Golden Diamond and Sapphire Attacks

Kerberoasting Basics

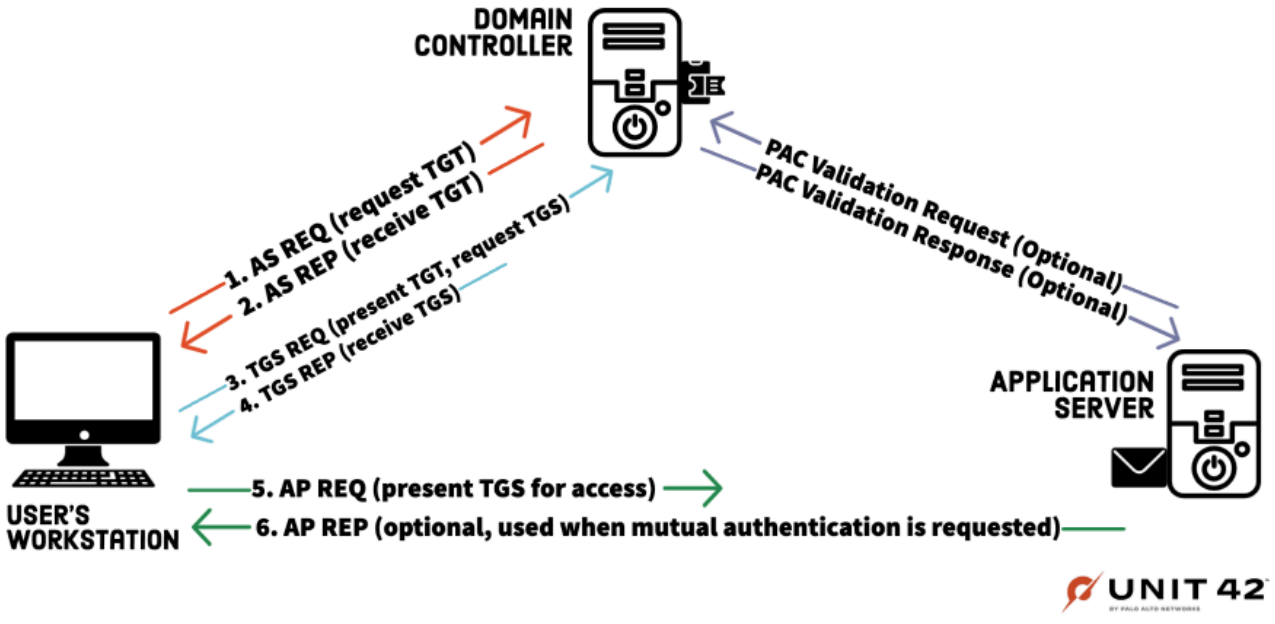
Kerberos is a network authentication protocol that is primarily used in Active Directory (AD) environments. Thousands of companies across different industries use Active Directory technology for managing user accounts and other resources within an organization. Active Directory's first version was released in Windows Server 2000 and since then, it has become particularly common in businesses and other large organizations that have a significant number of users and resources to manage.

Kerberos provides strong authentication by issuing tickets to authenticate users and allow access to services. The tickets are distributed by the key distribution center (KDC). In most environments, the KDC is installed on the domain controller (DC).

During the initial authentication, a Ticket Granting Ticket (TGT) is a ticket assigned to a user. The TGT is later used to authenticate the user to the KDC and request a service ticket from the Ticket Granting Service (TGS). Service tickets are granted for authentication against services.

A Kerberos authentication would consist of the following steps:

1. The user requests (AS-REQ) a TGT from the KDC and the KDC verifies and validates the credentials and user information.
2. After authenticating the user, the KDC sends an encrypted TGT back to the requester (AS-REP).
3. The user presents the TGT to the DC and requests a TGS (TGS-REQ).
4. The TGS is encrypted and sent back to the requesting user (TGS-REP).
5. The user connects to the server hosting the service requested and presents the TGS (AP-REQ) in order to access the service.
6. The application server sends an (AP-REP) to the client. 2



Kerberos uses authentication tokens, or tickets, to verify identities of Active Directory entities. This includes users, service accounts, domain admins, and computers. All of those entities have a password in Active Directory (AD), even though you might not have actually created or changed it manually. 4

Golden Ticket Attacks

To execute a golden ticket attack, an attacker must already have domain administrator privileges to extract the KRBTGT account hash. The attack follows these steps:
1. **Obtain domain information**: The attacker gathers key details, including:
- Fully qualified domain name (FQDN)
- Domain security identifier (SID)
- KRBTGT account hash (critical for ticket forging)
2. **Steal the KRBTGT hash**: Once an attacker gains domain controller (DC) access, they extract the KRBTGT NTLM hash using tools like Mimikatz. This hash allows them to sign their own Kerberos tickets, bypassing authentication mechanisms.
3. **Forge a Kerberos ticket**: With the KRBTGT hash, attackers create a TGT with arbitrary permissions, effectively granting themselves domain admin access.
4. **Persistent access**: The forged TGT can be set to remain valid for years, enabling the attacker to
- Access any system or service within the domain
- Create or modify user accounts
- Evade detection by appearing as a legitimate user

Unlike Pass-the-Hash or Pass-the-Ticket attacks, a golden ticket attack does not require re-authentication, making it significantly harder to detect. 1

Create Golden Kerberos Ticket

Metasploit Framework

> The following workflow requires a Meterpreter session to be followed exactly as written

Meterpreter Steps:

1. `load kiwi`
2. `golden_ticket_create -d LIFELINE.local -k 43460d636f269c709b20049cee36ae7a -s S-1-5-21-1304479805-1949792208-3801273171 -u daethyra -t goldenkrb.ticket`

Find Domain SID

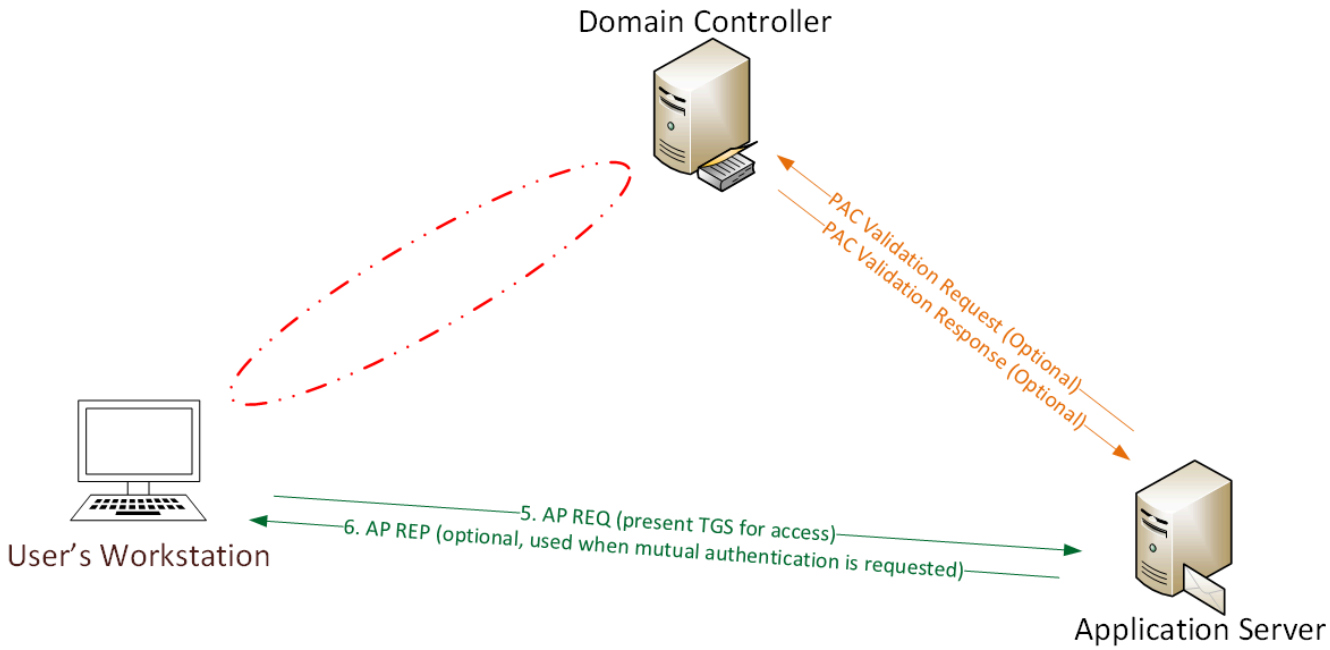
`nxc ldap 10.0.2.7 -u daethyra -p Password1@ --get-sid`

- Running with `-k` flag, as shown here, repeatedly failed to extract the SID

Silver Ticket Attacks

Silver Tickets are forged Kerberos Ticket Granting Service (TGS) tickets, also called service tickets. As shown in the following graphic, there is no AS-REQ / AS-REP (steps 1 & 2) and no TGS-REQ / TGS-REP (steps 3 & 4) communication with the Domain Controller. Since a Silver Ticket is a forged TGS, there is no

communication with a Domain Controller.



- Alluded to at BlackHat during the "Golden Ticket" presentation (Duckwall/Delpy) and discussed partly during Tim Medin's DerbyCon 2014 talk. Skip & Benjamin have provided additional information on Silver Tickets since, but confusion remains.
- The Kerberos Silver Ticket is a valid Ticket Granting Service (TGS) Kerberos ticket since it is encrypted/signed by the service account configured with a Service Principal Name for each server the Kerberos-authenticating service runs on.
- While a Golden ticket is a forged TGT valid for gaining access to any Kerberos service, the silver ticket is a forged TGS. This means the Silver Ticket scope is limited to whatever service is targeted on a specific server.
- While a Golden ticket is encrypted/signed with the domain Kerberos service account (KRBTGT), a Silver Ticket is encrypted/signed by the service account (computer account credential extracted from the computer's local SAM or service account credential).
- Most services don't validate the PAC (by sending the PAC checksum to the Domain Controller for PAC validation), so a valid TGS generated with the service account password hash can include a PAC that is entirely fictitious – even claiming the user is a Domain Admin without challenge or correction.
- The attacker needs the service account password hash
- TGS is forged, so no associated TGT, meaning the DC is never contacted.
- Any event logs are on the targeted server.

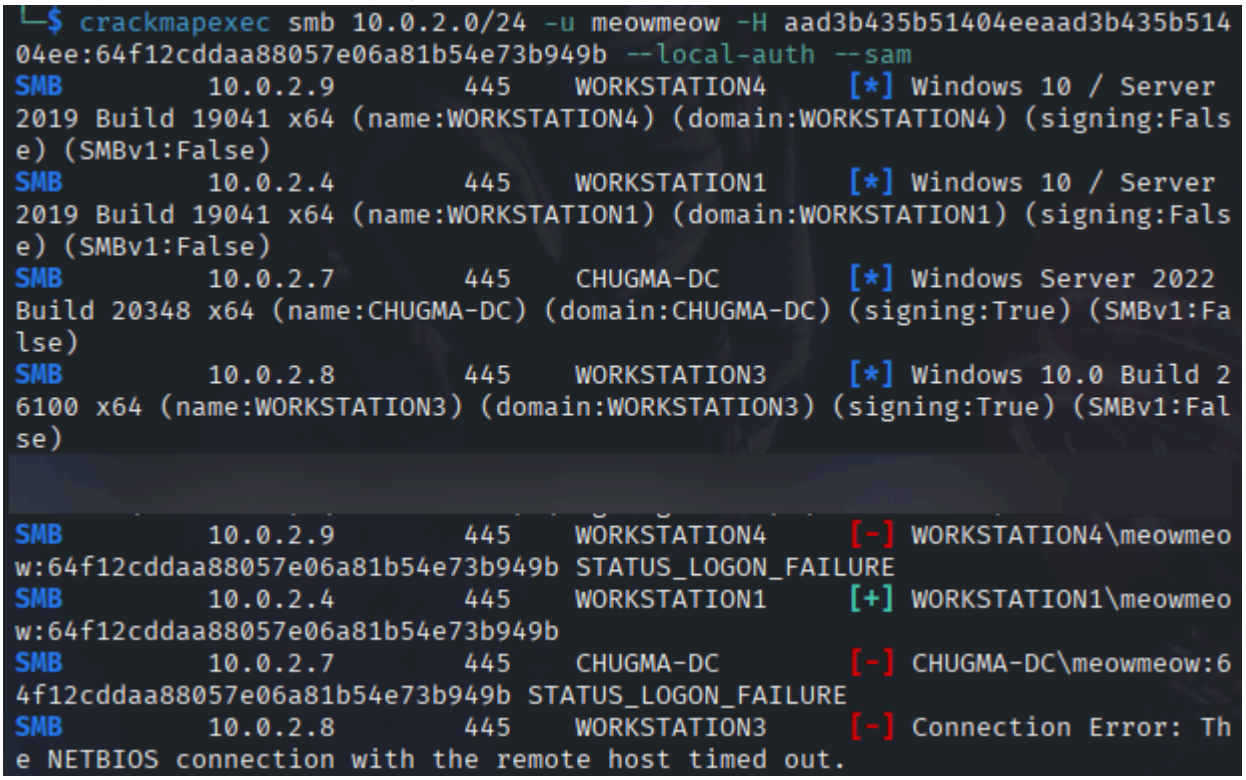
In my opinion, Silver Tickets can be more dangerous than Golden Tickets – while the scope is more limited than Golden Tickets, the required hash is easier to get and there is no communication with a DC when using them, so detection is more difficult than Golden Tickets. [7]

Creating Silver Tickets

Why Silver Tickets?

So when would it make sense to make a silver ticket? Often when dumping a machine's local Security Account Manager (SAM) hashes, it is observed that a local administrator’s logon is denied.

You may wish to attempt spraying a local SAM hash to look for valid access within the network:



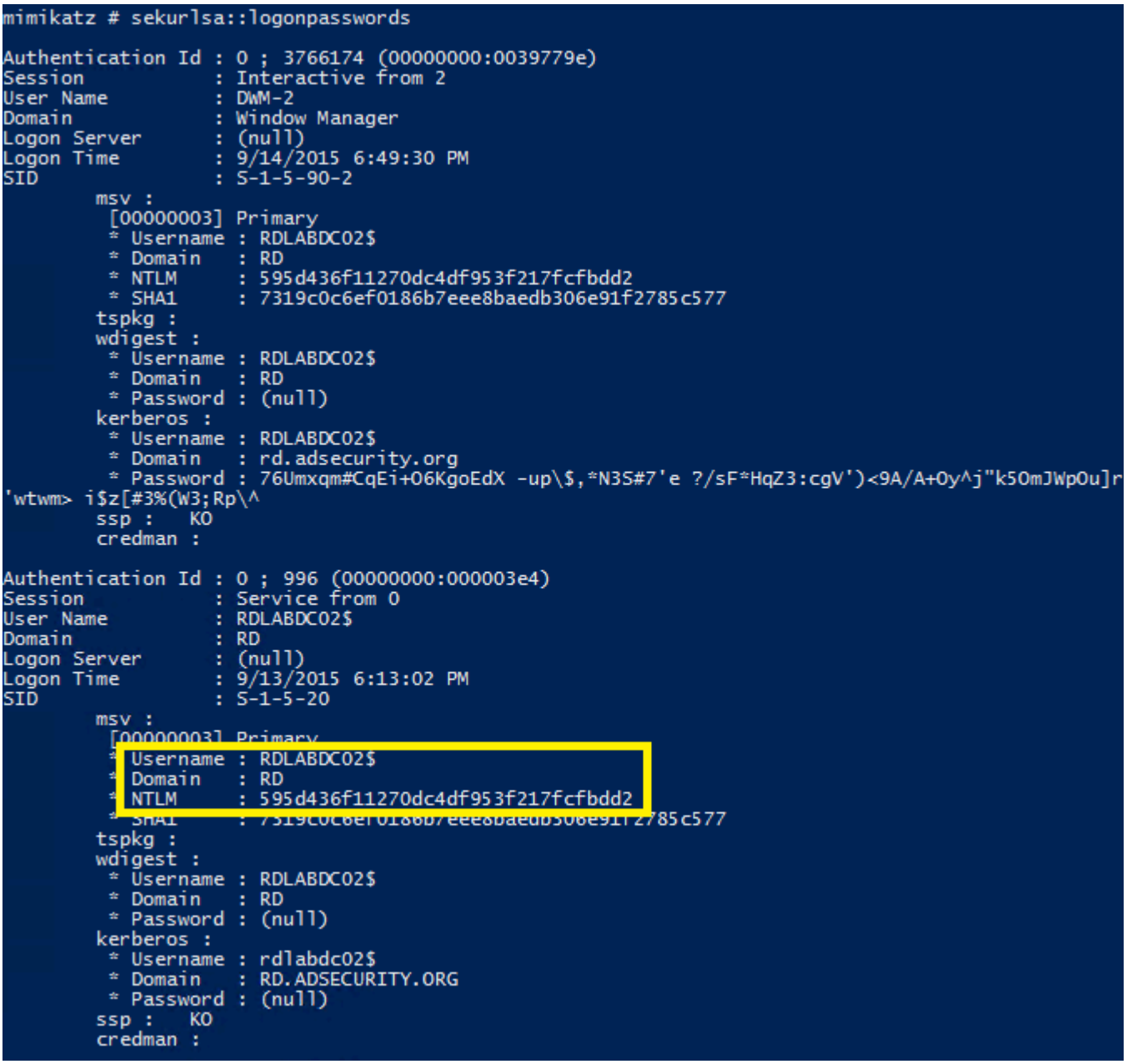
In order to create or forge a Silver Ticket, the attacker has to gain knowledge of the password data (password hash) for the target service. If the target service is running under the context of a user account, like MS SQL, then that Service Account’s password hash is required in order to create a Silver Ticket.

> Cracking Service Account passwords (Kerberoasting) is one potential method for identifying a target service’s associated password data.

How are Silver Tickets created?
Computers host services as well with the most common one being the Windows file share which leverages the “CIFS” service. Since the computer itself hosts this service, the password data required to create a Silver Ticket is the associated computer account’s password hash.
When a computer is joined to Active Directory, a *new* computer account object is created and linked to the computer. The password and associated hash is stored on the computer that owns the account and the NTLM password hash is stored in the Active Directory database on the Domain Controllers for the domain.

If an attacker can gain admin rights to the computer (to gain debug access) or be able to run code as local System, the attacker can dump the AD computer account password hash from the system using Mimikatz (the NTLM password hash is used to encrypt RC4 Kerberos tickets):

```
Mimikatz> “privilege::debug” “sekurlsa::logonpasswords” exit
```



Tools

Mimikatz

Silver Ticket Command Reference:

The Mimikatz command to create a golden or silver ticket is “kerberos::golden”

- /domain – the fully qualified domain name. In this example: “lab.adsecurity.org”.
- /sid – the SID of the domain. In this example: “S-1-5-21-1473643419-774954089-2222329127”.
- /user – username to impersonate
- /groups (optional) – group RIDs the user is a member of (the first is the primary group) default: 513,512,520,518,519 for the well-known Administrator’s groups (listed below).
- /ticket (optional) – provide a path and name for saving the Golden Ticket file to for later use or use /ptt to immediately inject the golden ticket into memory for use.
- /ptt – as an alternate to /ticket – use this to immediately inject the forged ticket into memory for use.

- /id (optional) – user RID. Mimikatz default is 500 (the default Administrator account RID).
- /startoffset (optional) – the start offset when the ticket is available (generally set to –10 or 0 if this option is used). Mimikatz Default value is 0.
- /endin (optional) – ticket lifetime. Mimikatz Default value is 10 years (~5,262,480 minutes). Active Directory default Kerberos policy setting is 10 hours (600 minutes).
- /renewmax (optional) – maximum ticket lifetime with renewal. Mimikatz Default value is 10 years (~5,262,480 minutes). Active Directory default Kerberos policy setting is 7 days (10,080 minutes).

Example Mimikatz Command to Create a Silver Ticket

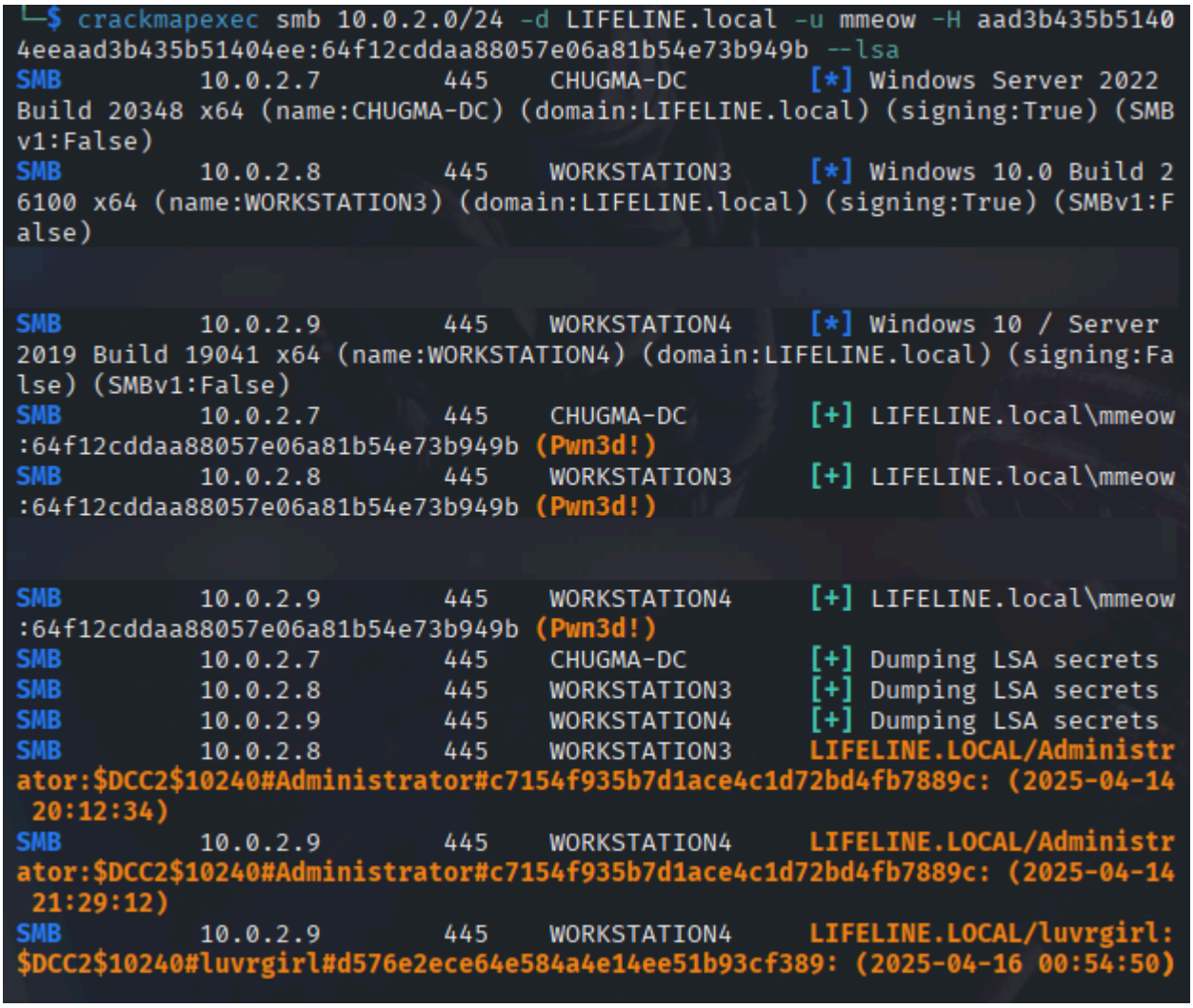
The following Mimikatz command creates a Silver Ticket for the CIFS service on the server `adsmswin2k8r2.lab.adsecurity.org`. In order for this Silver Ticket to be successfully created, the AD computer account password hash for `adsmswin2k8r2.lab.adsecurity.org` needs to be discovered, either from an AD domain dump or by running Mimikatz on the local system as shown above (`Mimikatz> "privilege::debug" "sekurlsa::logonpasswords" exit`). The NTLM password hash is used with the `/rc4` parameter. The service SPN type also needs to be identified in the `/service` parameter. Finally, the target computer's fully-qualified domain name needs to be provided in the `/target` parameter. Don't forget the domain SID in the `/sid` parameter.

```
mimikatz> "kerberos::golden /admin:LukeSkywalker /id:1106 /domain:lab.adsecurity.org
/sid:S-1-5-21-1473643419-774954089-2222329127 /target:adsmswin2k8r2.lab.adsecurity.org
/rc4:d7e2b80507ea074ad59f152a1ba20458 /service:cifs /ptt" exit
```

Crackmapexec & [Impacket's](#) [ticketer.py](#)

Dump LSA Using Domain Account and NTLM Hash:

```
crackmapexec smb 10.0.2.0/24 -d LIFELINE.local -u mmeow -H
aad3b435b51404eeaad3b435b51404ee:64f12cddaa88057e06a81b54e73b949b --lsa
```



> Note: A password may be used in place of the hash.

Create Silver Tickets:

Requirements:

- SPN for the service we want to access
- Domain name
- Computer NTLM hash
- Domain Controller IP address
- Domain SID
- Domain User account to impersonate

Creating our Silver Ticket with ticketer.py:

```
ticketer.py -spn HOST/WORKSTATION4.LIFELINE.local -domain LIFELINE.local -nthash
3c0ca92a8fa19374d19bf887543a5e19 -dc-ip
10.0.2.7 -domain-sid S-1-5-21-3010306422-4235424541-3945354764 <domain user>
```

Our silver ticket file located in Administrator.ccache is now saved on our machine and we can start using it with tools supporting Kerberos authentication, without being prompted for authentication. Using the following

command, we can tell Impacket (among other tools) to use our Silver Ticket for Kerberos authentication.

```
export KRB5CCNAME=Administrator.ccache
```

8

Silver Ticket Required Parameters

- `/target` – the target server's FQDN.
- `/service` – the kerberos service running on the target server. This is the Service Principal Name class (or type) such as cifs, http, mssql.
- `/rc4` – the NTLM hash for the service (computer account or user account)

Silver Ticket Default Groups:

- Domain Users SID: S-1-5-21-513
- Domain Admins SID: S-1-5-21-512
- Schema Admins SID: S-1-5-21-518
- Enterprise Admins SID: S-1-5-21-519
- Group Policy Creator Owners SID: S-1-5-21-520

Persistence With Computer Accounts

Once the attacker has access to the computer account password hash, the account can be used as a "user" account to query Active Directory, but the more interesting use case is to create Silver Tickets to access computer hosted services with admin rights. Since the Domain computer account password change policies are more of a guideline since they aren't forced to change by the Domain Controllers (set to 30 days by default but up to the computer to actually change the password), it's possible that once an attacker gains knowledge of the computer account password, it could be used for a long time. Active Directory does not prevent a computer account from accessing AD resources even if the computer account password hasn't changed in years.

The attacker could also prevent the computer account password from changing:

1. the following registry key:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Netlogon\Parameters\DisablePasswordChange = 1
```

2. There's a client Group Policy setting to prevent the computer from changing the password most often used to support VDI (virtual desktops). Enabling the group policy setting `Domain member: Disable machine account password changes` stops computers applying this GPO from changing their AD computer account password.
3. The domain Group Policy `The Domain member: Disable machine account password changes` policy setting determines whether a domain member periodically changes its computer account password. Setting its value to Enabled prevents the domain member from changing the computer account password. Setting it to Disabled allows the domain member to change the computer account password as specified by the value of the Domain member: Maximum machine account password age policy setting, which is

every 30 days by default." The domain Group Policy "Domain member: Maximum machine account password age" which tells the domain joined computers how often they should change their computer account password (though this is more of a guideline, than a rule). By default, this value is set to "30", but if it is set to "0", computers are unable to change their passwords.

4. There's a Domain Controller Group Policy setting `Domain controller: Refuse machine account password changes` that sets the Domain Controller to prevent clients from updating their computer account password in AD. "This security setting determines whether domain controllers will refuse requests from member computers to change computer account passwords. By default, member computers change their computer account passwords every 30 days.
    1. If enabled, the domain controller will refuse computer account password change requests. If it is enabled, this setting does not allow a domain controller to accept any changes to a computer account's password." This is a valid (and current) method for an attacker to persist access even after all the user, admin, and service account passwords are changed.

In normal Kerberos operations, the authentication ticket (TGT) is used to request service tickets (TGS) for each Kerberos enabled service. Silver Tickets bypass this normal process by injecting the forged Kerberos TGS tickets directly. Multiple Silver Tickets may be required to access the target service(s). 7

Advanced Techniques

Diamond and Sapphire Attacks

Both the Sapphire and Diamond Ticket attacks decrypt a legitimate TGT and change its PAC, and in order to do that, the adversary needs to have access to the KRBTGT account's key (the password hash). The KRBTGT account, as described by Microsoft TechNet (now Microsoft Docs), is a local default account that acts as a service account for the KDC service.

Sapphire Ticket

The Sapphire ticket attack (introduced by Charlie Bromberg), requires getting credentials of any user in the domain. Let's call that user Joe. Joe's credentials will be used to obtain a TGT (using a regular KRB_AS_REQ) and later to decrypt the PAC of a high-privileged user.

Once the TGT is received, the adversary will use the U2U + S4U2Self:

1. Decide on the user they want to impersonate (high-privileged user)
2. Generate a KRB_TGS_REQ with the following attributes:
    1. The PA_FOR_USER struct contains the impersonated user
    2. The service name (sname) is Joe's username
    3. Joe's TGT will be added to the additional-tickets field
    4. ENC-TKT-IN-SKEY flag in the KDC Option is set
3. Obtain a service ticket for the impersonated user 2

Diamond Ticket

The first part of the Diamond Ticket attack (introduced by Charlie Clark and Andrew Schwartz) is obtaining a TGT. This can be done by using one of the following techniques:

- Adversaries will use the low-privileged user's credentials to initiate a KRB_AS_REQ, which will result in a legitimate TGT.
- A tool called Rubeus has an option called tgtdeleg that abuses the Kerberos Generic Security Services Application Program Interface (GSS-API) to retrieve a usable TGT for the current user without needing elevation on the host.

Once receiving the TGT in the KRB_AS_REP, the adversary can now decrypt the TGT using the KRBTGT account's key and modify each part of the ticket.

After modifying the ticket, an attacker can escalate their privileges using one of the following approaches:

- Generating a TGT to impersonate a domain admin or any other user in the domain.

In this scenario the attacker will need to change the cname field in the ticket as well as the PAC. This method will result in a forged ticket under the name of the domain admin (or any other user in the domain), which is pretty similar to the Golden Ticket attack.

- Elevating a normal domain user's privileges to domain admin privileges by modifying the PAC. 2

Why are these attacks useful?

Diamond and Sapphire Tickets are forged TGTs created by modifying a legitimate TGT, which gives it additional privileges or a new identity. While many Golden Ticket detections are based on the absence of a TGT creation by a legitimate DC, the new attacks manipulate a legitimate TGT that was issued by the DC, which makes them harder to detect.

After the TGT is forged, the attacker uses it to request a TGS to any service or resource they desire. For example, they can request access to all computers, files and folders in the domain. 2

Unconstrained Delegation

Resources

1. https://www.crowdstrike.com/en-us/cybersecurity-101/cyberattacks/golden-ticket-attack/
2. https://unit42.paloaltonetworks.com/next-gen-kerberos-attacks/
3. https://www.tarlogic.com/blog/how-to-attack-kerberos/ (Includes info on SPN retrieval, Golden and Silver ticket creation)
4. https://www.varonis.com/blog/kerberos-attack-silver-ticket
5. YouTube: Modern Active Directory Attacks, Detection, and Protection
6. From Azure AD to Active Directory (via Azure) – An Unanticipated Attack Path
7. https://www.ivoidwarranties.tech/posts/pentesting-tuts/ad/kerberos-silver-tickets/
8. https://www.optiv.com/insights/source-zero/blog/kerberos-domains-achilles-heel

#kerberos #kerberoasting #netexec #lateral-movement #meterpreter