Detecting Forged Kerberos Ticket (Golden Ticket & Silver Ticket) Use in Active Directory

By Sean Metcalf

Published: 2015-05-03 · Archived: 2026-04-05 22:30:18 UTC

Over the last 6 months, I have been researching forged Kerberos tickets, specifically Golden Tickets, Silver Tickets, and TGTs generated by MS14-068 exploit code (a type of Golden Ticket). I generated forged Kerberos tickets using Mimikatz (Mimikatz Command Reference) and MS14-068 exploits and logged the results. Over the course of several weeks, I identified anomalies in the event logs that are clear indication of forged ticket use in an Active Directory environment.

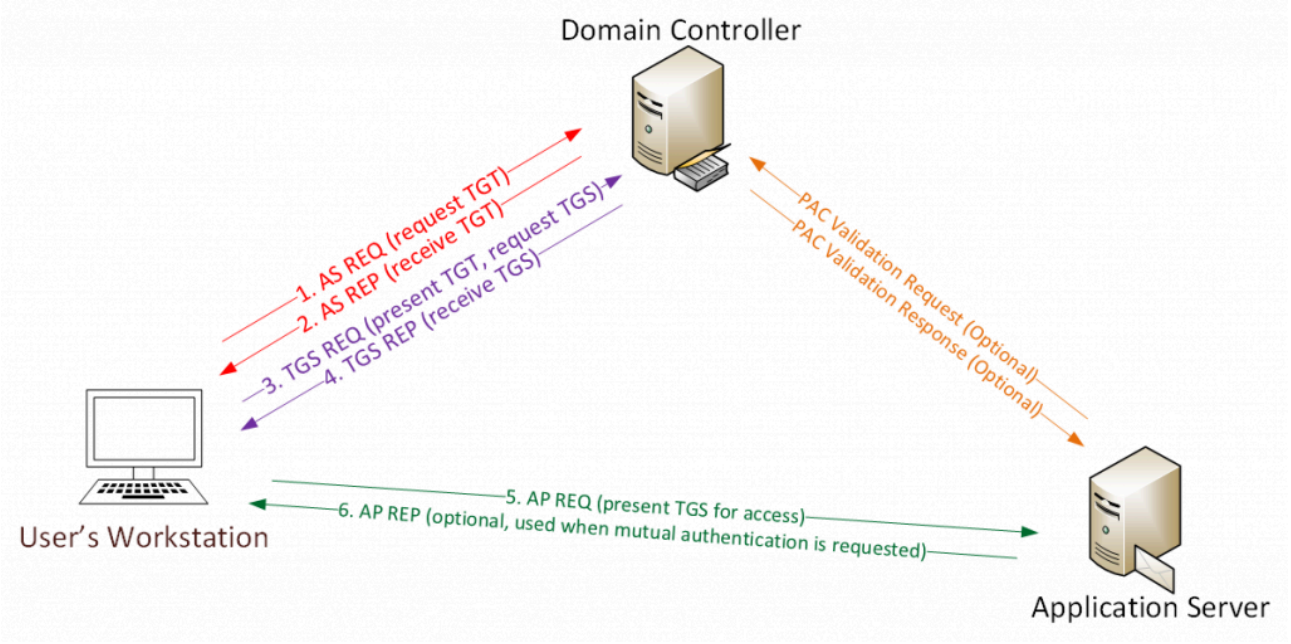
Update1/5/2016:
Around this time last year (early January 2015), I shared with customers these indicators for detecting forged Kerberos tickets and subsequently presented this information at BSides Charm 2015. Soon after, Mimikatz was updated with a domain field that was set to static values, usually containing the string “eo.oe”. As of the Mimikatz update dated 1/5/2016, forged Kerberos tickets no longer include a domain anomaly since the netbios domain name is placed in the domain component of the Kerberos ticket.

Mimikatz code diff:

```
588  601        KIWI_NEVERTIME(&validationInfo.PasswordMustChange);
589       -     RtlInitUnicodeString(&validationInfo.LogonDomainName, L"<3 eo.oe ~ ANSSI E>");
     602  +     RtlInitUnicodeString(&validationInfo.LogonDomainName, LogonDomainName);
```

More information on the difficulty of detecting forged Kerberos tickets (Golden Tickets, Silver Tickets, etc) in the in the Detecting Forged Kerberos Tickets section below.

Kerberos Overview & Communication Process:



User logs on with username & password.

1a. Password converted to NTLM hash, a timestamp is encrypted with the hash and sent to the KDC as an authenticator in the authentication ticket (TGT) request (AS-REQ).
1b. The Domain Controller (KDC) checks user information (logon restrictions, group membership, etc) & creates Ticket-Granting Ticket (TGT).

2. The TGT is encrypted, signed, & delivered to the user (AS-REP). *Only the Kerberos service (KRBTGT) in the domain can open and read TGT data.*

3. The User presents the TGT to the DC when requesting a Ticket Granting Service (TGS) ticket (TGS-REQ). The DC opens the TGT & validates PAC checksum – If the DC can open the ticket & the checksum check out, TGT = valid. The data in the TGT is effectively copied to create the TGS ticket.

4. The TGS is encrypted using the target service accounts' NTLM password hash and sent to the user (TGS-REP).

5.The user connects to the server hosting the service on the appropriate port & presents the TGS (AP-REQ). The service opens the TGS ticket using its NTLM password hash.

6. If mutual authentication is required by the client (think MS15-011: the Group Policy patch from February that added UNC hardening).

Unless PAC validation is required (rare), the service accepts all data in the TGS ticket with no communication to the DC.

Active Directory Kerberos Key Points:

- Microsoft uses the NTLM password hash for Kerberos RC4 encryption.
- Kerberos policy is only checked when the TGT is created & the TGT is the user authenticator to the DC.

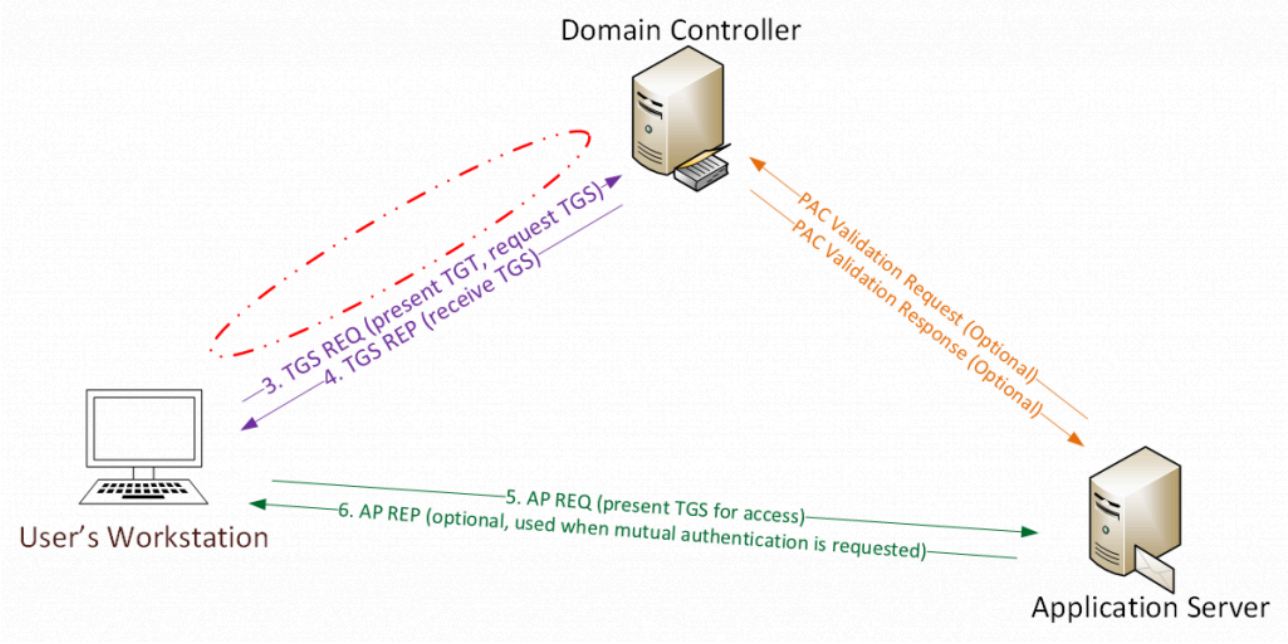
- The DC only checks the user account after the TGT is 20 minutes old to verify the account is valid or enabled. TGS PAC Validation only occurs in specific circumstances. When it does, LSASS on the server sends the PAC Validation request to the DC's netlogon service (using NRPC)
- If it runs as a service, PAC validation is optional (disabled). If a service runs as System, it performs server signature verification on the PAC (computer account long-term key).

Forging Kerberos Tickets:

- Forging Kerberos tickets depends on the password hash available to the attacker
- Golden Ticket requires the KRBTGT password hash.
- Silver ticket requires the Service Account (either the computer account or user account) password hash.
- Create anywhere and user anywhere on the network, without elevated rights.
- Spoof access without modifying AD groups.
- Once the KRBTGT account password is disclosed, the only way to prevent Golden Tickets is to change the KRBTGT password twice, since the current and previous passwords are kept for this account.

Golden Tickets:

Golden Tickets are forged Ticket-Granting Tickets (TGTs), also called authentication tickets.
As shown in the following graphic, there is no AS-REQ or AS-REP (steps 1 & 2) communication with the Domain Controller. Since a Golden Ticket is a forged TGT, it is sent to the Domain Controller as part of the TGS-REQ to get a service ticket.



- The **Kerberos Golden Ticket** is a valid TGT Kerberos ticket since it is encrypted/signed by the domain Kerberos account (KRBTGT). The TGT is only used to prove to the KDC service on the Domain Controller that the user was authenticated by another Domain Controller. The fact that the TGT is encrypted by the KRBTGT password hash and can be decrypted by any KDC service in the domain proves it is valid.

- Golden Ticket Requirements:
  * **Domain Name** [AD PowerShell module: (Get-ADDomain).DNSRoot]
  * **Domain SID** [AD PowerShell module: (Get-ADDomain).DomainSID.Value]
  * **Domain KRBTGT Account NTLM password hash**
  * **UserID for impersonation**.
- The Domain Controller KDC service doesn't validate the user account in the TGT until the TGT is older than 20 minutes old, which means the attacker can use a disabled/deleted account or even a fictional account that doesn't exist in Active Directory.

  Microsoft's MS-KILE specification (section 5.1.3):

  "Kerberos V5 does not provide account revocation checking for TGS requests, which allows TGT renewals and service tickets to be issued as long as the TGT is valid even if the account has been revoked. KILE provides a check account policy (section 3.3.5.7.1) that limits the exposure to a shorter time. KILE KDCs in the account domain are required to check accounts when the TGT is older than 20 minutes. This limits the period that a client can get a ticket with a revoked account while limiting the performance cost for AD queries."

- Since the domain Kerberos policy is set on the ticket when generated by the KDC service on the Domain Controller, when the ticket is provided, systems trust the ticket validity. This means that even if the domain policy states a Kerberos logon ticket (TGT) is only valid for 10 hours, if the ticket states it is valid for 10 *years*, it is accepted as such.
- The KRBTGT account password is never changed* and the attacker can create Golden Tickets until the KRBTGT password is changed (twice). Note that a Golden Ticket created to impersonate a user persists even if the impersonated user changes their password.
- It bypasses SmartCard authentication requirement since it bypasses the usual checks the DC performs before creating the TGT.
- This crafted TGT requires an attacker to have the Active Directory domain's KRBTGT password hash (typically dumped from a Domain Controller).
- The KRBTGT NTLM hash can be used to generate a valid TGT (using RC4) to impersonate any user with access to any resource in Active Directory.
- The Golden Ticket (TGT) be generated and used on any machine, even one not domain-joined.
- Used to get valid TGS tickets from DCs in the AD forest and provides a great method of persisting on a domain with access to EVERYTHING!

Mitigation:
Limit Domain Admins from logging on to any other computers other than Domain Controllers and a handful of Admin servers (don't let other admins log on to these servers) Delegate all other rights to custom admin groups. This greatly reduces the ability of an attacker to gain access to a Domain Controller's Active Directory database. If the attacker can't access the AD database (ntds.dit file), they can't get the KRBTGT account password data.

The KRBTGT account is disabled and stores the current password as well as the previous one. The KRBTGT password hash is used to sign the PAC in Kerberos tickets as well as encrypt the TGT (Authentication ticket). If a

ticket is signed/encrypted with a different key (password) then the DC (KDC) is expecting, it checks the KRBTGT previous password to see if that is successful. This is the reason why both passwords are kept.

It's advisable to regularly change the KRBTGT password (it is an admin account after all). Changing it once, then letting AD replicate, and changing it a second time about 12 to 24 hours later, will update both of the KRBTGT passwords (current and previous) in a manner that doesn't invalidate every existing Kerberos ticket. This process should have no negative impact on the environment (but as always, test first). This process should be the standard method for ensuring the KRBTGT password changes at least once a year (and when an AD admin leaves the organization, loss of control of AD backup media, etc).

Once an attacker has gained access to the KRBTGT account password hash(es), they can create Golden Tickets at will. Invalidate any existing Golden Tickets (and all active Kerberos tickets) by changing the KRBTGT password twice rapidly (aka "double-tap"). This invalidates all Kerberos tickets and removes the attacker ability to create valid Golden Tickets with their KRBTGT (assuming they don't have the ability to pull the updated KRBTGT pw hashes). This KRBTGT password "double-tap" is required when in a breach scenario and there's an active attacker operating on the network. Contact an Incidence Response company if you are in this situation first.

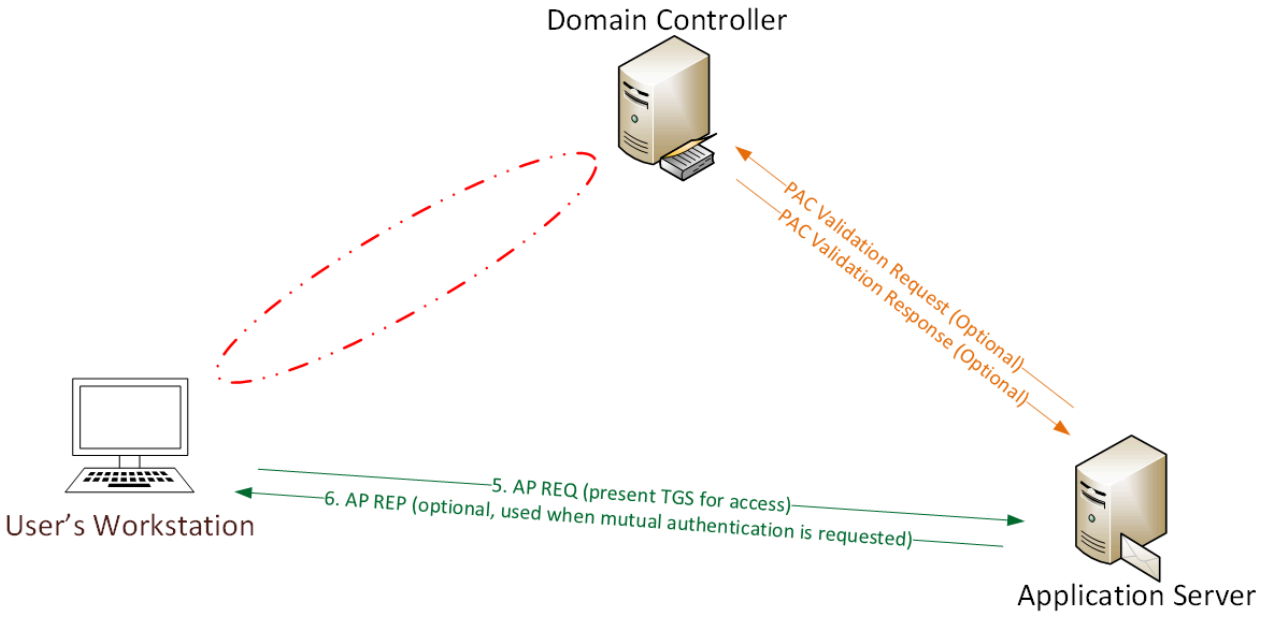
Using Mimikatz to Forge Kerberos Tickets:

The Mimikatz command Kerberos:Golden is used to create the "Golden Ticket" (forged TGT authentication ticket).
Mimikatz command example:
kerberos::golden /admin:ADMIINACCOUNTNAME /domain:DOMAINFQDN /id:ACCOUNTRID /sid:DOMAINSID /krbtgt:KRBTGTPASSWORDHASH /ptt

```
mimikatz(commandline) # kerberos::golden /admin:DarthVader /domain:lab.adsecurity.org /id:2601 /sid:S-1-5-21-1387203482-
2957264255-828990924 /krbtgt:8a2f1adcdd519a2e515780021d2d178a /startoffset:0 /endin:600 /renewmax:10080 /ptt
User      : DarthVader
Domain    : lab.adsecurity.org
SID       : S-1-5-21-1387203482-2957264255-828990924
User Id   : 2601
Groups Id : *513 512 520 518 519
ServiceKey: 8a2f1adcdd519a2e515780021d2d178a - rc4_hmac_nt
Lifetime  : 3/12/2015 9:44:08 PM ; 3/13/2015 7:44:08 AM ; 3/19/2015 9:44:08 PM
-> Ticket : ** Pass The Ticket **

 * PAC generated
 * PAC signed
 * EncTicketPart generated
 * EncTicketPart encrypted
 * KrbCred generated

Golden ticket for 'DarthVader @ lab.adsecurity.org' successfully submitted for current session

mimikatz(commandline) # exit
Bye!
PS C:\Users\JoeUser> klist

Current LogonId is 0:0xdac83

Cached Tickets: (1)

#0>     Client: DarthVader @ lab.adsecurity.org
        Server: krbtgt/lab.adsecurity.org @ lab.adsecurity.org
        KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
        Ticket Flags 0x40e00000 -> forwardable renewable initial pre_authent
        Start Time: 3/12/2015 21:44:08 (local)
        End Time:   3/13/2015 7:44:08 (local)
        Renew Time: 3/19/2015 21:44:08 (local)
        Session Key Type: RSADSI RC4-HMAC(NT)

PS C:\Users\JoeUser> net use \\adsdc02.lab.adsecurity.org\c$\windows\ntds
The command completed successfully.

PS C:\Users\JoeUser> whoami
adseclab\joeuser
PS C:\Users\JoeUser>
```

Silver Tickets:

Silver Tickets are forged Ticket Granting Service 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.

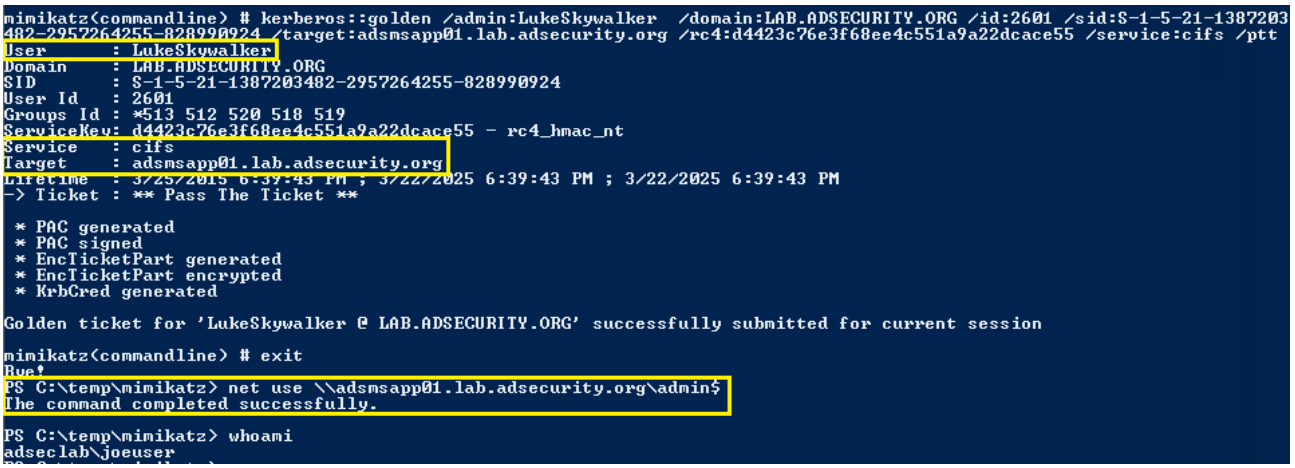


- Alluded to at BlackHat during the "Golden Ticket" presentation (Duckwall/Delpy) and discussed partly during Tim Medin's DerbyCon 2014 talk. Skip & Benjamin have provided additional information on Silver Tickets since, but confusion remains.
- The **Kerberos Silver Ticket** is a valid Ticket Granting 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

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 paramteer. 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



Detecting Forged Kerberos Tickets:

Most logon & logoff events include the following detail. Normal, valid account logon event data structure:

- Security ID: DOMAIN\AccountID
- Account Name: AccountID
- Account Domain: DOMAIN

I discovered that the domain field in many events in the Windows security event log are not properly populated when forged Kerberos tickets are used. The key indicator is that the domain field is blank or contains the FQDN instead of the short (netbios) name and depending on the tool used to generate the Kerberos tickets, other domain field anomalies may be present in the events.
The likely reason for the anomalies is that third party tools that create Kerberos tickets (TGT & TGS) don't format the tickets exactly the same way as Windows does.

The following includes some of the events I have identified that are logged when forged Kerberos tickets are used. Note that Silver Ticket events could be logged on any computer in the AD domain depending on what the target is, workstations, member servers, or Domain Controllers. Golden Tickets and MS14-068 exploit tickets, all of which are TGTs, will have events logged on the Domain Controller.

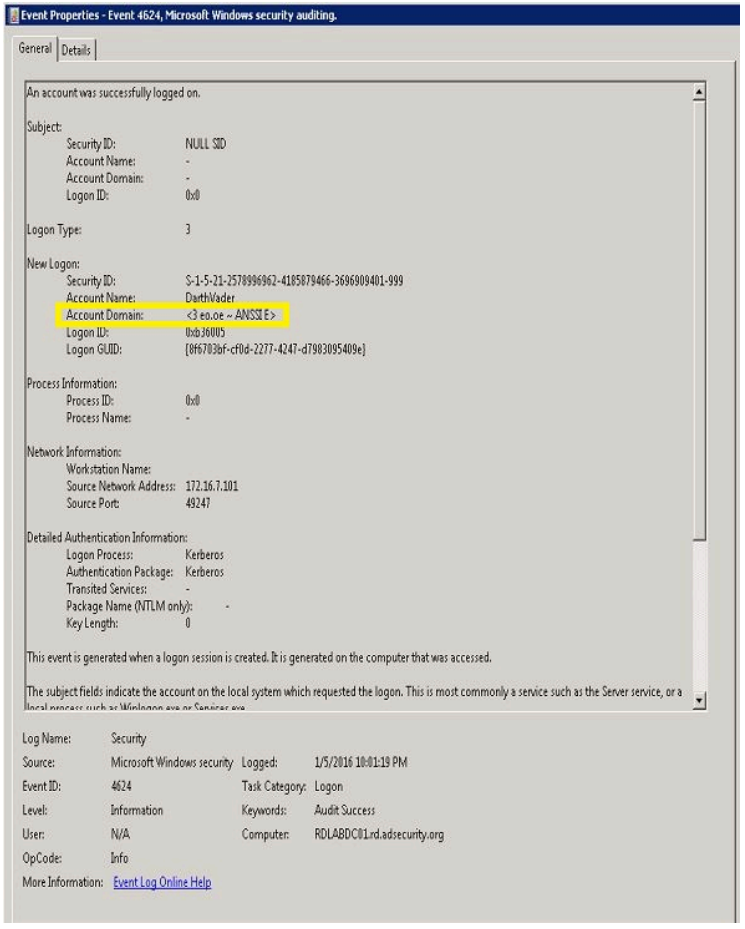
NOTE: As of 4/16/2015: Mimikatz generated tickets may include the string "eo.oe.kiwi :) " in the domain field.
NOTE: As of 6/29/2015: Mimikatz generated tickets may include the string "<3 eo.oe – ANSSI E>" in the domain field.

As stated at the top of this post, as of January 5th, 2016, Mimikatz no longer includes static values in Kerberos ticket domain fields which previously may have had anomalies from being blank to containing the string "eo.oe". As of the Mimikatz update dated 1/5/2016, forged Kerberos tickets no longer include a domain anomaly since the netbios domain name is placed in the domain component of the Kerberos ticket. This means that attackers using the Mimikatz version dated 1/5/2016 and/or Invoke-Mimikatz with this updated DLL will not trigger alerts based on the invalid domain fields I identified in the past.

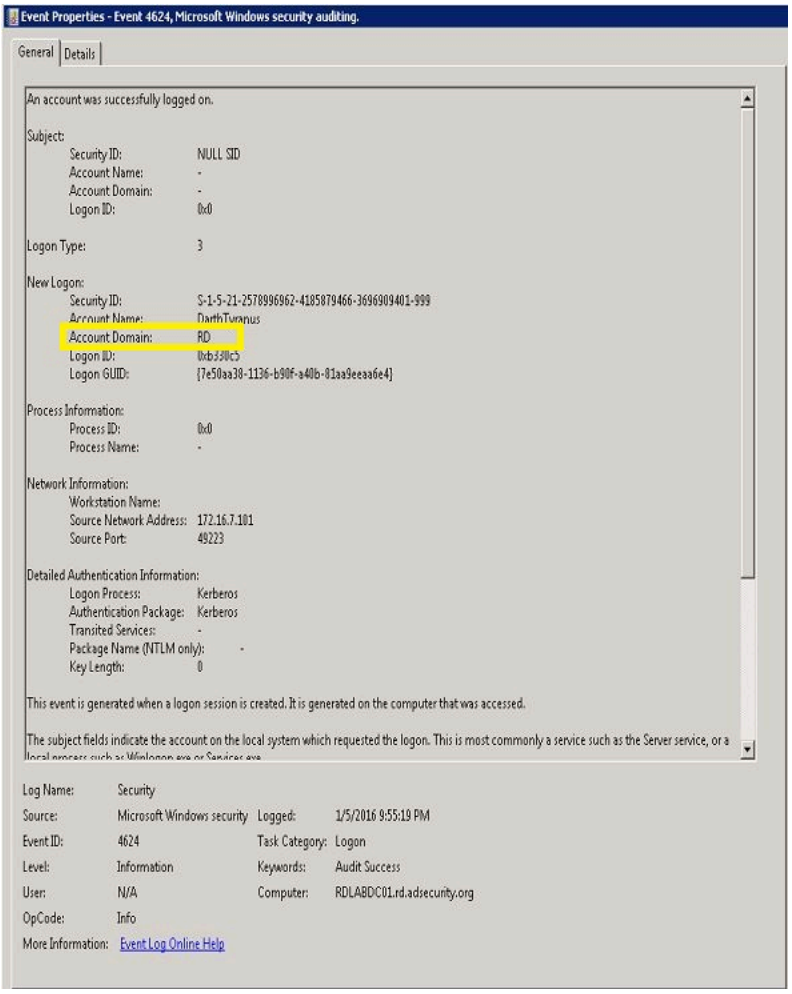
One method that is reliable is to look for RC4 encrypted Kerberos ticket usage. These should be rare on a modern network since Windows Vista & Windows Sever 2008 and newer support AES Kerberos encryption.
Note that if the attacker uses the NTLM password hash when creating the Golden Ticket, the TGT ticket will have RC4 encryption. If the Golden Ticket is created using the AES string, the TGT ticket will use AES, and will be very difficult to find. Read more about this in the Mimikatz guide.

User behavior analysis tools such as Microsoft Advanced Threat Analytics (ATA) is the best current method to detect this and other attack types *(though these methods also tend to involve ticket encryption type in the detection techniques).*
The best way to detect Golden Tickets is to correlate TGS requests to prior TGT requests. Since a TGT request should always preceed a TGS request, if there's no prior TGT request (within a threshold), then the TGS request may be related to a Golden Ticket.

Golden Ticket event from using Mimikatz dated (11/2015): Has the an invalid domain value ("*<3 eo.oe – ANSSI E>*")



Golden Ticket event from using Mimikatz dated (1//05/2015): Has the correct domain value (“RD”)



Expect that Mimikatz will continue to have different values in this field and that attackers may update this field to match the targeted environment. ***While these indicators may continue to have some value, they can't be relied on as primary detection of forged Kerberos ticket use. With that noted, monitoring events for domain field anomalies may still be the best and easiest way to detect forged Kerberos tickets other than looking for "special logon events" involving non-admins – these events are logged when accounts with admin rights log on.***

The following is left here for historic purposes and may be removed at a later date.

SILVER TICKET DETECTION

Silver Ticket events may have one of these issues:

- The Account Domain field is blank when it should be DOMAIN
- The Account Domain field is DOMAIN FQDN when it should be DOMAIN.

Event ID: 4624 (Account Logon)

Account Domain is FQDN & should be short domain name

Account Domain: LAB.ADSECURITY.ORG [ADSECLAB]
Event ID: 4634 (Account Logoff)

Account Domain is blank & should be short domain name

Account Domain: _____ [ADSECLAB]
Event ID: 4672 (Admin Logon)

Account Domain is blank & should be short domain name

Account Domain: _____ [ADSECLAB]

GOLDEN TICKET DETECTION

Golden Ticket events may have one of these issues:

- The Account Domain field is blank when it should be DOMAIN
- The Account Domain field is DOMAIN FQDN when it should be DOMAIN.

Event ID: 4624 (Account Logon)

Account Domain is FQDN & should be short domain name

Account Domain: LAB.ADSECURITY.ORG [ADSECLAB]

Event ID: 4672 (Admin Logon)

Account Domain is blank & should be short domain name

Account Domain: _____ [ADSECLAB]

MS14-068 Exploit Ticket Detection

MS14-068 events may have one of these issues:

- The Account Domain field is blank when it should be DOMAIN
- The Account Domain field is DOMAIN FQDN when it should be DOMAIN.
- Account Name is a different account from the Security ID.

PYKEK Events

Event ID: 4624 (Account Logon)

The Account Domain field is DOMAIN FQDN when it should be DOMAIN.

Account Name is a different account from the Security ID

Event ID: 4672 (Admin Logon)

The Account Domain field is DOMAIN FQDN when it should be DOMAIN.

Account Name is a different account from the Security ID

Event ID: 4768 (Kerberos TGS Request)

The Account Domain field is DOMAIN FQDN when it should be DOMAIN.

KEKEO Events

Event ID: 4624 (Account Logon)

The Account Domain field is DOMAIN FQDN when it should be DOMAIN.

Event ID: 4672 (Admin Logon)

Account Domain is blank & should be DOMAIN.

Event ID: 4768 (Kerberos TGS Request)

The Account Domain field is DOMAIN FQDN when it should be DOMAIN.
