Introduction and Terminology

What is Active Directory?

`Active Directory` (`AD`) is a directory service for Windows enterprise environments that Microsoft officially released in 2000 with Windows Server 2000. Microsoft has been incrementally improving AD with the release of each new server OS version. Based on the protocols x.500 and LDAP that came before it (which are still utilized in some form today), AD is a distributed, hierarchical structure that allows centralized management of an organization's resources, including users, computers, groups, network devices and file shares, group policies, devices, and trusts. AD provides authentication, accounting, and authorization functionalities within a Windows enterprise environment. It also allows administrators to manage permissions and access to network resources.

Active Directory is so widespread that it is by a margin the most utilized Identity and Access management (`IAM`) solution worldwide. For this reason, the vast majority of enterprise applications seamlessly integrate and operate with Active Directory. Active Directory is the most critical service in any enterprise. A compromise of an Active Directory environment means unrestricted access to all its systems and data, violating its `CIA` (`Confidentiality`, `Integrity`, and `Availability`). Researchers constantly discover and disclose vulnerabilities in AD. Via these vulnerabilities, threat actors can utilize malware known as ransomware to hold an organization's data hostage for ransom by performing cryptographic operations (`encryption`) on it, therefore rendering it useless until they either pay a fee to purchase a decryption key (`not advised`) or obtain the decryption key with the help of IT Security professionals. However, if we think back, an Active Directory compromise means the compromise of all and any applications, systems, and data instead of a single system or service.

Let's look at publicly disclosed vulnerabilities for the past three years (2020 to 2022). Microsoft has over 3000, and around 9000 since 1999, which signifies an incredible growth of research and vulnerabilities in the past years. The most apparent practice to keep Active Directory secure is ensuring that proper `Patch Management` is in place, as patch management is currently posing challenges to organizations worldwide. For this module, we will assume that Patch Management is done right (Proper Patch Management is crucial for the ability to withstand a compromise) and focus on other attacks and vulnerabilities we can encounter. We will focus on showcasing attacks that abuse common misconfigurations and Active Directory features, especially ones that are very common/familiar yet incredibly hard to eliminate. Additionally, the protections discussed here aim to arm us for the future, helping us create proper cyber hygiene. If you are thinking `Defence in depth`, `Network segmentation`, and the like, then you are on the right track.

If this is your first time learning about Active Directory or hearing these terms, check out the Intro to Active Directory module for a more in-depth look at the structure and function of AD, AD objects, etc. And also Active Directory - Enumeration and Attacks for strengthening your knowledge and gaining an overview of some common attacks.

Refresher

To ensure we are familiar with the basic concepts, let's review a quick refresher of the terms.

A `domain` is a group of objects that share the same AD database, such as users or devices.

A `tree` is one or more domains grouped. Think of this as the domains `test.local`, `staging.test.local`, and `preprod.test.local`, which will be in the same tree under `test.local`. Multiple trees can exist in this notation.

A `forest` is a group of multiple trees. This is the topmost level, which is composed of all domains.

`Organizational Units` (`OU`) are Active Directory containers containing user groups, Computers, and other OUs.

My Workstation

OFFLINE

Start Instance

∞ / 1 spawns left

`Trust` can be defined as access between resources to gain permission/access to resources in another domain.

`Domain Controller` is (generally) the Admin of the Active Directory used to set up the entire Directory. The role of the Domain Controller is to provide Authentication and Authorization to different services and users. In Active Directory, the Domain Controller has the topmost priority and has the most authority/privileges.

`Active Directory Data Store` contains Database files and processes that store and manages directory information for users, services, and applications. Active Directory Data Store contains the file `NTDS.DIT`, the most critical file within an AD environment; domain controllers store it in the `%SystemRoot%\NTDS` folder.

A `regular 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`
- `Default Domain Policy`
- `Domain Functional Levels`
- `Password Policy`
- `Group Policy Objects` (GPOs)
- `Kerberos Delegation`
- `Domain Trusts`
- `Access Control Lists` (ACLs)

Although the settings of AD allow this default behavior to be modified/disallowed, its implications can result in a complete breakdown of applications, services, and Active Directory itself.

`LDAP` is a protocol that systems in the network environment use to communicate with Active Directory. Domain Controller(s) run LDAP and constantly listen for requests from the network.

`Authentication` in Windows Environments:

- Username/Password, stored or transmitted as password hashes (`LM`, `NTLM`, `NetNTLMv1`/`NetNTLMv2`).
- `Kerberos` tickets (Microsoft's implementation of the Kerberos protocol). Kerberos acts as a trusted third party, working with a domain controller (DC) to authenticate clients trying to access services. The Kerberos authentication workflow revolves around tickets that serve as cryptographic proof of identity that clients exchange between each other, services, and the DC.
- `Authentication over LDAP`. Authentication is allowed via the traditional username/password or user or computer certificates.

`Key Distribution Center` (`KDC`): a Kerberos service installed on a DC that creates tickets. Components of the KDC are the authentication server (AS) and the ticket-granting server (TGS).

`Kerberos Tickets` are tokens that serve as proof of identity (created by the KDC):

- `TGT` is proof that the client submitted valid user information to the KDC.
- `TGS` is created for each service the client (with a valid TGT) wants to access.

`KDC key` is an encryption key that proves the TGT is valid. AD creates the KDC key from the hashed password of the `KRBTGT` account, the first account created in an AD domain. Although it is a disabled user, KRBTGT has the vital purpose of storing secrets that are randomly generated keys in the form of password hashes. One may never know what the actual password value represents (even if we try to configure it to a known value, AD will automatically override it to a random one).

Each domain contains the groups `Domain admins` and `Administrators`, the most privileged groups in broad access. By default, AD adds members of Domain admins to be Administrators on all Domain joined machines and therefore grants the rights to log on to them. While the 'Administrators' group of the domain can only log on to Domain Controllers by

default, they can manage any Active Directory object (e.g., all servers and therefore assign themselves the rights to log on to them). The topmost domain in a forest also contains an object, the group `Enterprise Admins`, which has permissions over all domains in the forest.

Default groups in Active Directory are heavily privileged and carry a hidden risk. For example, consider the group `Account Operators`. When asking AD admins what the reason is to assign it to users/super users, they will respond that it makes the work of the 'Service Desk' easier as then they can reset user passwords. Instead of creating a new group and delegating that specific right to the Organizational Units containing user accounts, they violate the principle of least privilege and endanger all users. Subsequently, this will include an escalation path from Account Operators to Domain Admins, the most common one being through the 'MSOL_' user accounts that Azure AD Connect creates upon installation. These accounts are placed in the default 'Users' container, where 'Account operators' can modify the user objects.

It is essential to highlight that Windows has multiple logon types: ' how' users log on to a machine, which can be, for example, interactive while a user is physically present on a device or remotely over RDP. Logon types are essential to know about because they will leave a 'trace' behind on the system(s) accessed. This trace is the username and password used. As a rule of thumb, logon types except 'Network logon, type 3' leave credentials on the system authenticated and connected to. Microsoft provides a complete list of logon types here.

To interact with Active Directory, which lives on Domain Controllers, we must speak its language, LDAP. Any query happens by sending a specifically crafted message in LDAP to a Domain Controller, such as obtaining user information and a group's membership. Early in its life, Microsoft realized that LDAP is not a 'pretty' language, and they released Graphical tools that can present data in a friendly interface and convert 'mouse clicks' into LDAP queries. Microsoft developed the `Remote Server Administration Tools` (RSAT), enabling the ability to interact with Active Directory locally on the Domain Controller or remotely from another computer object. The most popular tools are `Active Directory Users and Computers` (which allows for accessible viewing/moving/editing/creating objects such as users, groups, and computers) and `Group Management Policy` (which allows for the creation and modification of Group policies).

Important network ports in any Windows environment include (memorizing them is hugely beneficial):

- 53: `DNS`.
- 88: `Kerberos`.
- 135: `WMI/RPC`.
- 137-139 & 445: `SMB`.
- 389 & 636: `LDAP`.
- 3389: `RDP`
- 5985 & 5896: `PowerShell Remoting (WinRM)`

Real-world view

Every organization, which has (attempted) at some point to increase its maturity, has gone through exercises that classify its systems. The classification defines the `importance` of each system to the business, such as `ERP`, `CRM`, and `backups`. A business relies on this to successfully meet its objectives and is significantly different from one organization to another. In Active Directory, any additional roles, services, and features that get 'added' on top of what comes out of the box must be classified. This classification is necessary to ensure that we set the bar for which service, if compromised, poses an escalation risk toward the rest of Active Directory. In this design view, we need to ensure that any service allowing for direct (or indirect) escalation is treated similarly as if it was a Domain Controller/Active Directory. Active Directory is massive, complex, and feature-heavy - potential escalation risks are under every rock. Active Directory will provide services such as DNS, PKI, and Endpoint Configuration Manager in an enterprise organization. If an attacker were to obtain administrative rights to these services, they would indirectly have means to escalate their privileges to those of an Administrator of the `entire forest`. We will demonstrate this through some attack paths described later in the module.

Active Directory has limitations, however. Unfortunately, these limitations are a 'weak' point and expand our attack surface - some born by complexity, others by design, and some due to legacy and backward compatibility. For the sake

of completeness, below are three examples of each:

1. `Complexity` - The simplest example is figuring out nested group members. It is easy to get lost when looking into who is a member of a group, a member of another group, and a member of yet another group. While you may think this chain ends eventually, many environments have every 'Domain user' indirectly a member of 'Domain Admins'.

2. `Design` - Active Directory allows managing machines remotely via Group Policy Objects (GPOs). AD stores GPOs in a unique network share/folder called `SYSVOL`, where all domain-joined devices pull settings applied to them. Because it is a network-shared folder, clients access SYSVOL via the SMB protocol and transfer stored information. Thus, for a machine to use new settings, it has to call a Domain Controller and pull settings from SYSVOL - this is a systematic process, which by default occurs every 90 minutes. Every device must have a Domain Controller 'in sight' to pull this data from. The downside of this is that the SMB protocol also allows for code execution (a remote command shell, where commands will be executed on the Domain Controller), so as long as we have a set of valid credentials, we can consistently execute code over SMB on the Domain Controllers remotely. This port/protocol is available to all machines toward Domain Controllers. (Additionally, SMB is not well fit (generally Active Directory) for the zero-trust concepts.) If an attacker has a good set of privileged credentials, they can execute code as that account on Domain Controllers over SMB (at least!).

3. `Legacy` - Windows is made with a primary focus: it works out of the box for most of Microsoft's customers. Windows is `not` secure by default. A legacy example is that Windows ships with the broadcasting - DNS-like protocols `NetBIOS` and `LLMNR` enabled by default. These protocols are meant to be used if DNS fails. However, they are active even when it does not. However, due to their design, they broadcast user credentials on the wire (usernames, passwords, password hashes), which can effectively provide privileged credentials to anyone listening on the wire by simply being there. This blog post demonstrates the abuse of capturing credentials on the wire.