Attacking and Defending Windows Active Directory

0.Introduction:

In this short research we are going to talk about one of the most common attack vectors that adversaries use in order to compromise Windows Active Directory, and most importantly how to prevent these attacks and alert when they are being executed. Nearly every larger corporation or a company utilizes AD in their environment. This is the main reason why security awareness is crucial when dealing with this kind of technology. Active Directory security is vital to protect user credentials, company systems, sensitive data, software applications, and more from unauthorized access. A security compromise of AD can essentially undermine the integrity of your identity management infrastructure, leading to catastrophic levels of data leakage or system corruption/destruction . If a cyber attacker is able to access the AD system, they can potentially access all connected user accounts, databases, applications, and all types of crucial files and information. Therefore, an Active Security compromise, particularly those that are not caught early and are left unsanitized, can lead to widespread fallout from which it may be difficult to recover [1].

1.Windows Active Directory:

Large organizations depend on Windows Active Directory (AD) to maintain order in the chaos that is managing users, computers, permissions, and file servers. AD is a directory service provided by Microsoft. A directory service is a hierarchical arrangement of objects which are structured in a way that makes access easy. However, functioning as a locator service is not AD's exclusive purpose. It also helps

organizations have a central administration over all the activities carried out in their networks. This means that AD is centralized, i.e. it has a centralized computer which is called a Domain Controller. DC is a heart of the Windows Active Directory. Organizations primarily use AD to perform authentication and authorization. It is a central database that is contacted before a user is granted access to a resource or a service. Once the user is authenticated, then the AD performs authorization checks if that user can access specific resources. If the user checks out on both counts, access is granted [2].AD relies on some of the common networking protocols such as DNS and SMB, RPC but also uses some which deal especially with authentication of domain-joined computer like Kerberos .

2. Common threats to Active Directory systems:

In this research we are only going to cover some of the most exploited protocols and methods that APT's (Advanced Persistent Threat) use in order to compromise AD because the list is certainly not exhaustive.

Default Security Settings - AD has a set of predetermined, default security settings created by Microsoft which initialize when you install Active directory. These security settings may not be ideal for your organization's needs. Additionally, these default security settings are well-understood by hackers. [1].

Broken administrative and privileged user access roles - What we see very often are users that are present on the AD and have misconfigured access roles which can lead to devastating results if an attacker manages to compromise these accounts. Domain user accounts and other administrative users may have full, privileged access to AD. It is very likely that most employees, even those in IT, do not need high-level or superuser privileges.

Weak password policies, password reuse and passwords in general – Password were and will remain to be the main reason why attackers have high success rates when it comes to compromise. Passwords are very important when it comes to AD environments because one password can unlock many doors and users often use the same password for most of their accounts. Brute force attacks are also very common because password policies and lockout thresholds are not present. Password complexity is also a major concern when we talk about AD.

Unpatched vulnerabilities in AD servers - System administrators can't evade zero days and this is something that is inevitable but staying up to date with patches and security updates is crucial when operating in important system environments. This is often a method that attackers use in order to compromise Domains. Recent vulnerability named Zerologon showed how neglecting updates and patching can be damaging when a few APT's where found using this CVE in the wild.[1]

Not isolating network resources such as critical servers - Most internal networks are flat. Any computer can typically connect to any other, meaning that any workstation can often connect to any server, including critical assets such as financial or Human resources databases. Given the current threat profile, this paradigm needs to change. It is not appropriate for critical servers to be directly accessible from any random computer on the network. This is only one of the reasons why network segmenting should be implemented in such systems.[3]

Too many Domain Admins – This can easily be appended to some of the previously mentioned security issues but I think it deserves a place of it's own. Active Directory administration is typically performed by a small number of people. The number of Domain admins (DA) typically exceeds the number of actual AD admins. Domain Admins members have FULL administrative rights to all workstations, servers, Domain Controllers, Active Directory, Group Policy by default [3]. This is too much power for any one account in today's modern enterprise and what researches find is that a great number of these accounts exist inside domains which itself presents a massive security issue because it allows attackers to have multiple attack vectors when compromising the domain.

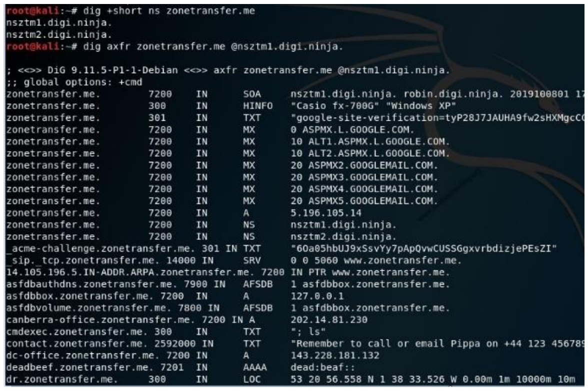
Over-permissioned Service Accounts – This also falls under the category of misconfigured access roles inside AD, but this one is very important because attackers love service accounts. One of the most famous AD attacks is actually "*kerberoasting"* which we will cover later in this paper, and this attack in particular targets service accounts. By gaining access to these accounts with improper permissions can lead to full domain compromise. Vendors have historically required Domain Admin rights for Service Accounts even when the full suite of rights DA provides is not actually required, though It makes the product easier to test and deploy. Keep in mind that a service running under the context of a service account has that credential in LSASS (protected memory) which can be extracted by an attacker. If the stolen credential has admin rights, the domain may be quickly compromised due to a single Service Account. [3] We have also seen some interesting attacks which leverage Service account privileges such as *SeImpersonatePrivilege* which is by default enabled on all Service accounts in case the server needs to impersonate the user

Insufficient logging and monitoring – this one is self-explanatory and is required in almost every system, even smaller ones. Logging is not meant only for security but also for monitoring user performance and things of that nature. Common events should be monitored such as unnatural account creation or unusual Kerberos traffic that can indicate compromise. Most of these options are built in Windows Server but many organizations use other tools that help them make more "surgical" analysis.

3. Commonly exploited protocols used in AD:

In this section we will cover some of the most common protocols that Active Directory uses and which are interesting when we talk about security of AD.

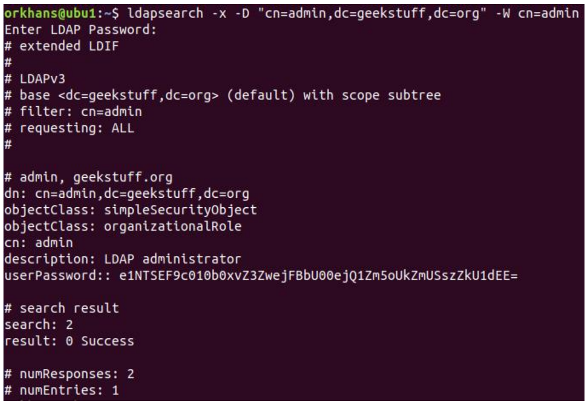
DNS - Domain Name System (DNS) is one of the industry-standard suite of protocols that comprise TCP/IP, and together the DNS Client and DNS Server provide computer name-to-IP address mapping name resolution services to computers and users. In Windows Server 2016, DNS is a server role that you can install by using Server Manager or Windows PowerShell commands. Active Directory Domain Services (AD DS) uses DNS as its domain controller location mechanism. When any of the Active Directory operations is performed, such as authentication or authorization, updating, or searching, computers use DNS to locate Active Directory domain controllers. If the DNS fails it resorts to some other methods. In addition, domain controllers use DNS to locate each other [4]. When DNS queries fail the computers use NetBIOS and LLMNR which are protocols used to resolve host names on local networks. Their main function is to resolve host names to facilitate communication between hosts on local networks. NetBIOS is generally outdated and can be used to communicate with legacy systems. LLMNR is designed for consumer-grade networks in which a domain name system (DNS) server might not exist. NetBIOS is enabled by default on Microsoft Windows 2000 machines and above (while existing independently of DNS in older versions), and LLMNR is enabled on Microsoft Windows Vista machines and above [5].DNS commonly uses port 53 though it can use any port number specified in it's configuration file. When it runs over TCP this is when it becomes tricky. If you see DNS running on TCP then it's probably configured to use DNS zone transfers. DNS zone transfers using the AXFR protocol are the simplest mechanism to replicate DNS records across DNS servers. To avoid the need to edit information on multiple DNS servers, you can edit information on one server and use AXFR to copy information to other servers. We can issue a simple command such as *"dig"* to get information about all the internal domain names of a certain domain using DNS Zone Transfers.



```
root@kali:~# dig +short ns zonetransfer.me
nsztm1.digi.ninja.
nsztm2.digi.ninja.
root@kali:~# dig axfr zonetransfer.me @nsztm1.digi.ninja.

; <<>> DiG 9.11.5-P1-1-Debian <<>> axfr zonetransfer.me @nsztm1.digi.ninja.
;; global options: +cmd
zonetransfer.me.        7200    IN      SOA     nsztm1.digi.ninja. robin.digi.ninja. 2019100801 17
zonetransfer.me.        300     IN      HINFO   "Casio fx-700G" "Windows XP"
zonetransfer.me.        301     IN      TXT     "google-site-verification=tyP28J7JAUHA9fw2sHXMgcCC
zonetransfer.me.        7200    IN      MX      0 ASPMX.L.GOOGLE.COM.
zonetransfer.me.        7200    IN      MX      10 ALT1.ASPMX.L.GOOGLE.COM.
zonetransfer.me.        7200    IN      MX      10 ALT2.ASPMX.L.GOOGLE.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX2.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX3.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX4.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX5.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      A       5.196.105.14
zonetransfer.me.        7200    IN      NS      nsztm1.digi.ninja.
zonetransfer.me.        7200    IN      NS      nsztm2.digi.ninja.
_acme-challenge.zonetransfer.me. 301 IN TXT     "6Oa05hbUJ9xSsvYy7pApQvwCUSSGgxvrbdizjePEsZI"
_sip._tcp.zonetransfer.me. 14000 IN     SRV     0 0 5060 www.zonetransfer.me.
14.105.196.5.IN-ADDR.ARPA.zonetransfer.me. 7200 IN PTR www.zonetransfer.me.
asfdbauthdns.zonetransfer.me. 7900 IN   AFSDB   1 asfdbbox.zonetransfer.me.
asfdbbox.zonetransfer.me. 7200  IN      A       127.0.0.1
asfdbvolume.zonetransfer.me. 7800 IN    AFSDB   1 asfdbbox.zonetransfer.me.
canberra-office.zonetransfer.me. 7200 IN A      202.14.81.230
cmdexec.zonetransfer.me. 300    IN      TXT     "; ls"
contact.zonetransfer.me. 2592000 IN     TXT     "Remember to call or email Pippa on +44 123 45678
dc-office.zonetransfer.me. 7200 IN      A       143.228.181.132
deadbeef.zonetransfer.me. 7201  IN      AAAA    dead:beaf::
dr.zonetransfer.me.     300     IN      LOC     53 20 56.558 N 1 38 33.526 W 0.00m 1m 10000m 10m
```

Figure 3.1 DNS Zone transfer performed on the domain "reydes.com"

***<u>LDAP</u>* -** Lightweight Directory Access Protocol is an open and cross platform protocol used for directory services authentication. LDAP provides the communication language that applications use to communicate with other directory services servers. Directory services store the users, passwords, and computer accounts, and share that information with other entities on the network [6]. We can look at LDAP as a way to talk to Active Directory. We construct LDAP queries in order to communicate with the Domain Controller. LDAP uses three ways of authentication. *Anonymous authentication* is especially dangerous since we give permission to anyone to construct queries and enumerate AD. *Unauthenticated authentication* is used for logging purposes only and should not be granted to clients. *Name/Password authentication* grants access to the server based on the credentials supplied. SASL authentication binds the LDAP server to another authentication mechanism, like Kerberos. The LDAP server uses the LDAP protocol to send an LDAP message to the other authorization service. That initiates a series of challenge response messages that result in either a successful authentication or a failure to authenticate. It's important to note that LDAP passes all of those messages in clear text by default, so anyone with a network sniffer like *Wireshark* for example can read the packets. You need to add TLS encryption or similar to keep your usernames and passwords safe [6]. Tools such as "*ldapsearch*" can be used to enumerate AD systems using LDAP queries.

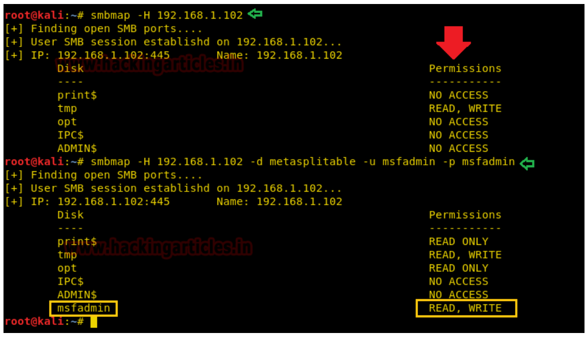


```
orkhans@ubu1:~$ ldapsearch -x -D "cn=admin,dc=geekstuff,dc=org" -W cn=admin
Enter LDAP Password:
# extended LDIF
#
# LDAPv3
# base <dc=geekstuff,dc=org> (default) with scope subtree
# filter: cn=admin
# requesting: ALL
#

# admin, geekstuff.org
dn: cn=admin,dc=geekstuff,dc=org
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: e1NTSEF9c010b0xvZ3ZwejFBbU00ejQ1Zm5oUkZmUSszZkU1dEE=

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

Figure 3.2: LDAP enumeration using ldapsearch

Using these queries we can enumerate server naming context and domain name, users, users in Domain Admins groups, Enterprise admins or Remote Desktop users. If someone prefers a graphical interface in order to interact with LDAP you can use "*jxplorer*".

***<u>SMB</u>* -**The Server Message Block (SMB) is a network protocol that enables users to communicate with remote computers and servers — to use their resources or share, open, and edit files. It's also referred to as the server/client protocol, as the server has a resource that it can share with the client. Like any network file sharing protocol, SMB needs network ports to communicate with other systems. Originally, it used port 139 that allowed computers to communicate on the same network. But since Windows 2000, SMB uses port 445 and the TCP network protocol to "talk" to other computers over the internet. The SMB protocol creates a connection between the server and the client by sending multiple request response messages back and forth. Imagine your team is working on a large project. You might want to be able to share and edit files that are stored in one place. The SMB protocol allows your team members to use these shared files as if they were on their own hard drives. They can choose to mount the whole system as it were a real hard drive or choose tools that directly communicate with the service [7]. Like many other protocols SMB also uses authentication, which can be anonymous but usually you would require some sort of credentials to access shares on a shared network. On Windows, SMB can run directly over TCP/IP without the need for NetBIOS over TCP/IP. This will use, as you point out, port 445. On other systems, you'll find services and applications using port 139. This means that SMB is running with NetBIOS over TCP/IP**.** This is the reason an port scan of a domain-joined computer or a Domain controller shows both port 139 and 445.



```
root@kali:~# smbmap -H 192.168.1.102
[+] Finding open SMB ports....
[+] User SMB session establishd on 192.168.1.102...
[+] IP: 192.168.1.102:445       Name: 192.168.1.102
        Disk                                                   Permissions
        ----                                                   -----------
        print$                                                 NO ACCESS
        tmp                                                    READ, WRITE
        opt                                                    NO ACCESS
        IPC$                                                   NO ACCESS
        ADMIN$                                                 NO ACCESS
root@kali:~# smbmap -H 192.168.1.102 -d metasplitable -u msfadmin -p msfadmin
[+] Finding open SMB ports....
[+] User SMB session establishd on 192.168.1.102...
[+] IP: 192.168.1.102:445       Name: 192.168.1.102
        Disk                                                   Permissions
        ----                                                   -----------
        print$                                                 READ ONLY
        tmp                                                    READ, WRITE
        opt                                                    READ ONLY
        IPC$                                                   NO ACCESS
        ADMIN$                                                 NO ACCESS
        msfadmin                                               READ, WRITE
root@kali:~#
```

Figure 3.3 Mapping SMB shares using "smbmap".Note: this can be done with many other tools such as smbclient.

After successful authentication you can enter shares and write inside of them, if you have sufficient permissions to do so.

To look for possible exploits to the SMB version it is important to know which version is being used.You can use Metasploit in order to enumerate the SMB version or use a NMAP script. For example, one would execute the following Metasploit module : "*auxiliary/scanner/smb/smb_version*".

Many previous versions of SMB are vulnerable, because of the pure nature of this protocol. *EternalBlue* is probably one of the most famous bugs because it was used by WannaCry malware that infected more than 250 000 computers in 150 different countries. Some of the most recent exploits targeting SMB are also to be considered like *SMBGhost* or *SMBleed* vulnerability. Most penetration testers and attacker use tools such as *crackmapexec* that have more "evil" capabilities such as gaining a shell through pass the hash techniques, password spraying or brute-forcing.

MSRPC/RPC - Microsoft Remote Procedure Call, also known as a function call or a subroutine call, is a protocol that uses the client server model in order to allow one program to request service from a program on another computer without having to understand the details of that computer's network. MSRPC was originally derived from open source software but has been developed further and copyrighted by Microsoft [8]. Depending on the host configuration, the RPC endpoint mapper can be accessed through TCP and UDP port 135, via SMB with a null or authenticated session (TCP 139 and 445), and as a web service listening on TCP port 593.

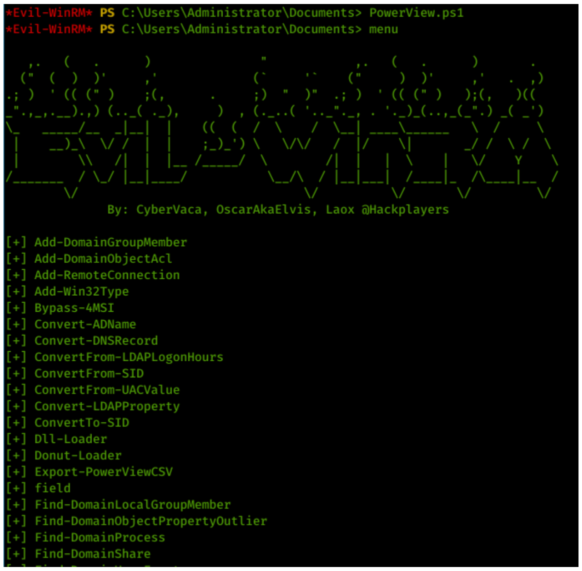
One tools mentioned in the book "Network security Assessment 3rd edition" is *rpcdump*, which "dumps" all the open RPC service on windows machines and enumerates them. It also helps in further enumerations with tools mentioned below.

You can query RPC Servics on a machine with tools like *"rpcclient"* and "*rpcdump.py*". Rpcclient utility is used to interact with RPC endpoints via named pipes. The following lists commands that you can issue to SAMR, LSARPC, and LSARPC-DS interfaces upon establishing a SMB session (often requiring credentials): *querydispinfo, enumdomusers, queryuser <0<rid>, queryusergroups <0xrid>, lookupnames, enumdomgroups, enumdomains.*

WinRM -Windows Remote Management (WinRM) is the Microsoft implementation of WS-Management-Protocol, a standard Simple Object Access Protocol , firewall-friendly protocol that allows hardware and operating systems, from different vendors, to interoperate. The WS-Management protocol specification provides a common way for systems to access and exchange management information across an IT infrastructure [9]. The easiest way to detect whether WinRM is available is by seeing if the port is opened. WinRM will listen on one of two ports:

5985/tcp (HTTP), 5986/tcp (HTTPS)**.** If one of these ports is open, WinRM is configured and you can try entering a remote session. We can interact with WinRM using Powershell by adding a client to a list of trusted hosts and then using "*Invoke-Command*" to execute certain commands on the remote computer. If you want to drop right into an interactive PowerShell session, use the "*Enter-PSSession*" function. One can easily brute-force WinRM protocol using previously mentioned tools such as "*crackmapexec*" .

Another great tool that attackers and penetration testers use is "*evil-winrm*" which is written in Ruby. It has download, upload and pass-the-hash capabilities.



```
*Evil-WinRM* PS C:\Users\Administrator\Documents> PowerView.ps1
*Evil-WinRM* PS C:\Users\Administrator\Documents> menu

              By: CyberVaca, OscarAkaElvis, Laox @Hackplayers

[+] Add-DomainGroupMember
[+] Add-DomainObjectAcl
[+] Add-RemoteConnection
[+] Add-Win32Type
[+] Bypass-4MSI
[+] Convert-ADName
[+] Convert-DNSRecord
[+] ConvertFrom-LDAPLogonHours
[+] ConvertFrom-SID
[+] ConvertFrom-UACValue
[+] Convert-LDAPProperty
[+] ConvertTo-SID
[+] Dll-Loader
[+] Donut-Loader
[+] Export-PowerViewCSV
[+] field
[+] Find-DomainLocalGroupMember
[+] Find-DomainObjectPropertyOutlier
[+] Find-DomainProcess
[+] Find-DomainShare
```

Figure 3.4 Using evil-winrm

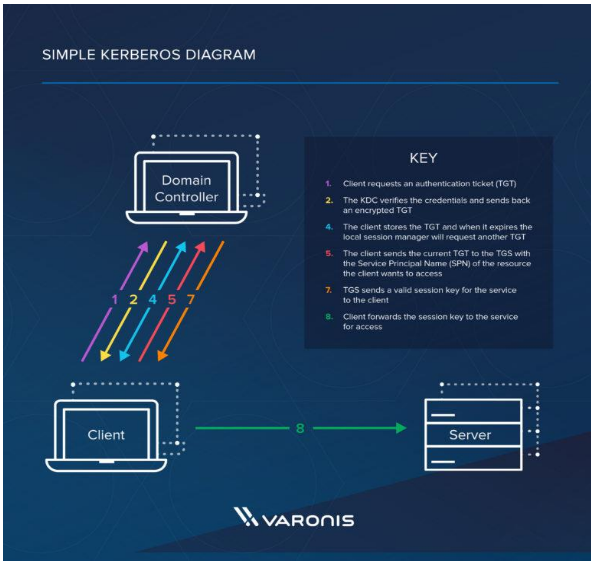
Kerberos – According to a myth, Cerberus guards the Gates to the Underworld. He's a big 3 headed dog with a snake for a tail and a really bad temper. Kerberos is probably the most important protocol that is implemented inside AD, and its also the most exploited one. Kerberos is a network authentication protocol. It is designed to provide strong authentication for client server applications by using secret-key cryptography [10]. Kerberos uses symmetric-key encryption and requires trusted third-party authorization to verify user identities. Since Kerberos requires 3 entities to authenticate and has an excellent track record of making computing safer, the name really does fit. It is currently the default authorization technology used by Microsoft Windows, and implementations of Kerberos exist in Apple OS, FreeBSD, UNIX, and Linux. There is also an implementation of this protocol by MIT. [11]. Before Kerberos, Microsoft used an authentication technology called NTLM. NTLM stands for NT Lan Manager and is a challenge-response authentication protocol. The target computer or domain controller challenge and check the password, and store password hashes for continued use. The biggest difference between the two systems is the third-party verification and stronger encryption capability in Kerberos. This extra step in the process provides a significant additional layer of security over NTLM. This is the main reason why Windows uses Kerberos over NTLM, but NTLM is still supported because of legacy systems that are used in many organizations. [10]. In order to better understand Kerberos and it's misuses we will briefly explain the inner workings of the protocol.

Here are the most basic steps taken to authenticate in a Kerberos environment.

1. Client requests an authentication ticket (TGT ticket) from the Key Distribution Center (KDC)
2. The KDC verifies the credentials provided and sends back an encrypted TGT and session key
3. The TGT is encrypted using the Ticket Granting Service (TGS) secret key (*krbtgt*)
4. The client stores the TGT and when it expires the local session manager will request another TGT (this process is transparent to the user). TGT is usually stored in cache.

If the Client is requesting access to a service or other resource on the network, this is the process:

5. The client sends the current TGT to the TGS with the Service Principal Name (SPN) of the resource the client wants to access.
6. The KDC verifies the TGT of the user and that the user has access to the service
7. TGS sends a valid session key for the service to the client
8. Client forwards the session key to the service to prove the user has access, and the service grants access.



KEY

1. Client requests an authentication ticket (TGT)
2. The KDC verifies the credentials and sends back an encrypted TGT
4. The client stores the TGT and when it expires the local session manager will request another TGT
5. The client sends the current TGT to the TGS with the Service Principal Name (SPN) of the resource the client wants to access
7. TGS sends a valid session key for the service to the client
8. Client forwards the session key to the service for access

Figure 3.5: Simple Kerberos authentication

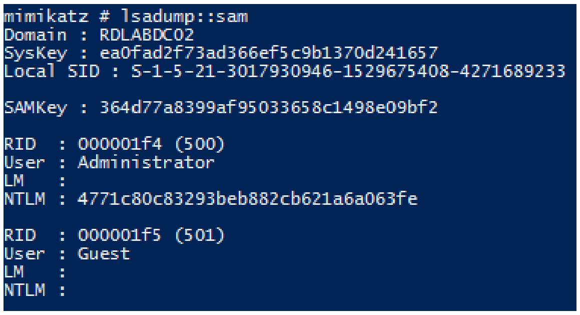
Note: KDC or a key distribution center is a part of the Domain controller and it manages “tickets”.It is the main service of Kerberos, responsible of issuing the tickets, installed on the DC (Domain Controller). It is supported by the AS (Authentication Service), which issues the TGTs. TGT or Ticket Granting Tickets are main tickets that Kerberos uses and are needed for further communication. TGT’s are signed by the KDC using a krbtgt hash which is machine-generated and nearly impossible to break. TGS are Ticket Granting Service tickets also issued by the KDC which are issued in order to communicate and interact with system resources, Services such as SQL, WEB server, FTP and such have so called Service accounts which are identified by their SPN(Service Principal Number). The TGS is signed by the hash of the Service account user. We will see why this is important later.

Another very important part of Kerberos, when talking about security is PAC. The PAC (Privilege Attribute Certificate) is a structure included in almost every ticket. This structure contains the privileges of the user and it is signed with the krbtgt key. It is possible for services to verify the PAC by communicating with the KDC, although this does not happen often. Nevertheless, the PAC verification consists of checking only its signature, without inspecting if privileges inside of PAC are correct.

4. Attacking Windows Active Directory:

In this section we will cover some of the most common attacks that APT’s and hackers use to compromise Active directory, exfiltrate data or just corrupt or destroy these systems. This list, as mentioned, is certainly not exhaustive and many other attacks are utilized in the wild. After explaining each attack I will explain methods that defense and blue teams can apply in order to prevent them or even alert when these attack are being executed.

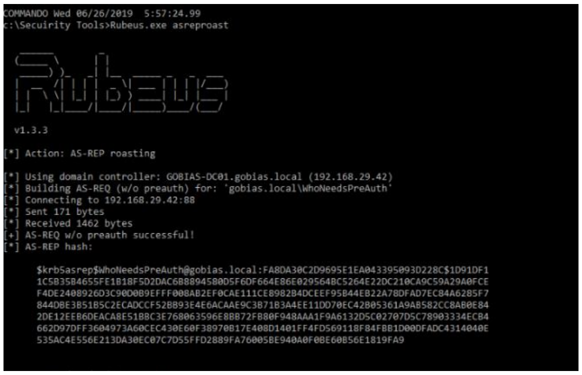
Password attacks: Passwords, as mentioned are one of the most controversial themes when it comes to cyber security just because people don’t understand how easy an attacker can crack them or guess them suing OSINT techniques or some other methods such as phishing campaigns or using social engineering. Brute force attacks are common but what is more common when it comes to AD is password spraying. Password spraying is basically trying one passwords that the attacker gathered using previous methods, against many domain user accounts, which again, can be gathered enumerating MSRPC, LDAP, SMB or other protocols. This method is useful because users tend to re-use passwords on different services. More critical passwords from Domain Admins or Administrators that log onto User machines can be easily extracted from lssas.exe process using tools such as Mimikatz, developed by Benjamin Delpy. Locally, passwords and hashes are stored inside a file called a SAM file. They are encrypted using a boot key that is stored inside the SYSTEM file. By extracting these files one can easily crack the passwords offline or even use hashes for pass-the-hash attacks. Domain Controller has something called a NTDS.dit file, which when obtained, provides the attacker with all the passwords and hashes of domain-joined computers, including a Domain controller. Setting lockout thresholds is something that is obligatory when talking about securing AD and is often neglected. Password complexity is also critical. Group policies should be used inside the AD that require passwords of greater lengths that use symbols, numbers and capital letters.



```
mimikatz # lsadump::sam
Domain : RDLABDC02
SysKey : ea0fad2f73ad366ef5c9b1370d241657
Local SID : S-1-5-21-3017930946-1529675408-4271689233

SAMKey : 364d77a8399af95033658c1498e09bf2

RID  : 000001f4 (500)
User : Administrator
LM   :
NTLM : 4771c80c83293beb882cb621a6a063fe

RID  : 000001f5 (501)
User : Guest
LM   :
NTLM :
```

Figure 4.1 dumping SAM file contents using mimikatz.exe

ASREP Roasting: This one is more of an enumeration technique used to determine valid users on the Domain but is valuable from the attackers perspective. Even though it's uncommonly seen in the wild because it's not something that is default on Domain joined computers it is certainly worth mentioning. The ASREP Roast attack looks for users without Kerberos pre-authentication required attribute**.** Pre-authentication is the first step in Kerberos authentication, and is designed to prevent brute-force password guessing attacks [12]. That means that anyone can send an AS_REQ request to the DC on behalf of any of those users, and receive an AS_REP message. Users that have this attribute don't have to encrypt the challenge that the DC sends to a user when the user asks for a TGT. This means we can obtain a valid TGT whose part is signed with the users hash. Run that through a hash rainbow table and you have yourself a valid password of a domain joined computer. Users that have this attribute set can easily be enumerated using LDAP queries or programs such as *"GetNPUsers.py"* from the impacket tool suite. This attack not only gives you a user password, but can be used by ATP's like a persistence mechanism. Administrators should always check for these attributes and disable them if present.



```
COMMANDO Wed 06/26/2019  5:57:24.99
c:\Secuirity Tools>Rubeus.exe asreproast

  v1.3.3

[*] Action: AS-REP roasting

[*] Using domain controller: GOBIAS-DC01.gobias.local (192.168.29.42)
[*] Building AS-REQ (w/o preauth) for: 'gobias.local\WhoNeedsPreAuth'
[*] Connecting to 192.168.29.42:88
[*] Sent 171 bytes
[*] Received 1462 bytes
[+] AS-REQ w/o preauth successful!
[*] AS-REP hash:

      $krb5asrep$WhoNeedsPreAuth@gobias.local:FA8DA30C2D9695E1EA043395093D228C$1D91DF1
      1C5B35B4655FE1B18F5D2DAC6B8894580D5F6DF664E86E029564BC5264E22DC210CA9C59A29A0FCE
      F4DE2408926D3C90D0B9EFFF008AB2EF0CAE111CE8982B4DCEEF95B44EB22A78DFAD7EC84A6285F7
      844DBE3B51B5C2ECADCCF52BB93E4E6ACAAE9C3B71B3A4EE11DD70EC42B05361A9AB582CC8AB0E84
      2DE12EEB6DEACA8E51BBC3E768063596E8BB72FB80F948AAA1F9A6132D5C02707D5C78903334ECB4
      662D97DFF3604973A60CEC430E60F38970B17E408D1401FF4FD569118F84FBB1D00DFADC4314040E
      535AC4E556E213DA30EC07C7D55FFD2889FA76005BE940A0F0BE60B56E1819FA9
```

Figure 4.2 Performing ASREP roast using Rubeus

Golden Ticket: This is one of the hardest attacks to do because it requires access to the Domain Controller. The security of the Kerberos protocol is rooted in the use of shared secrets to encrypt and sign messages. Some of these secrets are known to the trusted third-party (the *Key Distribution Center (KDC)* in Kerberos) and clients, but one in particular is known only to the KDC: the secret to the krbtgt user, whose password hash is used to sign or encrypt Kerberos tickets issued by the KDC. In other words, compromising the krbtgt hash allows an adversary to behave as if they were Active Directory, more precisely a Domain Controller issuing valid TGT tickets [13].Once the attacker takes control of the DC, he can dump the lsass memory and extract the krbtgt hash, and then it's game over. With this they can issue tickets for users that don't exist, add users to groups in which they don't belong, or issue tickets with lifetimes far beyond the configured maximum. In effect, the adversary in Active Directory can access any resource they choose. This capability is both extremely powerful and difficult to detect [13].

Kerberoasting: This type of attack is probably the most known one, and it exploits the configuration of Kerberos. Kerberoasting is a pervasive attack technique targeting Active Directory service account credentials. Advanced and lesser-skilled attackers alike favor Kerberoasting because the technique can be carried out by any user on a domain—not just administrators. It is also an "offline" attack that doesn't require any packets be sent to the targeted service—traffic that would be logged and quite possibly trigger alerts. Kerberoasting, instead, takes advantage of human nature nearly as much as it exploits known security weaknesses in Kerberos authentication for Active Directory. At its core, Kerberoasting is a password-cracking attack in which credentials are stolen from memory and cracked offline [14] . The technique allows the attackers, as valid domain users, to request a Kerberos service ticket for any service, capture that ticket granting service (TGS) ticket from memory, which is signed by the Service account password's hash and then attempt to crack the service credential hash offline using any number of password-cracking tools, such as Hashcat, John the Ripper, and others. The attacker first needs to acquire the SPN(Service Principal Number) of the service accounts. This means that these accounts have users created for them and are not machine called accounts, whose passwords are generated by the Domain controller and nearly impossible to crack. In addition they change every 30 days. SPN's can be easily gather using impacket's *GetUserSPNs.py* script.

Silver Ticket : This is another attack that uses TGS and Service accounts and although it is lesser known than the golden counterpart it is in some cases even more deadlier. It is basically the same as the Golden Ticket attack, in which we forged TGT's with unlimited access. This time we create our own TGS tickets after gaining the Service account hash. Part of the TGS is also encrypted with the krbtgt account's hash and that part is the PAC which we mentioned earlier. PAC contains information about the user requesting access to the service, for example if that user is an administrator. We can easily forge a PAC giving it a user of our choice granting him full administrative permission over that service because the PAC is often not checked inside the domain for performance issues. Password complexity is crucial when dealing with these kinds of attacks and they can prevent not just this, but the majority of the attacks explained here.

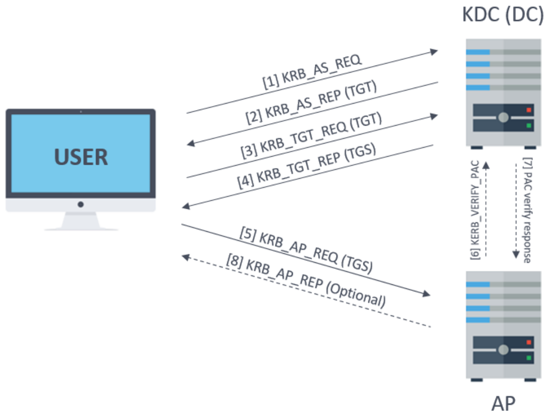


Figure 4.3 PAC verification and Kerberos cycle

Skeleton key: A skeleton key is a key that is used to open all the doors inside a house. This is exactly what this attack is about. The Skeleton Key is a particularly scary piece of malware targeted at Active Directory domains to make it alarmingly easy to hijack any account. This malware injects itself into LSASS and creates a master password that will work for any account in the domain.It uses Windows API hooking to inject itself. Existing passwords will also continue to work, so it is very difficult to know this attack has taken place unless you know what to look for. In order to perpetrate this attack, the attacker must have Domain Admin rights**.** This attack must be performed on each and every domain controller for complete compromise, but even targeting a single domain controller can be effective [15]. This attack is easily performed using *mimikatz* created by Benjamin Delpy by issuing a command: *misc::skeleton.* It sets the password as "mimikatz" but this can be easily changed. This attack can be easily mitigated by runing lsass.exe as a protected process, it forces an attacker to load a kernel mode driver. It can be spotted by parsing event log files for events 7045 **-** A service was installed in the system. (Type Kernel Mode driver), 4673 – Sensitive Privilege Use ("Audit privilege use" must be enabled), 4611 – A trusted logon process has been registered with the Local Security Authority ("Audit privilege use" must be enabled).

LLMNR poisoning – This is an attack that one can use when they have access to the internal network. An attacker can use tools such as "responder" to poison the network. When a user on a computer types an incorrect DNS name, for example by using an ftp server the DNS server won't be able to provide the answer. Tools like responder can act as a DNS server and provide the user with the place to authenticate. Then the attacker can grab the user's hash and attempt to crack it offline or even use ntlm-hash-relaying by abusing smb signing on a different computer and gett a shell. Note that in order to do this user that is authenticating needs to be an administrator on the targeted computer and SMB signing must be disabled. Defenders should always check for rouge computers on their networks and smb signing should always be enabled.

DCSync attack **-** The DCSync attack simulates the behavior of a Domain Controller by impersonating it so to say and asks other Domain Controllers to replicate information with protocols such as Directory Replication Service Remote Protocol (MS-DRSR). This is hard to mitigate since MS-DRSR is a valid and necessary function of Active Directory which in turn means it cannot be disabled. By default only Domain Admins, Enterprise Admins, Administrators, and Domain Controllers groups have the required privileges, but attacks such as Zerologon showed how easy it is to acquire these privileges inside a domain [16]. This is very powerful because the attacker can user tools such as "secretsdump.py" to to the DCSync attack remotely using just the Domain Admin's hash. This leads to obtaining the krbtgt hash, which then leads to golden ticket or skeleton key post-exploitation attacks which we covered. Blue teams should look inside the Event logs for Event ID's: 4662 – An operation was performed on an object, 5136 – A directory service object was modified, 4670 – Permissions on an object were changed.

5. Conclusion

We covered only some of the possible attacks that can be accomplished inside Windows Active Directory and many more are exploited by attackers and used by malware. Active Directory security is a delicate balancing act. In the last couple of years, as the cloud was getting more and more used, a lot of large corporations also moved their infrastructures to the cloud, in form of Azure Active directory which itself provides a bigger attack surface. Users are crucial when talking about AD because users find ways to work around security measures they find too inconvenient: Require them to create complex passwords that must be changed every thirty days, and you'll soon find a lot of sticky notes on their desks, which undermines your goal of protecting their accounts from unauthorized access. Securing your AD environment should be a top priority. Not only should you worry about attackers stealing your sensitive data but also rogue employees, thing that we see more and more often in the IT industry. Even one successful attack on your Active Directory can cause long-lasting damage to your organization, or even put it out of business altogether.

6. References

[1]. Article url: https://www.beyondtrust.com/resources/glossary/active-directory-security
Article title: What is Active Directory Security? (Security Definitions/Glossary)

[2]. Article url: https://www.windows-active-directory.com/active-directory-for-beginners.html
Article title : What is Active Directory? | What is Active Directory? Microsoft Active Directory Fundamentals with Video Tutorials

[3]. Article url: https://adsecurity.org/?p=1684

Article title: The Most Common Active Directory Security Issues and What You Can Do to Fix Them

Author: Sean Metcalf

[4]. Article url: https://docs.microsoft.com/en-us/windows-server/networking/dns/dns-top
Article title: Domain Name Systems (DNS)

[5]. Article url: https://www.crowe.com/cybersecurity-watch/netbios-llmnr-giving-away-credentials
Article title: NetBIOS and LLMNR: The Gifts That Keep on Giving (Away Credentials)

[6]. Article url: https://www.varonis.com/blog/the-difference-between-active-directory-and-ldap
Article title: The Difference Between Active Directory and LDAP

[7]. Article url: https://nordvpn.com/blog/what-is-smb/
Article title: What is SMB and how does it work?

[8]. Article url: https://www.extrahop.com/resources/protocols/msrpc/
Article title: MSRPC Protocol: Definition & How It Works

[9]. Article url: https://docs.microsoft.com/en-us/windows/win32/winrm/portal
Article title: Windows Remote Management

[10]. Article url: https://web.mit.edu/kerberos/
Article title: Kerberos: The Network Authentication Protocol

[11]. Article url:
https://www.varonis.com/blog/kerberos-authentication-explained/
Article title: Kerberos Authentication Explained

[12]. Article url:
https://stealthbits.com/blog/cracking-active-directory-passwords-with-as-rep-roasting/
Article title: -

[13]. Article url:
https://attack.stealthbits.com/how-golden-ticket-attack-works
Article title:

[14]. Article url:
https://www.qomplx.com/qomplx-knowledge-kerberoasting-attacks-explained/
Article title: -

[15]. Article url:
https://book.hacktricks.xyz/windows/active-directory-methodology/skeleton-key
Article title: -

[16]. Article url:
https://book.hacktricks.xyz/windows/active-directory-methodology/dcsync
Article title: -