Vulnerabilità in Kerberos v5

(Secondo Mitr3 Att4ck nel 2023)

Author

LORENZO FALLANI

Date

12/09/2023

Cosa è Kerberos e cosa garantisce?

KERBEROS SECONDO RFC 1510

Kerberos fornisce un modo per verificare le identità dei principali (Autenticazione) su una rete aperta senza fare affidamento sul sistema operativo host, basare la fiducia sugli indirizzi degli host o richiedere sicurezza fisica. Utilizza una chiave segreta condivisa per effettuare l'autenticazione come servizio di terze parti affidabile.

Le parti coinvolte sono: il client che desidera autenticarsi, il server che deve autenticare il client e il KDC (Key Distribution Center) che garantisce l' autenticazione distribuita sulla rete aperta come sistema centralizzato e affidabile.

Il KDC è diviso in AS (Authentication Server) e TGS (Ticket-Granting Server).

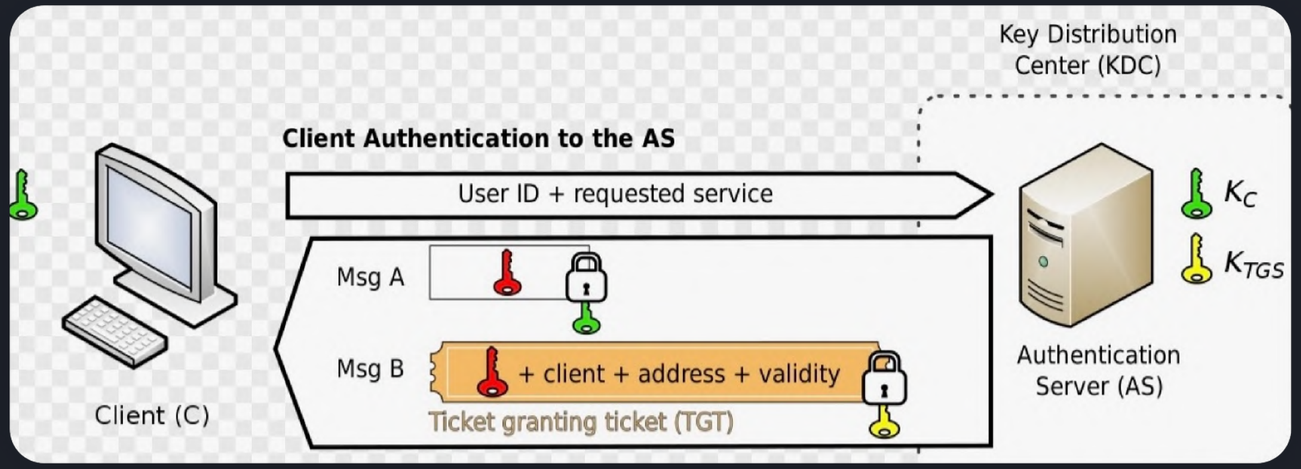
SCAMBIO DI MESSAGGI

1. KRB_AS_REQ e KRB_AS_REP (client - AS e AS - client)
2. KRB_TGS_REQ e KRB_TGS_REP (client - TGS e TGS - client)
3. KRB_AP_REQ e KRB_AP_REP (client - server e server - client)

KRB_AS_REQ

KRB_AS_REQ

E' una richiesta fatta dal client al AS per ottenere il TGT (Ticket-Granting Ticket) che verrà utilizzato dal client per ottenere i Ticket di Servizio che sono utilizzati per autenticarsi ai Server applicativi.



La richiesta contiene le informazioni utili al AS per identificare il client da autenticare (User ID, cname, crealm, etc...) e per identificare il server che hosta il servizio richiesto (sname, srealm, server IP Address, etc...).

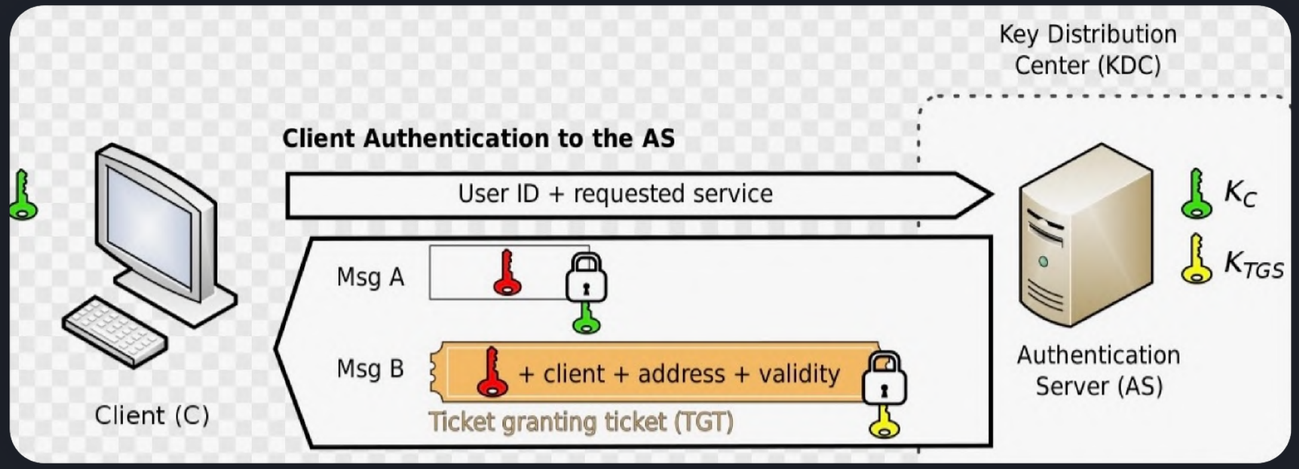
Attenzione! Il testo inviato è in ClearText!

Nota: Nel KRB_AS_REQ è possibile specificare all' interno della sezione PADATA (Pre-Auth Data) un Authenticator che consiste (in questo caso) di un timestamp criptato con la chiave segreta del Client (condivisa con il KDC) che consente di evitare attacchi replay.

KRB_AS_REP

KRB_AS_REP

E' la risposta dall' AS e contiene il TGT e una Session Key temporanea generata dall' AS randomicamente, essa verrà utilizzata per criptare la conversazione tra Client e TGS.



Come l' AS genera il TGT e la Session Key?

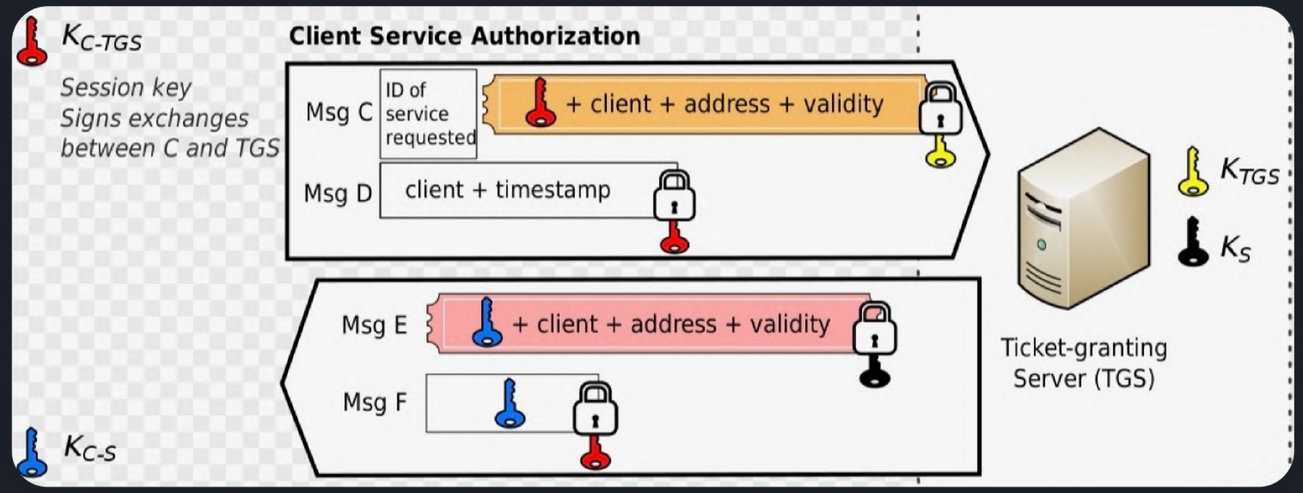
1. Estrae le chiavi segrete Kc e Ktgs dal DB del KDC (Kc è la chiave segreta del Client e Ktgs è la chiave segreta del TGS).
2. Genera pseudorandomicamente la Session Key.
3. La Session Key viene appesa alle info che identificano il client insieme ai timestamp di inizio transazione ed il tutto viene criptato con la Ktgs. Il risultato è il TGT. (Ad esempio con RC4-HMAC o DESCBC-MD5)
4. Una copia della Session Key viene criptata dalla Kc.

Infine il TGT e la Session Key criptata vengono inviate al Client.

KRB_TGS_REQ

KRB_TGS_REQ

E' una richiesta fatta dal client al TGS per ottenere il Ticket di Servizio relativo ad un servizio su un server applicativo. Essa viene fatta inviando il TGT ottenuto dalla AS_REQ e da un Authenticator.



Come il client genera la KRB_TGS_REQ?

1. Ricava la Session Key decriptandola dal KRB_AS_REP con la sua chiave segreta Kc.
2. Genera l' Authenticator concatenando il timestamp con le info relative al client e criptando il tutto con la Session Key (usata da ora in poi per criptare i messaggi dal Client al TGS).

Attenzione! Il TGT non può essere decriptato dal client che non possiede la chiave Ktgs.

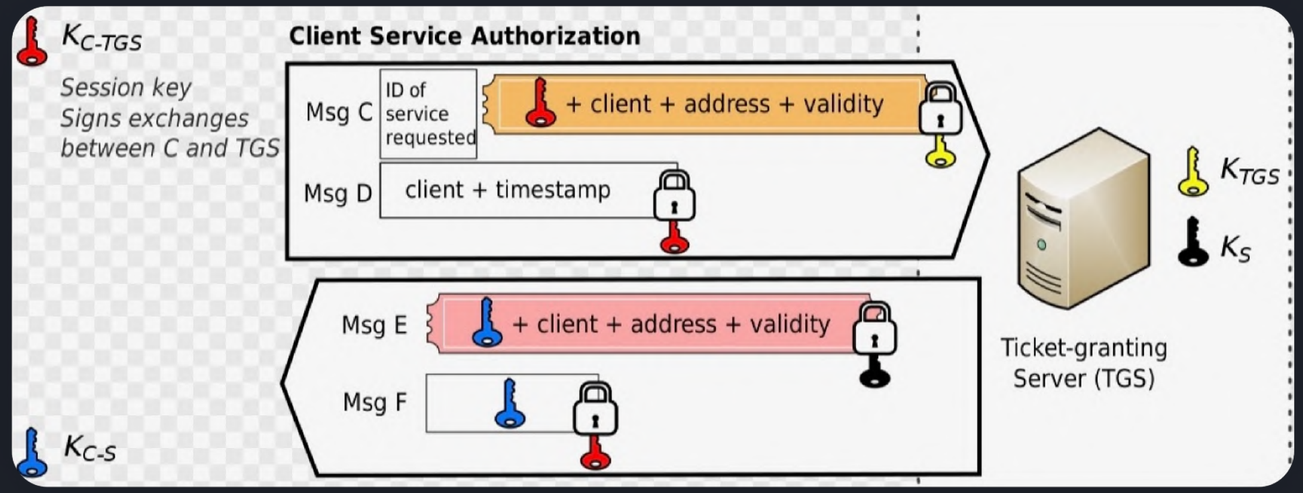
Nota: Anche in questo caso l' Authenticator è utilizzato per evitare attacchi replay grazie al timestamp criptato.

Infine viene inviato il TGT ottenuto dalla KRB_AS_REP concatenato alle info per relative al server applicativo e all' Authenticator.

KRB_TGS_REP

K R B _ T G S _ R E P

E' la risposta del TGS e contiene il Ticket di Servizio e una Session key temporanea generata dal TGS randomicamente, essa verrà utilizzata per criptare la conversazione tra client e server applicativo.



Come il TGS genera il TS e la Session Key?

1. Estrae la chiave Ks dal DB del KDC.
2. Genera pseudorandomicamente la Session Key.
3. La nuova Session Key viene appesa alle info che identificano il client insieme ai timestamp di inizio transazione ed il tutto viene criptato con la Ks. Il risultato è il Ticket di Servizio.
4. Una copia della nuova Session Key viene criptata dalla Session Key precedente

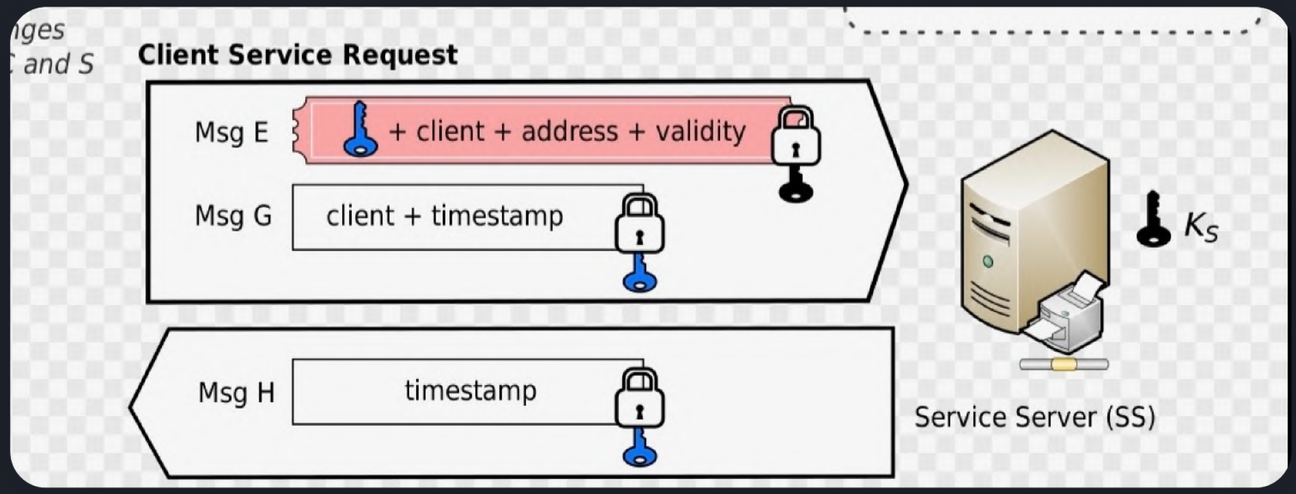
Nota: Il TGS autentica il client dal momento che può decriptare l'Authenticator con la chiave di Sessione presente nel TGT inviato dalla KRB_TGS_REQ (il quale può essere letto solo dal TGS).

Infine il Ticket di Servizio e la nuova Session Key criptata vengono inviati al client.

KRB_AP_REQ e KRB_AP_REP

KRB_AP_REQ

E' una richiesta fatta dal client al server applicativo utilizzando il Ticket di Servizio preso dalla KRB_TGS_REP e dall' Authenticator, ovvero il timestamp concatenato alle info relative al client criptate con la nuova Session Key (ottenuta estraendola dal KRB_TGS_REP).



KRB_AP_REP

E' la risposta fatta dal server applicativo una volta che è riuscito ad autenticare il client e serve ad autenticare se stesso (questo messaggio infatti è opzionale) inviando il timestamp presente nell' Authenticator.

Come il server autentica il client?

1. Decripta il Ticket di Servizio con la chiave Ks (condivisa tra KDC e server)e si salva la nuova Session Key.
2. La Session Key viene usata per decriptare l' Authenticator e legge il timestamp => client autenticato!

Quali algoritmi di cifratura vengono utilizzati?

Una volta capito il protocollo Kerberos è utile conoscere le implementazioni di tale protocollo nel mondo reale per conoscere le eventuali vulnerabilità sfruttabili.
Kerberos è utilizzato ampiamente in Active Directory di Microsoft ma la sua implementazione lascia un pò a desiderare. Leggo testuali parole presenti nel documento ufficiale di Microsoft.
"By default, user account do not have a value set so unless you have manually enabled AES on them, tickets for service accounts will be encrypted with RC4"

Questo significa che di default i Ticket di Servizio degli account di servizio (che sono gli account che hanno i permessi più alti all'interno di una rete e controllano i servizi applicativi) sono generati utilizzando RC4.

Type value	Encryption type used
0x3	DES-CBC-MD5
0x12	AES256-CTS-SHA196
0x17	RC4-HMAC

In questa tabella è possibile vedere quali sono alcuni degli etype possibili per criptare i ticket con relativo type value.
Nel testo precedente è possibile capire che di default gli account di servizio implementano il type value 0x17 su AD quando sarebbe preferibile implementare il type value 0x12.

Cosa ci dice il MITRE Att&ck?

STEAL OR FORGE KERBEROS TICKETS

Se andiamo a controllare nel famoso framework MITRE Att&ck notiamo che gli attacchi relativi ai Ticket Kerberos sono 4:

1. Golden Ticket e Silver Ticket
2. Kerberoasting e AS-REP Roasting

ID	Name
T1558.001	Golden Ticket
T1558.002	Silver Ticket
T1558.003	Kerberoasting
T1558.004	AS-REP Roasting

Kerberoasting

IN COSA CONSISTE E COME MAI FUNZIONA?

Il Kerberoasting è una pratica comune nell' attacco a Kerberos in Active Directory e consente ad un avversario che ha ottenuto un account di dominio valido nella rete di ottenere la password in chiaro di un account di servizio legato ad un SPN (Service Principal Name), ovvero l' id del server applicativo, nel caso in cui i Ticket di Servizio siano stati generati con una cifratura debole come RC4 o DES.

COME AVVIENE?

L' Attaccante possedendo un account di dominio valido può richiedere il TGT all' AS. Con il TGT può richiedere il Ticket di Servizio utile ad autenticarsi al server applicativo. Ma come viene generato il Ticket di Servizio?
Il Ticket di Servizio viene generato (di default in AD per gli account di servizio) con la cifratura RC4 utilizzando come chiave la password NTLM dell' account di servizio legato al server applicativo.
Questo significa che un avversario può effettuare un attacco Brute Force sul Ticket di Servizio sapendo che è cifrato usando RC4. Questo permette di provare tante combinazioni in poco tempo e trovare la password dell' account di Servizio facilmente. (con AES ci vorrebbe molto più tempo)

AS-REP Roasting

COSA CAMBIA DAL KERBEROASTING?

AS-REP Roasting sfrutta, come il Kerberoasting, l' utilizzo di cifrature deboli per la generazione dei Ticket, ma questa volta l'avversario interviene in una fase diversa.
Come ho spiegato nella Nota del KRB_AS_REQ è possibile inviare un Authenticator nella richiesta ma non è obbligatorio (dovrebbe esserlo in quanto protegge dall' offline cracking).
L'Attaccante infatti nel caso conoscesse i client che non hanno fatto una KRB_AS_REQ con Authenticator potrebbe generare lui stesso una KRB_AS_REQ con i dati di quei client vittima senza Authenticator e inviarla all' AS. Il quale sapendo che l' Authenticator è opzionale risponderebbe con il TGT del client vittima.

Ma come è generato il TGT del client vittima?
Il TGT del client vittima viene generato (nel caso in cui si utilizzi RC4 come cifratura) con la cifratura RC4 utilizzando come chiave la password NTLM dell' account del client vittima.
Facendo quindi lo stesso discorso del Kerberoasting è possibile crackare il TGT e ottenere la password dell' account del client vittima.

Golden Ticket

A COSA SERVE?

Questo attacco è un attacco di lateral movement per accedere ad altre risorse dove l'avversario è già riuscito ad ottenere accesso al KRBTGT account o conosce la sua password NTLM in Active Directory.

COSA È IL KRBTGT ACCOUNT?

E' l'account di servizio legato al server TGS ed è in grado di generare i TGT per tutti client che vogliono autenticarsi in rete.

COME GENERARE IL GOLDEN TICKET

Come sappiamo dalla slide del KRB_AS_REP il TGT è generato con la chiave segreta Ktgs ovvero la chiave del TGS che non è altro che la password del KRBTGT account o una sua trasformazione. Quindi se l'avversario ha il controllo o comunque conosce la password NTLM del KRBTGT account allora può generare qualsiasi TGT valido.

Il Golden Ticket non è altro che un TGT generato dall' avversario con all' interno le informazioni di qualsiasi account AD e criptato con la password NTLM del KRBTGT account.

Viene chiamato Golden perché adesso l'avversario può accedere a qualsiasi risorsa di AD autenticandosi come qualsiasi account di AD senza conoscere le password dei relativi account di Servizio o account AD.

Silver Ticket

COSA È E COSA CAMBIA DAL GOLDEN TICKET?

Questo attacco è il fratello minore del precedente in quanto è comunque un attacco che permette lateral movement ma permette l'accesso soltanto ad un sottoinsieme degli account di Active Directory. Questo si ha perché l' attaccante ha accesso (solo) alla password NTLM dell' account di Servizio di un server applicativo.

COME FARE A GENERARLO?

Come sappiamo dalla slide del KRB_TGS_REP il Ticket di Servizio è generato con la chiave segreta Ks ovvero la password NTLM dell' account di Servizio del server applicativo. Quindi se l'avversario ha la password NTLM dell' account di Servizio di quel server applicativo potrà generare dei Ticket di Servizio validi per autenticarsi in quel servizio.

Il Silver Ticket non è altro che un Ticket di Servizio generato dall' avversario con all' interno le informazioni di qualsiasi account AD e criptato con la password NTLM dell' account di Servizio del server applicativo.

Viene chiamato Silver perché adesso l'avversario può autenticarsi al server applicativo come qualsiasi account AD senza conoscere la password dei relativi account AD.

Nota: Questo attacco è più difficile da individuare in quanto l' avversario non interagisce con il KDC.

Kerberoasting nella vita di tutti i giorni

COSA USANO GLI HACKER PER EFFETTUARE QUESTO ATTACCO?

Abbiamo detto che sostanzialmente l' attacco è diviso in due parti:

1. Richiesta dei Ticket di Servizio tramite un script open source python usando un account di dominio valido

```
:~# GetUserSPNs.py example.local/user:password -request > TGS.hash
```

2. Cracking del TGS criptato con etype 0x17 (RC4-HMAC) ottenuto da GetUserSPNs.py utilizzando hashcat dove l'opzione -m 13100 specifica l'etype e il file rockyou.txt contiene la wordlist per il Dictionary Attack.

```
:~# hashcat -m 13100 TGS.hash /usr/share/wordlist/rockyou.txt
```

Golden Ticket con Mimikatz

COSA USANO GLI HACKER PER EFFETTUARE QUESTO ATTACCO?

Una volta che l' avversario ha accesso all' account KRBTGT può installare nel sistema un tool open source che si chiama Mimikatz e permette il dump della password NTLM dell' account KRBTGT tramite il comando:

```
mimikatz # lsadump::dcsync /domain:example.local /user:krbtgt
```

A questo punto nell' output è presente la password NTLM (hash) dell' account KRBTGT che viene usata qui:

```
mimikatz # kerberos::golden /domain:example.local /sid:XX /rc4:YY /id:500 /user:Admin
```

XX è il vero SID (Security ID) ottenibile dalla Powershell, YY è l' hash della password NTLM specificato nell' opzione rc4 (possibile cambiare opzione di cifratura), 500 è l' id dell' account Administrator.
A questo punto mimikatz avrà generato il golden ticket (ticket.kirbi) e potrà essere caricato in memoria con:

```
mimikatz # kerberos::ptt ticket.kirbi
```

Fonti utilizzate

- documento ufficiale di Microsoft: https://techcommunity.microsoft.com/t5/core-infrastructure-and-security/decrypting-the-selection-of-supported-kerberos-encryption-types/ba-p/1628797
- pagina di MITRE Att4ck: https://attack.mitre.org/techniques/T1558/001/
- RFC 1510 (Kerberos v5): https://www.rfc-editor.org/rfc/rfc1510
- Golden Ticket Video: https://www.youtube.com/watch?v=v0xKYSkyI6Q
- Kerberoasting Video: https://www.youtube.com/watch?v=BMBNteDRKHA&t=283s
- Mimikats tool: https://github.com/gentilkiwi/mimikatz
- GetUserSPNs.py tool: https://github.com/fortra/impacket/blob/master/examples/GetUserSPNs.py
- Hashcat tool: https://github.com/hashcat/hashcat

Fine

GRAZIE PER L'ATTENZIONE

Pitch

Want to make a presentation like this one?

Start with a fully customizable template, create a beautiful deck in minutes, then easily share it with anyone.

Create a presentation (It's free)