Kerberos in Focus: Strategies for Robust Cyber Risk Mitigation

A KAMIND IT Educational Whitepaper

OCTOBER 1, 2024

KAMIND IT
www.kamind.com

Table of Contents

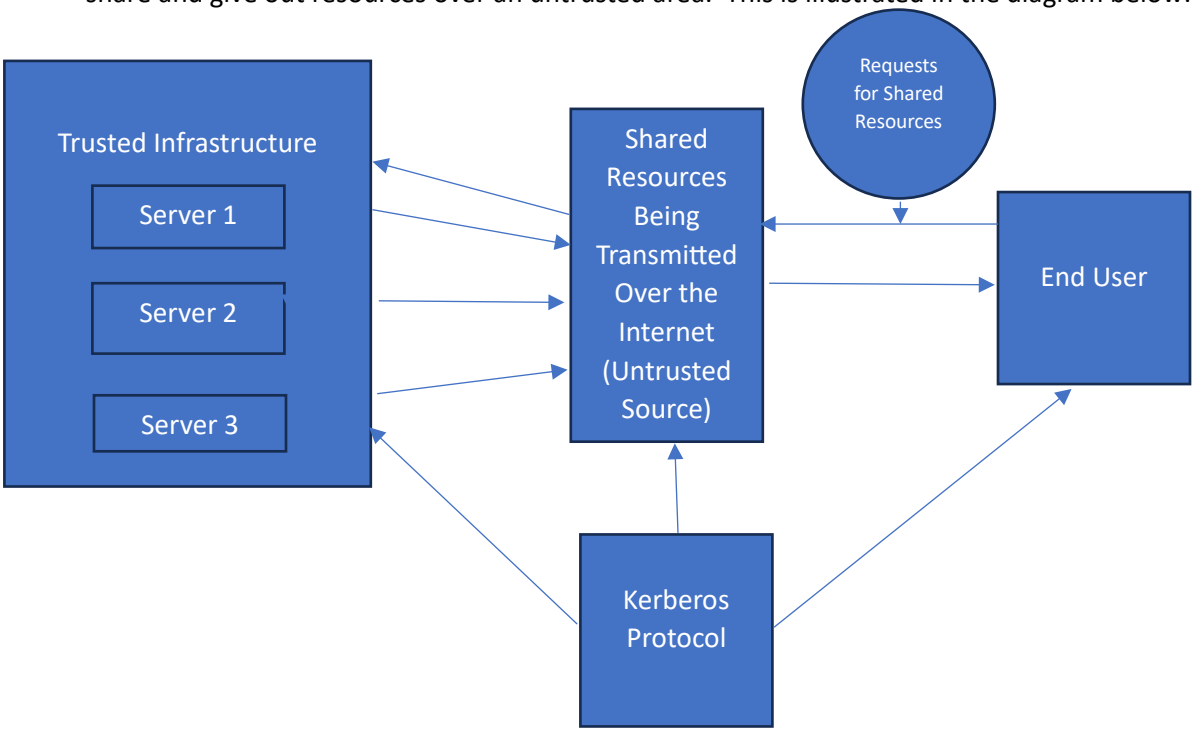
Kerberos in Focus: Strategies for Robust Cyber Risk Management

In today's cybersecurity landscape, confirming the identity of people with whom you are communicating with and those that are trying to access shared resources is a prime security concern today. Of course, there are many ways that this can be done, such as implementing the Zero Trust Framework, the concepts of Identity and Access Management and Privileged Access Management (known as “IAM” and “PAM”, respectively), and perhaps even using the Biometric Modalities such as those of Fingerprint Recognition and Iris Recognition.

But there is another tool out there as well, and in fact, it has been in use for quite some time. It is known as “Kerberos”, and this is the focal point of this whitepaper.

What Is Kerberos?

In non-technical terms, Kerberos is actually a network protocol that is used to serve as the bridge, or even the gateway between the end users and the Internet. It allows for two more trusted servers to share and give out resources over an untrusted area. This is illustrated in the diagram below:



This graphic demonstrates how Kerberos is used, at its most basic level. First, the end user requests access to shared resources across multiple servers. He or she is obviously trusted, because they have

been authenticated either by using Two Factor Authentication (2FA), or Multifactor Authentication (MFA). The request is being sent over an untrusted source, which is the Internet. Once these specific requests have reached the Trusted Infrastructure, the Servers then determine which of the Shared Resources need to be transmitted back to the end user. But again, they will be transmitted back over an Untrusted Source, the Internet.

This midway point travel (back and forth), is the most vulnerable, as any Cyberattacker can easily intercept these Shared Resources and exploit them for malicious purposes. Therefore, the goal of the Kerberos Protocol is to provide a safe means of travel across this Untrusted Source.

There is nothing really new about the Kerberos Protocol, it was first developed by researchers at the Massachusetts Institute of Technology. It is deployed across all of the major Operating System platforms, which include all of the versions of Windows being used today, the Apple MacOS, Free BSD, and all of the flavors of Linux. It can also be used on mobile devices as well, most notably those of the iOS and Android. In the world of Microsoft Azure, it is an integral part of the Azure Active Directory, recently renamed Entra (also commonly referred to as the "AAD").

As mentioned previously, the above illustration is only a very simple example of how the Kerberos Protocol actually works. But today, given the interconnectivity of devices, both physical and virtual, the Kerberos Protocol can actually become much more complex. In these scenarios, there are more entities that are involved in the transfer of requests and access to the Shared Resources. These are as follows:

1) The KDC:

   This is an acronym that stands for the "Key Distribution Center". It acts as a third-party authentication service for the Kerberos Protocol. This is typically used when a business has a large amount of network traffic, along an equal or greater amount of requests. There is also a subcomponent to this, which is known as the "Kerberos Database". This is a centralized repository that contains all of the transactions and processes that have occurred with the Kerberos Protocol in a specified time frame.

2) The Tickets:

   Although the Trusted Sources (which are the Servers and the end users) have proven their identity, "tickets" are also used as an extra layer of authentication, for all of the entities that are involved in a Shared Resource transaction. Cryptography is also used here, to make sure that all communications, especially those over the Internet, remain in a scrambled state, so that they become useless if they were to be intercepted.

An example of a ticket is below:

```
PS C:Windowssystem32> klist tickets


Current LogonId is 0:0xe67df
```

```
Cached Tickets: (4)

#0>     Client: Joe @ domain.local

        Server: cifs/fileserver1.domain.local/domain.local @ DOMAIN.LOCAL

        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96

        Ticket Flags 0x60a10000 -> forwardable forwarded renewable
pre_authent name_canonicalize

        Start Time: 7/10/2020 12:33:49 (local)

        End Time:   7/10/2020 22:32:13 (local)

        Renew Time: 7/17/2020 12:32:13 (local)

        Session Key Type: AES-256-CTS-HMAC-SHA1-96

        Cache Flags: 0x40 -> FAST

        Kdc Called: DC1.domain.local
```

(SOURCE: 1).

When these services are used, there are more steps that are involved in the Kerberos Authentication process, and are:

1) The end user places a request for access to the shared resources. At this time also, a ticket is also automatically requested, as a means for further authentication. This sub process is technically known as the "Ticket Granting Service".

2) An Authentication Server then confirms the identity of the user, and examines the level of rights and permissions they have for access. The ticket is also issued in this step. All activities up to this step are fully encrypted.

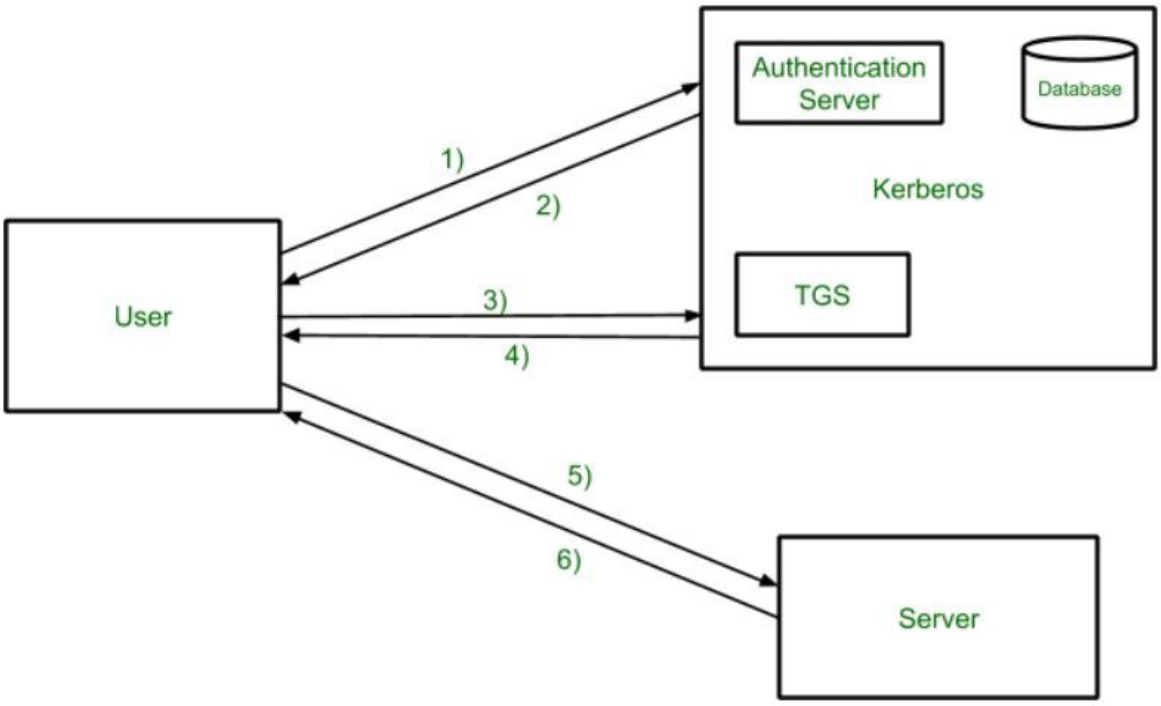
3) A decryption process is now initiated, and the ticket that has been generated in the last step is now transmitted to the to another entity known as the "Ticket Granting Server", also known as the "TGS"

4) The TGS now confirms the validity of the ticket, and fully verifies the request for access to the Shared Resources.

5) The confirmed ticket is then transmitted to the Servers that have the repositories of the Shared Resources that the end user is trying to gain access to.

6) The Servers (as mentioned in the last step) do one more validity check on the ticket, and if everything proves to be legitimate and authentic, access to the Shared Resources is granted, and the end user can ultimately download what they need.

This entire process is illustrated in the diagram below:



(SOURCE: 2).

As it was stated earlier in this whitepaper, both 2FA and MFA are used the most for a Kerberos environment. But unfortunately, one of the credentials that is used to authenticate the end user is the traditional password. Because of the many flaws of it that can be exploited, there are certain rules that have to be followed to mitigate this risk:

1) The passwords of the end user cannot be sent over the network.

2) The passwords cannot be stored in any configuration of the Kerberos environment.

3) The passwords can never be sent as Plaintext. At all times, they must be encrypted.

4) The password can only be used once in a Kerberos transaction. For any subsequent ones, a new password will be created.

Other kinds of vulnerabilities and weaknesses of the Kerberos Protocol will be examined later in this whitepaper.

How To Deploy Kerberos Into The Microsoft Azure Active Directory

The last section reviewed into detail as to how the Kerberos works in both a simple and a much more complex environment. In this one, we now explore how to set up the Kerberos infrastructure into Microsoft Azure, and integrate it with the Active Directory.

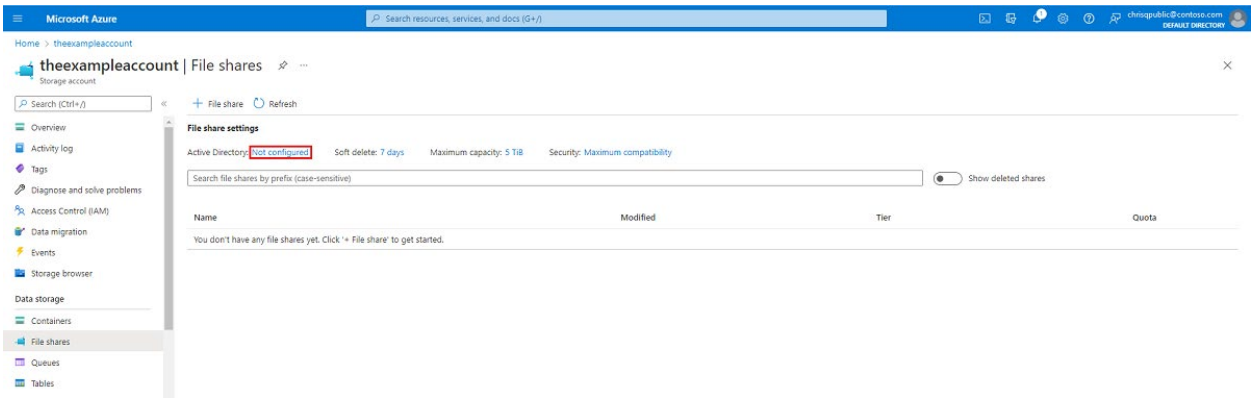
Deploying The Kerberos

To set up the process, follow these steps:

1) Log into your Azure Portal (portal.azure.com).

2) Hover over to “Data Storage”, and select “File Shares”.

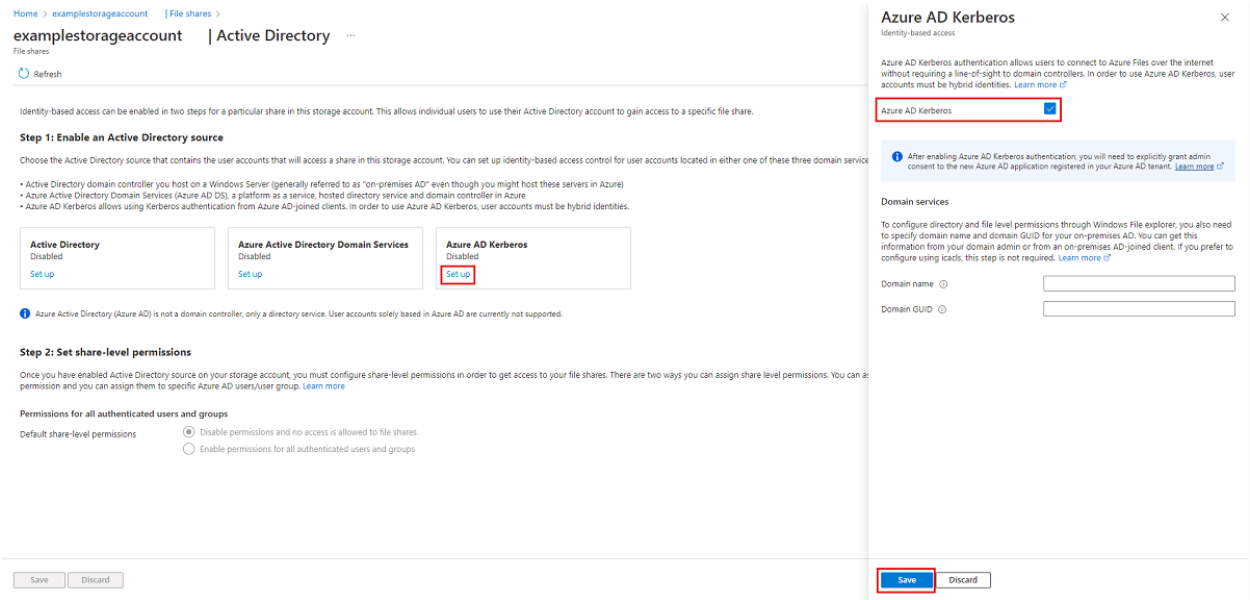
3) Next to the Active Directory module, select the appropriate Configuration Status. This is illustrated below:



(SOURCE: 3).

4) Under the Active Directory, select "Set Up".

5) Select the "Azure Active Directory Kerberos" checkbox. This is illustrated below:



(SOURCE: 3).

6) Click on "Select".

NOTE: You can also deploy Kerberos through:

- The Azure Power Shell: https://learn.microsoft.com/en-us/azure/storage/files/storage-files-identity-auth-hybrid-identities-enable?tabs=azure-powershell

- The Azure Command Line Interface (CLI): https://learn.microsoft.com/en-us/azure/storage/files/storage-files-identity-auth-hybrid-identities-enable?tabs=azure-cli

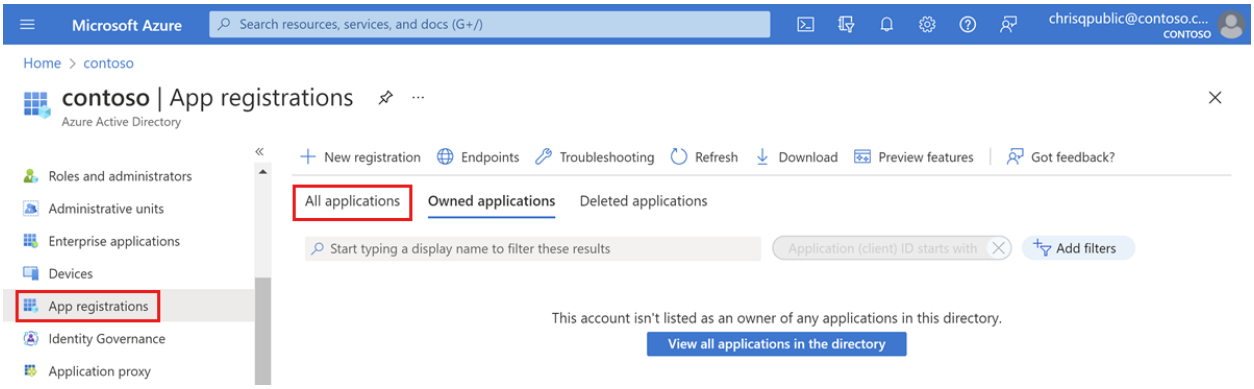
Assigning Administrative Privileges In Kerberos

Now that you have Kerberos firmly established into your Azure Active Directory, the next major step to be completed is to establish the Administrative Privileges that go with it. To do this, follow these steps:

1) Log into the Azure Active Directory (portal.azure.com).

2) Go to the "Apps Registrations" on the left pane.

3) Select upon "All Applications". This is illustrated in the diagram below:



4) Click on the application with the file name matching this syntax:

[Storage Account] <your-storage-account-name>.file.core.windows.net

5) Select the "API Permissions" on the left-hand side of the screen.

6) Select the "Grant Admin Consent" for the name of the sub directory to give the rights, privileges, and permission for these three specific APIS:

   - Openid
   - Profile
   - User.Read

7) Click on "Yes" to confirm all in order Administrative Privileges to be established.

NOTE: If you would like to also configure and implement Administrative Privileges for specific User Groups and their corresponding Profiles in the Azure Active Directory, click on the link below:

https://learn.microsoft.com/en-us/azure/storage/files/storage-files-identity-ad-ds-assign-permissions?tabs=azure-portal

Establishing The Kerberos Infrastructure For The Virtual Desktops/Client Devices

The final step here now is to set up the Kerberos with those devices that will be requesting access to the Shared Resources from the Virtual Machines (VMs) in Microsoft Azure. To do this, follow these steps:

1) Configure the Microsoft Intune Policy CSP to this setting:

   "Kerberos/CloudKerberosTicketRetrievalEnabled, set to 1"

2) Configure the Group Policy to "Enabled":

   "Administrative Templates\System\Kerberos\Allow retrieving the Azure AD Kerberos Ticket Granting Ticket during logon"

3) Set up the appropriate Registry Values for the client devices:

   "reg add HKLM\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Parameters /v CloudKerberosTicketRetrievalEnabled /t REG_DWORD /d 1"

NOTE: The above configurations do not happen instantaneously, and you may have to reboot certain parts of your Azure Cloud deployment in order for them to take effect.

Security Threats To The Azure Active Directory Kerberos

Apart from the vulnerabilities that are posed by passwords, there are a lot of other security threats that can breach an entire Kerberos Infrastructure. For purposes of this whitepaper, we focus on the more prevalent and damaging ones, which are as follows:

- The Lateral Attacks
- The Golden Ticket
- Kerberoasting
- Other threats: Bounce The Ticket and Silver Iodide

The Lateral Attacks

The technical definition for Lateral Attack is:

"Lateral movement refers to the techniques that a cyberattacker uses, after gaining initial access, to move deeper into a network in search of sensitive data and other high-value assets. After entering the network, the attacker maintains ongoing access by moving through the compromised environment and obtaining increased privileges using various tools."

(SOURCE: 4).

In simpler terms, once a Cyberattacker has gained a foothold into the IT/Network Infrastructure, he or she assumes the identity of a legitimate end user. Because of that, they can move very discreetly without raising any suspicion. Typically, they move from one device to another, in a "sideways fashion" in order to achieve their objective.

This is why this kind of breach is known formally as a "Lateral Attack", because the Cyberattacker is not moving towards a straight, linear path towards their intended target. Although the exact mechanics of a of this will vary, the following is a methodology of how it is generally carried out:

1) Reconnaissance:

   In this first phase, the Cyberattacker takes note of and maps the entire IT/Network Infrastructure, end users, and all of the devices that are being fully activated and being used. This allows for the Cyberattacker to discover any hidden backdoors so that they can not only gain entrance, but also deploy any malicious payloads. A popular tool used for exploiting purposes is known as "Netstat", which shoes the current network connections on a real time basis. This can also be used for identifying mission critical assets or for gaining further knowledge about the Network Infrastructure.

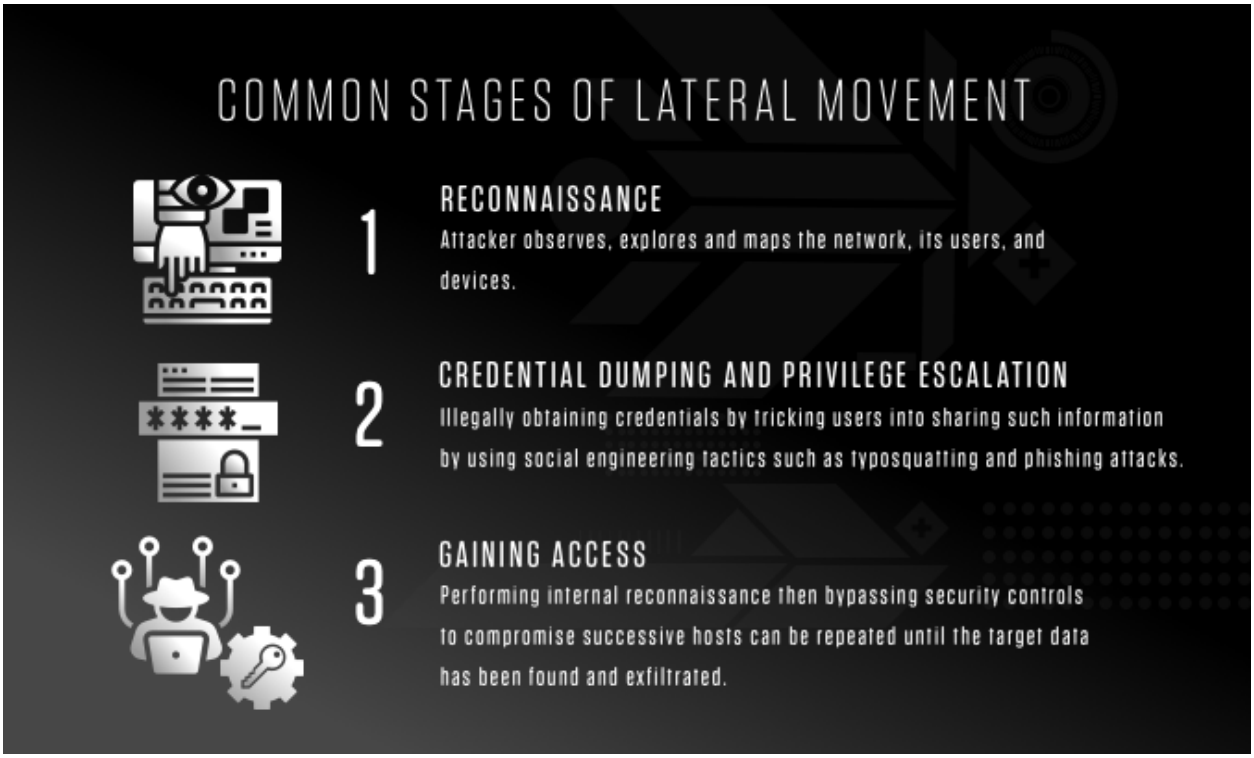
2) Credential Dumping/Privilege Escalation:

   At some point in their journey, the Cyberattacker will need to gain the login credentials of an unsuspecting victim. In this situation, Social Engineering is used quite often, in tricking others to give out their usernames and passwords. This is technically referred to as "Credential Dumping", and can also be achieved through Phishing based emails. A tool here that is also used is known as "Mimikatz". It can be used to hijack passwords that are stored in Cache, or even the authentication certificates from the memory of a compromised device. Keyloggers can also be deployed, which can secretly record the keystrokes of the victim.

3) Gaining Access:

   Once the last two steps have been completed, the Cyberattacker can now gain access to whatever will offer the most rewards. Typically, these are the Privileged Access Accounts, because with these kinds of credentials, the Cyberattacker can penetrate much deeper into the IT/Network Infrastructure. In order to stay discreet, they will still continue to employ the tactics of Social Engineering. At this stage, a common security breach is that of Data Exfiltration. This is where the Personal Identifiable Information (PII) datasets are heisted, a little bit at a time, in order to avoid any sort of detection.

These steps are illustrated below:



(SOURCE: 4).

One of the keyways to mitigate the risk of a Lateral Attack from happening is to drastically cut down on a metric known as the "Breakout Time". This simply reflects the time it takes for the IT Security team to detect that a Cyberattacker is present, and moving laterally. At the present time, it takes almost two hours for this kind of detection to be noticed. Thus, vigilance and human intervention are also needed in this regard. This metric is illustrated below:

(SOURCE: 4).

The Golden Ticket

As was explored previously, the Kerberos Protocol makes use of various tickets in order fully authenticate the identity of the end user, who is trying to gain access to the Shared Resources. But this kind of system does have its vulnerabilities as well, and it is quite possible for a Cyberattacker to actually hijack and forge one of these tickets. By doing this, they will be able to gain access to the Privileged Access Accounts. Because of this, the emulated ticket becomes a "Golden Ticket", given the amount of infiltration that a Cyberattacker can do.

The technical definition for it is as follows:

"A Golden Ticket Attack exploits Kerberos, the default authentication for Active Directory, by extracting a user's Ticket Granting Ticket (TGT) within the domain . . . the ultimate goal is to grant the attacker unrestricted access to the network that can last up to 10 years."

(SOURCE: 5).

The general methodology that is followed for launching a Golden Ticket Attack is:

1) Investigation:

   The Cyberattacker examines closely the weaknesses in the Kerberos Protocol that they are trying to break through. Once this has been done, then a Golden Ticket is created.

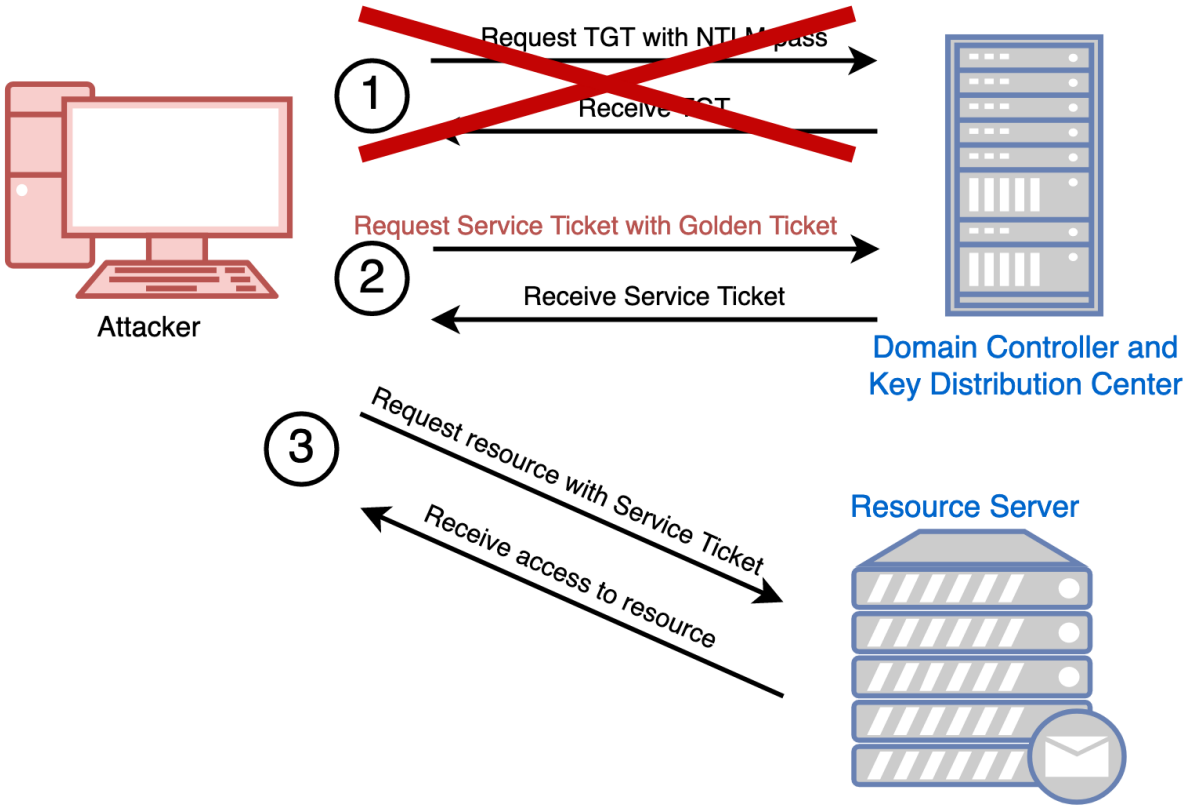
2) Gaining Access:

   In order to start gaining access to the IT/Network Infrastructure, a corresponding "Service Ticket" is allocated, pending successful receipt of the Golden Ticket by the Kerberos Protocol.

3) Infiltration:

   Once the Cyberattacker has actual access to the "Service Ticket", he or she can then start to gain access and penetrate further to where they want to, very often going unnoticed.

These steps are illustrated in the diagram below:



(SOURCE: 6).

The Kerberoasting

This kind of attack is a more sophisticated version of the Golden Ticket. In this situation, the Cyberattacker is trying to break the encryption that has been provided into the generated tickets, and from there, trying to gain access to key digital assets.

A technical definition of it is as follows:

"Kerberoasting is a post-exploitation attack (an attack that's carried out on a system that's already been compromised) hackers use for persistence, privilege escalation, and lateral movement in a compromised system. It targets the Kerberos protocol to get password hashes for Active Directory service accounts with Service Principal Names (SPNs)."

(SOURCE: 7).

So as you can see from this, the specific encryption that is attempted to be broken into are the "Password Hashes". This is where the password is scrambled into a garbled state. But by breaking this,

the Cyberattacker can then gain full access to it. The “Service Principal Name” are the unique identifiers that are associated with all of the Shared Resources.

In order to launch this type of exploitation, the Cyberattacker usually gains access to the IT/Network Infrastructure following the same steps as detailed for the Golden Ticket. Subsequently, the process then becomes:

Steps 1-3: See last subsection.

4) The Cyberattacker attempts to crack the password attack, using the various tools at their disposal.

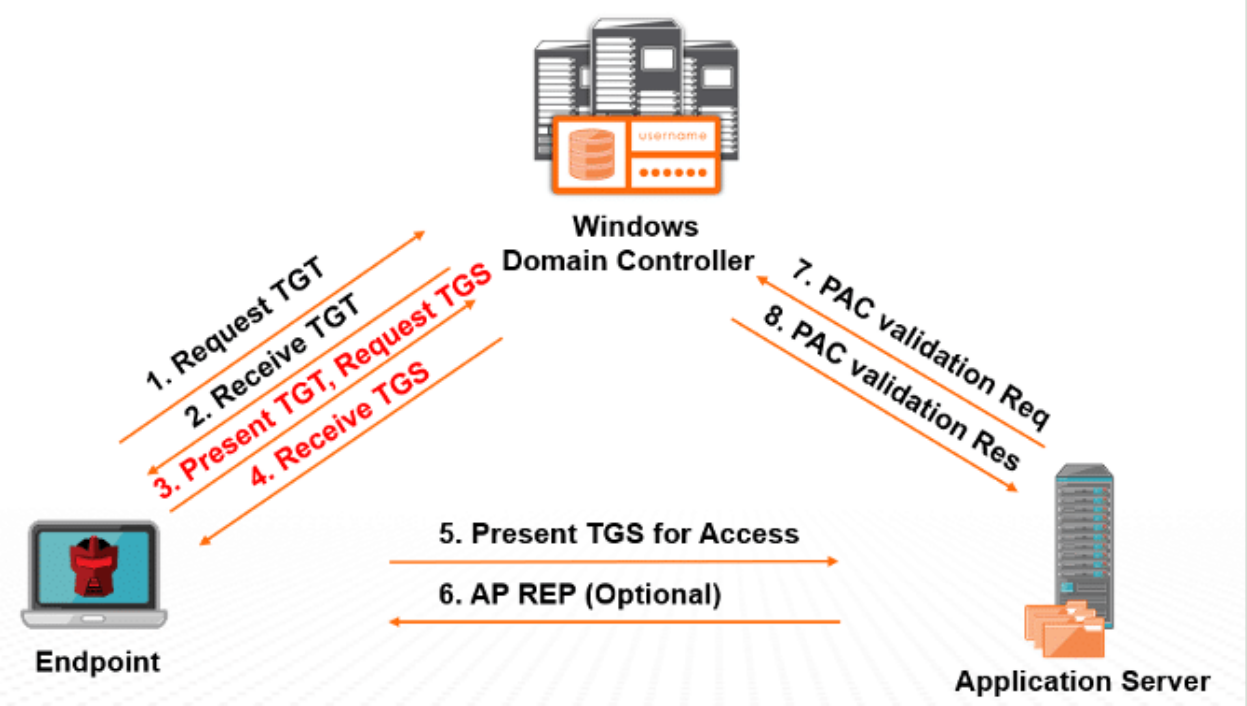
5) Once the last step has occurred, the Cyberattacker can then gain access to more fortified points, especially if the Zero Trust Framework has been implemented.

6) Once successful, the Cyberattacker can then heist the most vulnerable of the digital assets, or Shared Resources.

7) The above process can be repeated for multiple tickets, which represent the different end users and their corresponding access requests.

8) At this point, once the Privileged Access accounts have been reached, more destructive attacks can happen, even including the total breakdown of the entire IT/Network Infrastructure.

This kind of threat vector is illustrated in the diagram below:



(SOURCE: 8).

Other Threats To The Azure Active Directory Kerberos: Bounce The Ticket And Silver Iodide

It is expected that attacks to the Kerberos Protocol will continue, and are even expected to get worse. Two of the more modern threat vectors in this regard are as follows:

- Bounce The Ticket
- Silver Iodide

The Bounce The Ticket

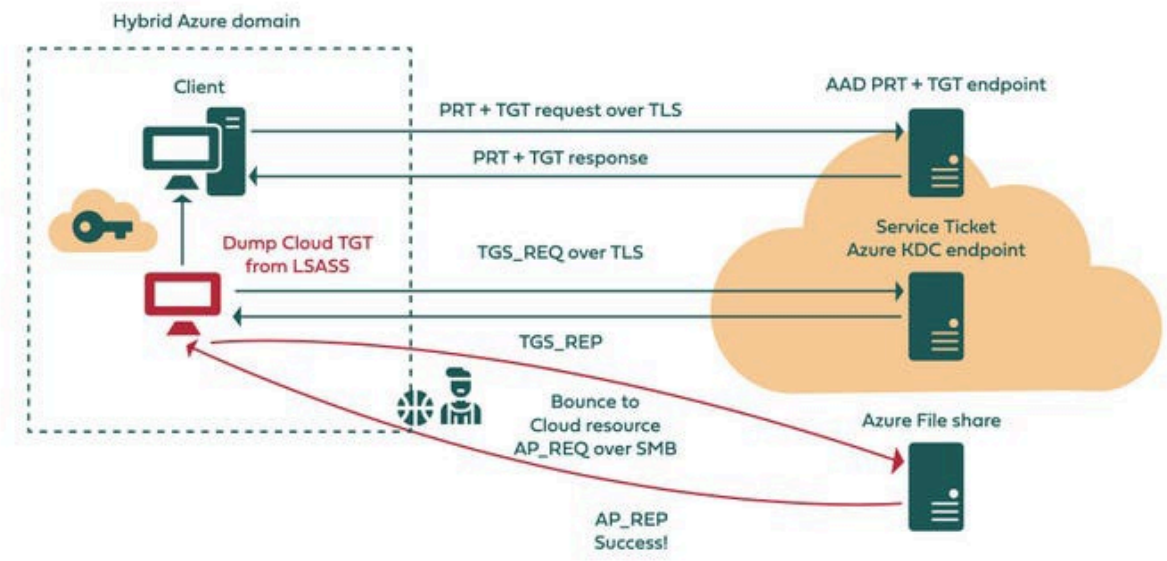
This kind of attack is similar to the Kerberoasting, but instead of going after the "Password Hashes" and the "Service Principal Names", the main target of the Cyberattacker is to now gain access to specific Azure Cloud Workloads and compromising them, by breaking the encryption that is associated with them. A Cloud Workload is technically defined as:

"A logical bundle of software and data that is present in, and processed by, a cloud computing technology."

(SOURCE: 9).

In these specific instances, the Cyberattacker attempts to hijack a ticket from the memory of the compromised device, and then break the corresponding hashes in an attempt to gain access to the Cloud Workloads. A sophisticated version of this kind of attack is illustrated below:

((



(SOURCE: 10).

The Silver Iodide

In this kind of threat vector, the Cyberattacker is specially targeting the Azure Active Directory, which is connected to a specific service, which is associated with a specific Shared Resource. From here, Access Keys can then be created to compromise and exfiltrate all of the PII datasets which are contained in the designated Azure Storage Accounts. It should be noted that this kind of breach is targeted towards Microsoft Azure, and not the AWS or the Google Cloud Platform (GCP).

How To Mitigate The Risks

The bottom line is that no individual or business is 100% immune to a Cyberattack. The only thing that can be done is to mitigate, or minimize, that risk from actually happening. Thus, the preventative actions that one can take to protect their Kerberos Infrastructure from the threat vectors outlined in this whitepaper are virtually the same as you would take to practice good levels of Cyber Hygiene.

But when it comes to protecting it from attacks when it is integrated with the Azure Active Directory takes some more specialized steps. The following is a plan that you should implement, according to Microsoft.

You should implement a three-tier model in order to protect the credentials of all of your end users, especially the Privileged Access Accounts. This methodology is as follows:

1) Tier 0:

   All of the user profiles, accounts and servers in this tier have a direct path to the domain administrator and other super user privileges. To be assigned to this level, having any sort of Privileged Access will be classified as an account as Tier 0.

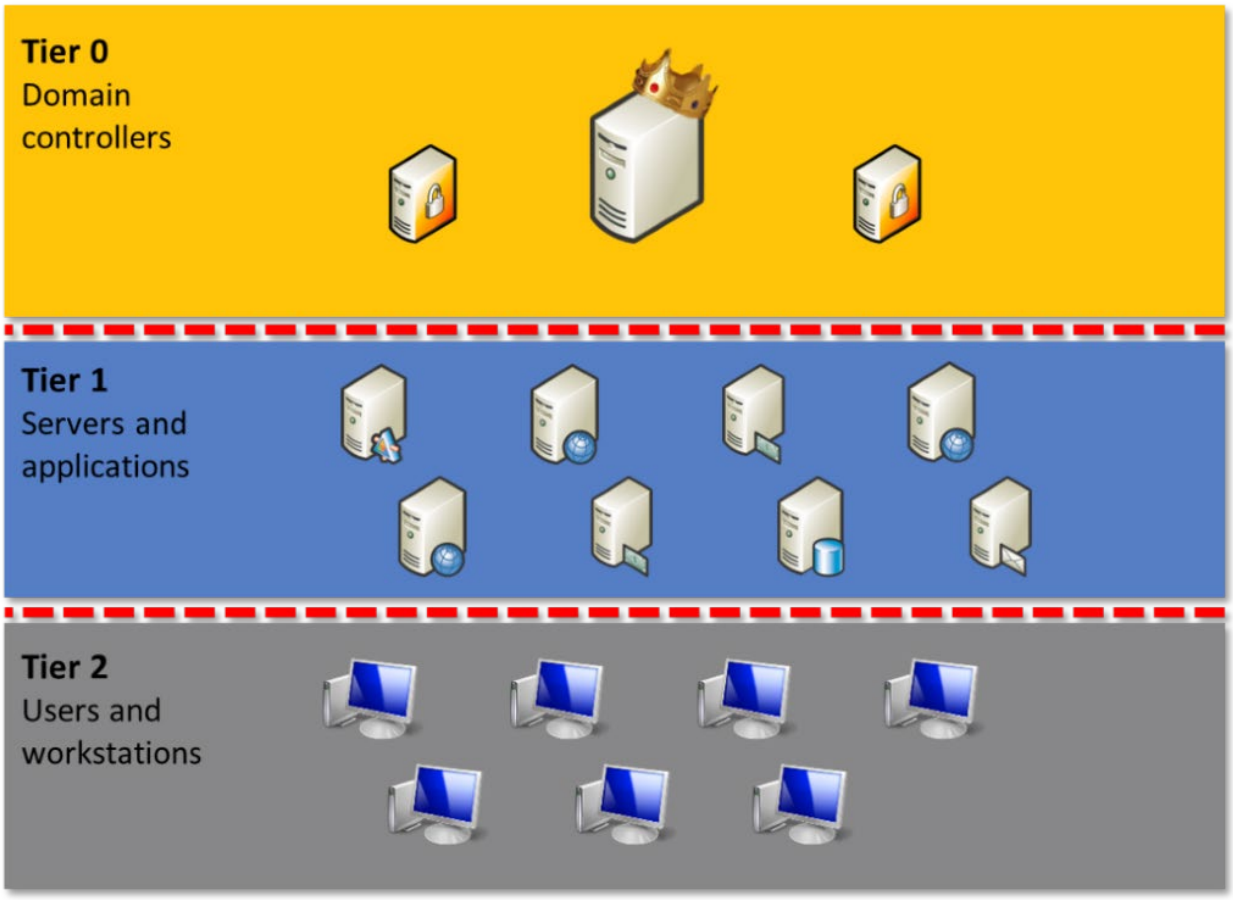
2) Tier 1:

   This particular level will contain the rights, privileges, and permissions to access mission critical digital assets. All accounts and even applications include file shares, application servers, and database servers.

3) Tier 2:

   This category refers to any account or application that does not belong to the other two tiers, and will be the place where standard user accounts will reside.

This model is illustrated in the diagram below:



(SOURCE: 11).

It is important to note at this point that all of the tiers just described have to be totally segregated from one another. In other words, an account cannot be shared across the tiers. This can be accomplished by setting up Group Policy Objects (GPOs) or configuring and deploying Privileged Access Workstations. More information can be found at this link:

https://learn.microsoft.com/en-us/security/privileged-access-workstations/privileged-access-devices

Other considerations you should consider for fortifying your Azure Active Directory/Kerberos Infrastructure:

1) Deploy AES Kerberos Encryption:

   More information about this can be seen at this link:

   https://learn.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/network-security-configure-encryption-types-allowed-for-kerberos

2) Implement Group Managed Service Accounts:

More information about this can be seen at this link:

https://learn.microsoft.com/en-us/windows-server/security/group-managed-service-accounts/group-managed-service-accounts-overview

3) Upgrade your Azure Active Directory to a Forest Level:

More information about this can be seen at this link:

https://adsecurity.org/?p=3377

Other recommendations include the following:

- Make sure that you deploy all needed software patches and updates on a regular basis.
- Keep a constant eye for any abnormal or suspicious activity in your network traffic. A tool like a Next Gen Firewall would be an excellent choice.
- Once again, create very strong and harder to break passwords. For this very reason, you should require the use of a Password Manager, and deploy it across all areas of your business.
- Make sure that all of your Network Security Devices (especially your Firewalls, Routers, Network Intrusion Devices, etc.) are all optimized.
- Enable Multifactor Authentication (MFA) across all levels. Make sure to use different authenticating mechanisms.
- Deploy the concepts of Privileged Access Management (PAM).
- Create an Incident Response (IR) plan if you don't have one in hand, and practice it on a regular basis (at minimum, once a quarter).
- Deploy Endpoint Response (EDR) and if possible, even Extended Detection Response (XDR).
- Conduct Penetration Testing and Vulnerability Scanning exercises on a regular basis (at minimum, once a quarter).
- Enforce the concept of Least Privilege. This is where you assign the rights, permissions, and privileges at the bare minimum possible.

- Consider seriously the deployment of the Zero Trust Framework. This is where you segment out your entire IT/Network Infrastructure into different "Zones". Each one of them has their own lines of defense, supported by Multifactor Authentication (MFA).

- Give serious consideration to using Artificial Intelligence (AI) and Machine Learning (ML). These technologies can be used to automate the routine and mundane tasks that your IT Security team does. They can also be used to filter out false positives, so that only the authentic alerts and warnings are presented.

Conclusions

Overall, this whitepaper has examined the following:

- What Kerberos is.

- How to deploy Kerberos in an Azure Active Directory environment.

- The major threat vectors posed to a Kerberos Infrastructure.

- How to mitigate the risks of your Kerberos Infrastructure from being targeted by a Cyberattacker.

If you have any questions, or need help in deploying a Kerberos Infrastructure with Azure Active Directory, please contact us today.

Sources

1) https://blog.netwrix.com/what-is-kerberos/
2) https://www.geeksforgeeks.org/kerberos/
3) https://learn.microsoft.com/en-us/azure/storage/files/storage-files-identity-auth-hybrid-identities-enable?tabs=azure-portal
4) https://www.crowdstrike.com/cybersecurity-101/lateral-movement/
5) https://www.hornetsecurity.com/us/security-information-us/golden-ticket-attack/
6) https://www.optiv.com/insights/source-zero/blog/kerberos-domains-achilles-heel
7) https://nordvpn.com/blog/kerberoasting-attack/
8) https://www.sentinelone.com/cybersecurity-101/what-is-kerberoasting-attack/#:~:text=In%20a%20Kerberoasting%20attack%2C%20an,or%20network%20resources%20if%20successful.
9) https://csrc.nist.gov/glossary/term/cloud_workload#:~:text=Definitions%3A,by%2C%20a%20cloud%20computing%20technology.
10) https://www.darkreading.com/cloud-security/microsoft-azure-kerberos-attacks-open-cloud-accounts

11) https://www.microsoft.com/en-us/security/blog/2022/10/26/how-to-prevent-lateral-movement-attacks-using-microsoft-365-defender/