CERT-EU Security Whitepaper 2014-007

Kerberos Golden Ticket Protection

Mitigating Pass-the-Ticket on Active Directory

Miguel SORIA-MACHADO, Didzis ABOLINS,
Ciprian BOLDEA, Krzysztof SOCHA
ver. **1.4**
April 26, 2016

1 Introduction

Kerberos authentication protocol is the preferred authentication mechanism used by Windows in a domain-based environment, and interoperates with Kerberos implementations supported by other operating systems. While the *pass-the-hash* (PtH) technique is still used by *Advanced Persistent Threats* (APT), the equivalent technique misusing the Kerberos protocol, known as pass-the-ticket (PtT), is increasing[1].

The Kerberos protocol, invented by MIT and used by multiple operating systems, relies on a secret key in order to protect the authentication. If the server that stores the secret key is hacked, it may be possible for a malicious actor to generate credentials to appear to be other users.

A recent release of Mimikatz[2] provides a proof of concept of this pass-the-ticket attack called the **golden ticket**. Before the golden ticket is possible, the malicious actor must first hack the system with the secret key (Active Directory, the domain controller), then hack to become a full system administrator on the same domain controller. The adversary uses this access to steal the secret key, effectively a golden-ticket that enables the adversary to impersonate anybody in a Windows-domain based environment until the Kerberos secret key is reset.

This white-paper provides the required steps to prevent and block attacks based on the golden-ticket.

The tests mentioned in this document were done in a Windows 7 and Windows Server 2008R2 environment. New security controls have been put in place in Windows 8.1 and 2012R2 which mitigate the risks described in this paper. See [4] for further details. However, CERT-EU strongly recommends putting in place the controls described in this paper.

2 The Golden Ticket

2.1 What is a Golden Ticket

Mimikatz is a tool used by security researchers for pen-testing and studies purposes. It is publicly available and can potentially also be used with malicious intent.

After an adversary hacks a system and then hacks to obtain full administrator privileges, the tool can dump Windows credentials, like NT hashes and Kerberos tickets, from memory and perform pass-the-hash and pass-the-ticket attacks. A recent release of Mimikatz includes a new feature called golden ticket. If the adversary is able to hack to gain full administrator privileges on a Windows Domain Controller this feature allows creating a special **Kerberos TGT ticket** which has the following properties[3]:

[1] Refer to [1] for more details about credential theft attacks, like pass-the-hash or pass-the-ticket. You can also find a short explanation in Annex A – Definitions and Annex B – Introduction to pass-the-ticket.

[2] January 2014, http://blog.gentilkiwi.com/securite/mimikatz/golden-ticket-kerberos (French)

[3] Refer to Annex B – Introduction to pass-the-ticket for a short explanation of this type of attack.

- **the golden ticket is a method to arbitrarily generate Kerberos TGT tickets for any user of the target domain**[4]. Therefore, it can be used to **impersonate anybody**, Domain Administrators accounts are the most interesting but potentially any legitimate user can be impersonated;
- golden tickets can be **created off-line**. Therefore, one does not need to be connected to the domain once you have collected all the data required to create the ticket (see next section);
- the golden ticket is valid for an **arbitrary lifetime, Mimikatz default is 10 years – or until a Domain Administrator resets the Kerberos key used to generate the TGT**. This is the current setting implemented by Mimikatz but it should be possible to create tickets of any lifetime at anytime (arbitrary start, renewal and end time is possible);
- **Kerberos lifetime policy** (default renewal lifetime 10h and total lifetime is 7 days) **does not have any impact on the golden ticket**. Indeed, KDC validates TGT tickets based on the lifetime settings embedded in the protected core of the ticket and not on the policy set on the Domain Controller. Nevertheless, even if such control is in place, it cannot be used to block golden tickets. The attacker can generate any ticket with the appropriate lifetime in line with the local policy and so bypass the control;
- once created, the golden ticket can be replayed with **pass-the-ticket attack** technique. This will allow the adversary accessing other resources available to the impersonated user;
- as any pass-the-ticket, there is **no need of privilege access** to replay and use the golden ticket;
- resetting the password of the impersonated account **does not make the golden ticket invalid**;
- resetting the Kerberos secret key does make all golden tickets invalid.
- Windows event logs does not distinguish the use of legitimate TGT ticket versus a golden ticket, so there is **no universal rule to detect** the use of a golden ticket;

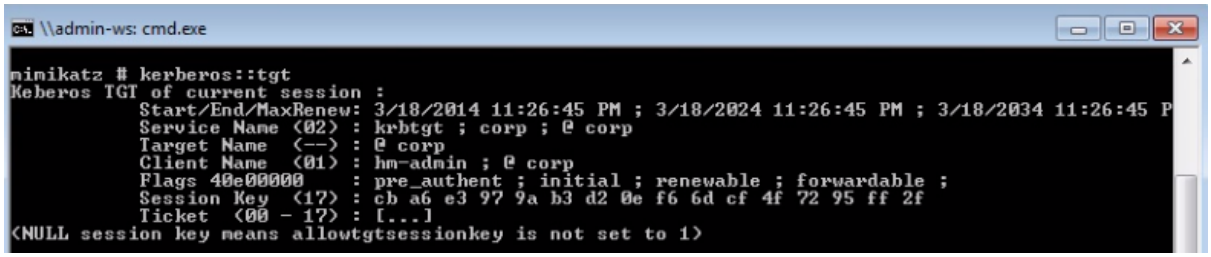

Figure 1: Example of a golden ticket with life time until 2024

2.2 Requirements for Golden Ticket

This is the list of items required to create the golden ticket:

- the NT hash of the domain `KRBTGT` account:

[4]NT hash is derived from the password.

This hash is the secret key used for Kerberos and is only hosted in the Active Directory of the Domain Controller. Therefore, **the adversary must have both compromised the Domain Controller and elevated privilege to get full administrator privileges first**. This can be achieved with lateral movements like described in Annex B – Introduction to pass-the- ticket. Example of a KRBTGT hash:

```
KRBTGT:502:NO PASSWORD**********************:a54932c2b0560e12386a0214767fbd5c:::
```

- the **targeted domain account name**, usually a domain admin;
- the **domain name**, this is trivial to obtain once the attacker has access to a host in the domain;
- the **domain's SID**:

  This can be obtained from a domain user SID or sysinternals' `psGetsid.exe`.

3 Countermeasures against the Golden Ticket

3.1 Prevention by Protecting the DC and Sensitive Accounts

The main requirement to create a golden ticket is NT hash of the KRBTGT account which implies having full administrator access to the Domain Controller. This may be achieved by stealing sensitive domain accounts and do lateral movements with pass-the-hash or pass-the-tickets attacks as explained in Annex B – Introduction to pass-the-ticket.

Microsoft has released a white paper that provides the controls to put in place to protect your domain against credential-theft-based attacks. The proposed countermeasures may require a re-engineering of your Domain Controller and its management but we strongly recommend implementing them. Indeed, pass-the-[credential] attacks are always used in APTs and Domain Controller is one of their main targets.

See [1] Mitigating Pass-the-Hash (PtH) Attacks and Other Credential Theft Techniques for the full list of controls.

3.2 Containment by Resetting Twice the KRBTGT Account Password

As mentioned earlier, resetting the password of the impersonated user does not block the usage of the related golden ticket. However, resetting **twice** the built-in Key Distribution Service account (KRBTGT) password will make invalid any golden tickets created with the previously stolen KRBTGT hash as well as all other Kerberos tickets.

3.2.1 Reasoning

The following simplified steps describe the Kerberos protocol when requesting access to a resource:

1. The requestor authenticates to the KDC. If the authentication is successful, the KDC creates the TGT ticket and encrypts the core of the TGT ticket with the NT hash of `KRBTGT` account before sending it back to the requestor. Note that this step is optional if a valid TGT ticket is already in memory (SSO feature) of the requestor host;
2. The TGT ticket does not provide direct access to a resource. The requestor send first the TGT ticket to the KDC to request a Service Ticket for the specific resource (ex, remote host);
3. The KDC decrypts the core of TGT ticket with its hash, if the KDC fails to decrypt with the hash, it will try with the former hashes stored in the Active Directory (i.e. the password history feature);
4. The KDC validates the content before creating and issuing the related Service Ticket (ST) to the requestor; If this latter step succeed, the requestor will send the ST to the resource to be accessed (ex, to the remote host of step 2);

Consequently, changing the `KRBTGT` account password[5] will make the step-2 fail. In addition, as the KDC retries with former hashes, the golden ticket will be invalid only if the whole history of password is reset, 2 by default.

3.2.2 Impact

Changing the `KRBTGT` account hash may have an impact on the availability of legitimate session used by account in the target domain. The exact impact level will depend on the criticality of the related users, services and application. If done as part of the recovery from a domain controller administrator-level compromise, the impact should be no different from other steps (such as resetting user and service accounts). Here are some examples:

- End-users including smart-card users will be automatically requested to authenticate to receive a new valid ticket. This might create some load on the Domain controller;
- Services and applications that require manual startup with a password and use Kerberos may stop working properly until next manual restart.

Consequently, it is recommended to do a first assessment before resetting the `KRBTGT` account password.

3.2.3 How to Reset KRBTGT Account Password

The following instructions depicts the way to reset the `KRBTGT` account password in a Domain Controller (Windows Server 2008):

[5] NT hash is derived from the password.

1. Open Active Directory Users and Computers on you Domain Controller;
2. Select Advanced Features in View menu;
3. Select Users in the target domain and find the KRBTGT user account;
4. Right-click the account, and then click “Reset password”;
5. Uncheck all options and enter the new password twice;
6. Redo step 4 and 5 to reset the password a second time.

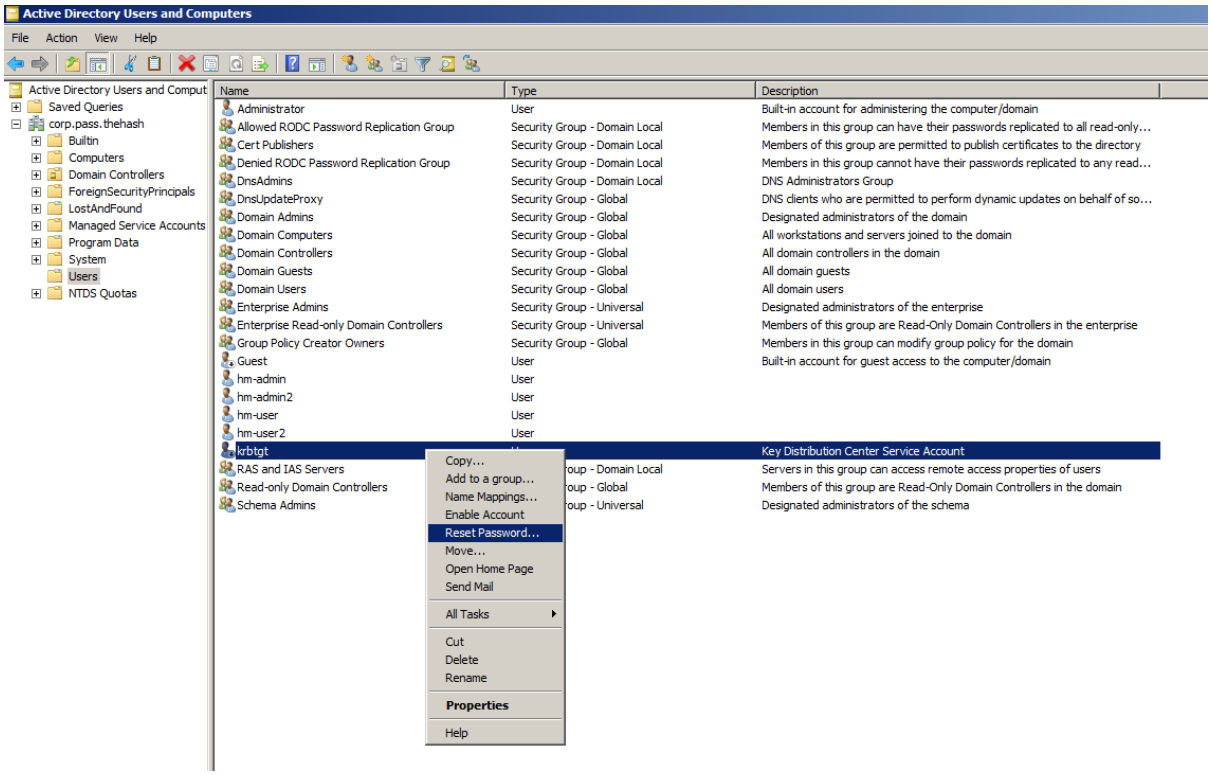

Figure 2: Reset password window

Alternatively, it is possible to use KRBTGT Account Password Reset Scripts from Microsoft[6] to reset KRBTGT password across all writable domain controllers that minimizes the hassle to do it by hand on each one, and possibly minimizing the availability impact. Also, it allows to follow a step-by-step list and to validate results.

3.3 Eradication by Recovering the Domain Controller

As the domain controller is accessed to extract the required NT hash of the KRBTGT account, the usual procedure to recover from compromised Active Directory applies here.

CERT-EU recommends following your recovery procedure. We also recommend to read Microsoft guideline [3] Chapter 3 - Recovering from Active Directory Attacks.

[6] https://blogs.microsoft.com/cybertrust/2015/02/11/krbtgt-account-password-reset-scripts-now-available-for-customer

3.4 Detection

3.4.1 Security Events when Using a Valid Golden Ticket

As any pass-the-ticket attack, the attacker replays the golden ticket in a standard Kerberos protocol. Therefore, there is no clear indication of such attack in Windows logs. Still there might be some anomalies if the golded ticket is not properly prepared. Such anomalies were observed in the Account Domain field in following events : `Event ID: 4624` (Account Logon), `Event ID: 4672` (Admin Logon), `Event ID: 4634` (Account Logoff).

Account Domain might be `<3 eo.oe ~ ANSSI E>`, a FQDN, blank, or other value, while in fact it should be short domain name. The string `<3 eo.oe ~ ANSSI E>` is hardcoded in the source code of the tool generating the ticket and can be modified. For instance in `Mimikatz 2.0 alpha (x86)` release *Kiwi en C*, in the file:

```
/mimikatz/modules/kerberos/kuhl_m_kerberos.c:589
```

we can find:

```
RtlInitUnicodeString(&validationInfo.LogonDomainName, L"<3 eo.oe ~ ANSSI E>");
```

While such indicators may be looked for when investigating potential cases of golden ticket use, they are not guranteed to be present. For example Kerberos tickets forged using Mimikatz January 2016 update, no longer include a domain anomaly since the netbios domain name is placed in the domain component of the Kerberos ticket [6].

3.4.2 Detecting a Golden Ticket Based on Lifetime

The method presented in [5] is looking for Ticket Granting Tickets (TGT) which have a duration (lifetime) that is different from the value set per domain.

3.4.3 Windows Security Event after Resetting `KRBTGT` Password

The following `Event ID: 4769` (A Kerberos service ticket was requested) is generated on the Domain Control when using a golden ticket after the **double** reset of the `KRBTGT` password. The status code `0x1f` indicates a failed attempt with the following reason:

```
0x1F: Integrity check on decrypted field failed
```

Unfortunately, there is no additional information like the username linked to the TGT ticket or the resource that was requested. Consequently, the monitoring of this event is very limited and cannot be used efficiently to detect a potential use of a golden ticket.

Nevertheless, this type of error should be rare and might be worth to monitor and investigate.

```
Event id: 4769 - A Kerberos service ticket was requested.
Account Information:
 - Account Name:
 - Account Domain:
 - Logon GUID:  {00000000-0000-0000-0000-000000000000}
```

```
Service Information:
 - Service Name:
 - Service ID:  NULL SID
Network Information:
 - Client Address:  ::ffff:192.168.89.101
 - Client Port:  49278
Additional Information:
 - Ticket Options:  0x40810000
 - Ticket Encryption Type: 0xffffffff
 - Failure Code:  0x1f
 - Transited Services: -
```

`Client address` is the source IP address from where the ticket was used. In some circumstance, we have seen events with the same error code but without any IP address.

4 Conclusion

If an adversary is able to compromise and gain full administrative privileges on a domain controller, then the adversary has a significant degree of control over the domain and can take many malicious actions. If this type of compromise is not properly contained then the golden ticket is a real threat and is a dangerous asset in APTs:

- The adversary can use the technique to impersonate anybody in the domain and gain access to sensitive resources;
- They also retain privilege access in the domain after a first eradicated wave of attack. Indeed, if during a first eradicated wave of attack, the adversary was able to get a copy of the Active Directory and this latter has not been properly reset as explained in this paper, this same entity can immediately get privileged access in a second wave of attack. Only a simple but successful spear-phishing attempt against one internal workstation would be enough to gain again full control in a domain. There is no need of a privilege account on the local host to perform a pass-the-ticket attack.

We recommend the following guideline to protect against this type of attack:

- Prevention by protecting the DC and sensitive accounts (section 3.1);
- Containment by resetting **twice** the `KRBTGT` account password (section 3.2);
- Potentially detect the attack by monitoring event `4769` with error code `0x1f` (section 3.4.2). However, this event may lead to false-positives;

5 Annex A – Definitions

Pass-the-hash : is a hacking technique that allows an attacker to authenticate to a remote server/service by using the previously stolen underlying NTLM and/or LanMan hash of a user's password, instead of requiring the associated plaintext password as is normally the case.

Pass-the-ticket : similar to Pass-the-hash but with Kerberos tickets instead of NTLM/LanMan hashes;

Credentials : the identity and the related secret (authenticator) that can be used to proof someone identity. Consequently, a type of credential is not limited to plain-text passwords but it can be a Windows NTLM hash or a Kerberos Ticket depending on the Windows Authentication protocol that is used. In some circumstances, Windows caches the credentials to provide the single-sign-on feature. This paper will focus on Kerberos ticket called Ticket-Granting-Tickets (TGT). See [1] for more details about Windows Credential types (table 4) and caching.

TGT and ST tickets : Ticket-Granting-Tickets (TGT) and Service Tickets are part of the Kerberos protocol, refer to section 3.2.1. for a short description of these tickets. For more details about Kerberos and related tickets see [2].

KDC : Key Distribution Center.

Key Distribution Service account (`KRBTGT`) : this is a built-in account in the Domain Controller used by the KDC. This account has a hash which is used to create Kerberos tickets.

6 Annex B – Introduction to Pass-the-Ticket

The golden ticket exploits the pass-the-ticket technique. We provide below a short introduction to pass-the-ticket.

Pass-the-[Kerberos]-ticket is a similar attack than pass-the-hash attack. In both cases, the attack is a 2-steps-based scenario:

1. Capture the credential from memory of a compromised host, the Kerberos ticket (TGT or ST) in this case. This scenario requires 1.1. having control on a compromised host in the target network. This is usually obtained via spear phishing. 1.2. having high privileges or `sedebug` privileges on the compromised host. A local privilege escalation vulnerability can be used for this if the compromised user does not have such privileges yet. This privilege will allow the attack to access the memory (ie, `LSAss` process) and extract the credentials. The attacker can then take any credential that is stored in memory. The most interesting ones are privileged accounts of your domain like help-desk or domain admin account.
2. Replay the ticket to access resources: Once the credential is stolen, the attacker can use it to access another resource like another host or server (ex, Exchange email accounts). This is called lateral movement. The advantage of this attack is that the attacker can capture a credential and use it latter on with specialized tools like Mimikatz.

It is important to note that those credentials, including domain admins one, are cached in memory of the workstation where they connected to or run commands, for instance when doing RDP or runas commands. Therefore if these privileged accounts are connected to an infected machine, their credentials are exposed. Usually, APTs will target other workstation where it can obtain higher and higher privilege accounts until they get the credential of a domain admin account. The final step of the attack will be to access the Domain Controller and get a copy of the entire Active Directory.

The following picture presents the typical scenario of an APT with lateral movement:

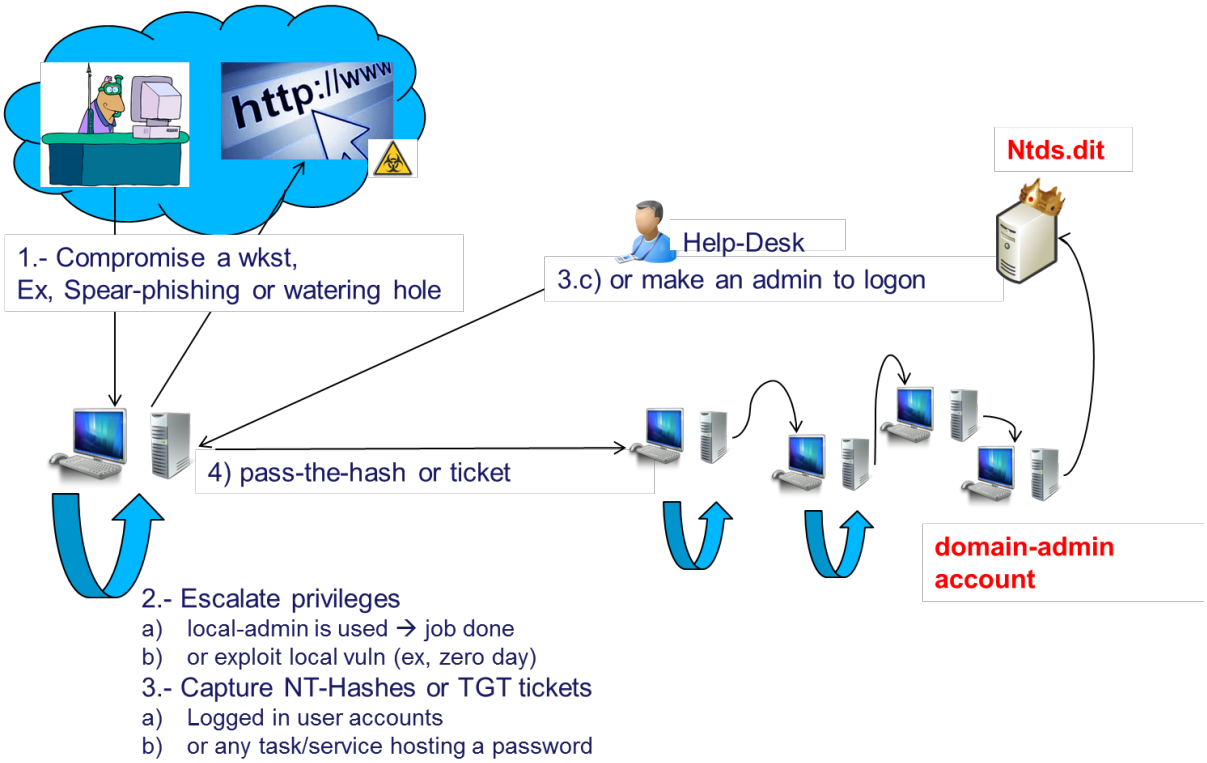

Figure 3: Typical lateral movements in APTs

You can find more details about pass-the-hash(-ticket) and credential theft from memory in [1]. This document published by Microsoft provides as well counter-measures to protect your environment against such attack scenarios.

7 Annex C – References

[1] http://www.microsoft.com/en-us/download/details.aspx?id=36036 Mitigating Pass-the-Hash (PtH) Attacks and Other Credential Theft, Version 1 and 2

[2] http://technet.microsoft.com/en-us/library/cc739058(v=ws.10).aspx Kerberos Authentication Technical Reference

[3] http://technet.microsoft.com/en-us/library/bb727066.aspx#ECAA Chapter 3 - Recovering from Active Directory Attacks

[4] http://technet.microsoft.com/en-us/library/dn408190.aspx Credential Protection and Management

[5] https://gallery.technet.microsoft.com/scriptcenter/Kerberos-Golden-Ticket-b4814285

[6] https://adsecurity.org/?p=1515#DetectingForgedKerberosTickets

8 Annex D - Additional Details

8.1 How to Create a Golden Ticket

This is an example of how to create a golden ticket with Mimikatz for the account `myadmin` and save it in `myadmin-golden.kiribi` file. The attacker can then safe the file and load it back in memory on any compromised host in the target domain whenever he wants[7]:

```
mimikatz # kerberos::golden /admin:myadmin /domain:corp
  /sid:S-1-5-21-2976932740-3244455291-537790045 /KRBTGT:a54932c2b0560e12386a0214767fbd5c
  /ticket:myadmin-golden.kiribi
```

Here is the related profile of the just that was created. As you can see the lifetime is of 10 years. Notice the renewal lifetime is longer than the actual lifetime of the ticket, this should avoid the need to constantly renew the ticket and make it usable at any time before the end time:

```
mimikatz # kerberos::tgt
Keberos TGT of current session :
           Start/End/MaxRenew: 3/18/2014 11:26:45 PM ; 3/18/2024 11:26:45 PM ; 3/18/2034
     11:26:45 PM
           Service Name (02) : KRBTGT ; corp ; @ corp
           Target Name  (--) : @ corp
           Client Name  (01) : myadmin ; @ corp
           Flags 40e00000    : pre_authent ; initial ; renewable ; forwardable ;
           Session Key  (17) : cb a6 e3 97 9a b3 d2 0e f6 6d cf 4f 72 95 ff 2f
           Ticket  (00 - 17) : [...]
(NULL session key means allowtgtsessionkey is not set to 1)
```

The following screen shot provides an example simulating the usage of the golden ticket[8] 6 years after its creation. The picture does not show the load in memory, this was done with the following command:

```
mimikatz # kerberos::tgt ticket:myadmin-golden.kiribi
```

8.2 Additional Events Generated by a Golden Ticket

The following section depicts the events that are recorded on the DC and target host while using a golden ticket. The test scenario was as follows:

- The golden ticket was created for `myadmin` user who is part of the domain admin group.
- The attacker used the ticket from a workstation called `user-ws` (`192.168.89.101`) to connect to a workstation call `admin-ws` (`192.168.86.102`).

[7] The `KRBTGT` NTLM hash is not directly accessible on the AD but tools are available to extract the required files and all hashes, like combining `ntdsutil` and `NTDSXtract 1.0` (http://www.ntdsxtract.com/).

[8] We used Mimikatz to upload the golden ticket in memory. The ticket is then automatically used by the Windows command (`net use` here) thanks to the *single sign-on* feature of Windows. Note also that privilege was escalated in this example (NT SYSTEM), but a normal account would have been enough to load the ticket in memory.

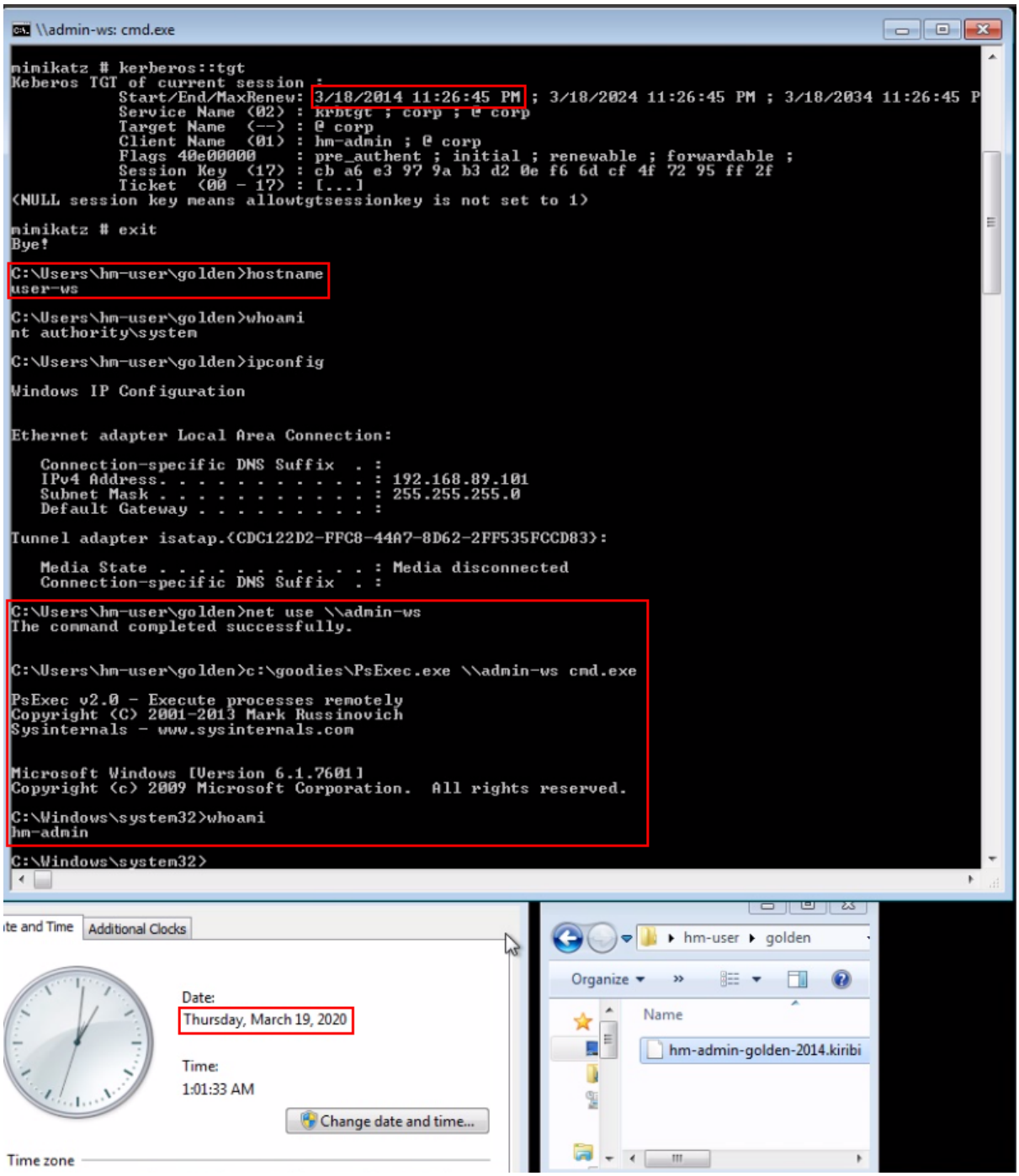

Figure 4: Example of golden ticket usage

- The domain name is `corp`

The *command* column shows the instructions run on user-ws that generated the event.

8.2.1 Domain Controller Events

We see on the Domain Controller that a Kerberos service ticket was requested to access the `admin-ws` host from the IP address of the `user-ws` machine (`192.168.86.101` – `event-id 4769`). However, there is no indication that this is a golden ticket, a legitimate use of `myadmin` account will create a similar entry in the security log.

Notice that there is no `event-id 4768` (Kerberos TGT request). This is in line with the fact that the golden ticket was created off-line.

```
Command: net use \\admin-ws
Date: 03/18/2020
Time: 14:11:12
Event: 4769
Event Content:
    - TargetUserName = myadmin@corp
    - TargetDomainName = corp
    - ServiceName = ADMIN-WS$
    - ServiceSid = S-1-5-21-2976932740-3244455291-537790045-1107
    - TicketOptions = 0x40810000
    - TicketEncryptionType = 0x00000012
    - IpAddress = ::ffff:192.168.89.101 IpPort = 49407
    - Status = 0x00000000
    - LogonGuid = {B757831E-D810-CDCC-C1C2-804BB3A2FB2C}
    - TransmittedServices = -
```

8.2.2 Target Host Events (`admin-ws`)

We see two events related to accounts successfully logged on the target machine (`event-id 4624`). As on the DC, none of them indicate it is a golden ticket:

```
Command: net use \\admin-ws
Date: 03/18/2020
Time: 14:11:12
Event: 4624
Event Content:
    - SubjectUserSid = S-1-0-0 SubjectUserName = - SubjectDomainName = - SubjectLogonId =
     0x0000000000000000
    - TargetUserSid = S-1-5-21-2976932740-3244455291-537790045-500 TargetUserName = myadmin
     TargetDomainName = corp
    - TargetLogonId = 0x000000000051f916
    - LogonType = 3 LogonProcessName = Kerberos AuthenticationPackageName = Kerberos
    - WorkstationName =
    - LogonGuid = {A0706C8D-9BC6-F4D5-1226-FA2A48BB58D9} TransmittedServices = - LmPackageName
      = - KeyLength = 0 ProcessId = 0x0000000000000000 ProcessName = -
    - IpAddress = 192.168.89.101 IpPort = 49406

Command: net use \\admin-ws
Date: 03/18/2020
Time: 14:11:12
```

```
Event: 4672
Event Content:
    - SubjectUserSid = S-1-5-21-2976932740-3244455291-537790045-500 SubjectUserName = myadmin
    - SubjectDomainName =
    - SubjectLogonId = 0x000000000051f916
    - PrivilegeList = SeSecurityPrivilege SeBackupPrivilege SeRestorePrivilege
     SeTakeOwnershipPrivilege SeDebugPrivilege SeSystemEnvironmentPrivilege
     SeLoadDriverPrivilege SeImpersonatePrivilege

Command: psexec \\admin-ws cmd
Date: 03/18/2020
Time: 14:11:39
Event: 4624
Event Content:
    - SubjectUserSid = S-1-0-0 SubjectUserName = - SubjectDomainName = - SubjectLogonId =
     0x0000000000000000
    - TargetUserSid = S-1-5-21-2976932740-3244455291-537790045-500 TargetUserName = myadmin
     TargetDomainName = corp
    - TargetLogonId = 0x00000000005204ad
    - LogonType = 3 LogonProcessName = Kerberos AuthenticationPackageName = Kerberos
    - WorkstationName =
    - LogonGuid = {B504E2E8-3007-1C03-F480-011559C08D34} TransmittedServices = - LmPackageName
      = - KeyLength = 0 ProcessId = 0x0000000000000000 ProcessName = -
    - IpAddress = 192.168.89.101 IpPort = 49409

Command: psexec \\admin-ws cmd
Date: 03/18/2020
Time: 14:11:39
Event: 4672
Event Content:
    - SubjectUserSid = S-1-5-21-2976932740-3244455291-537790045-500 SubjectUserName = myadmin
     SubjectDomainName =
    - SubjectLogonId = 0x00000000005204ad
    - PrivilegeList = SeSecurityPrivilege SeBackupPrivilege SeRestorePrivilege
     SeTakeOwnershipPrivilege SeDebugPrivilege SeSystemEnvironmentPrivilege
     SeLoadDriverPrivilege SeImpersonatePrivilege
```