ACTIVE DIRECTORY FOR PENTESTERS IN 5 MINUTES

Active Directory for pentesters in 5 minutes

By Miguel Ángel Villalobos

https://www.linkedin.com/in/m7villalobos/

1. Understanding Active Directory: What It Is and Why It Matters

Imagine a large company. They need an organized and centralized way to manage:

- **Who works there:** User accounts (e.g., `john.doe`) with their identities and credentials.
- **What computers and servers exist:** Machine identities (e.g., `PC-Accounting`, `WebServer`), each with its own account in AD.
- **Who has permission to do what:** Access control (Authorization - AuthZ) to files, printers, applications, shared resources, etc.
- **What configuration and security rules apply:** Group Policies (GPOs) that define everything from password complexity to what software can run or how desktops are configured.

Active Directory (AD) is Microsoft's solution for managing all of this. It is fundamentally a **hierarchical and distributed database** (stored on Domain Controllers) along with a set of **services** that act as the central nervous system for Windows-based networks.

Its importance lies in:

- **Centralization:** Greatly simplifies the administration of resources and users.
- **Security:** Provides robust mechanisms for **Authentication** (AuthN - verifying who you are, primarily using Kerberos) and **Authorization** (AuthZ - defining what you can do, based on permissions and group memberships).
- **Scalability:** Designed to work for networks ranging from small businesses to global organizations with millions of objects.
- **Integration:** Countless enterprise applications and services (Exchange, SharePoint, SQL Server, many third-party applications) rely on AD for identity and access management.

Why is it a key target in offensive security (Red Team / Pentesting)?

Because AD literally holds the "keys to the digital kingdom." Compromising Active Directory at an administrator level (`Domain Admin` in a domain or, the holy grail, `Enterprise Admin` in the forest) means having practically absolute control over the organization's technological infrastructure. It allows an attacker to:

- Access confidential data on file servers.
- Deploy software (including malware or ransomware) across the entire network.
- Create user accounts (including hidden or highly privileged accounts).
- Reset passwords for any user (including other administrators).
- Modify security policies.
- Move laterally without restrictions throughout the network.

Therefore, obtaining elevated privileges in AD is almost always the **primary objective** of Red Team operations and internal penetration tests.

2. Essential AD Components

Understanding these parts is fundamental to identifying attack vectors and weak points.

- **Forest:** The highest-level container structure in AD. It is the complete security and administrative boundary. Contains one or more Domain Trees. The `Enterprise Admins` account (which exists only in the **forest root domain**) has authority over all domains in the forest.
- **Tree:** A set of one or more domains that share a contiguous and hierarchical DNS namespace (e.g., `company.local` as the tree root and `sales.company.local`, `dev.company.local` as child domains).
- **Domain:** The main administrative and replication partition unit. Each domain has its own AD database (though replicated among its DCs), specific policies, users, groups, and computers. Being a `Domain Admin` gives you total control over *that specific* domain.
- **Domain Controllers (DCs):** The **most critical** servers in the infrastructure. They are responsible for:
  - Storing a full (read/write, except on RODCs) copy of the domain's database (`NTDS.dit`). This file contains information about all objects in the domain, including the **password hashes** of users and computers.
  - Processing authentication requests (Kerberos, NTLM).
  - Replicating AD changes among themselves.
  - Applying Group Policies (GPOs).
  - They are the number one target within a domain once certain privileges are obtained.
- **Organizational Units (OUs):** Containers *within a domain* used to logically organize objects (users, groups, computers) (e.g., by department, geographic location). Their main purpose is:
  - **Delegate administrative control:** Specific permissions over an OU can be assigned to a group or user, allowing them to manage only the objects within that OU (e.g., a helpdesk that can only reset passwords for users in the "Sales" OU). *Misconfigured delegations are a common source of privilege escalation.*
  - **Apply Group Policies (GPOs):** GPOs are linked to OUs (as well as domains or sites) to apply specific configurations to the objects contained within them.

- **Objects:** The entities managed by AD:
  - **Users:** Represent people or service accounts. Key attributes: `samAccountName` (username), `SID` (Security Identifier - unique and immutable), `userAccountControl` (status flags), `memberOf` (groups they belong to), and their password hash (not directly visible).
  - **Groups:** Collections of users, computers, or other groups. Used to simplify permission assignment. There are security groups (with SIDs, used for permissions) and distribution groups (for email). **Privileged groups** are crucial, such as `Domain Admins`, `Enterprise Admins` (only in the root domain), `Administrators` (local group on each machine, but also a Builtin group in AD), `Schema Admins`, `Backup Operators`, `Account Operators`, `Server Operators`, `DNSAdmins`, etc. *The ultimate goal is often obtaining membership in one of these groups.*
  - **Computers:** Represent workstations and servers joined to the domain. They have their own identity, password (managed automatically by default), and SID in AD.
- **Group Policy Objects (GPOs):** Sets of configurations and rules that define aspects of the user environment and operating system (password policies, drive mappings, software installation, security restrictions, firewall settings, etc.). They are applied to containers (Sites, Domains, OUs). They are extremely powerful. *If an attacker with the right permissions can modify a GPO linked to many systems (or to the DCs), they can compromise them en masse.*
- **LDAP (Lightweight Directory Access Protocol):** The standard protocol used to query and (if permissions allow) modify information stored in Active Directory. It runs over TCP/IP, typically on port **389/TCP** (often unencrypted, be careful!) or **636/TCP (LDAPS)** for encrypted connections using SSL/TLS. *Essential for the reconnaissance and enumeration phase.*
- **Kerberos:** The **default and preferred** authentication protocol in modern AD environments. It relies on a system of "tickets" issued by the Key Distribution Center (KDC, a service running on DCs) to verify the identity of users and services without sending passwords over the network. Although conceptually secure, specific attacks exist against its implementation and usage (see below).
- **NTLM (NT LAN Manager):** An older authentication protocol based on a challenge-response mechanism. It is considered **less secure** than Kerberos. It is still used for compatibility with legacy systems or as a fallback mechanism if Kerberos fails. It is vulnerable to relay attacks (NTLM Relay) and Pass-the-Hash.
- **Trusts:** Relationships established between domains (within the same forest or between different forests) that allow users and groups from one domain to access resources in the other. Trusts can be transitive or non-transitive, one-way or two-way. *They can be an avenue for an attacker to "jump" from a compromised domain to another if the trust is misconfigured or if valid credentials in the trusted domain are obtained.*

3. Typical Phases of an AD Attack

An attack aimed at compromising AD usually follows these general steps, although variations exist:

1. **Reconnaissance & Enumeration:**
   - **Objective:** Map the domain/forest structure. Identify users (especially administrators and service accounts), privileged groups, Domain Controllers, critical servers, trust relationships, password policies, potential misconfigurations, and known vulnerabilities.
   - **Methods:**
     - LDAP queries (using tools like `PowerView` (PowerShell), `SharpHound` (C#, BloodHound's collector), `AdExplorer` (Sysinternals), `ldapsearch` (Linux), or custom scripts).
     - DNS record analysis to find DCs and other servers.
     - Network scanning (`nmap`) to identify open ports (LDAP, Kerberos, SMB, WinRM, RDP, etc.) on DCs and other targets.
     - Intensive use of **`BloodHound`**: This tool is fundamental. It uses collected information (usually via `SharpHound`) to visualize permission relationships (ACLs), group memberships, active sessions, and GPO policies in a graph. It allows for the rapid identification of attack paths for privilege escalation and lateral movement that would be very difficult to find manually.
2. **Initial Access:**
   - **Objective:** Gain an initial foothold on the network, usually by compromising a workstation or server with credentials of a low-privileged user.
   - **Methods:** Phishing (obtaining credentials or executing malware), exploiting vulnerabilities in internet-facing services or user software, **Password Spraying** attacks (trying common or seasonal passwords like `Summer2024!` against a large list of users), guessing weak passwords.
3. **Credential Access:**
   - **Objective:** Once inside a machine, extract credentials (cleartext passwords, NTLM hashes, Kerberos tickets) that may be stored in memory, registry, or files. These credentials can allow access to other machines or privilege escalation.
   - **Methods:**
     - Using tools like **`Mimikatz`**: Capable of extracting secrets from the LSASS (Local Security Authority Subsystem Service) process. Requires local Administrator or `SYSTEM` privileges on the target machine. Can obtain cleartext passwords, NTLM hashes, and Kerberos tickets.
     - **Kerberoasting:** Technique to request service tickets (TGS) for user accounts configured as Service Accounts (with a Service Principal Name - SPN). The encrypted part of the ticket (containing information signed with the service account's password hash) can be extracted and cracked offline. *Does not require elevated privileges to request the ticket initially*, only being an authenticated user in the domain.

- **AS-REP Roasting:** Similar to Kerberoasting, but targets user accounts that have the "Do not require Kerberos preauthentication" option enabled. Allows directly requesting part of the initial TGT (AS-REP) encrypted with the user's hash and cracking it offline. *Also does not require elevated privileges initially.*
- Manual or automated search for passwords in cleartext or configuration files (`web.config`, PowerShell scripts, history files, notes, etc.).
- Dumping the local SAM database (for local hashes, if not domain-joined or as an intermediate step) or the `NTDS.dit` file from a DC (if privileged access to the DC is already obtained).

4. **Privilege Escalation:**
   - **Objective:** Increase the permission level, either locally on the compromised machine (from standard user to Administrator/`SYSTEM`) or, more importantly, at the domain level (from standard user to a member of a privileged group like `Domain Admins`).
   - **Methods:**
     - Using stolen credentials (hashes, tickets, passwords) from accounts with higher privileges.
     - **Abuse of Permissions/ACLs (Access Control Lists):** Finding objects in AD (users, groups, computers, OUs, GPOs) where the compromised account (or a group it belongs to) has dangerous write permissions (e.g., `WriteMembers` on a group, `WriteProperty` on certain user attributes, `GenericAll`/`GenericWrite` on an object, permission to modify a GPO). Tools like **`BloodHound`** and **`PowerView`** are essential for finding these abuse paths.
     - **GPO Abuse:** If control over a GPO (or the OU/domain where it's linked) is obtained, policies can be modified to run scripts, install software, add users to local groups, etc., on all machines affected by that GPO.
     - Exploitation of local operating system vulnerabilities or specific AD vulnerabilities (e.g., Zerologon, PrintNightmare, if unpatched).
     - Abuse of insecure service configurations (e.g., services with `Unquoted Service Paths` or weak permissions).
5. **Lateral Movement:**
   - **Objective:** Use the obtained credentials or privileges to authenticate and execute code on other machines in the network, expanding control.
   - **Methods:**
     - **Pass-the-Hash (PtH):** Using a stolen NTLM hash (e.g., with `Mimikatz`) to authenticate to another machine that accepts NTLM authentication (common with SMB, WMI). Tools from the **`Impacket`** framework (Python) like `psexec.py`, `wmiexec.py`, `smbexec.py` are widely used for this.
     - **Pass-the-Ticket (PtT):** Using a stolen Kerberos ticket (TGT or TGS, e.g., with `Mimikatz` or extracted from memory) to authenticate to services that use Kerberos. Tools like **`Mimikatz`** ("kerberos::ptt") or **`Rubeus`** (C#) facilitate this.

    - **Overpass-the-Hash (OtH):** Using an NTLM hash to *request* a Kerberos ticket (TGT) and then using that ticket for Kerberos authentication (PtT).
    - Using stolen cleartext credentials with standard remote administration tools like `PsExec` (Sysinternals), WMI (Windows Management Instrumentation), WinRM (Windows Remote Management), RDP (Remote Desktop Protocol).
6. **Persistence & Domain Dominance:**
    - **Objective:** Establish mechanisms to maintain long-term access, even if passwords are changed or the initial access is detected, and finally achieve the ultimate goals (data exfiltration, ransomware deployment, sabotage, etc.).
    - **Methods (Advanced AD Persistence):**
        - **Golden Ticket:** A devastating attack. If the NTLM hash of the `krbtgt` account (a special domain account whose hash is used to sign all Kerberos TGTs) is obtained, an attacker can _forge TGTs_for any user (existing or not), with any privilege level (e.g., `Enterprise Admins`), and with long validity periods. Requires `Domain Admin`-level access to obtain the `krbtgt` hash. It is very stealthy at the authentication level.
        - **Silver Ticket:** Similar to a Golden Ticket, but a service ticket (TGS) is forged for a *specific service* (e.g., CIFS for file access, HOST for remote execution) using the password hash of that service's service account. Allows access to that specific service as any user. Requires the service account's hash.
        - **DCSync:** Abusing directory replication permissions (normally assigned to DCs and sometimes other accounts) to directly request a DC to replicate credential information (hashes) for any user, including the `krbtgt` account. Tools like `Mimikatz` ("lsadump::dcsync") implement this.
        - Modifying ACLs of critical objects: Adding full control for an attacker-controlled account over the `AdminSDHolder` object (whose permissions propagate to all protected accounts and groups), the Domain object, important GPOs, or key OUs.
        - Creating hidden accounts, adding accounts to privileged groups, modifying GPOs to run code periodically, creating scheduled tasks on DCs, installing backdoors or rootkits.
    - **Actions on Objectives:** Once control and persistence are established, perform the final actions: exfiltrate sensitive data, encrypt systems with ransomware, destroy information, etc.

4. Common Attacks Explained (Quick Summary)

- **Password Spraying:** Trying 1-3 passwords (`Summer2024`, `Welcome1!`, `Password123`) against HUNDREDS or THOUSANDS of accounts. Avoids lockouts from failed attempts on a single account. Very effective against weak password policies.
- **Kerberoasting:** Enumerate service accounts (SPNs). Request a service ticket (TGS) for them (any authenticated user can do this). Extract the part encrypted with the service

account's hash. Crack that hash offline. If the password is weak, access to the service account is gained.

- **AS-REP Roasting:** Find user accounts configured without Kerberos pre-authentication. Request an AS-REP (part of the initial TGT) encrypted with the user's hash. Crack offline. If the password is weak, access to the user account is gained.
- **Mimikatz / LSASS Dumping:** If you are a Local Administrator / `SYSTEM` on a machine, run `Mimikatz` (or similar tools) to dump the memory of the LSASS process and extract credentials (cleartext, NTLM hashes, Kerberos tickets) of users who have logged into that machine.
- **Pass-the-Hash (PtH):** Instead of using a password, directly use the stolen NTLM hash to authenticate to another system via the NTLM protocol (e.g., with `wmiexec.py –hashes :<ntlm_hash> target_ip`).
- **Pass-the-Ticket (PtT):** Inject a stolen Kerberos ticket (TGT or TGS) into the current session and use it to access resources/services that use Kerberos authentication.
- **Abuse of Permissions (ACLs / GPOs):** The "art" of finding misconfigurations or excessive delegations. Does a normal user have permission to add members to the `Domain Admins` group? Can a low-privilege group modify a critical GPO? **`BloodHound`** is the key tool for visualizing and finding these toxic relationships.
- **Golden Ticket:** The Kerberos "master key". Requires the NTLM hash of the `krbtgt` account (the domain's best-kept secret). Once obtained (usually requiring DA), it allows forging fake TGTs for any user, granting unlimited access and stealthy persistence.

5. What Can Stop an Attacker? (Key Defenses)

It is crucial for a pentester to know the defenses, both to understand how to attempt bypassing them and to be able to make useful recommendations.

- **Strong, Unique Passwords + Multi-Factor Authentication (MFA):** The absolute foundation. Robust password policies and, above all, MFA everywhere (VPN, OWA, privileged access) make the use of stolen credentials and brute-force/spraying attacks significantly harder.
- **Constant Patching and Vulnerability Management:** Applying security patches promptly, especially those affecting AD and DCs (e.g., Zerologon, PrintNightmare), closes the doors to direct exploitation.
- **Principle of Least Privilege:** Grant each account (user or service) *exactly* the permissions needed to perform its function, and nothing more. Periodically review memberships in privileged groups and permission delegations.
- **LAPS (Local Administrator Password Solution):** A free Microsoft tool that manages random, unique passwords for the local Administrator account on each domain-joined workstation and server, rotating them periodically. Greatly complicates lateral movement based on reusing the same local admin password.

- **Tier Model / Secure Administration (PAWs - Privileged Access Workstations):** Strictly segregate the accounts and workstations used to manage critical systems (Tier 0: DCs, AD; Tier 1: Servers; Tier 2: Workstations) from the normal user environment. AD administrators should only use dedicated, highly secured accounts and machines. Very effective if implemented correctly.
- **Credential Guard & Remote Credential Guard:** Windows features (based on Virtualization-Based Security - VBS) that isolate the LSASS process and protect stored credentials (NTLM hashes, Kerberos TGTs) against theft, even by code with `SYSTEM` privileges. Makes `Mimikatz` and similar attacks much harder.
- **Advanced Monitoring and Detection (EDR, SIEM, Microsoft Defender for Identity - MDI):** Use Endpoint Detection and Response (EDR) tools on endpoints, a Security Information and Event Management (SIEM) to correlate logs, and specific AD monitoring solutions like **Microsoft Defender for Identity (MDI)** (formerly Azure ATP) that analyze authentication traffic and AD behavior to detect known attack patterns (Pass-the-Hash, Golden Ticket, suspicious Kerberoasting, etc.) and alert in real-time.
- **Network Segmentation and Firewalls:** Limit communication between different network segments (e.g., preventing workstations from communicating directly with each other via SMB, allowing only specific machines to contact DCs on necessary ports).
- **System and AD Hardening:** Apply recommended security configurations: disable obsolete protocols (SMBv1, LM/NTLMv1), require SMB and LDAP signing, configure Kerberos securely, enable advanced logging, etc.

Active Directory is the heart of the IT infrastructure in the vast majority of organizations using Windows. Its inherent complexity and the multitude of ways it can be configured (and misconfigured) make it an extensive and attractive attack surface. For any offensive security professional (Red Team, pentesting), understanding AD architecture, its authentication protocols, its key objects, and, above all, the common techniques to enumerate it, exploit its weaknesses, escalate privileges, and move laterally, is absolutely fundamental. Mastering these techniques not only allows for effective assessment of an organization's security posture but also helps them strengthen their defenses against real threats.

The next logical step would be to delve deeper into the specific tools and commands for each phase: reconnaissance (`PowerView` , `BloodHound` / `SharpHound`), credential harvesting (`Mimikatz` , `Rubeus`), lateral movement (`Impacket` , `PsExec`), etc. We'll cover this later.
