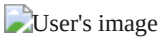
4768(S, F) A Kerberos authentication ticket (TGT) was requested. - Windows 10

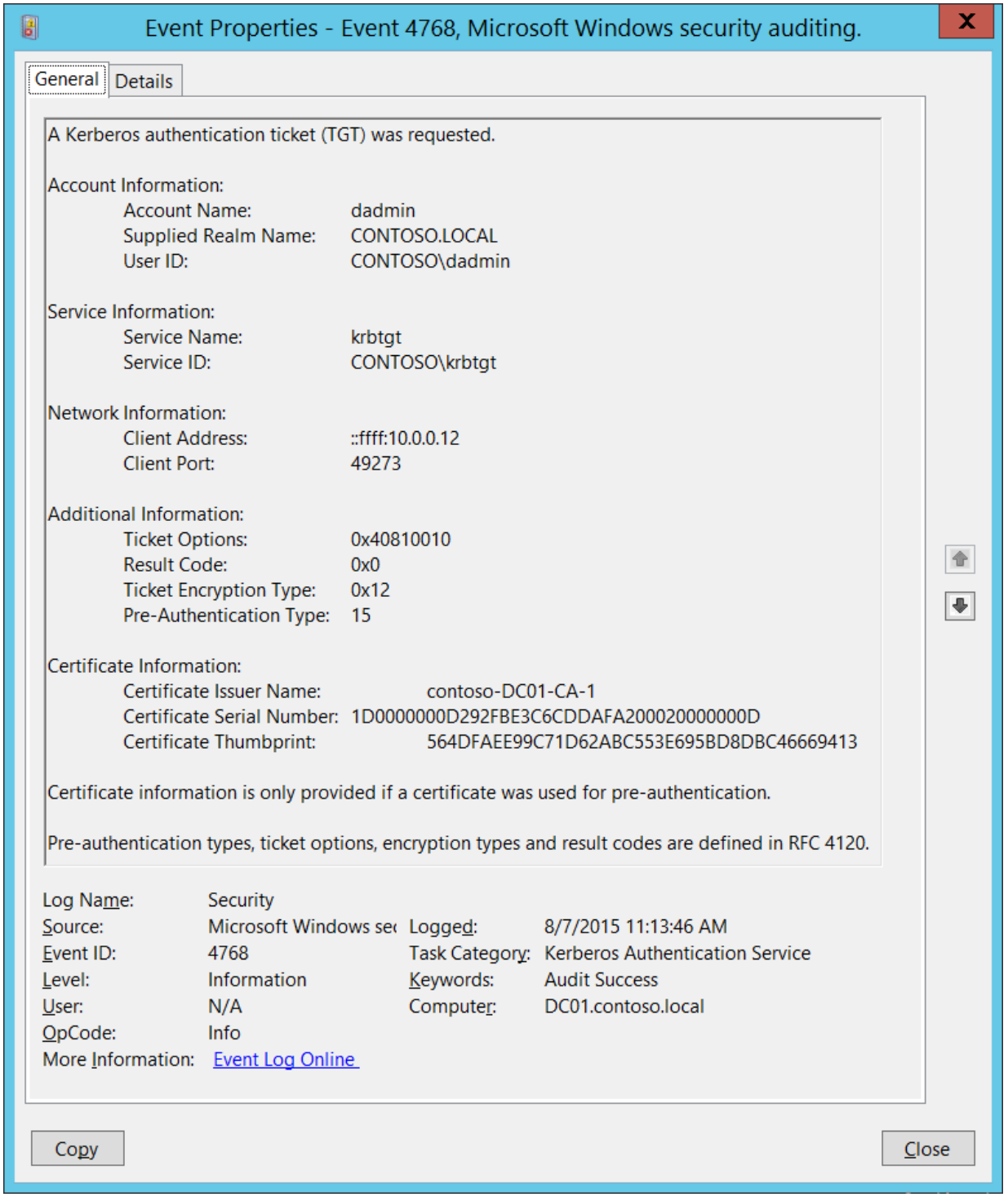
By vinaypamnani-msft

Archived: 2026-04-05 21:17:37 UTC

Update: Windows Server 2016 and later OSs will display an updated version of Event 4768 after getting the January 14th, 2025 or later Security Cumulative Update.



Previous version:



Subcategory: Audit Kerberos Authentication Service

Event Description:

This event generates every time Key Distribution Center issues a Kerberos Ticket Granting Ticket (TGT).

This event generates only on domain controllers.

If TGT issue fails then you will see Failure event with **Result Code** field not equal to "**0x0**".

This event doesn't generate for **Result Codes**: 0x10 and 0x18. Event "4771: Kerberos pre-authentication failed." generates instead.

Note

For recommendations, see Security Monitoring Recommendations for this event.

Event XML:

Updated Event 4768:

```
 <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
<System>
<Provider Name="Microsoft-Windows-Security-Auditing" Guid="{54849625-5478-4994-a5ba-3e3b0328c30d}" />
<EventID>4768</EventID>
<Version>2</Version>
<Level>0</Level>
<Task>14339</Task>
<Opcode>0</Opcode>
<Keywords>0x8020000000000000</Keywords>
<TimeCreated SystemTime="2025-02-26T00:11:39.8853919Z" />
<EventRecordID>2868</EventRecordID>
<Correlation />
<Execution ProcessID="696" ThreadID="2564" />
<Channel>Security</Channel>
<Computer>DC01.contoso.local</Computer>
<Security />
</System>
<EventData>
<Data Name="TargetUserName">duser</Data>
<Data Name="TargetDomainName">CONTOSO.LOCAL</Data>
<Data Name="TargetSid">S-1-5-21-3114123713-1067509961-1646476345-1104</Data>
<Data Name="ServiceName">krbtgt</Data>
<Data Name="ServiceSid">S-1-5-21-3114123713-1067509961-1646476345-502</Data>
<Data Name="TicketOptions">0x40810010</Data>
<Data Name="Status">0x0</Data>
<Data Name="TicketEncryptionType">0x12</Data>
<Data Name="PreAuthType">2</Data>
<Data Name="IpAddress">::ffff:172.27.248.104</Data>
<Data Name="IpPort">54393</Data>
<Data Name="CertIssuerName" />
<Data Name="CertSerialNumber" />
<Data Name="CertThumbprint" />
<Data Name="ResponseTicket">j2P3Uf3sxhIsE6N4+wMt0WDyhXdVBUKMoWzmRRpxqaI=</Data>
<Data Name="AccountSupportedEncryptionTypes">0x27 (DES, RC4, AES-Sk)</Data>
<Data Name="AccountAvailableKeys">AES-SHA1, RC4</Data>
<Data Name="ServiceSupportedEncryptionTypes">0x1F (DES, RC4, AES128-SHA96, AES256-SHA96)</Data>
<Data Name="ServiceAvailableKeys">AES-SHA1, RC4</Data>
<Data Name="DCSupportedEncryptionTypes">0x1F (DES, RC4, AES128-SHA96, AES256-SHA96)</Data>
<Data Name="DCAvailableKeys">AES-SHA1, RC4</Data>
<Data Name="ClientAdvertizedEncryptionTypes">AES256-CTS-HMAC-SHA1-96 RC4-HMAC-NT RC4-HMAC-OLD RC4-MD4 RC4-HMAC-NT-EXP RC
<Data Name="SessionKeyEncryptionType">0x12</Data>
<Data Name="PreAuthEncryptionType">0x12</Data>
</EventData>
</Event>
```

Previous Event:

```
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
- <System>
<Provider Name="Microsoft-Windows-Security-Auditing" Guid="{54849625-5478-4994-A5BA-3E3B0328C30D}" />
<EventID>4768</EventID>
<Version>0</Version>
<Level>0</Level>
<Task>14339</Task>
<Opcode>0</Opcode>
<Keywords>0x8020000000000000</Keywords>
<TimeCreated SystemTime="2015-08-07T18:13:46.074535600Z" />
<EventRecordID>166747</EventRecordID>
<Correlation />
<Execution ProcessID="520" ThreadID="1496" />
<Channel>Security</Channel>
<Computer>DC01.contoso.local</Computer>
<Security />
</System>
- <EventData>
<Data Name="TargetUserName">dadmin</Data>
<Data Name="TargetDomainName">CONTOSO.LOCAL</Data>
<Data Name="TargetSid">S-1-5-21-3457937927-2839227994-823803824-1104</Data>
<Data Name="ServiceName">krbtgt</Data>
<Data Name="ServiceSid">S-1-5-21-3457937927-2839227994-823803824-502</Data>
<Data Name="TicketOptions">0x40810010</Data>
<Data Name="Status">0x0</Data>
<Data Name="TicketEncryptionType">0x12</Data>
<Data Name="PreAuthType">15</Data>
<Data Name="IpAddress">::ffff:10.0.0.12</Data>
<Data Name="IpPort">49273</Data>
<Data Name="CertIssuerName">contoso-DC01-CA-1</Data>
<Data Name="CertSerialNumber">1D0000000D292FBE3C6CDDAFA200020000000D</Data>
<Data Name="CertThumbprint">564DFAEE99C71D62ABC553E695BD8DBC46669413</Data>
</EventData>
</Event>
```

Required Server Roles: Active Directory domain controller.

Minimum OS Version: Windows Server 2008.

Event Versions: 0.

Field Descriptions:

Account Information:

- **Account Name** [Type = UnicodeString]**:** the name of account, for which (TGT) ticket was requested. Computer account name ends with **$** character.
    - User account example: dadmin
    - Computer account example: WIN81$
- **Supplied Realm Name** [Type = UnicodeString]**:** the name of the Kerberos Realm that **Account Name** belongs to. This can appear in a variety of formats, including the following:

- Domain NETBIOS name example: CONTOSO
- Lowercase full domain name: contoso.local
- Uppercase full domain name: CONTOSO.LOCAL

Note

A **Kerberos Realm** is a set of managed nodes that share the same Kerberos database. The Kerberos database resides on the Kerberos master computer system, which should be kept in a physically secure room. Active Directory domain is the example of Kerberos Realm in the Microsoft Windows Active Directory world.

- **User ID** [Type = SID]**:** SID of account for which (TGT) ticket was requested. Event Viewer automatically tries to resolve SIDs and show the account name. If the SID cannot be resolved, you will see the source data in the event.

  For example: CONTOSO\dadmin or CONTOSO\WIN81$.

  - **NULL SID** – this value shows in 4768 Failure events.

  Note

  A **security identifier (SID)** is a unique value of variable length used to identify a trustee (security principal). Each account has a unique SID that is issued by an authority, such as an Active Directory domain controller, and stored in a security database. Each time a user logs on, the system retrieves the SID for that user from the database and places it in the access token for that user. The system uses the SID in the access token to identify the user in all subsequent interactions with Windows security. When a SID has been used as the unique identifier for a user or group, it cannot ever be used again to identify another user or group. For more information about SIDs, see Security identifiers.

- **[New] MSDS-SupportedEncryptionTypes** [Type = UnicodeString]**:** Hexadecimal and string values of the account's MsDs-SupportedEncryptionTypes as stored in Active Directory. This field is sometimes referred to as "msds-SET" for short. These values indicate what encryption type an account is able to use.

  - For computer accounts, this value is managed automatically based on the computer's encryption policy.
  - For Domain Controllers, Read-Only Domain Controllers, and Trusts, AES is supported-by-default unless the account's msds-SET has been manually changed.
  - For user, gMSA, and other accounts, if their msds-SET has not been manually configured then Windows Domain Controllers use the value of the DefaultDomainSupportedEncType policy.
  - The field displays "N/A" if the information could not be queried or was otherwise unavailable.

- **[New] Available Keys:** [Type = UnicodeString]: List of available keys for the account stored in active directory. These keys are generated during password sets and password changes. If the domain has a DFL of 2008 or greater and accounts have had at least one password rotation, AES keys are available for the account.

Service Information:

- **Service Name** [Type = UnicodeString]: the name of the service in the Kerberos Realm to which TGT request was sent. Typically has value "**krbtgt"** for TGT requests, which means Ticket Granting Ticket issuing service.

  - For Failure events **Service Name** typically has the following format: **krbtgt/REALM_NAME**. For example: krbtgt/CONTOSO.

- **Service ID** [Type = SID]**:** SID of the service account in the Kerberos Realm to which TGT request was sent. Event Viewer automatically tries to resolve SIDs and show the account name. If the SID cannot be resolved, you will see the source data in the event.

  Domain controllers have a specific service account (**krbtgt**) that is used by the Key Distribution Center (KDC) service to issue Kerberos tickets. It has a built-in, pre-defined SID: S-1-5-21-DOMAIN_IDENTIFIER-502.

  - **NULL SID** – this value shows in 4768 Failure events.

- **[New] MSDS-SupportedEncryptionTypes** [Type = UnicodeString]**:** Hexadecimal and string values of the service's MsDs-SupportedEncryptionTypes as stored in Active Directory. This field is sometimes referred to as "msds-SET" for short. These values indicate what encryption type an account is able to use.

  - This value is based on the account in Active Directory where the service has been registered to.

- **[New] Available Keys:** [Type = UnicodeString]: List of available keys for the service account that's stored in active directory. These keys are generated during password sets and password changes. If the domain has a DFL of 2008 or greater and accounts have had at least one password rotation, AES keys are available for the service.

[New] Domain Controller Information:

- **[New] MSDS-SupportedEncryptionTypes** [Type = UnicodeString]**:** Hexadecimal and string values of the Domain Controller's MsDs-SupportedEncryptionTypes as stored in Active Directory. This field is sometimes referred to as "msds-SET" for short. This value is determined based on the Kerberos Encryption Policy that's set on the Domain Controller.

- **[New] Available Keys** [Type = UnicodeString]: List of available keys for the Domain Controller. These keys are generated during password sets and password changes, as well as during DCPROMO. If the domain has a DFL of 2008 or greater and accounts have had at least one password rotation, AES keys are available for the Domain Controller.

Network Information:

- **Client Address** [Type = UnicodeString]**:** IP address of the computer from which the TGT request was received. Formats vary, and include the following:

  - **IPv6** or **IPv4** address.

  - **::ffff:IPv4_address**.

  - **::1** - localhost.

- **Client Port** [Type = UnicodeString]: source port number of client network connection (TGT request connection).

  - 0 for local (localhost) requests.

- **[New] Advertized Etypes** [Type = UnicodeString]: A list of encryption types the Kerberos client has advertised support for as part of the Kerberos Protocol. If an encryption type is not recognized, it's numeral value is logged. Common types include:

Decimal Value	Hexidecimal Value	Name
0	0x0	NULL
1	0x1	DES-CBC-CRC

Decimal Value	Hexidecimal Value	Name
2	0x2	DES-CBC-MD5
12	0xC	RC2-CBC-ENV
15	0xF	DES-EDE3-CBC-ENV
17	0x11	AES128-CTS-HMAC-SHA1-96
18	0x12	AES256-CTS-HMAC-SHA1-96
23	0x18	RC4-HMAC-NT
24	0x19	RC4-HMAC-NT-EXP
-128	0xFFFFFF80	RC4-MD4
-133	0xFFFFFF7B	RC4-HMAC-OLD
-135	0xFFFFFECB	RC4-HMAC-OLD-EXP

Note

Windows clients and servers advertise a "set" of encryption types based on the Kerberos Encryption Policy that's deployed on the machines.

Additional information:

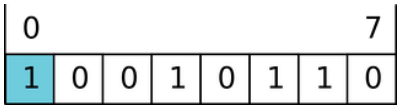
- **Ticket Options** [Type = HexInt32]: this is a set of different ticket flags in hexadecimal format.

  Example:

  - Ticket Options: 0x40810010
  - Binary view: 01000000100000010000000000010000
  - Using **MSB 0** bit numbering we have bit 1, 8, 15 and 27 set = Forwardable, Renewable, Canonicalize, Renewable-ok.

Note

In the table below **“MSB 0”** bit numbering is used, because RFC documents use this style. In “MSB 0” style bit numbering begins from left.



The most common values:

- 0x40810010 - Forwardable, Renewable, Canonicalize, Renewable-ok
- 0x40810000 - Forwardable, Renewable, Canonicalize
- 0x60810010 - Forwardable, Forwarded, Renewable, Canonicalize, Renewable-ok

Bit	Flag Name	Description
0	Reserved	-
1	Forwardable	(TGT only). Tells the ticket-granting service that it can issue a new TGT—based on the presented TGT—with a different network address based on the presented TGT.
2	Forwarded	Indicates either that a TGT has been forwarded or that a ticket was issued from a forwarded TGT.
3	Proxiable	(TGT only). Tells the ticket-granting service that it can issue tickets with a network address that differs from the one in the TGT.
4	Proxy	Indicates that the network address in the ticket is different from the one in the TGT used to obtain the ticket.
5	Allow-postdate	Postdated tickets SHOULD NOT be supported in KILE (Microsoft Kerberos Protocol Extension).
6	Postdated	Postdated tickets SHOULD NOT be supported in KILE (Microsoft Kerberos Protocol Extension).
7	Invalid	This flag indicates that a ticket is invalid, and it must be validated by the KDC before use. Application servers must reject tickets which have this flag set.
8	Renewable	Used in combination with the End Time and Renew Till fields to cause tickets with long life spans to be renewed at the KDC periodically.
9	Initial	Indicates that a ticket was issued using the authentication service (AS) exchange and not issued based on a TGT.
10	Pre-authent	Indicates that the client was authenticated by the KDC before a ticket was issued. This flag usually indicates the presence of an authenticator in the ticket. It can also flag the presence of credentials taken from a smart card logon.
11	Opt-hardware-auth	This flag was originally intended to indicate that hardware-supported authentication was used during pre-authentication. This flag is no longer recommended in the Kerberos V5 protocol. KDCs MUST NOT issue a ticket with this flag set. KDCs SHOULD NOT preserve this flag if it is set by another KDC.
12	Transited-policy-checked	KILE MUST NOT check for transited domains on servers or a KDC. Application servers MUST ignore the TRANSITED-POLICY-CHECKED flag.
13	Ok-as-delegate	The KDC MUST set the OK-AS-DELEGATE flag if the service account is trusted for delegation.
14	Request-anonymous	KILE not use this flag.
15	Name-canonicalize	In order to request referrals the Kerberos client MUST explicitly request the "canonicalize" KDC option for the AS-REQ or TGS-REQ.
16-25	Unused	-

Bit	Flag Name	Description
26	Disable-transited-check	By default the KDC will check the transited field of a TGT against the policy of the local realm before it will issue derivative tickets based on the TGT. If this flag is set in the request, checking of the transited field is disabled. Tickets issued without the performance of this check will be noted by the reset (0) value of the TRANSITED-POLICY-CHECKED flag, indicating to the application server that the transited field must be checked locally. KDCs are encouraged but not required to honor the DISABLE-TRANSITED-CHECK option. Should not be in use, because Transited-policy-checked flag isn't supported by KILE.
27	Renewable-ok	The RENEWABLE-OK option indicates that a renewable ticket will be acceptable if a ticket with the requested life cannot otherwise be provided, in which case a renewable ticket may be issued with a renew-till equal to the requested end time. The value of the renew-till field may still be limited by local limits, or limits selected by the individual principal or server.
28	Enc-tkt-in-skey	No information.
29	Unused	-
30	Renew	The RENEW option indicates that the present request is for a renewal. The ticket provided is encrypted in the secret key for the server on which it is valid. This option will only be honored if the ticket to be renewed has its RENEWABLE flag set and if the time in its renew-till field has not passed. The ticket to be renewed is passed in the padata field as part of the authentication header.
31	Validate	This option is used only by the ticket-granting service. The VALIDATE option indicates that the request is to validate a postdated ticket. Should not be in use, because postdated tickets are not supported by KILE.

Note

KILE **(Microsoft Kerberos Protocol Extension)** – Kerberos protocol extensions used in Microsoft operating systems. These extensions provide additional capability for authorization information including group memberships, interactive logon information, and integrity levels.

- **Result Code** [Type = HexInt32]**:** hexadecimal result code of TGT issue operation. The “Table 3. TGT/TGS issue error codes.” contains the list of the most common error codes for this event.

Code	Code Name	Description	Possible causes
0x0	KDC_ERR_NONE	No error	No errors were found.
0x1	KDC_ERR_NAME_EXP	Client's entry in KDC database has expired	No information.

Code	Code Name	Description	Possible causes
0x2	KDC_ERR_SERVICE_EXP	Server's entry in KDC database has expired	No information.
0x3	KDC_ERR_BAD_PVNO	Requested Kerberos version number not supported	No information.
0x4	KDC_ERR_C_OLD_MAST_KVNO	Client's key encrypted in old master key	No information.
0x5	KDC_ERR_S_OLD_MAST_KVNO	Server's key encrypted in old master key	No information.
0x6	KDC_ERR_C_PRINCIPAL_UNKNOWN	Client not found in Kerberos database	The username doesn’t exist.
0x7	KDC_ERR_S_PRINCIPAL_UNKNOWN	Server not found in Kerberos database	This error can occur if the domain controller cannot find the server’s name in Active Director This error is similar to KDC_ERR_C_PRINCIPAL_UNKNOWN excep that it occurs when the server name cannot be found.
0x8	KDC_ERR_PRINCIPAL_NOT_UNIQUE	Multiple principal entries in KDC database	This error occurs if duplicate principal names exist. Unique principal names are crucial for ensuring mutual authentication. Thus, duplicate principal names are strictly forbidden, even across multiple realms. Without unique principal names, the client has no way of ensuring that the server it is communicating with is the correct one.
0x9	KDC_ERR_NULL_KEY	The client or server has a null key (master key)	No master key was found for client or server. Usually it means that administrator should reset the password on the account.

Code	Code Name	Description	Possible causes
0xA	KDC_ERR_CANNOT_POSTDATE	Ticket (TGT) not eligible for postdating	This error can occur if a client requests postdating of a Kerberos ticket. Postdating is the act of requesting that a ticket's start time be set into the future. It also can occur if there is a time difference between the client and the KDC.
0xB	KDC_ERR_NEVER_VALID	Requested start time is later than end time	There is a time difference between the KDC and the client.
0xC	KDC_ERR_POLICY	Requested start time is later than end time	This error is usually the result of logon restrictions in place on a user's account. For example workstation restriction, smart card authentication requirement or logon time restriction.
0xD	KDC_ERR_BADOPTION	KDC cannot accommodate requested option	Impending expiration of a TGT. The SPN to which the client is attempting to delegate credentials isn't in its Allowed-to-delegate-to list
0xE	KDC_ERR_ETYPE_NOTSUPP	KDC has no support for encryption type	In general, this error occurs when the KDC or a client receives a packet that it cannot decrypt.
0xF	KDC_ERR_SUMTYPE_NOSUPP	KDC has no support for checksum type	The KDC, server, or client receives a packet for which it does not have a key of the appropriate encryption type. The result is that the computer i unable to decrypt the ticket.
0x10	KDC_ERR_PADATA_TYPE_NOSUPP	KDC has no support for PADATA type (pre-authentication data)	Smart card logon is being attempted and the proper certificate cannot be located. This can happen because the wrong certification authority (CA) is being queried or the proper CA cannot b contacted. It can also happen when a domain controller doesn't have a certificate installed for smart card (Domain Controller or Domain Controller Authentication templates). This error code cannot occur in event "4768. A Kerberos authentication ticket (TGT) was requested". It occurs in "4771. Kerberos pre-authentication failed" event.
0x11	KDC_ERR_TRTYPE_NO_SUPP	KDC has no support for transited type	No information.

Code	Code Name	Description	Possible causes
0x12	KDC_ERR_CLIENT_REVOKED	Client's credentials have been revoked	This might be because of an explicit disabling or because of other restrictions in place on the account. For example: account disabled, expired or locked out.
0x13	KDC_ERR_SERVICE_REVOKED	Credentials for server have been revoked	No information.
0x14	KDC_ERR_TGT_REVOKED	TGT has been revoked	Since the remote KDC may change its PKCROSS key while there are PKCROSS ticket still active, it SHOULD cache the old PKCROSS keys until the last issued PKCROSS ticket expires. Otherwise, the remote KDC will respon to a client with a KRB-ERROR message of type KDC_ERR_TGT_REVOKED. See RFC1510 fo more details.
0x15	KDC_ERR_CLIENT_NOTYET	Client not yet valid—try again later	No information.
0x16	KDC_ERR_SERVICE_NOTYET	Server not yet valid—try again later	No information.
0x17	KDC_ERR_KEY_EXPIRED	Password has expired—change password to reset	The user's password has expired.
0x18	KDC_ERR_PREAUTH_FAILED	Pre-authentication information was invalid	The wrong password was provided. This error code cannot occur in event "4768. A Kerberos authentication ticket (TGT) was requested". It occurs in "4771. Kerberos pre-authentication failed" event.
0x19	KDC_ERR_PREAUTH_REQUIRED	Additional pre-authentication required	This error often occurs in UNIX interoperability scenarios. MIT-Kerberos clients do not request pre-authentication when they send a KRB_AS_REQ message. If pre-authentication is required (the default), Windows systems will send this error. Most MIT-Kerberos clients will respond to this error by giving the pre-authentication, in which case the error can be ignored, but some clients might not respond in this way.

Code	Code Name	Description	Possible causes
0x1A	KDC_ERR_SERVER_NOMATCH	KDC does not know about the requested server	No information.
0x1D	KDC_ERR_SVC_UNAVAILABLE	KDC is unavailable	No information.
0x1F	KRB_AP_ERR_BAD_INTEGRITY	Integrity check on decrypted field failed	The authenticator was encrypted with something other than the session key. The result is that the client cannot decrypt the resulting message. The modification of the message could be the result o an attack or it could be because of network noise
0x20	KRB_AP_ERR_TKT_EXPIRED	The ticket has expired	The smaller the value for the “Maximum lifetim for user ticket” Kerberos policy setting, the more likely it is that this error will occur. Because ticket renewal is automatic, you should not have to do anything if you get this message.
0x21	KRB_AP_ERR_TKT_NYV	The ticket is not yet valid	The ticket presented to the server isn't yet valid (in relationship to the server time). The most probable cause is that the clocks on the KDC and the client are not synchronized. If cross-realm Kerberos authentication is being attempted, then you should verify time synchronization between the KDC in the target realm and the KDC in the client realm, as well.
0x22	KRB_AP_ERR_REPEAT	The request is a replay	This error indicates that a specific authenticator showed up twice — the KDC has detected that this session ticket duplicates one that it has already received.
0x23	KRB_AP_ERR_NOT_US	The ticket is not for us	The server has received a ticket that was meant for a different realm.
0x24	KRB_AP_ERR_BADMATCH	The ticket and authenticator do not match	The KRB_TGS_REQ is being sent to the wrong KDC. There is an account mismatch during protocol transition.
0x25	KRB_AP_ERR_SKEW	The clock skew is too great	This error is logged if a client computer sends a timestamp whose value differs from that of the server’s timestamp by more than the number of minutes found in the “Maximum tolerance for computer clock synchronization” setting in Kerberos policy.

Code	Code Name	Description	Possible causes
0x26	KRB_AP_ERR_BADADDR	Network address in network layer header doesn't match address inside ticket	Session tickets MAY include the addresses from which they are valid. This error can occur if the address of the computer sending the ticket is different from the valid address in the ticket. A possible cause of this could be an Internet Protocol (IP) address change. Another possible cause is when a ticket is passed through a proxy server or NAT. The client is unaware of the address scheme used by the proxy server, so unless the program caused the client to request a proxy server ticket with the proxy server's sourc address, the ticket could be invalid.
0x27	KRB_AP_ERR_BADVERSION	Protocol version numbers don't match (PVNO)	When an application receives a KRB_SAFE message, it verifies it. If any error occurs, an error code is reported for use by the application. The message is first checked by verifying that th protocol version and type fields match the curre version and KRB_SAFE, respectively. A mismatch generates a KRB_AP_ERR_BADVERSION. See RFC4120 for more details.
0x28	KRB_AP_ERR_MSG_TYPE	Message type is unsupported	This message is generated when target server finds that message format is wrong. This applies to KRB_AP_REQ, KRB_SAFE, KRB_PRIV an KRB_CRED messages. This error also generated if use of UDP protocol is being attempted with User-to-User authentication.
0x29	KRB_AP_ERR_MODIFIED	Message stream modified and checksum didn't match	The authentication data was encrypted with the wrong key for the intended server. The authentication data was modified in transit by a hardware or software error, or by an attacke The client sent the authentication data to the wrong server because incorrect DNS data caused the client to send the request to the wrong server The client sent the authentication data to the wrong server because DNS data was out-of-date on the client.
0x2A	KRB_AP_ERR_BADORDER	Message out of order (possible tampering)	This event generates for KRB_SAFE and KRB_PRIV messages if an incorrect sequence number is included, or if a sequence number is expected but not present. See RFC4120 for more details.

Code	Code Name	Description	Possible causes
0x2C	KRB_AP_ERR_BADKEYVER	Specified version of key isn't available	This error might be generated on server side during receipt of invalid KRB_AP_REQ message. If the key version indicated by the Ticket in the KRB_AP_REQ isn't one the server can use (e.g., it indicates an old key, and the server no longer possesses a copy of the old key) the KRB_AP_ERR_BADKEYVER error is returned.
0x2D	KRB_AP_ERR_NOKEY	Service key not available	This error might be generated on server side during receipt of invalid KRB_AP_REQ message. Because it is possible for the server to be registered in multiple realms, with different keys in each, the realm field in the unencrypted portion of the ticket in the KRB_AP_REQ is use to specify which secret key the server should use to decrypt that ticket. The KRB_AP_ERR_NOKEY error code is returned the server doesn't have the proper key to decipher the ticket.
0x2E	KRB_AP_ERR_MUT_FAIL	Mutual authentication failed	No information.
0x2F	KRB_AP_ERR_BADDIRECTION	Incorrect message direction	No information.
0x30	KRB_AP_ERR_METHOD	Alternative authentication method required	According to RFC4120 this error message is obsolete.
0x31	KRB_AP_ERR_BADSEQ	Incorrect sequence number in message	No information.
0x32	KRB_AP_ERR_INAPP_CKSUM	Inappropriate type of checksum in message (checksum may be unsupported)	When KDC receives KRB_TGS_REQ message decrypts it, and after that, the user-supplied checksum in the Authenticator MUST be verifie against the contents of the request. The message MUST be rejected either if the checksums do no match (with an error code of KRB_AP_ERR_MODIFIED) or if the checksun isn't collision-proof (with an error code of KRB_AP_ERR_INAPP_CKSUM).

Code	Code Name	Description	Possible causes
0x33	KRB_AP_PATH_NOT_ACCEPTED	Desired path is unreachable	No information.
0x34	KRB_ERR_RESPONSE_TOO_BIG	Too much data	The size of a ticket is too large to be transmitted reliably via UDP. In a Windows environment, th message is purely informational. A computer running a Windows operating system will automatically try TCP if UDP fails.
0x3C	KRB_ERR_GENERIC	Generic error	Group membership has overloaded the PAC. Multiple recent password changes have not propagated. Crypto subsystem error caused by running out o memory. SPN too long. SPN has too many parts.
0x3D	KRB_ERR_FIELD_TOOLONG	Field is too long for this implementation	Each request (KRB_KDC_REQ) and response (KRB_KDC_REP or KRB_ERROR) sent over the TCP stream is preceded by the length of the request as 4 octets in network byte order. The high bit of the length is reserved for future expansion and MUST currently be set to zero. If a KDC that does not understand how to interpret a set high bit of the length encoding receives a request with the high order bit of the length set, i MUST return a KRB-ERROR message with the error KRB_ERR_FIELD_TOOLONG and MUST close the TCP stream.
0x3E	KDC_ERR_CLIENT_NOT_TRUSTED	The client trust failed or isn't implemented	This typically happens when user’s smart-card certificate is revoked or the root Certification Authority that issued the smart card certificate (i a chain) isn't trusted by the domain controller.
0x3F	KDC_ERR_KDC_NOT_TRUSTED	The KDC server trust failed or could not be verified	The trustedCertifiers field contains a list of certification authorities trusted by the client, in the case that the client does not possess the KDC's public key certificate. If the KDC has no certificate signed by any of the trustedCertifiers, then it returns an error of type KDC_ERR_KDC_NOT_TRUSTED. See RFC1510 for more details.
0x40	KDC_ERR_INVALID_SIG	The signature is invalid	This error is related to PKINIT. If a PKI trust relationship exists, the KDC then verifies the client's signature on AuthPack (TGT request signature). If that fails, the KDC returns an error message of type KDC_ERR_INVALID_SIG.

Code	Code Name	Description	Possible causes
0x41	KDC_ERR_KEY_TOO_WEAK	A higher encryption level is needed	If the clientPublicValue field is filled in, indicating that the client wishes to use Diffie-Hellman key agreement, then the KDC checks to see that the parameters satisfy its policy. If they do not (e.g., the prime size is insufficient for the expected encryption type), then the KDC sends back an error message of type KDC_ERR_KEY_TOO_WEAK.
0x42	KRB_AP_ERR_USER_TO_USER_REQUIRED	User-to-user authorization is required	In the case that the client application doesn't know that a service requires user-to-user authentication, and requests and receives a conventional KRB_AP_REP, the client will send the KRB_AP_REP request, and the server will respond with a KRB_ERROR token as described in RFC1964, with a msg-type of KRB_AP_ERR_USER_TO_USER_REQUIRED
0x43	KRB_AP_ERR_NO_TGT	No TGT was presented or available	In user-to-user authentication if the service does not possess a ticket granting ticket, it should return the error KRB_AP_ERR_NO_TGT.
0x44	KDC_ERR_WRONG_REALM	Incorrect domain or principal	Although this error rarely occurs, it occurs when a client presents a cross-realm TGT to a realm other than the one specified in the TGT. Typically, this results from incorrectly configured DNS.

- **Ticket Encryption Type** [Type = HexInt32]: the cryptographic suite that was used for issued TGT. These values are the same as the "Advertised Etypes" field, with some minor name changes.
- **[New] Session Encryption Type** [Type = HexInt32]: the cryptographic suite that was used for session key in the issued TGT. These values are the same as the "Advertised Etypes" field, with some minor name changes.

Table 4. Kerberos encryption types

Type	Type Name	Description
0x1	DES-CBC-CRC	Disabled by default starting from Windows 7 and Windows Server 2008 R2.
0x3	DES-CBC-MD5	Disabled by default starting from Windows 7 and Windows Server 2008 R2.
0x11	AES128-CTS-HMAC-SHA1-96	Supported starting from Windows Server 2008 and Windows Vista.
0x12	AES256-CTS-HMAC-SHA1-96	Supported starting from Windows Server 2008 and Windows Vista.

Type	Type Name	Description
0x17	RC4-HMAC	Default suite for operating systems before Windows Server 2008 and Windows Vista.
0x18	RC4-HMAC-EXP	Default suite for operating systems before Windows Server 2008 and Windows Vista.
0xFFFFFFFF or 0xffffffff	-	This type shows in Audit Failure events.

- **Pre-Authentication Type** [Type = UnicodeString]: the code number of pre-Authentication type which was used in TGT request.
- **[New] Pre-Authentication EncryptionType** [Type = HexInt32]: The Encryption Type that was selected for the Pre-Authentication flow. The values are the same as the "Ticket Encryption Type" field.

Table 5. Kerberos Pre-Authentication types

Type	Type Name	Description
0	-	Logon without Pre-Authentication.
2	PA-ENC-TIMESTAMP	This is a normal type for standard password authentication.
11	PA-ETYPE-INFO	The ETYPE-INFO pre-authentication type is sent by the KDC in a KRB-ERROR indicating a requirement for additional pre-authentication. It is usually used to notify a client of which key to use for the encryption of an encrypted timestamp for the purposes of sending a PA-ENC-TIMESTAMP pre-authentication value. Never saw this Pre-Authentication Type in Microsoft Active Directory environment.
15	PA-PK-AS-REP_OLD	Used for Smart Card logon authentication.
16	PA-PK-AS-REQ	Request sent to KDC in Smart Card authentication scenarios.
17	PA-PK-AS-REP	This type should also be used for Smart Card authentication, but in certain Active Directory environments, it is never seen.
19	PA-ETYPE-INFO2	The ETYPE-INFO2 pre-authentication type is sent by the KDC in a KRB-ERROR indicating a requirement for additional pre-authentication. It is usually used to notify a client of which key to use for the encryption of an encrypted timestamp for the purposes of sending a PA-ENC-TIMESTAMP pre-authentication value. Never saw this Pre-Authentication Type in Microsoft Active Directory environment.
20	PA-SVR-REFERRAL-INFO	Used in KDC Referrals tickets.
138	PA-ENCRYPTED-CHALLENGE	Logon using Kerberos Armoring (FAST). Supported starting from Windows Server 2012 domain controllers and Windows 8 clients.
-		This type shows in Audit Failure events.

Certificate Information:

- **Certificate Issuer Name** [Type = UnicodeString]**:** the name of the Certification Authority that issued the smart card certificate. Populated in **Issued by** field in certificate.
- **Certificate Serial Number** [Type = UnicodeString]**:** smart card certificate's serial number. Can be found in **Serial number** field in the certificate.
- **Certificate Thumbprint** [Type = UnicodeString]**:** smart card certificate's thumbprint. Can be found in **Thumbprint** field in the certificate.

[New] Ticket Information

- **[New] Response ticket hash** [Type = UnicodeString]: The hash of the ticket that the Windows Domain Controller replied back to the client as part of the Response.

Security Monitoring Recommendations

For 4768(S, F): A Kerberos authentication ticket (TGT) was requested.

Type of monitoring required	Recommendation
High-value accounts: You might have high-value domain or local accounts for which you need to monitor each action. Examples of high-value accounts are database administrators, built-in local administrator account, domain administrators, service accounts, domain controller accounts and so on.	Monitor this event with the **"User ID"** that corresponds to the high-value account or accounts.
Anomalies or malicious actions: You might have specific requirements for detecting anomalies or monitoring potential malicious actions. For example, you might need to monitor for use of an account outside of working hours.	When you monitor for anomalies or malicious actions, use the **"User ID"** (with other information) to monitor how or when a particular account is being used.
Non-active accounts: You might have non-active, disabled, or guest accounts, or other accounts that should never be used.	Monitor this event with the **"User ID"** that corresponds to the accounts that should never be used.
Account allowlist: You might have a specific allowlist of accounts that are the only ones allowed to perform actions corresponding to particular events.	If this event corresponds to an "allowlist-only" action, review the **"User ID"** for accounts that are outside the allowlist.
External accounts: You might be monitoring accounts from another domain, or "external" accounts that are not allowed to perform certain actions (represented by certain specific events).	Monitor this event for the **"Supplied Realm Name"** corresponding to another domain or "external" location.
Account naming conventions: Your organization might have specific naming conventions for account names.	Monitor "**User ID"** for names that don't comply with naming conventions.

- You can track all 4768 events where the **Client Address** isn't from your internal IP address range or not from private IP address ranges.
- If you know that **Account Name** should be used only from known list of IP addresses, track all **Client Address** values for this **Account Name** in 4768 events. If **Client Address** isn't from the allowlist, generate the alert.

- All **Client Address** = `::1` means local authentication. If you know the list of accounts which should log on to the domain controllers, then you need to monitor for all possible violations, where **Client Address** = `::1` and **Account Name** isn't allowed to log on to any domain controller.
- All 4768 events with **Client Port** field value > 0 and < 1024 should be examined, because a well-known port was used for outbound connection.
- Also consider monitoring the fields shown in the following table, to discover the issues listed:

Field	Issue to discover
Certificate Issuer Name	Certification authority name is not from your PKI.
Certificate Issuer Name	Certification authority name is not authorized to issue smart card authentication certificates.
Pre-Authentication Type	Value is **0**, which means that pre-authentication was not used. All accounts should use Pre-Authentication, except accounts configured with "Do not require Kerberos preauthentication," which is a security risk. For more information, see Table 5. Kerberos Pre-Authentication types.
Pre-Authentication Type	Value is **not 15** when account must use a smart card for authentication. For more information, see Table 5. Kerberos Pre-Authentication types.
Pre-Authentication Type	Value is **not 2** when only standard password authentication is in use in the organization. For more information, see Table 5. Kerberos Pre-Authentication types.
Pre-Authentication Type	Value is **not 138** when Kerberos Armoring is enabled for all Kerberos communications in the organization. For more information, see Table 5. Kerberos Pre-Authentication types.
Ticket Encryption Type	Value is **0x1** or **0x3**, which means the DES algorithm was used. DES should not be in use, because of low security and known vulnerabilities. It is disabled by default starting from Windows 7 and Windows Server 2008 R2. For more information, see Table 4. Kerberos encryption types.
Ticket Encryption Type	Starting with Windows Vista and Windows Server 2008, monitor for values **other than 0x11 and 0x12**. These are the expected values, starting with these operating systems, and represent AES-family algorithms. For more information, see Table 4. Kerberos encryption types.
Result Code	**0x6** (The username doesn't exist), if you see, for example N events in last N minutes. This can be an indicator of account enumeration attack, especially for highly critical accounts.
Result Code	**0x7** (Server not found in Kerberos database). This error can occur if the domain controller cannot find the server's name in Active Directory.
Result Code	**0x8** (Multiple principal entries in KDC database). This will help you to find duplicate SPNs faster.
Result Code	**0x9** (The client or server has a null key (master key)). This error can help you to identify problems with Kerberos authentication faster.
Result Code	**0xA** (Ticket (TGT) not eligible for postdating). Microsoft systems should not request postdated tickets. These events could help identify anomaly activity.

Field	Issue to discover
Result Code	**0xC** (Requested start time is later than end time), if you see, for example N events in last N minutes. This can be an indicator of an account compromise attempt, especially for highly critical accounts.
Result Code	**0xE** (KDC has no support for encryption type). In general, this error occurs when the KDC or a client receives a packet that it cannot decrypt. Monitor for these events because this should not happen in a standard Active Directory environment.
Result Code	**0xF** (KDC has no support for checksum type). Monitor for these events because this should not happen in a standard Active Directory environment.
Result Code	**0x12** (Client's credentials have been revoked), if you see, for example N events in last N minutes. This can be an indicator of anomaly activity or brute-force attack, especially for highly critical accounts.
Result Code	**0x1F** (Integrity check on decrypted field failed). The authenticator was encrypted with something other than the session key. The result is that the KDC cannot decrypt the TGT. The modification of the message could be the result of an attack or it could be because of network noise.
Result Code	**0x22** (The request is a replay). This error indicates that a specific authenticator showed up twice—the KDC has detected that this session ticket duplicates one that it has already received. It could be a sign of attack attempt.
Result Code	**0x29** (Message stream modified and checksum didn't match). The authentication data was encrypted with the wrong key for the intended server. The authentication data was modified in transit by a hardware or software error, or by an attacker. Monitor for these events because this should not happen in a standard Active Directory environment.
Result Code	**0x3C** (Generic error). This error can help you more quickly identify problems with Kerberos authentication.
Result Code	**0x3E** (The client trust failed or is not implemented). This error helps you identify logon attempts with revoked certificates and the situations when the root Certification Authority that issued the smart card certificate (through a chain) is not trusted by a domain controller.
Result Code	**0x3F**, **0x40**, **0x41** errors. These errors can help you more quickly identify smart-card related problems with Kerberos authentication.
