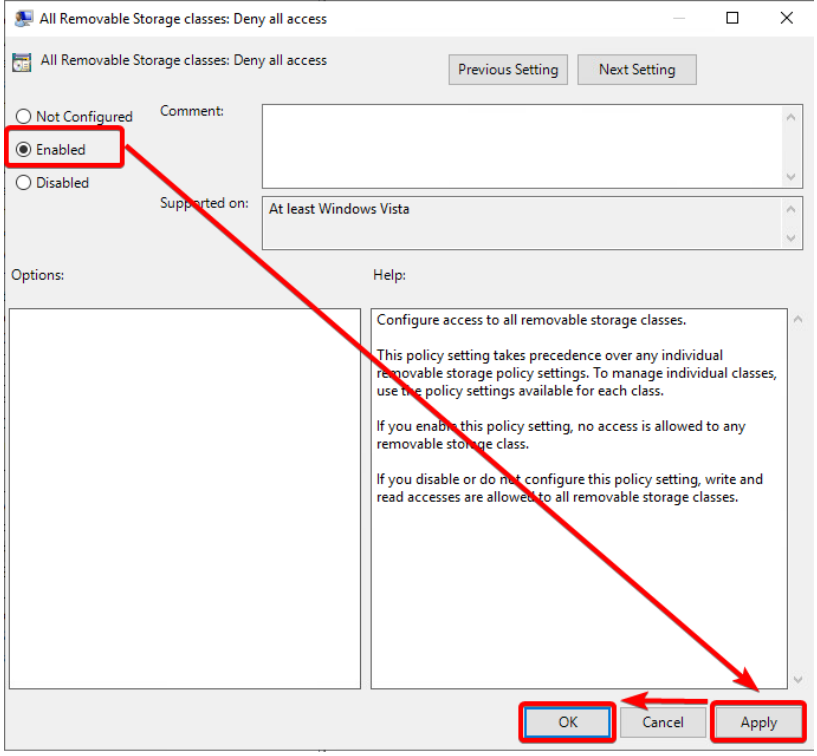
1. Right-click the GPO we wish to modify and select "Edit". This will bring up the Group Policy Configuration Editor window.



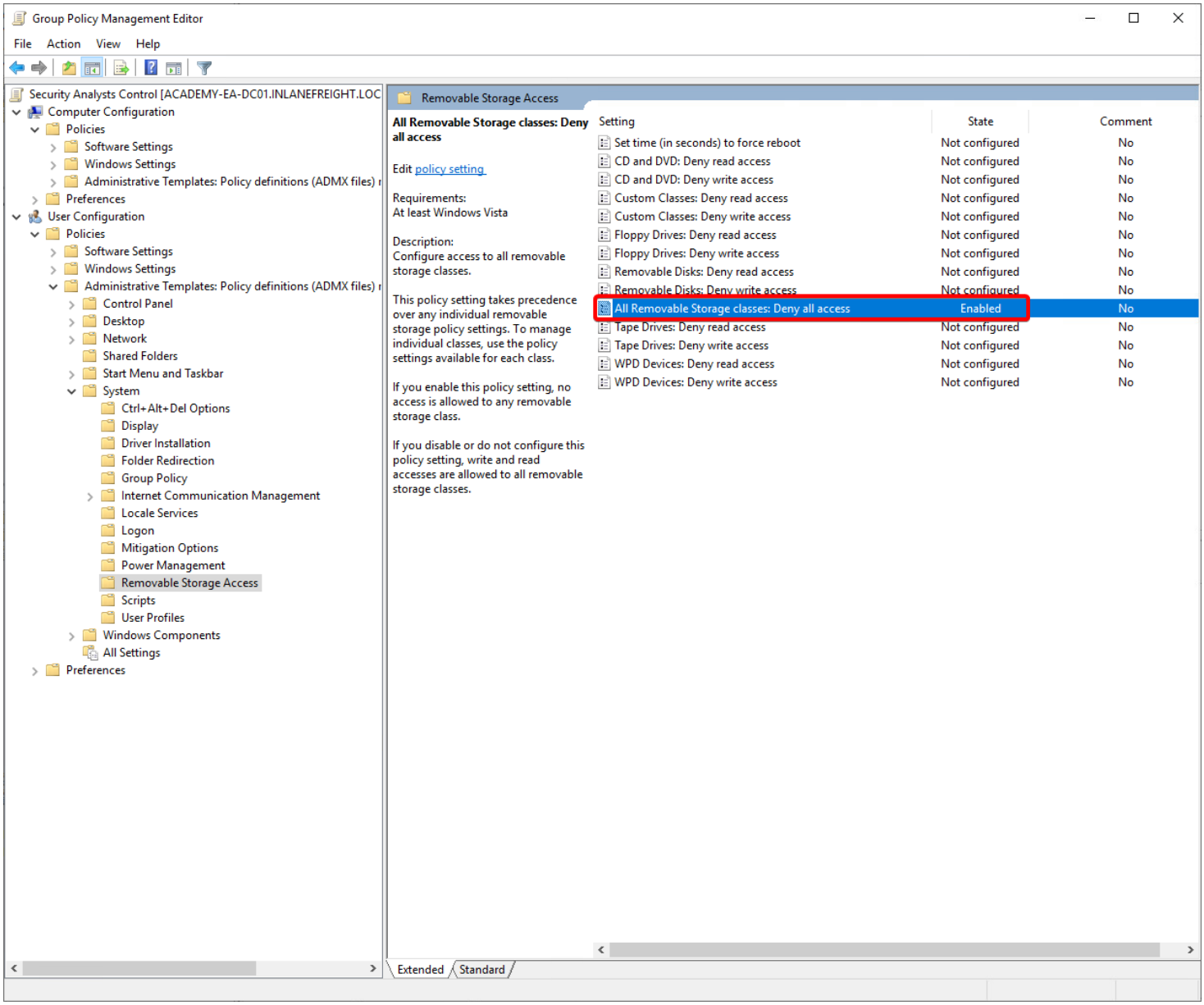
2. Drill down into the User Configuration policies to System > "Removable Storage Access". The policy we are going to edit is highlAdding an AD User via the GUlighted.



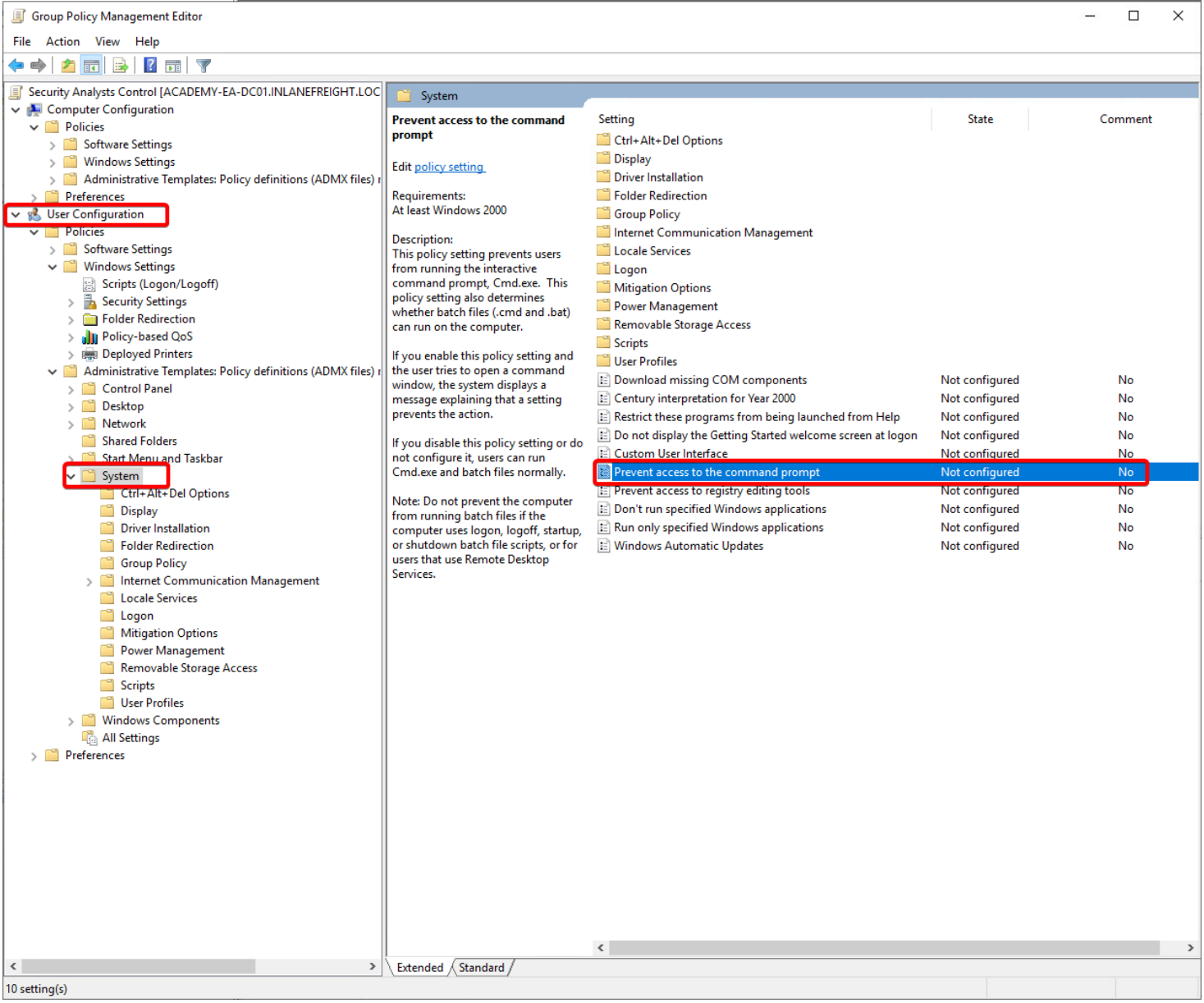
3. Right click the setting and select "Edit".



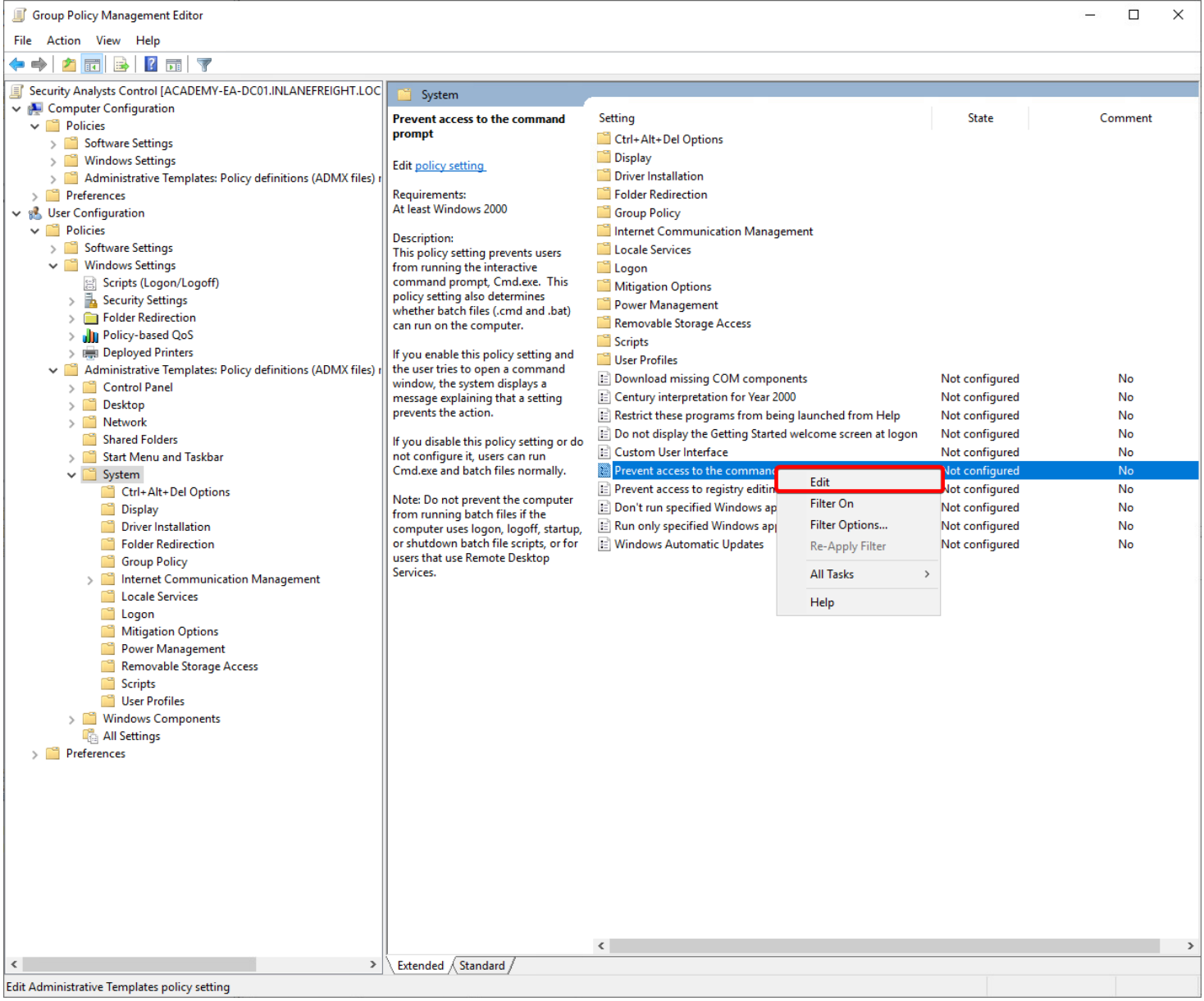
4. Check the radial button to enable the setting, hit "Apply" and then "OK".



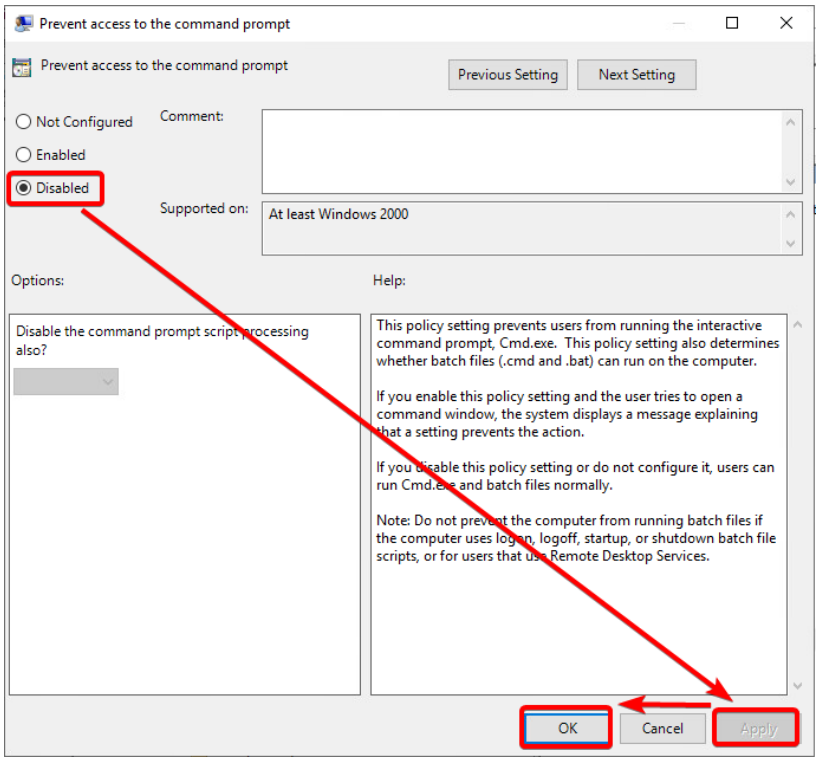
5. We can now see that our Policy setting is set to Enbled. Once we push policy to the domain, it will take effect.



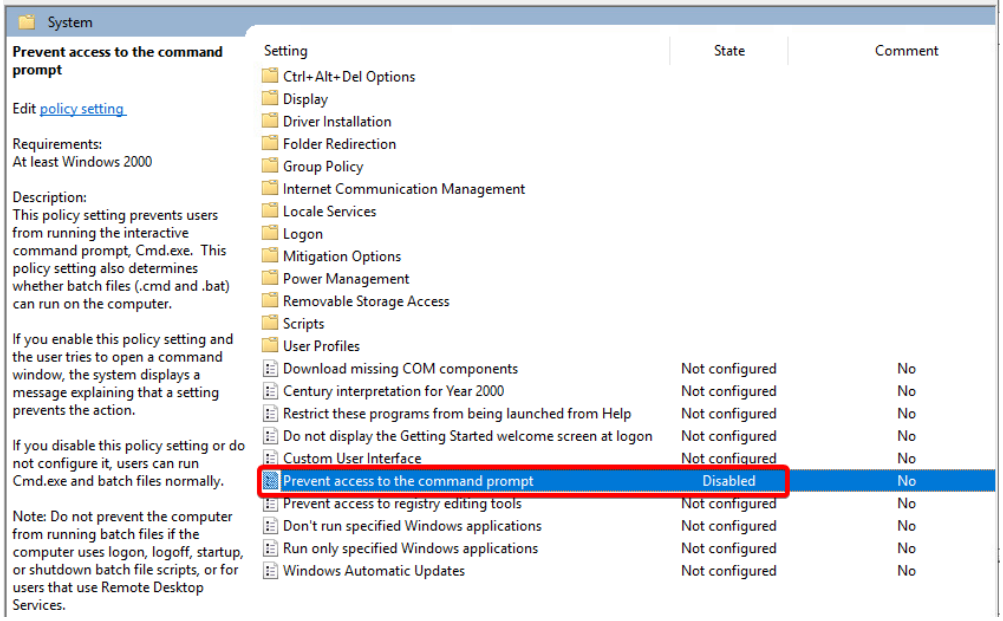
6. Next we are modifying the policy for Command Prompt access. Move to the System hive under User Configuration.



7. Right click and Edit the setting for "Prevent access to the command prompt".



8. We will select the radial button beside "Disabled" to explicitly allow the Security Analyst users to run command prompt and batch files as necessary for their role.



9. We can validate our policy settings are set in the view highlighted.

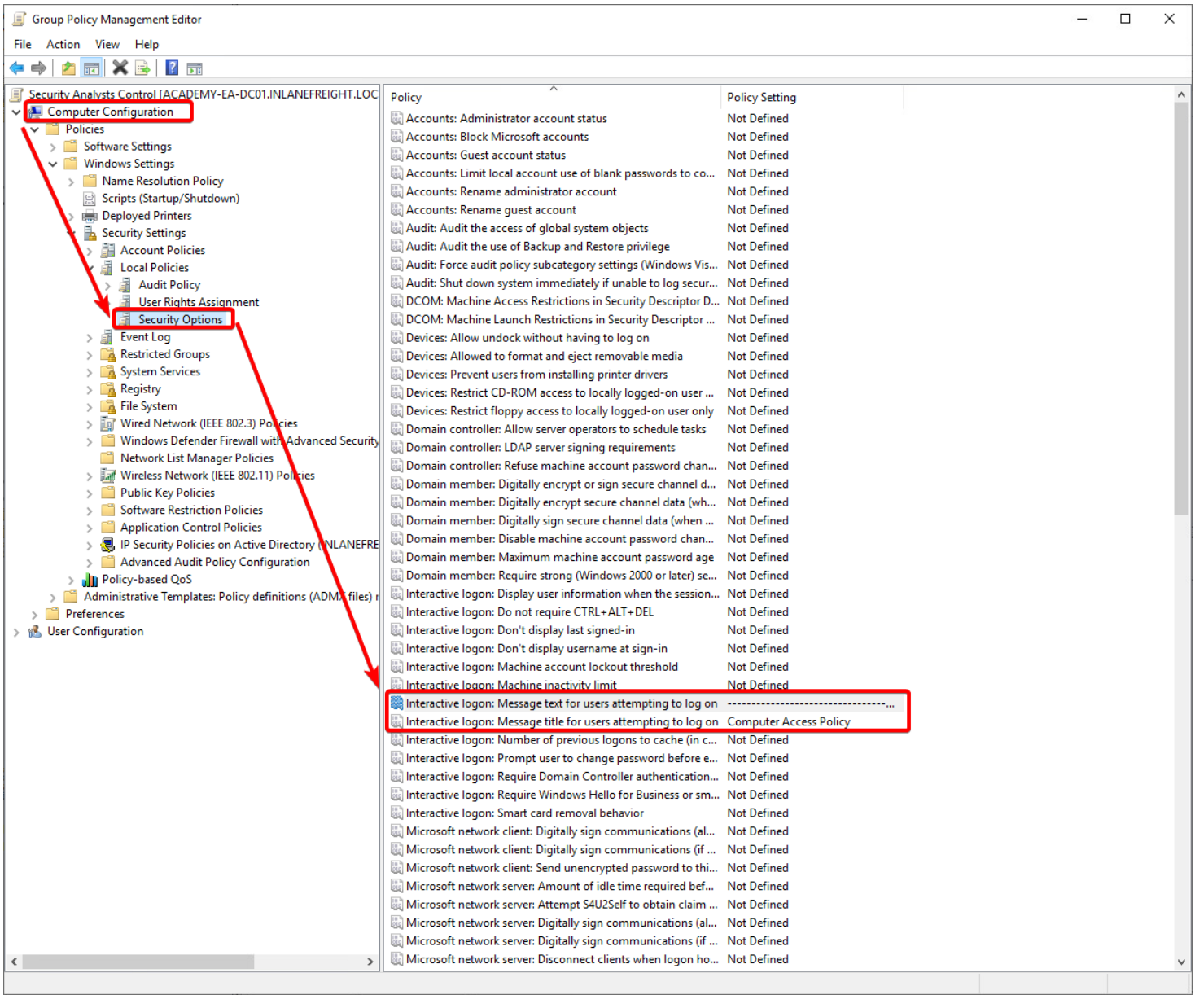
‹›

Now, let's modify the group policies affecting our `Computer` settings. We don't have to exit from the GPMC editor; we can just collapse the user configuration section and expand the Computer Configuration section.

W3.CSS

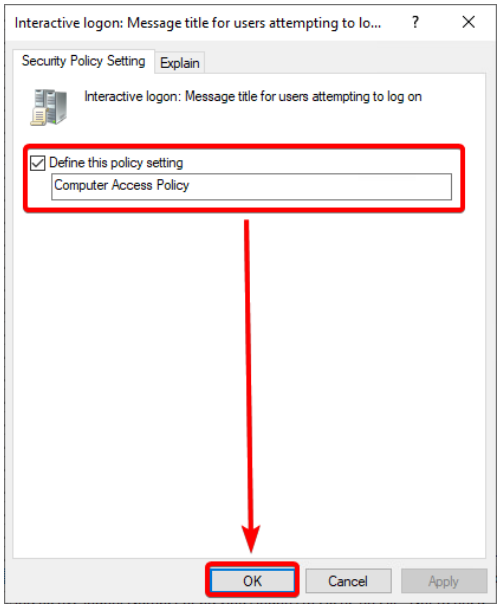
Computer Configuration Group Policies

This slideshow will walk us through modifying group policies that affect computers in the group. We will be modifying the policies affecting the Logon Banner for the host, and setting a more restrictive password policy.

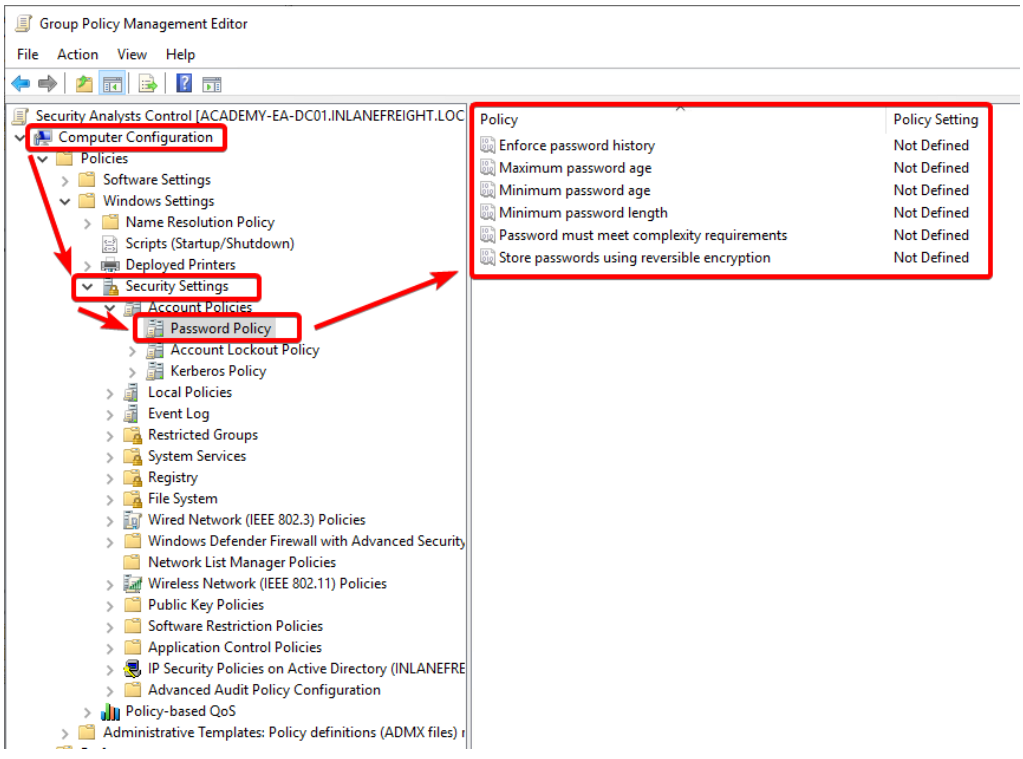


1. Move from the User Configuration hive into the Computer Configuration have. We will be validating the "Logon Banner" settings first. We validate the setting in "Interactive Logon Message Text" and "Interactive Logon Message Title".

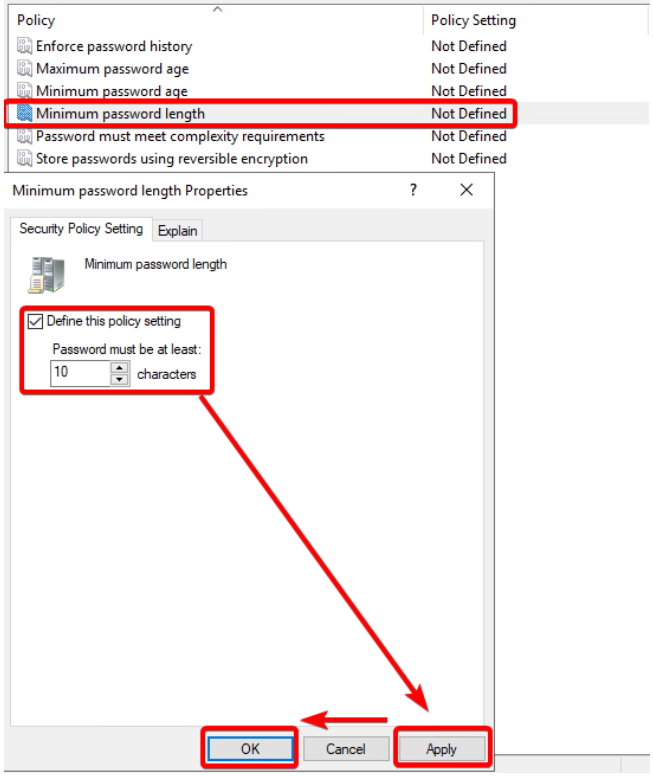
2. Right-click the setting and select Properties. Ensure the radial to define the policy setting is enabled and there is a Banner in the text box. If all appears good, hit OK.



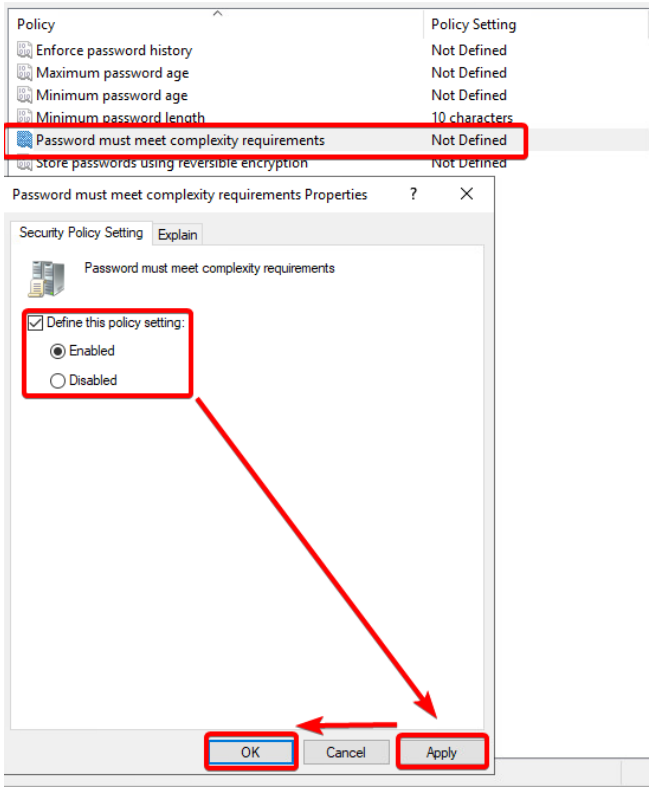
3. Change to the Message Title policy setting and validate the radial is selected, and a title of "Computer Access Policy" has been defined.



4. Now, we will modify the settings for the Password Policies. Move into the Security Settings hive and click on "Password Policy" under the Account Policies dropdown. The policies on the right are what we will modify.

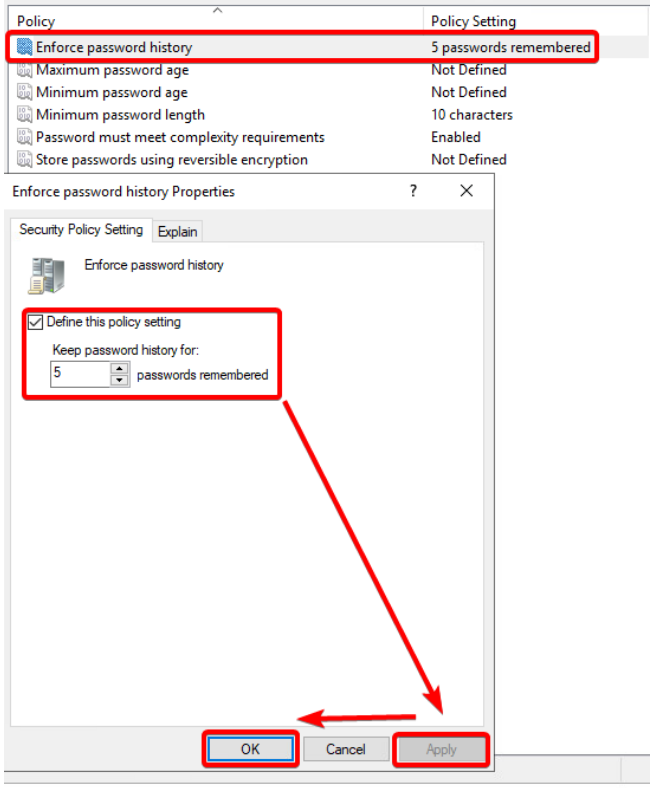


5. Starting with the "Minimum Password Length" setting. Right-click, select "Properties", and select the radial button to define the setting. Set the character count to ten. When done, apply and hit OK.

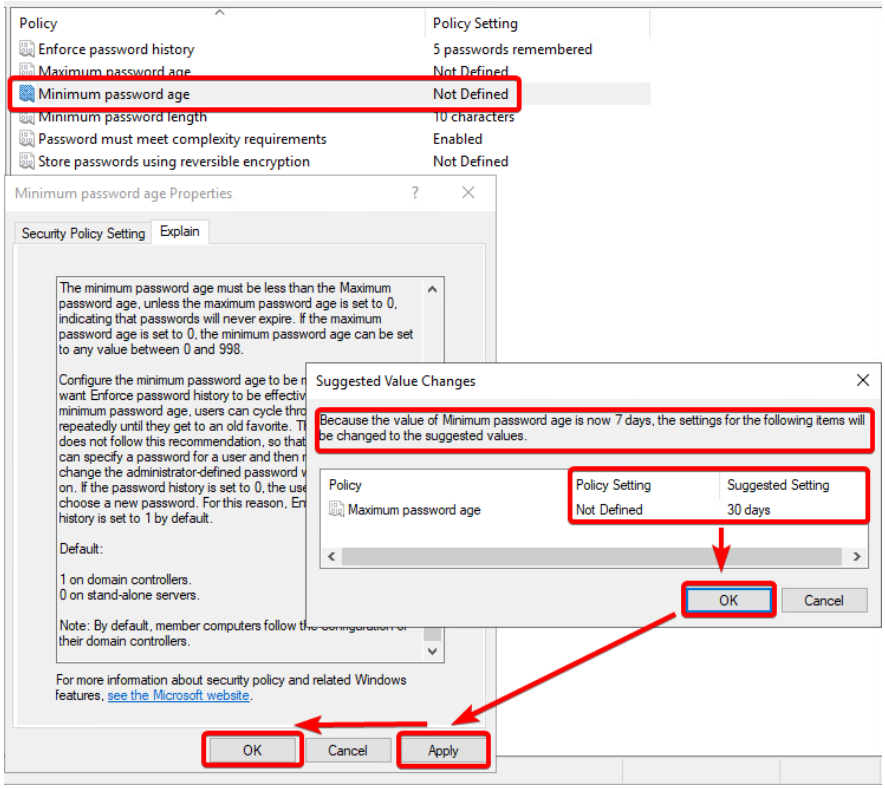


6. Now, we will enable "Password Complexity Requirements." Define the policy setting by clicking the radial button and then ensure "Enabled" is selected.

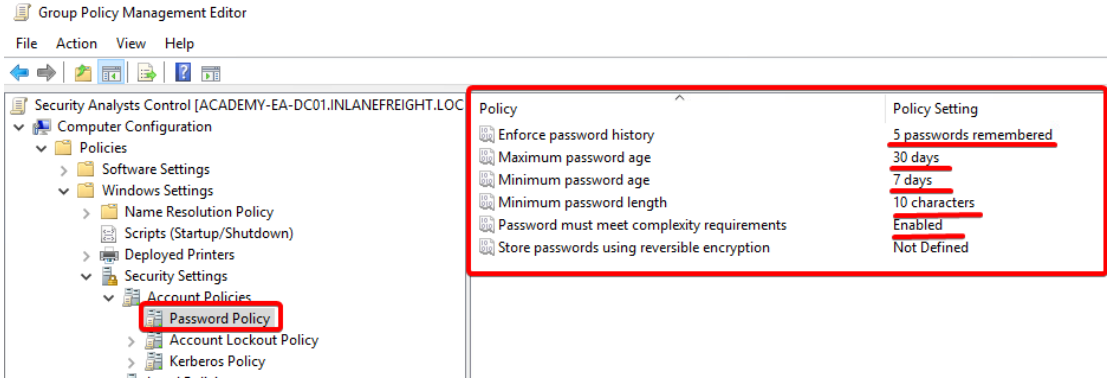
Local Policies > Security Options



7. Next, we want to enforce password history for resetting the account password. Define the setting, and set the password history count to 5 previous passwords remembered. Hit Apply and OK.



8. Set the Minimum Password Age setting by defining the setting and applying a minimum age of 7 days. A new window will pop up telling us that the setting for "Maximum Password Age" will be set as well.



9. Validate all the settings match what we wished to define. If all looks well, we have completed this task!

❮❯

Summary

This wraps it up for the first part of the guided lab. We covered how to manage users, groups, and Group Policy. In the next section, we will add a Computer to the INLANEFREIGHT domain, change the OU it exists in, ensuring that it is in the proper group to receive the Group Policy we created earlier.

Note: It may take 2-3 minutes for your target instance to spawn. If you receive the error message `Timeout waiting for activation` when attempting to connect via RDP, wait a few seconds and run the command again. Furthermore, loading the Active Directory PowerShell module or the MMC snap-ins may take a bit longer on the first run.

AD Administration: Guided Lab Part II

In this section of the guided lab, we will be completing the final tasks for the day. We have to add a computer to the domain and change the OU it resides in.

Connection Instructions

For this lab, you will utilize RDP and have access to a non-domain-joined Windows host from which you can perform any actions needed to complete the lab. You will be using an RDP connection, much like in Part one.

- Click below in the Questions section to spawn the target host and obtain an IP address.
  - IP ==
  - Username == `image`
  - Password == `Academy_student_AD!`

Task 4 Add and Remove Computers To The Domain

Our new users will need computers to perform their daily duties. The helpdesk has just finished provisioning them and requires us to add them to the INLANEFREIGHT domain. Since these analyst positions are new, we will need to ensure that the hosts end up in the correct OU once they join the domain so that group policy can take effect properly.

The host we need to join to the INLANEFREIGHT domain is named: `ACADEMY-IAD-W10` and has the following credentials for use to login and finish the provisioning process:

- User == `image`
- Password == `Academy_student_AD!`

Once you have access to the host, utilize your `htb-student_adm`: `Academy_student_DA!` account to join the host to the domain.

`Solution: Task 4`

To add the localhost to a domain via PowerShell, Open a PowerShell session as administrator, and then we can use the following command:

PowerShell Join a Domain

```
PS C:\htb> Add-Computer -DomainName INLANEFREIGHT.LOCAL -Credential INLANEFREIGHT\HTB-student_adm -Restart
```

This string utilizes the `domain` (INLANEFREIGHT.LOCAL) we wish to join the host to, and we must specify the `user` whose credentials we will use to authorize the join. (HTB-student_ADM). Specifying the restart at the string is necessary because the join will not occur until the host restarts again, allowing it to acquire settings and policies from the domain.

Add via the GUI

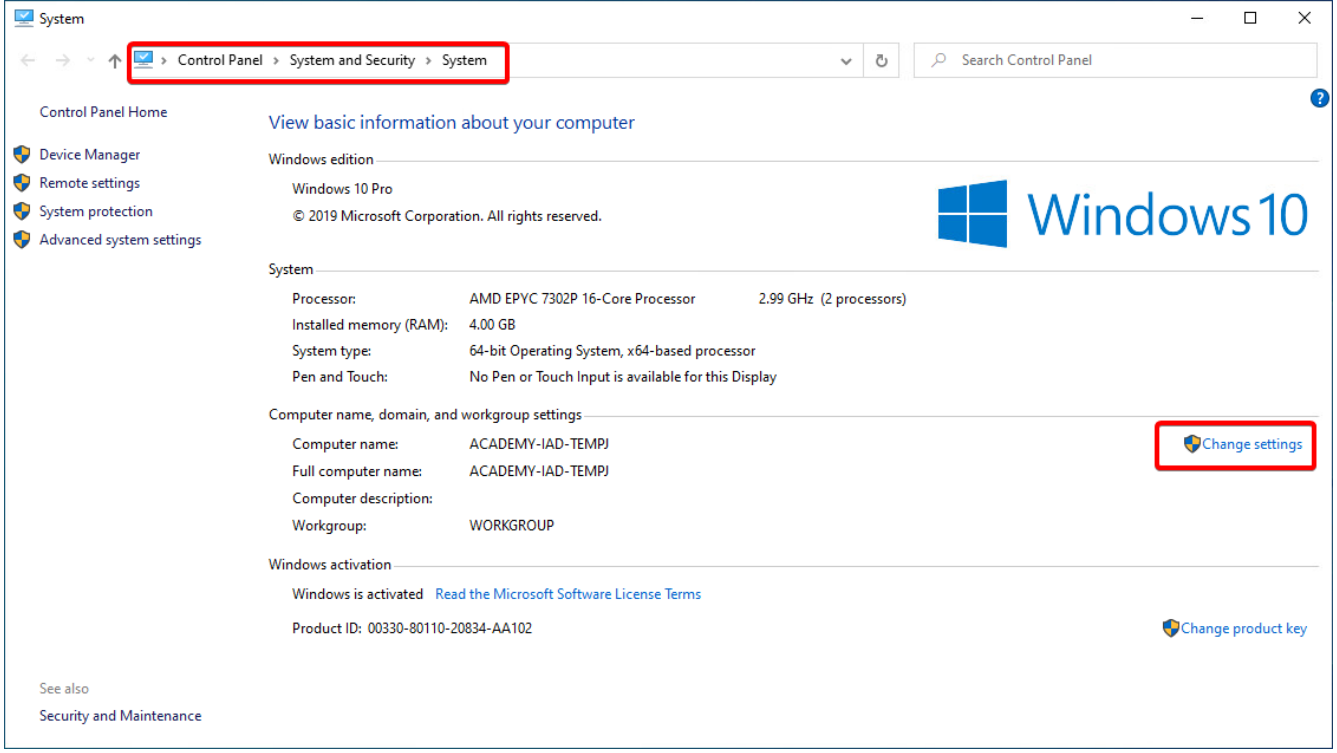
To add the computer to the domain from the localhost GUI is a bit different. Follow these steps to join it to the domain:

- From the computer you wish to join the domain, open the `Control Panel` and navigate to "System and Security > System."
- Now select the "Change Settings" icon in the `Computer name` section. Another dialog box will pop up asking you for administrator credentials.
  In the next window, we need to select the change icon next to the portion that says, "To rename this computer or change its domain or workgroup, click change" This will open yet another window for you to modify the computer's name, domain, and workgroup. Check that the computer's name matches the naming standard you wish to use for the domain before joining. Doing so will ease the administrative burden of renaming a domain-joined host later.
- next, we need to enter the name of the domain we wish to join the computer to (`INLANEFREIGHT.LOCAL`) and click OK. You may receive a warning about NetBIOS name resolution. That is an issue outside the scope of this lab. For now, move forward.
  - You will be prompted for domain credentials to complete this action. Utilize the domain administrator account you have been given at the beginning of this lab. (`htb-student_adm`).
  - If all goes well, you will be presented with a prompt welcoming you to the domain. The computer needs to restart to apply changes and new group policy settings it will receive from the domain.

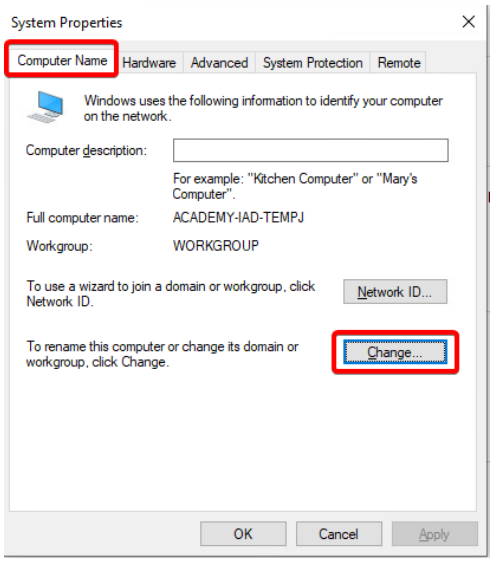
W3.CSS

Add A Computer To The Domain

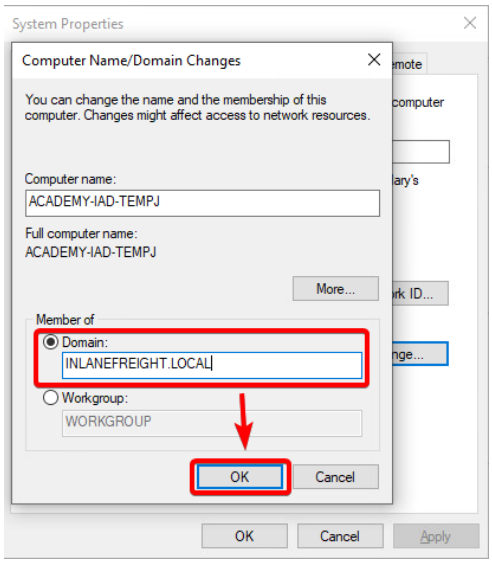
We are going to use the Windows GUI to add this PC to the domain.



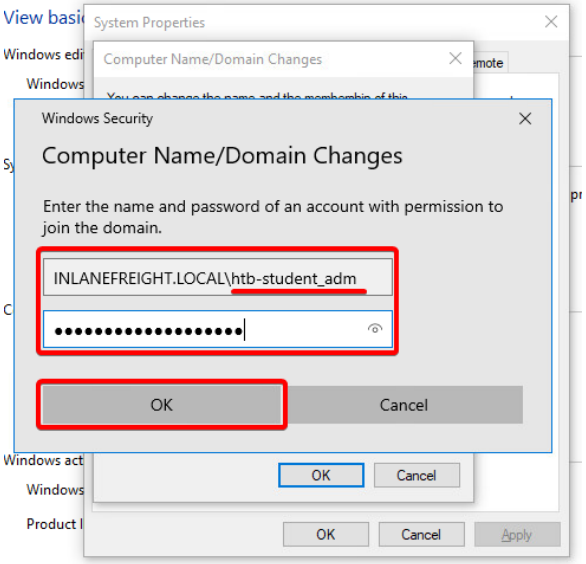
1. From the control panel, open up system properties for the pc. Click on Change Settings in the Computer name section.



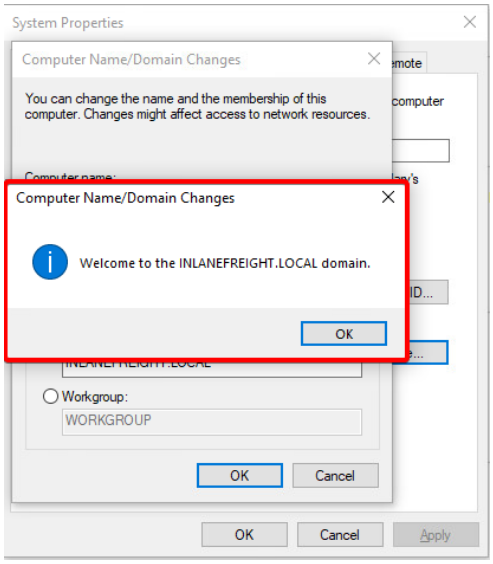
2. In this window, select the Change button beside "rename this computer or change its domain."



3. Enter the domain you wish to join the host to. (INLANEFREIGHT.LOCAL) Hit OK.



4. Enter the domain administrator credentials provided to join the host to the domain.



5. If all goes well, you will be prompted with a Welcome to the domain popup.

❮❯

Add a Remote Computer to a Domain

```
PS C:\htb> Add-Computer -ComputerName ACADEMY-IAD-W10 -LocalCredential ACADEMY-IAD-W10\image -DomainName INLANEFREIGHT.LOCAL -Credential INLANEFREIGHT\htb-student_adm -Restart
```

When we added the computer to the domain, we did not stage an AD object for it in the OU we wanted the computer in beforehand, so we have to move it to the correct OU now. To do so via PowerShell:

Check OU Membership of a Host

```
PS C:\htb> Get-ADComputer -Identity "ACADEMY-IAD-W10" -Properties * | select CN,CanonicalName,IPv4Address
```

The CanonicalName property (seen above) will tell us the full path of the host by printing out the name in the format "Domain/OU/Name." We can use this to locate the host and validate where it is in our AD structure.

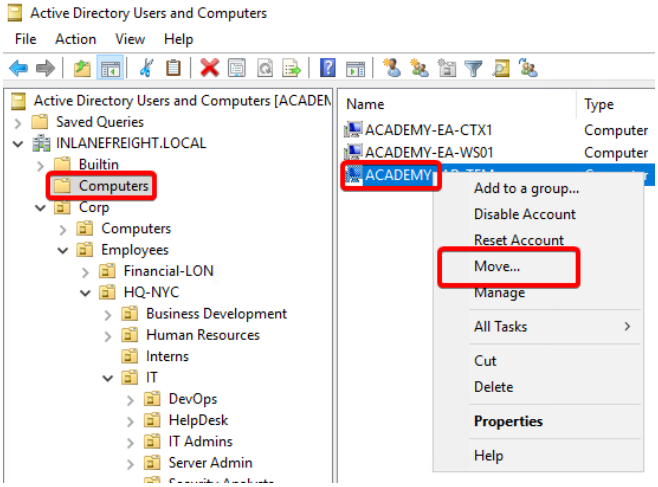
Utilizing the ADUC snap-in, you can also move computer objects pretty quickly. You do so by:

Add to a New OU

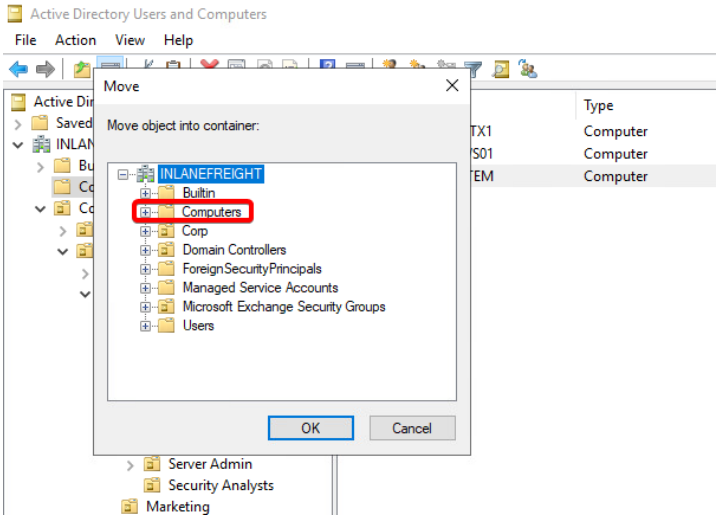
W3.CSS

Move A Computer Object To A New OU

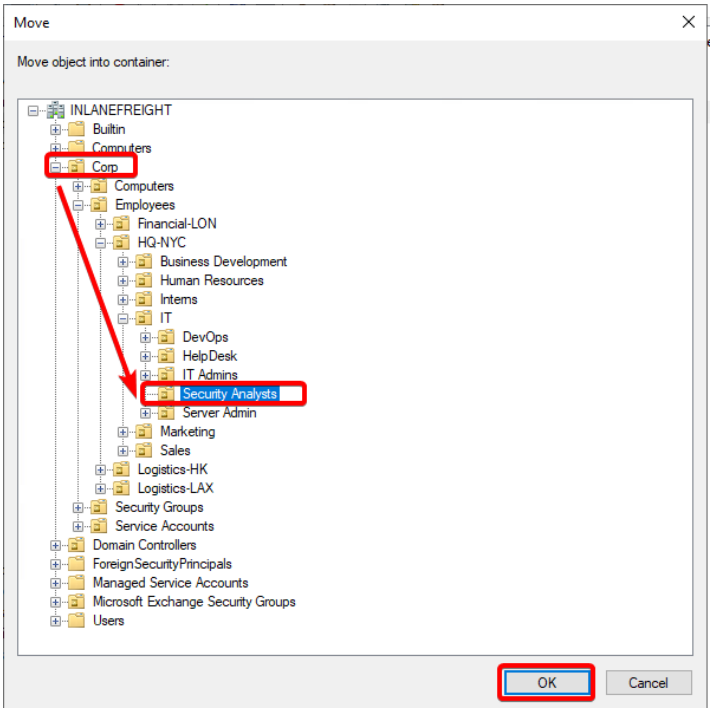
We need to find the new host, and move it to the "Security Analysts" OU in the same manner we moved the user account earlier.



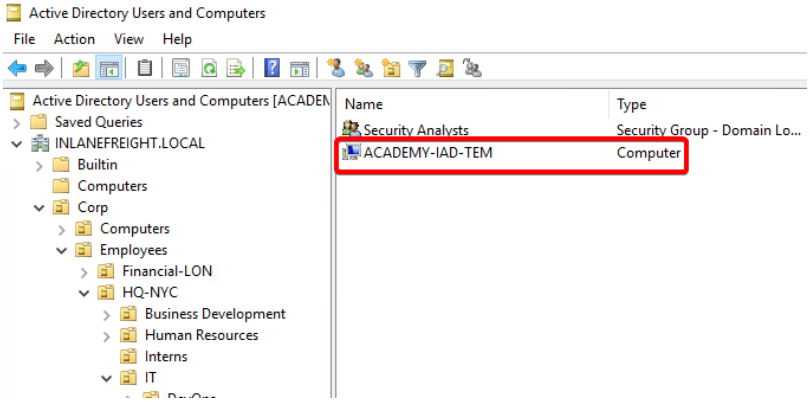
1. Looking in the Computers OU, select our newly joined host and right click it. Select the option to "Move"



2. In the popup, drill down to the Security Analysts OU.



3. Select the Security Analysts OU and hit OK.



4. If we look in that OU we will now see a new Computer object within.

❮❯

Summary

This wraps up our administration duties for the day. Hopefully, this lab helped reinforce the basic concepts surrounding AD management.
It is always great to get hands-on experience with topics and technologies like Active Directory. This experience provides a better understanding of how it functions and how it can possibly be taken advantage of. New vulnerabilities and attacks are being released every day that affect the Windows operating system, and by extension, Active Directory. A fundamental understanding of AD, the attacks that plague it, and defensive measures will take us a long way as security Professionals.

Note: It may take 2-3 minutes for your target instance to spawn. If you receive the error message `Timeout waiting for activation` when attempting to connect via RDP, wait a few seconds and run the command again. Furthermore, loading the Active Directory PowerShell module or the MMC snap-ins may take a bit longer on the first run.

Wrapping It Up

We've now started down the rabbit hole that is Active Directory. Love it or hate it, if we are to continue down a technical information security path, we will have to deal with Active Directory in some way -- hardening and administering as a security-minded sysadmin, attacking it as a network pentester, defending it as a threat hunter, or performing incident response or digital forensics. Working in any of these roles requires in-depth knowledge of Active Directory. Here we've laid the foundations for the common terms that will pop up again and again as you progress through other Academy modules, other training courses, and infosec in general. Where do we go from here? Check out the accompanying module in our Penetration Tester Path, `Active Directory Enumeration & Attacks`, that covers manual and automated enumeration and attack techniques that we've seen and used frequently in real-world environments.

Academy Skills Paths

We also have an Active Directory Enumeration path which features the following modules:

- Active Directory LDAP
- Active Directory PowerView
- Active Directory BloodHound

Boxes To Pwn, Videos To Help Visualize

The Hack The Box main platform has many targets for learning and practicing AD enumeration and attacks. Some boxes worth checking out are:

- Active
- Resolute
- Forest
- Cascade

Ippsec has recorded videos explaining the paths through many of these boxes and more. If you get stuck or want a great primer dealing with Active Directory and see how some of the tools work, check out the video links above.

More AD Learning Opportunities

Pro Labs are large simulated corporate networks that teach skills applicable to real-life penetration testing engagements. The Dante Pro Lab is an excellent place to start with varying vectors and some AD exposure. The Offshore Pro Lab is an intermediate-level lab that contains a wealth of opportunities for practicing AD enumeration and common and less common attacks.

- Dante Pro Lab
- Offshore Pro Lab

There are more advanced Pro Labs available too.

For more practice specific to AD, check out the AD Track as well. Tracks are curated lists of machines and challenges for users to work through and master a particular topic. The AD Track contains boxes of varying difficulties with a variety of attack vectors. Even if you are unable to solve these boxes on your own, it is still worth working with them with a walkthrough or video or just watching the video on the box by Ippsec. The more you expose yourself to these topics, the more comfortable and second nature enumeration and many attacks will become.

Closing Thoughts

Between the HTB Discord, Forums, and blogs, there are plenty of amazing write-ups to help advance your skills along the way. One to pay attention to would be 0xdf's walkthroughs. These are also a great resource to get an idea of how an Active Directory attack path may look in the real world. 0xdf writes about much more, and his blog is an excellent resource.

We would be remiss not to mention Sean Metcal's amazing Active Directory Security blog. This blog is a treasure trove of information and worth perusing.

It is also worth reading as much about Active Directory security as possible and becoming familiar with the tools and techniques mentioned in the Active Directory timeline earlier in the module. It is a vast topic and will take time to master. A fundamental understanding of AD and the tools surrounding the field, both as a pentester or defender, will make life immeasurably easier, though. Just because BloodHound is billed as a Pentesting tool, that doesn't mean that it can't be a great tool to leverage as a Defender to acquire better visibility into your enterprise and the relationships contained within. The more we understand the bigger picture, how things in enterprise networks are intertwined, and the various ways to attack and harden AD, the more powerful we will become as attackers and defenders, and the more value we can provide to our clients and the companies we work for.