



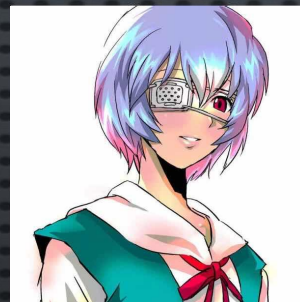
2018

360 企业安全集团

Attacking Kerberos

演讲人：吴方东, n1nty @ 360 A-TEAM

关于



吴方东, *n1nty* @ **360 A-TEAM**

资深失眠患者

安全研究员

360 A-TEAM

- 360 企业安全集团旗下的纯技术研究团队。
- 致力于 *Web* 渗透, *APT* 攻防对抗, 前瞻性攻防工具预研。
- 进行严肃、有深度的技术研究, 深入还原攻与防的技术本质。



n1nty @ 360 A-TEAM

什么是身份认证

服务端验证“*你是否真的是你所声称的那个人*”的过程

传统账号密码认证

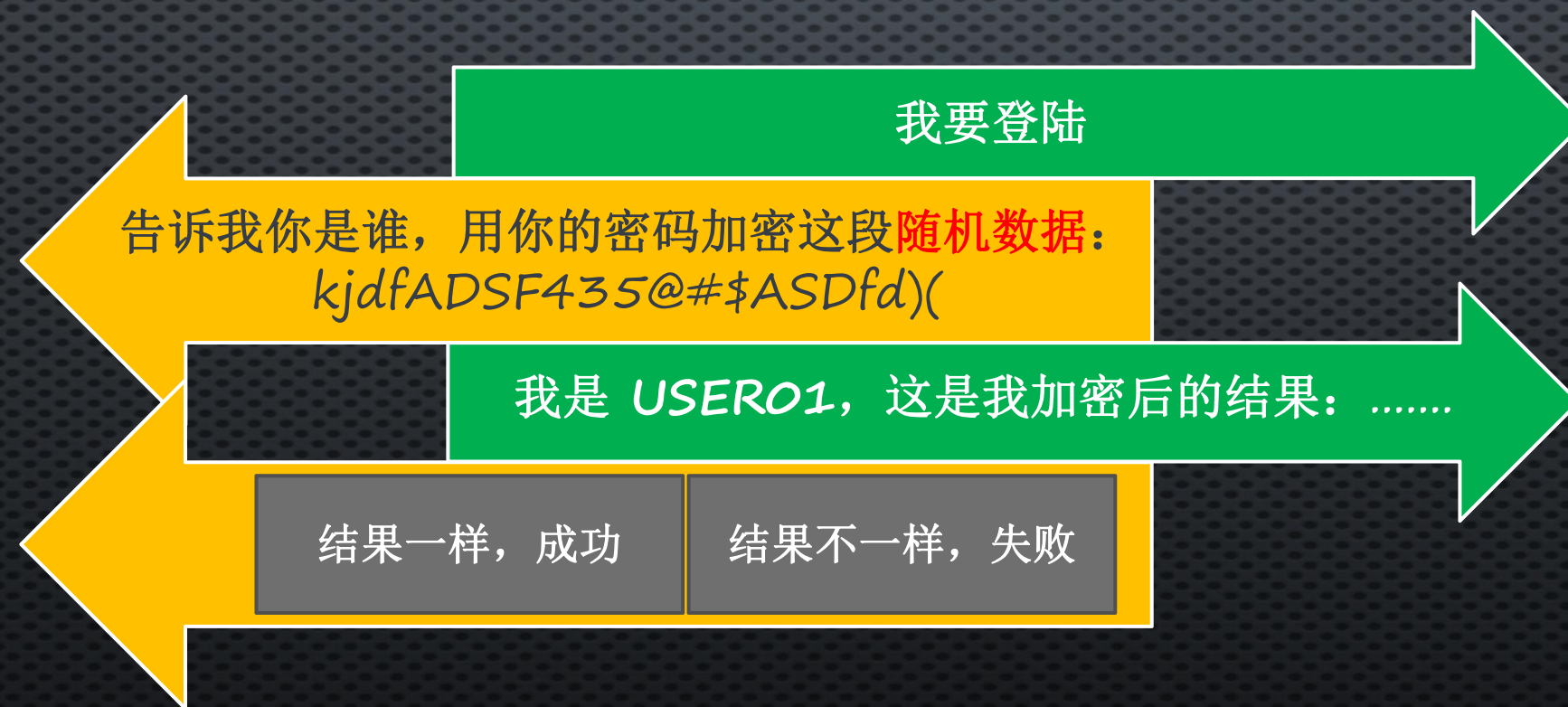


非加密通道会导致明文密码泄露

挑战认证



USER01



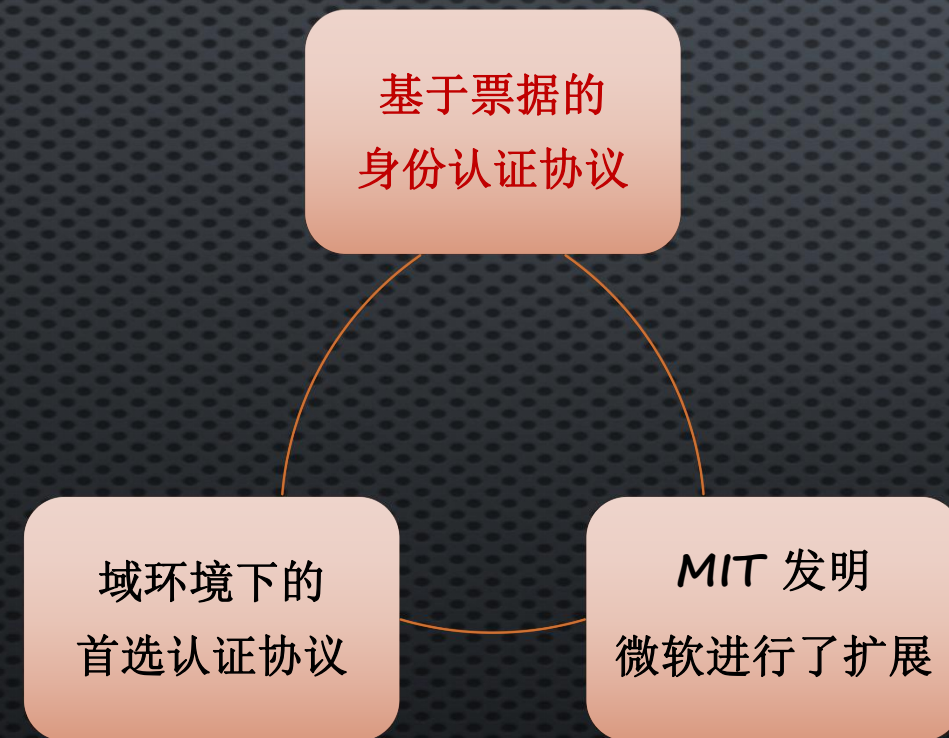
- 不需要传输密码本身
- 可在非加密通道使用

挑战认证 - NTLM

- WINDOWS 工作组环境下的默认身份验证协议
- [HTTPS://DOCS.MICROSOFT.COM/ZH-CN/WINDOWS/DESKTOP/SECAUTHN/MICROSOFT-NTLM](https://docs.microsoft.com/zh-cn/windows/desktop/SecAuthN/microsoft-ntlm)
- [HTTP://DAVENPORT.SOURCEFORGE.NET/NTLM.HTML](http://davenport.sourceforge.net/ntlm.html)



INTRO TO KERBEROS



什么是票据

- 账号密码认证

- 通过向服务端发送“账号密码”来证明自己的身份

- 挑战认证

- 通过向服务端发送“一段计算结果”来证明自己的身份

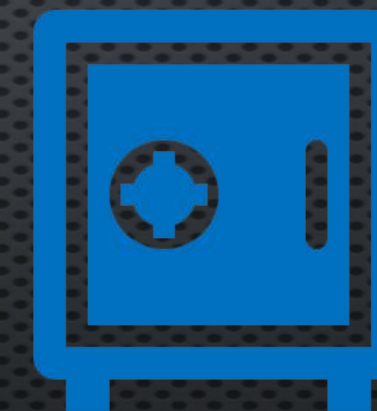
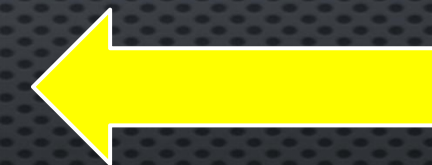
- *Kerberos* 认证

- 通过向服务端发送一张“票”，来证明自己的身份



票与 KDC

加密的 *Kerberos* 票据



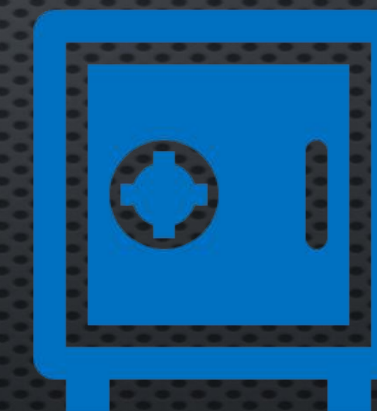
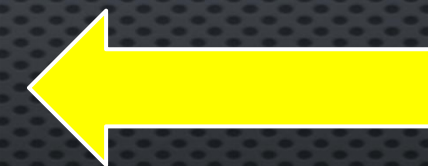
KDC

- 域内最重要的服务器
- 统一保存着所有账号的密码

票与 KDC

TGT – 访问 TGS 服务的票

普通票据 – 访问其他服务的票



KDC

- 域内最重要的服务器
- 统一保存着所有账号的密码

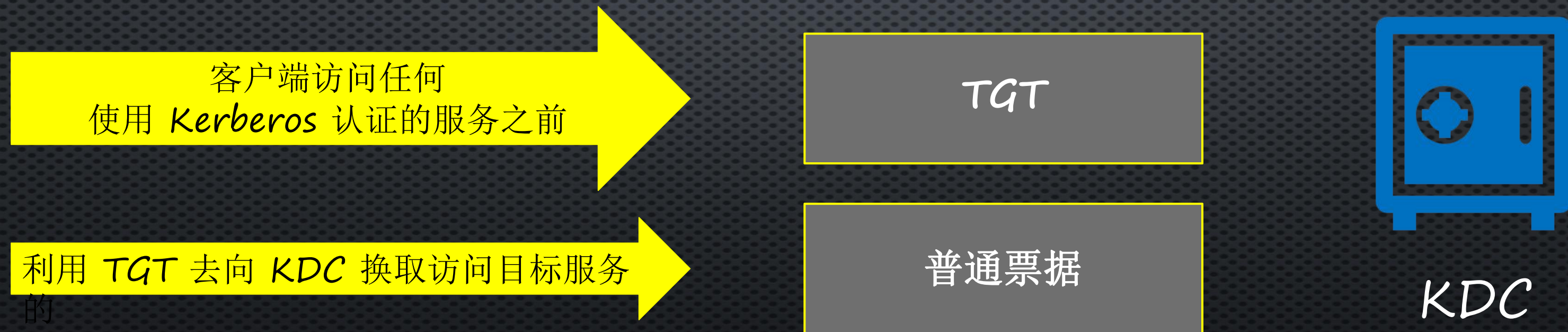
TGT - TICKET GRANTING TICKET



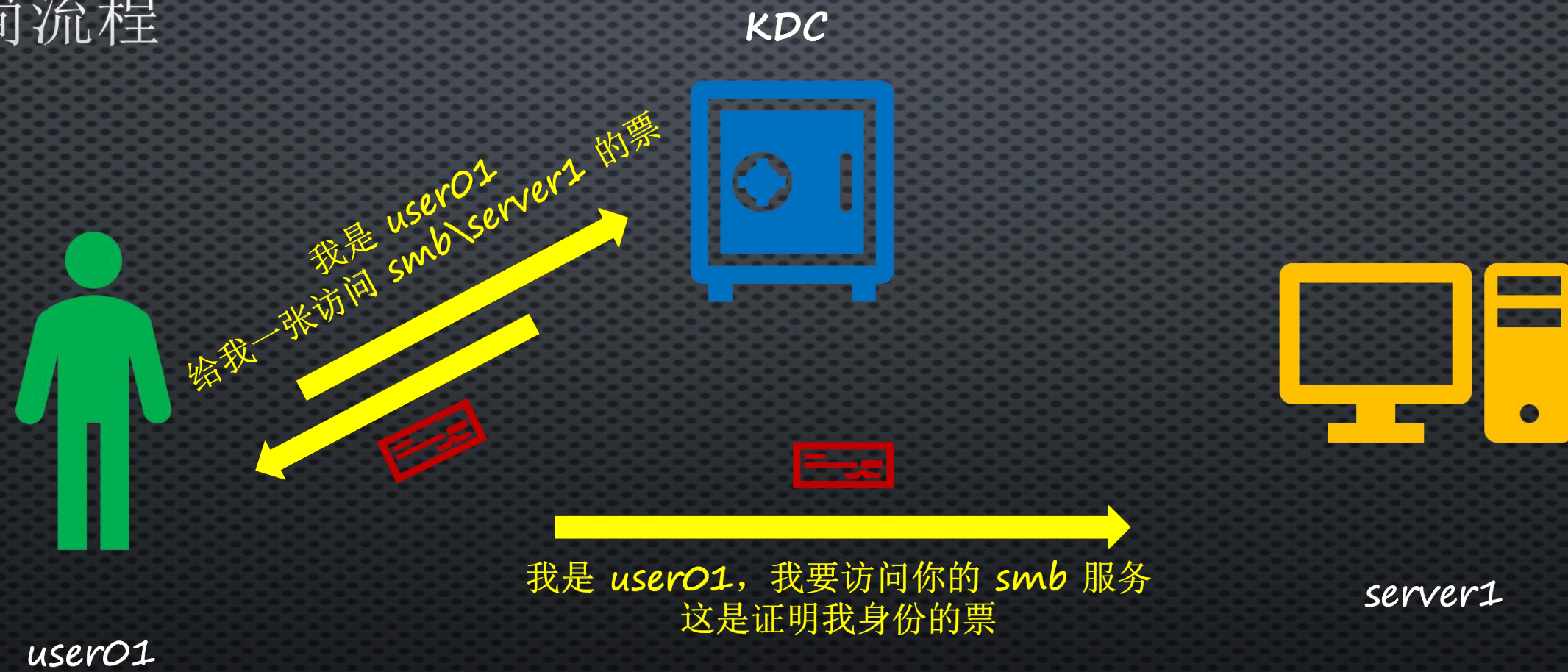
由 *KDC* 发放的最重要的一张票据。获取了代表某个用户的身份访问域内的资源



TGT 与其他票据的关系

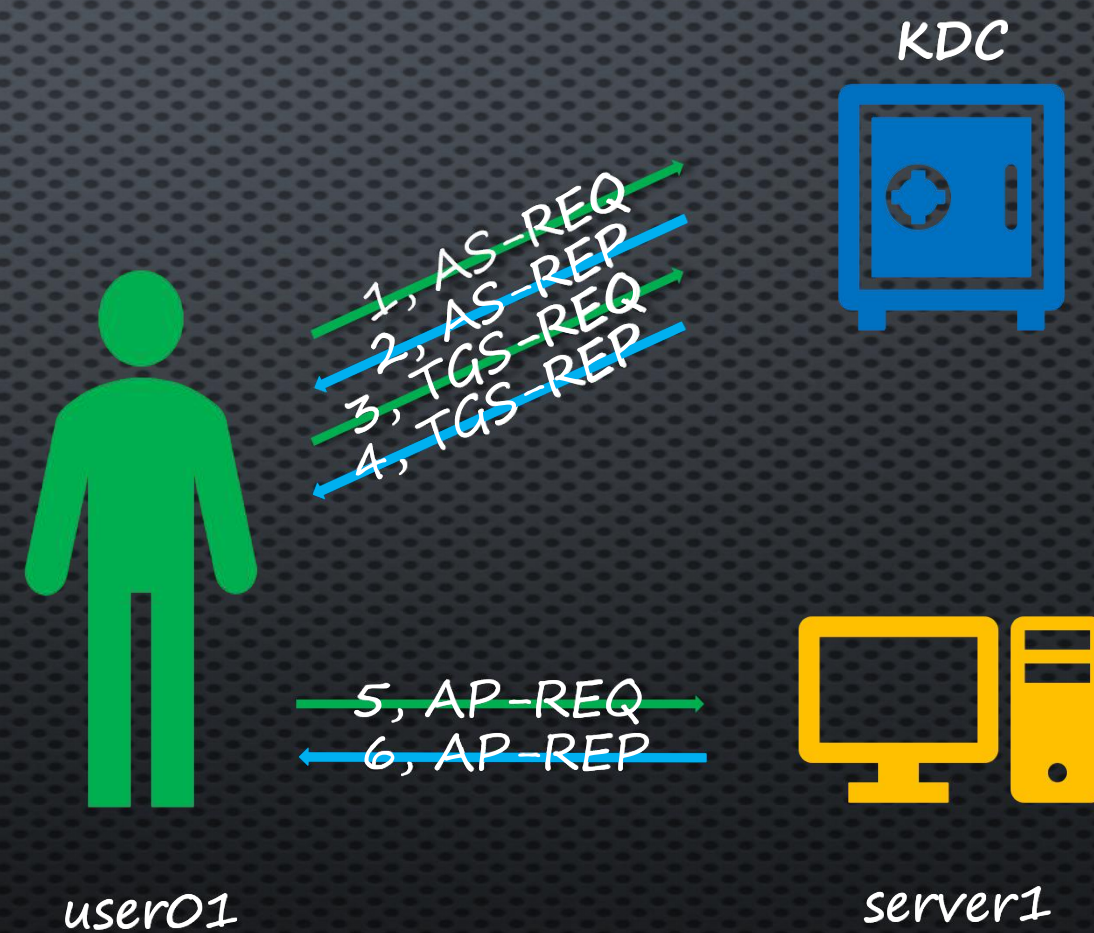


KERBEROS 极简流程



KERBEROS 子协议

- **AS-REQ & AS-REP**
 - 获取 TGT （身份证）
- **TGS-REQ & TGS-REP**
 - 用 TGT 去换目标服务的票
- **AP-REQ & AP-REP**
 - 将票发送给目标服务



AS 协议

TGT（身份证） 获取协议

AS-REQ & AS-REP



- **AS-REQ**
 - “我是 user01, 这是一段加密数据, 来证明我的身份”
- **AS-REP**
 - “身份验证成功, 给你一张能证明你是 user01 的票 (TGT), 以及对应的 Session Key”

```
[PDU Size: 308]
Kerberos
  Record Mark: 304 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: 3041a003020112a23a0438cc9e41085806106f6f810ebf7a...
            etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
            cipher: cc9e41085806106f6f810ebf7aa51f4c52e03b38824f8ad2...
        PA-DATA Unknown:119
      req-body
        Padding: 0
        kdc-options: 40800000 (forwardable, renewable)
        cname
          realm: W4EDOM-L4.BASE
        sname
          till: 2017-03-08 13:54:41 (UTC)
          rtime: 2017-04-06 13:54:41 (UTC)
          nonce: 769773897
        etype: 6 items
        addresses: 1 item FD23-73<20>
```

authenticator 验证因子

AS-REQ h1nty @ 360 A-TEAM

AS-REQ & AS-REP



- **AS-REQ**
 - “我是 user01, 这是一段加密数据, 来证明我的身份”
- **AS-REP**
 - “身份验证成功, 给你一张能证明你是 user01 的票 (TGT) 以及对应的 Session Key”

```
▼ Kerberos
  ► Record Mark: 1515 bytes
  ▼ as-rep
    pvno: 5
    msg-type: krb-as-rep (11)
    ▼ padata: 1 item
      ► PA-DATA PA-ENCTYPE-INF02
    crealm: W4EDOM-L4.BASE
    ► cname
    ▼ ticket
      tkt-vno: 5
      realm: W4EDOM-L4.BASE
      ► sname
      ▼ enc-part
        etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
        kvno: 2
        cipher: f788fbeb18790e549b60c5bdef04375439b9a35e8a44ced6...
    ▼ enc-part
      etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
      kvno: 82
      cipher: 19738d868597466c7b78bcf2c14b692989a9c27d617dcc92...
```

加密的 TGT

加密的 Session Key

AS-REP

h1nty @ 360 A-TEAM

ATTACKING AS-REQ & AS-REP

域内用户枚举

TGT SESSION
KEY

离线爆破

Overpass the
Hash(Pass the key)

域内用户枚举



- **AS-REQ** 不进行预身份认证时
 - “我是 *user01*，但是我不想提交 *authenticator* 来证明我的身份，请给我一张 *TGT*”
- 成功时，**AS-REP**
 - “*user01* 账号存在，并且配置了不要求进行预身份验证”
- 失败时，**KRB-ERROR**
 - **PREAUTH-REQUIRED**
 - **UNKNOWN-PRINCIPLE**

```
[PDU Size: 308]
Kerberos
  Record Mark: 304 bytes
  as-req
    pvno: 5
    msg-type: krb-as-req (10)
    padata: 2 items
      [Redacted]
    PA-DATA Unknown
  req-body
    Padding: 0
    kdc-options: 40800000 (forwardable, renewable)
    cname
      realm: W4EDOM-L4.BASE
    sname
      time: 2017-03-08 13:54:41 (UTC)
      rtime: 2017-04-06 13:54:41 (UTC)
      nonce: 769773897
    etype: 6 items
    addresses: 1 item FD23-73<20>
```

去掉这段数据

发送 AS-REQ 请求时不进行预身份验证
根据返回的错误码可判断用户是否存在

AS-REQ

域内用户枚举

- 发送不进行预身份验证的 *AS-REQ* 请求，对响应进行检查

响应类型	结果
<i>AS-REP</i> 响应	账号存在 且开启了 “不要求 <i>Kerberos</i> 预身份验证” 选项
<i>KRB-ERROR: PREAUTH-REQUIRED</i>	账号存在 未开启 “不要求 <i>Kerberos</i> 预身份验证” 选项
<i>KRB-ERROR: CLIENT-REVOKED</i>	账号存在 但是处于锁定状态
<i>KRB-ERROR: UNKNOWN-PRINCIPLE</i>	账号不存在

域内用户枚举-相关工具

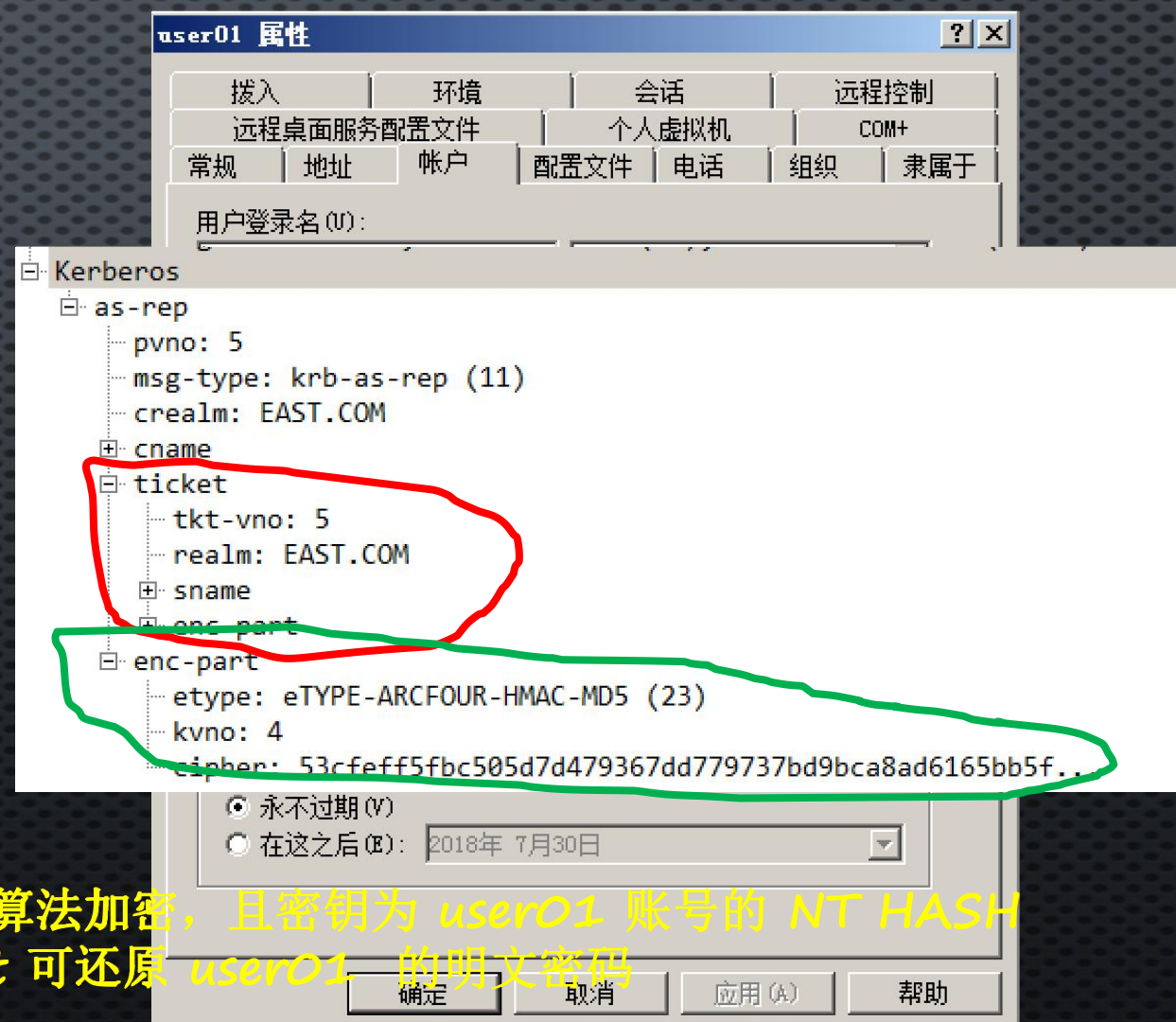
- [HTTPS://GITHUB.COM/360-A-TEAM/KERBEROSUSERENUM](https://github.com/360-A-Team/KERBEROSUSERENUM)
- NMAP KRB5-USER-ENUM
- MSF KERBEROS-ENUMUSERS

```
East@MacBook-Pro-9 > /private/tmp > p3 enum.py --file=/tmp/test --dcip=192.168.1.1 --domain=192.168.1.1 --port=88
  ▪ 192.168.1.1\liming True
  ▪ 192.168.1.1\wufangdong True
  ▪ 192.168.1.1\aaaaahahaha False
  ▪ 192.168.1.1\test11 False
```

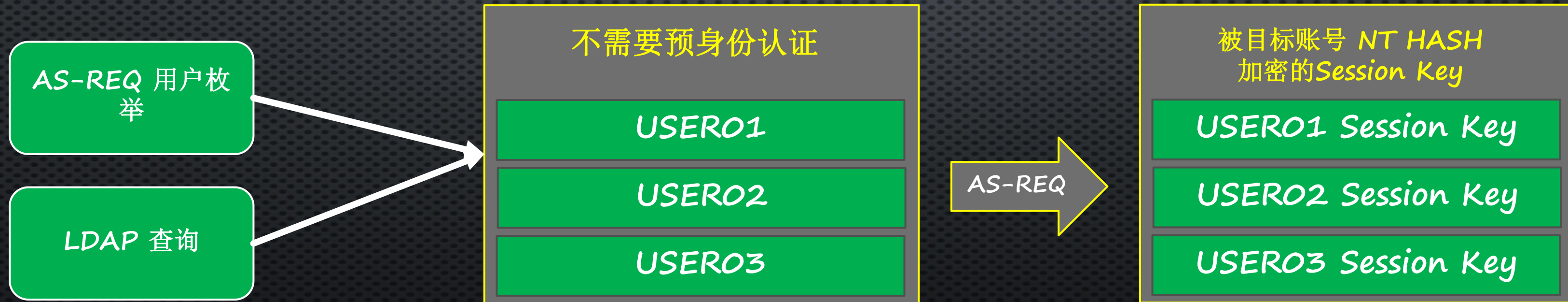

TGT SESSION KEY 离线爆破

- **AS-REQ**, 由客户端发送至 **KDC**
 - “我是 *user01*, 我不进行预身份验证, 请给我一张 *TGT*”
- **AS-REP**, 由 **KDC** 返回
 - “经过检查发现 *user01* 被配置为不需要进行预身份验证, 这里给你返回一个 *AS-REP*, 里面包含了你想要的 *TGT*, 以及与 *TGT* 配合使用的 *session key*”

🤪 **enc-part(session key)** 部分被 **RC4** 算法加密, 且密钥为 *user01* 账号的 **NT HASH**
离线暴力破解 **enc-part** 可还原 *user01* 的明文密码



TGT SESSION KEY 离线爆破



TGT SESSION KEY 离线爆破



- 离线爆破，不在域控留下日志
- 绕过密码锁定策略

明文密码

P@ssword

1qazXSW@

111222

离线爆破
Session Key

被目标账号 NT HASH
加密的 Session Key

USER01 Session Key

USER02 Session Key

USER03 Session Key

TGT 离线爆破 — 相关工具

ASREPROAST

- 利用 *LDAP* 自动查找域内关闭了预身份验证的 *KERBEROS* 账户
- 自动发送 *AS-REQ* 获取账户的 *AS-REP* 中的 *ENC-PART*，并输出可被 *JOHN THE RIPPER* 破解的格式
- [HTTP://WWW.HARMJOY.NET/BLOG/ACTIVEDIRECTORY/ROASTING-AS-REPS/](http://www.harmjoy.net/blog/activedirectory/roasting-as-reps/)

 HarmJ0y / ASREProast

insight.io Not Indexed

Watch 7

★ Unstar 42

Fork 16

<> Code

! Issues 0

🔗 Pull requests 1

📁 Projects 0

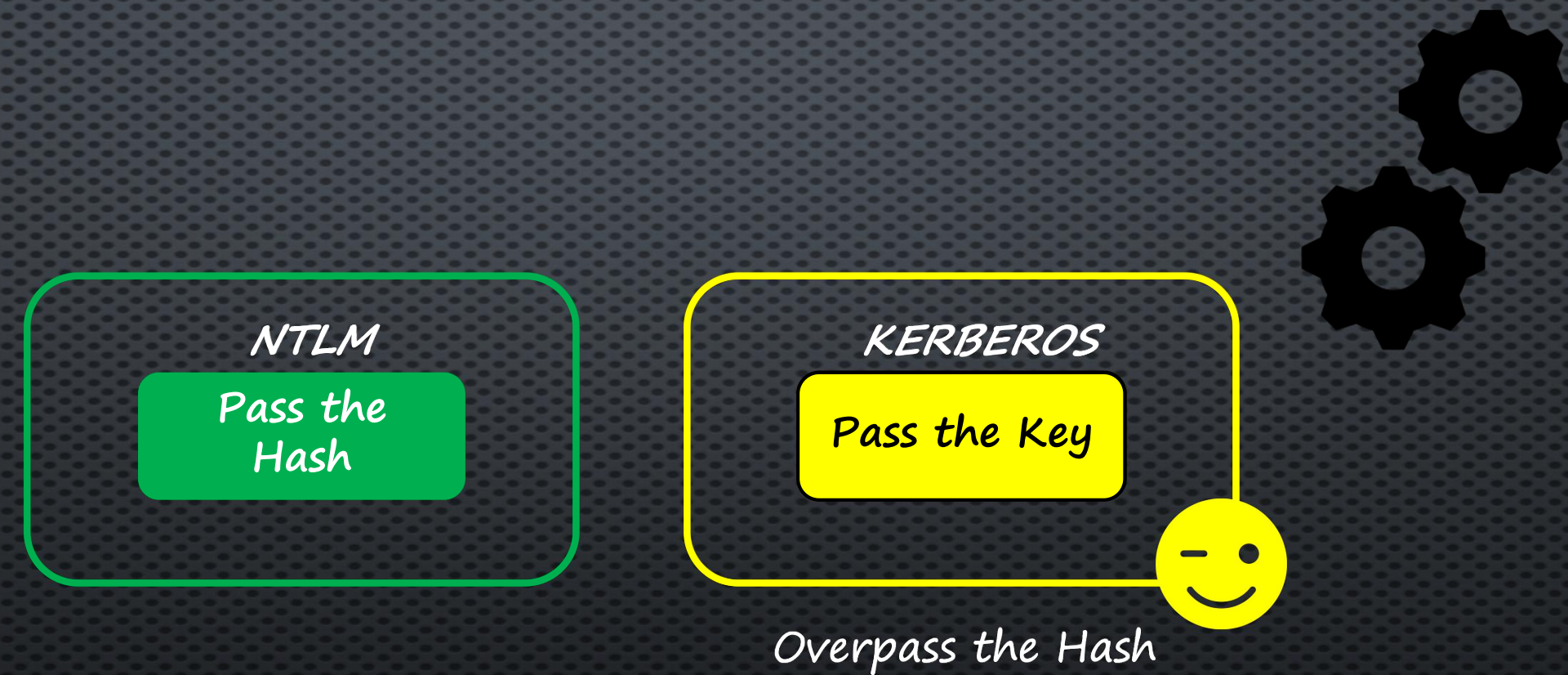
📖 Wiki

📊 Insights

Project that retrieves crackable hashes from KRB5 AS-REP responses for users without kerberoast preauthentication enabled.

n1nty @ 360 A-TEAM

PASS THE KEY



在没有明文密码的情况下进行 NTLM/Kerberos 身份认证

PASS THE KEY



- **AS-REQ**
 - “我是 *user01*，这是一段加密数据，来证明我的身份”
- **AS-REP**
 - “身份验证成功，给你一张能证明你是 *user01* 的票（*TGT*），以及对应的 *Session Key*”

加密算法

加密 Key

被加密数据

PASS THE KEY



- **AS-REQ**
 - “我是 `user01`，这是一段加密数据，来证明我的身份”
- **AS-REP**
 - “身份验证成功，给你一张能证明你是 `user01` 的票（`TGT`），以及对应的 `Session Key`”

Key”

加密算法	KEY
AES 256	密码的 AES 256 KEY
AES 128	密码的 AES 128 KEY
RC4	密码的 NT HASH 格式
其他很多	



被加密数据为当前时间戳

PASS THE KEY



- **AS-REQ**

- “我是 user01，这是一段**加密数据**，来证明我的身份”

- **AS-REP**

- “身份验证成功，给你一张能证明你是 user01 的票（TGT），以及对应的 Session Key”

- **AS-REQ**

- “我是 user01，我利用 **RC4 算法**并以我密码的 **NT HASH** 做为 **KEY**，加密了当前时间戳（**authenticator**），来证明我的身份”

- **AS-REP**

- “因为你利用的是 **RC4 算法**对 **authenticator** 进行加密，所以我将**同样利用 RC4 算法**对 **authenticator** 进行解密，并使用 **user01 账号的 NT HASH** 做为 **解密 Key**。”
- “解密成功，发现你传送过来的时间戳与服务器的当前时间戳误差在 5 分钟以内，所以你确实是 user01，给你一张 TGT，以及对应的 Session Key”



进行 Kerberos 验证的过程中，并没有涉及到账号的明文密码

TGS 协议

利用 TGT 换取其他服务票据的协议

TGS-REQ & TGS-REP



- 。。。前面已经通过 AS-REQ 与 AS-REP 获取了 TGT
- TGS-REQ**
 - 客户端：“我是 user01，我想申请一张访问 server1 的 smb 服务的票。这是我的 TGT，以及一段加密数据来证明我知道与 TGT 对应的 Session key”
- TGS-REP**
 - KDC：“您的 TGT 是合法的，并且加密数据校验成功，这是一张能够访问 server1 的 smb 服务的票（service ticket）以及 Session Key”

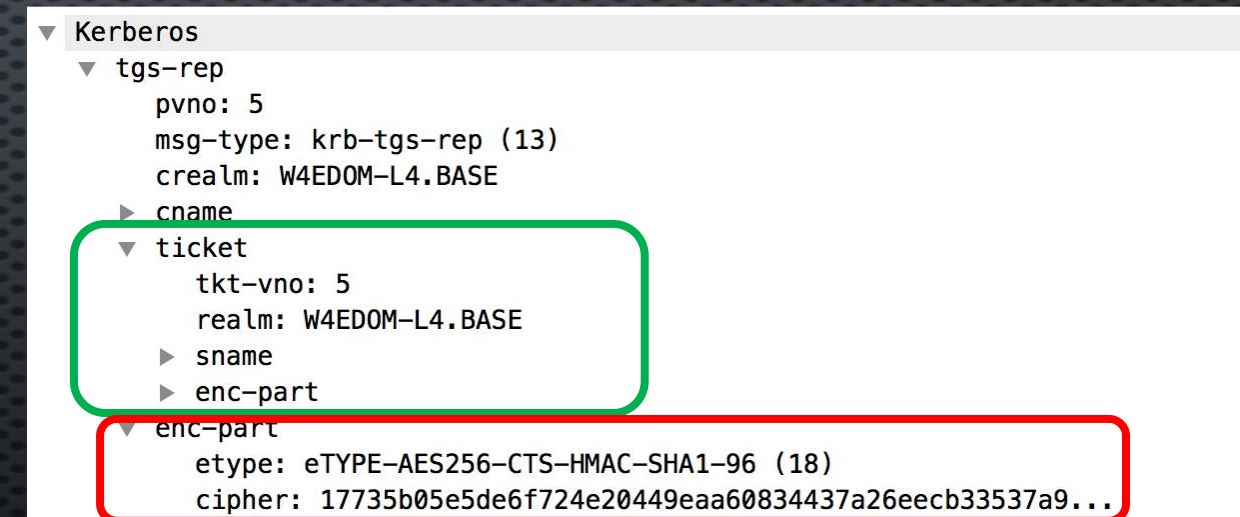


TGS-REQ

TGS-REQ & TGS-REP



- 。 。 。 前面已经通过 AS-REQ 与 AS-REP 获取了 TGT
- **TGS-REQ**
 - 客户端: “我是 user01, 我想申请一张访问 server1 的 smb 服务的票。这是我的 TGT, 以及一段加密数据来证明我知道 与 TGT 对应的 Session key”
- **TGS-REP**
 - KDC: “您的 TGT 是合法的, 并且加密数据校验成功, 这是一张能够访问 server1 的 smb 服务的票 (service ticket) 以及 Session Key”



TGS-REP

ATTACKING TGS-REQ & TGS-REP

GOLDEN TICKET

KERBEROAST

KERBEROS
DELEGATION

SPN 后门, <https://adsecurity.org/?p=3466>

GOLDEN TICKET



KERBEROS 世界中最重要的一张票据。获取了代表某个用户的 *TGT* 后，就可以以该用户的身份访问域内的资源。



伪造 *TGT*，冒充任意用户

GOLDEN TICKET



- 。。。前面已经通过 *AS-REQ* 与 *AS-REP* 获取了 *TGT*
- **TGS-REQ**
 - 客户端：“我是 *user01*，我想申请一张访问 *server1* 的 *smb* 服务的票。这是我的 *TGT*，以及一段加密数据来证明我知道与 *TGT* 对应的 *Session key*”
- **TGS-REP**
 - KDC：“您的 *TGT* 是合法的，并且加密数据校验成功，这是一张能够访问 *server1* 的 *smb* 服务的票（*service ticket*）以及 *Session Key*”



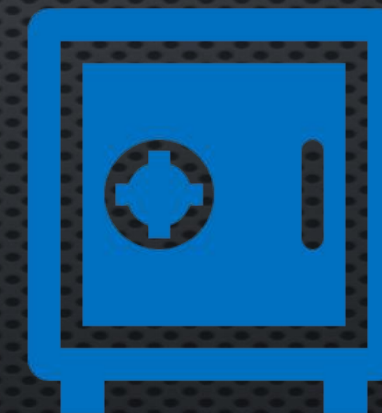
KDC 是如何判断客户端提交上来的 *TGT* 是合法的？

GOLDEN TICKET



加密的 TGT

加密的 Authenticator



KDC

利用 TGS-REQ 向 KDC 申请服务 Server1 SMB 服务的票

GOLDEN TICKET

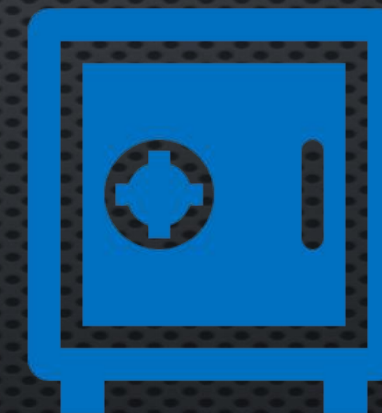


返回访问
SMB/Server1
的票与 Session
key

加密的 TGT

加密的 Authenticator

用 KRBTGT
Key
解密 TGT



KDC

GOLDEN TICKET



- 只要 *TGT* 能被 *KDC* 利用相关算法解密（以 *KRBTGT* 相关 *Key* 做为密钥），则 *TGT* 合法
- 盗取 *KRBTGT* 相关 *Key* 即可伪造 *TGT*

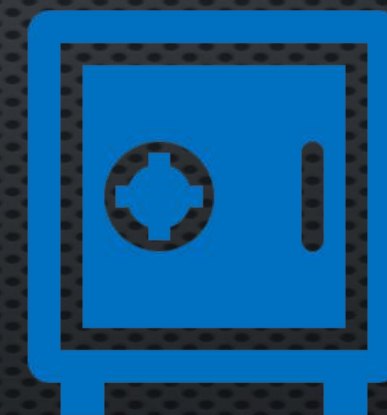
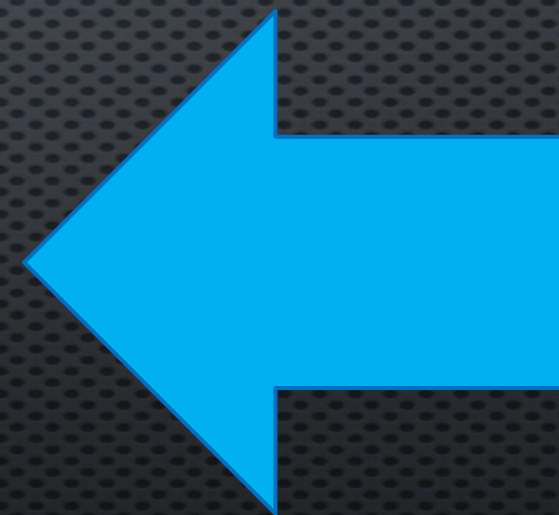
KERBEROAST



返回访问
SMB/Server1
的票与 *Session*
key

被 *SMB/Server1*
服务启动账号的 *NT HASH* 加密

Session Key



KDC

KERBEROAST – Kerberos 服务启动账号密码的离线爆破

KERBEROAST

USER01 TGT

MSSQL/SERVER
1 TICKET

HTTP/SERVER2
TICKET

KMC/SERVER3
TICKET



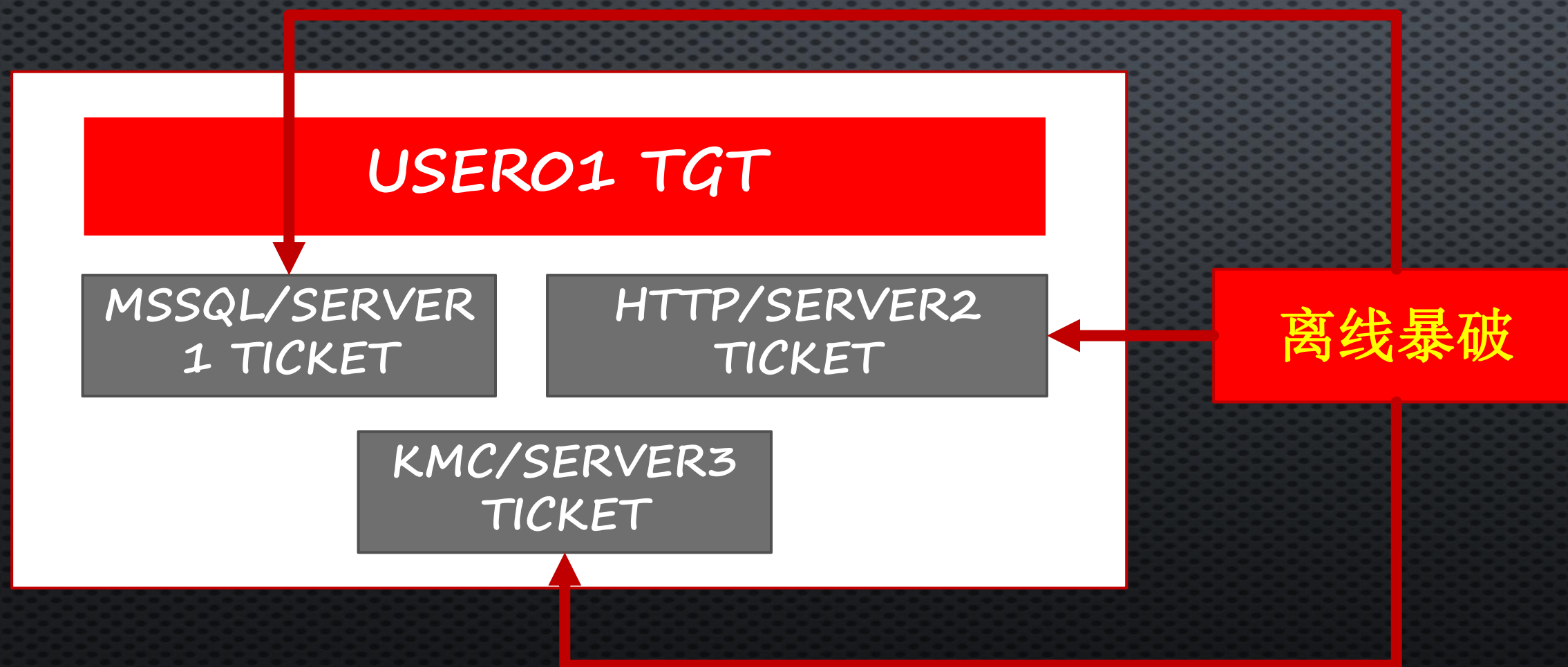
SPN 查询
域内 Kerberos 服务

MSSQL/SERVER1

HTTP/SERVER2

KMC/SERVER3

KERBEROAST



访问域内
资源

制作白银
票据

KERBEROS DELEGATION



后面再讲

AP 协议

将票据传送给目标服务的协议

AP-REQ & AP-REP



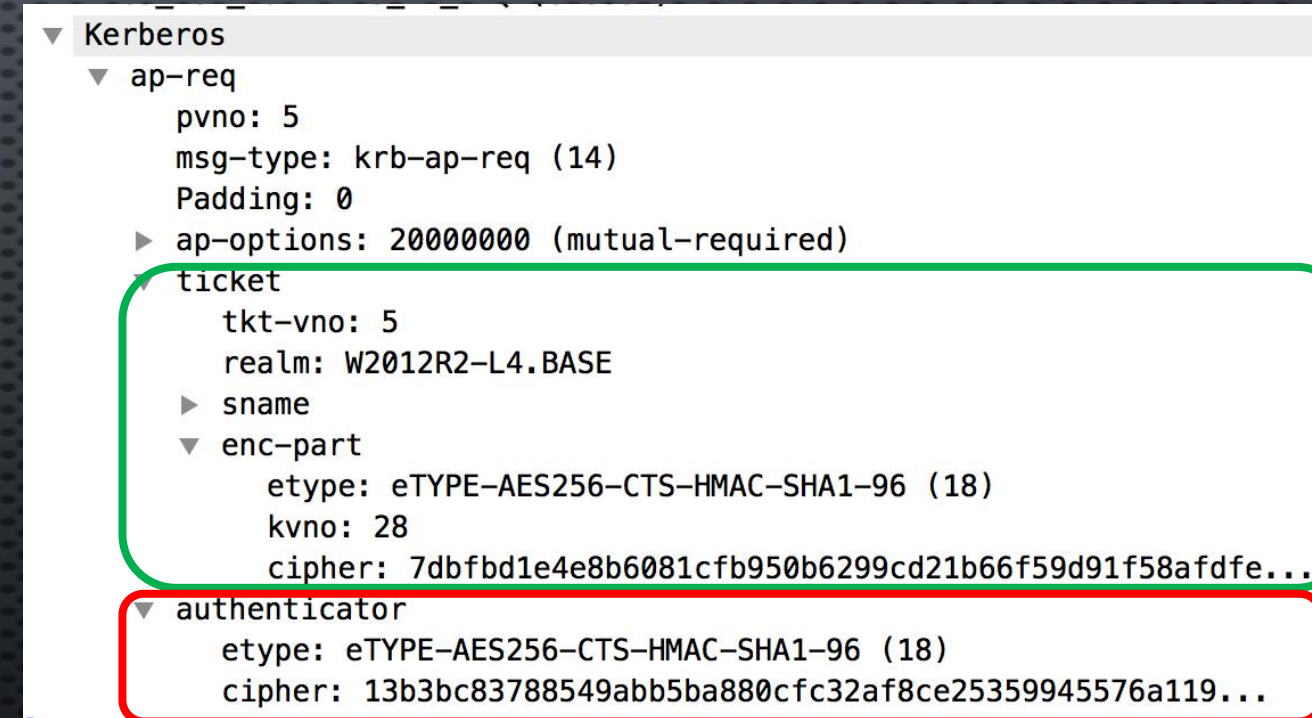
- 已通过 AS-REQ & AS-REP 获得 TGT
- 已通过 TGS-REQ & TGS-REP 获得 Service ticket

1. AP-REQ

- 客户端: “我是 user01, 我要访问你的 smb 服务, 这是一张能够证明我身份的票 (service ticket) 以及一段加密数据来证明我知道这张票所对应的 Session Key”

2. AP-REP

- SMB 服务: “您的票是合法的, 并且加密数据校验成功。已经确定您是 user01, 可以访问”



SMB 服务是如何验证票是否合法的?

AP-REQ & AP-REP



加密的 SERVICE
TICKET

加密的 Authenticator



Server1

利用 AP-REQ 访问 Server1 SMB 服务

AP-REQ & AP-REP



加密的 SERVICE
TICKET

加密的 Authenticator



Server1

ATTACKING AP-REQ & AP-REP

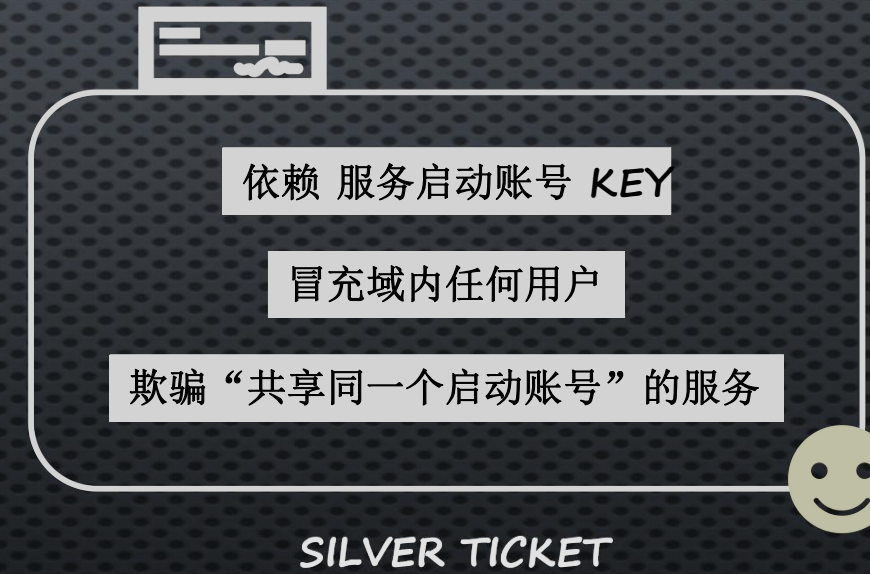
SILVER TICKET – 白银票据

SILVER TICKET



- 只要 *Service ticket* 能被目标服务利用相关算法解密（以服务启动账号的相关 *Key* 做为密钥），则 *Service ticket* 合法
- 盗取 “服务启动账号的相关 *Key*” 即可伪造 *Service ticket*

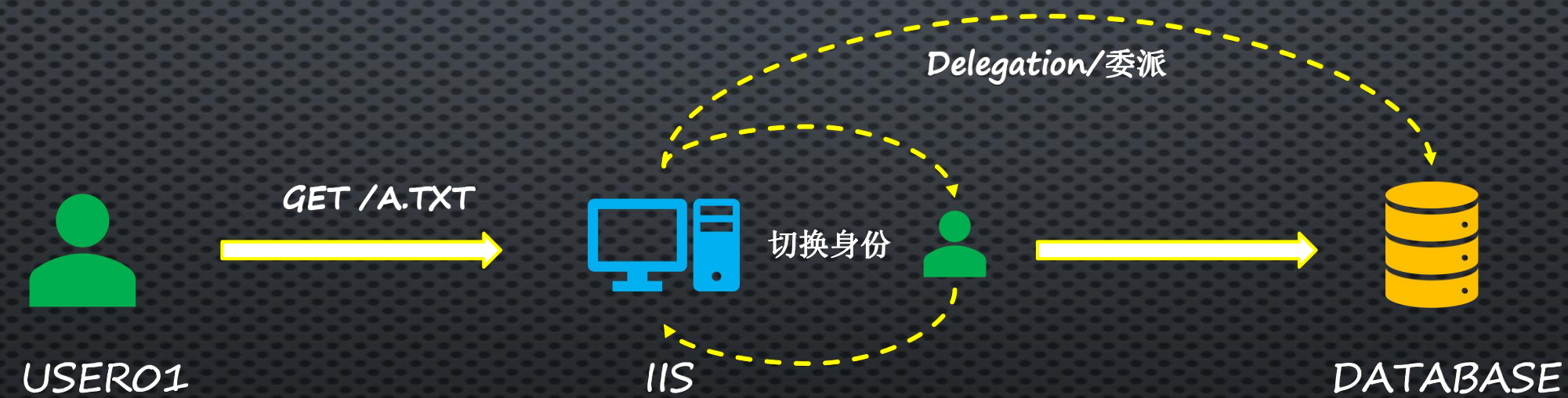
GOLDEN TICKET VS SILVER TICKET



KERBEROS DELEGATION

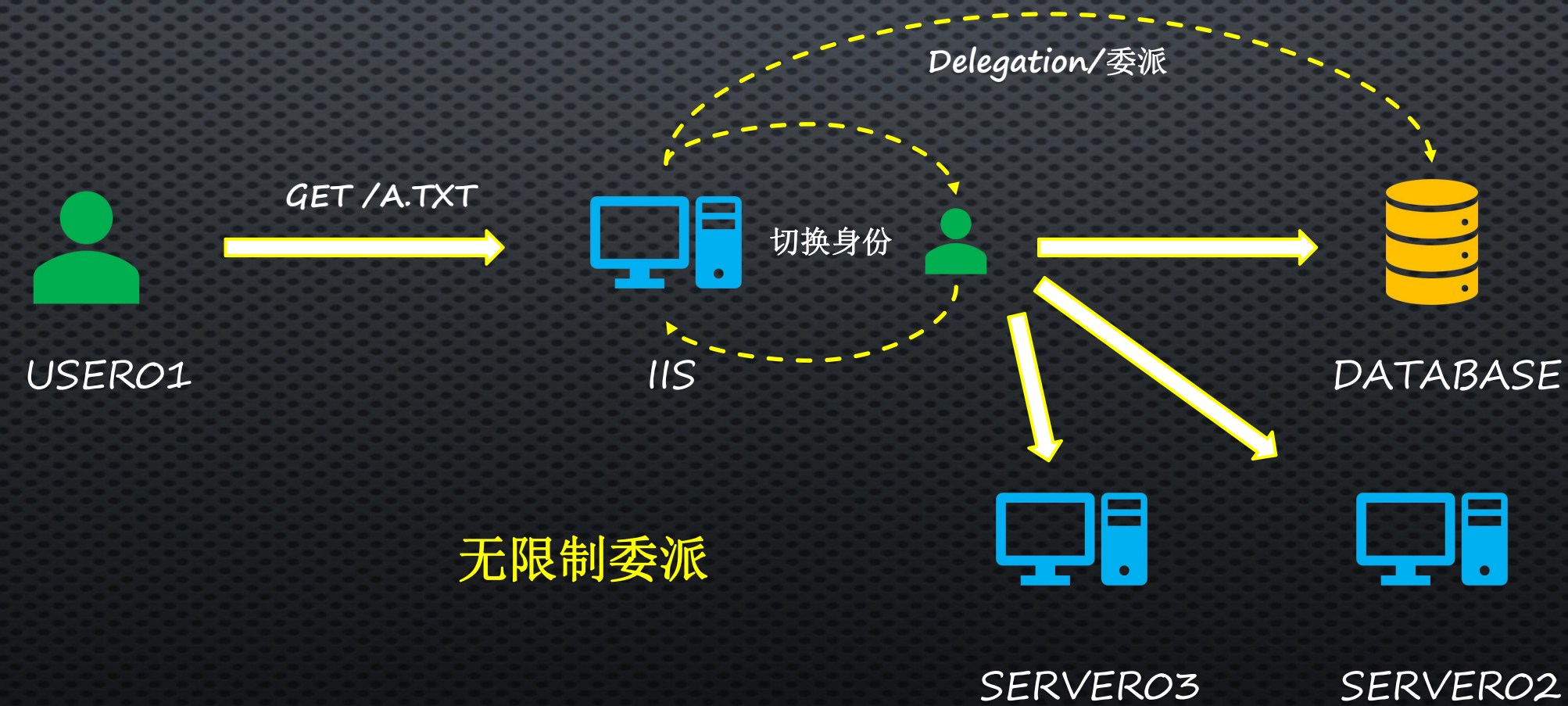
KERBEROS 委派

KERBEROS DELEGATION

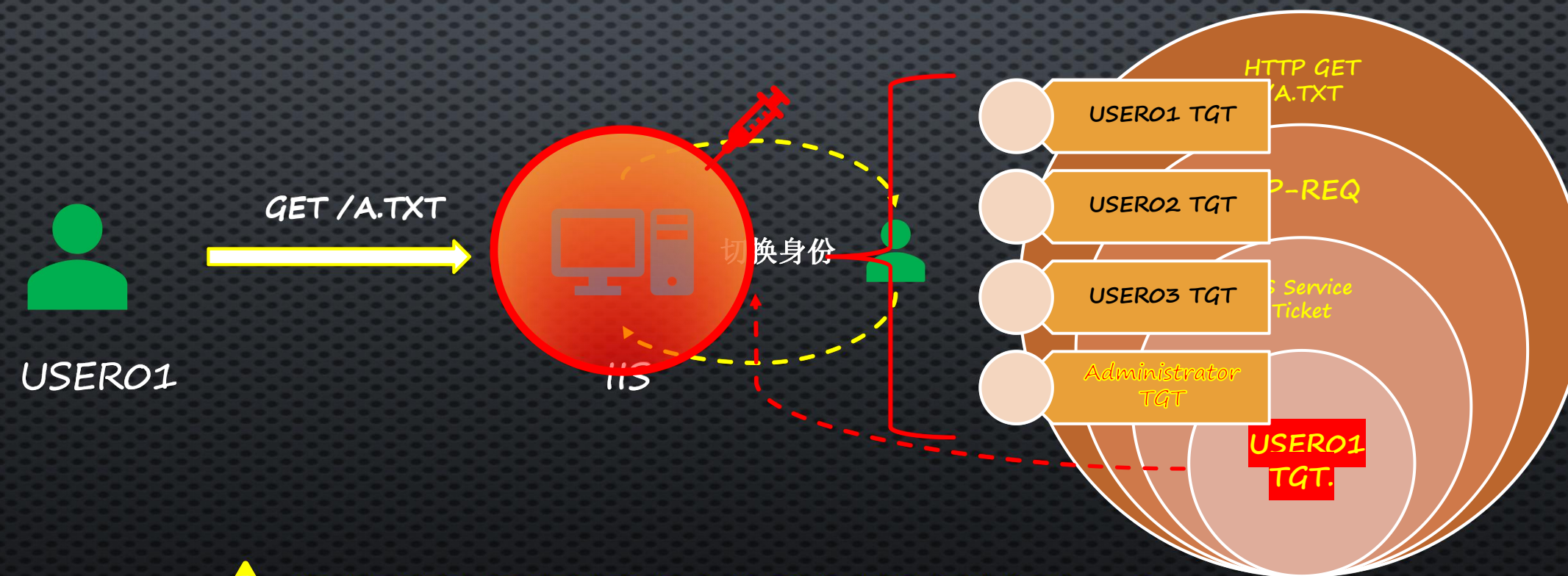


客户端 (USER01) 访问服务端 (IIS) 时, 服务端利用客户端的身份访问了其他资源

UNCONSTRAINED DELEGATION

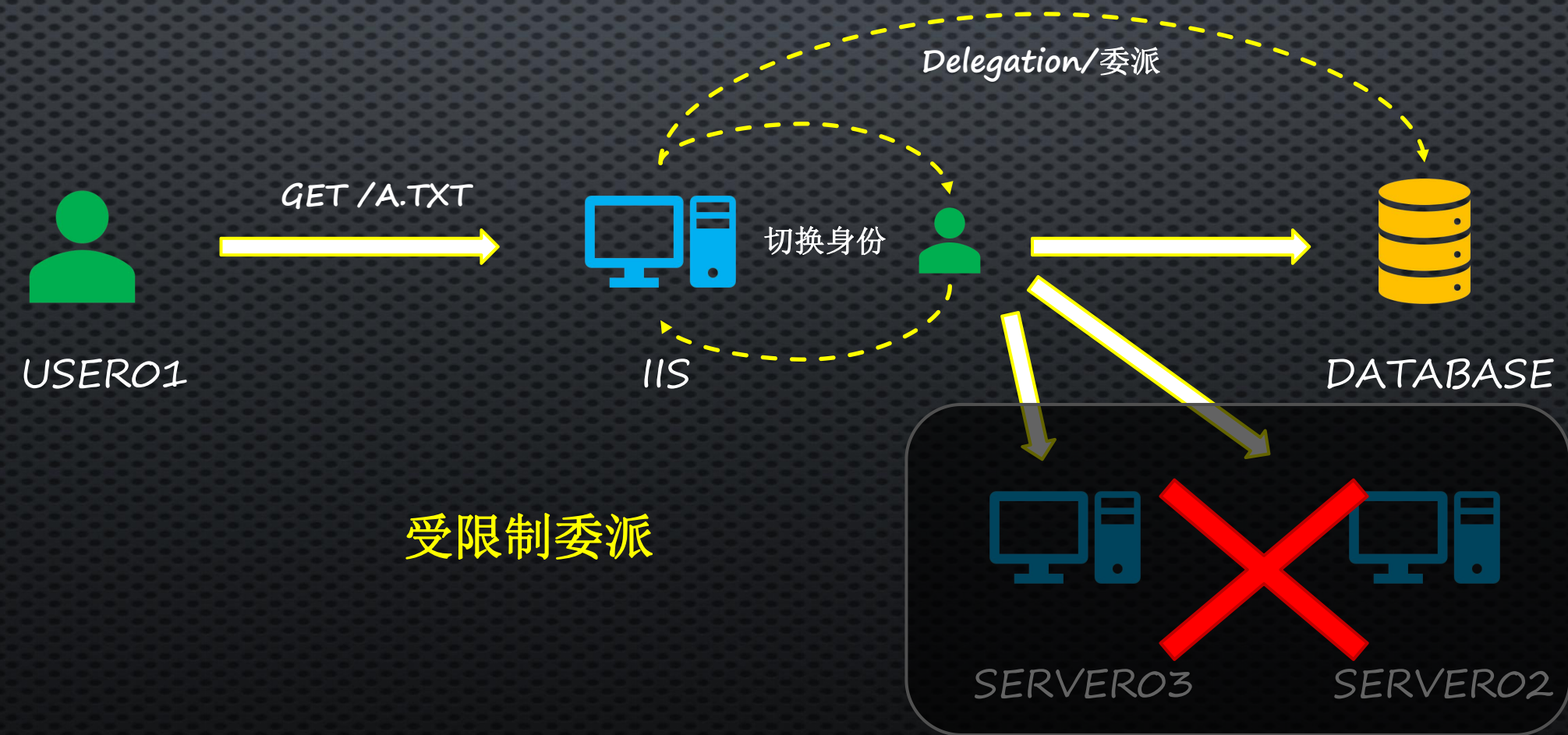


UNCONSTRAINED DELEGATION

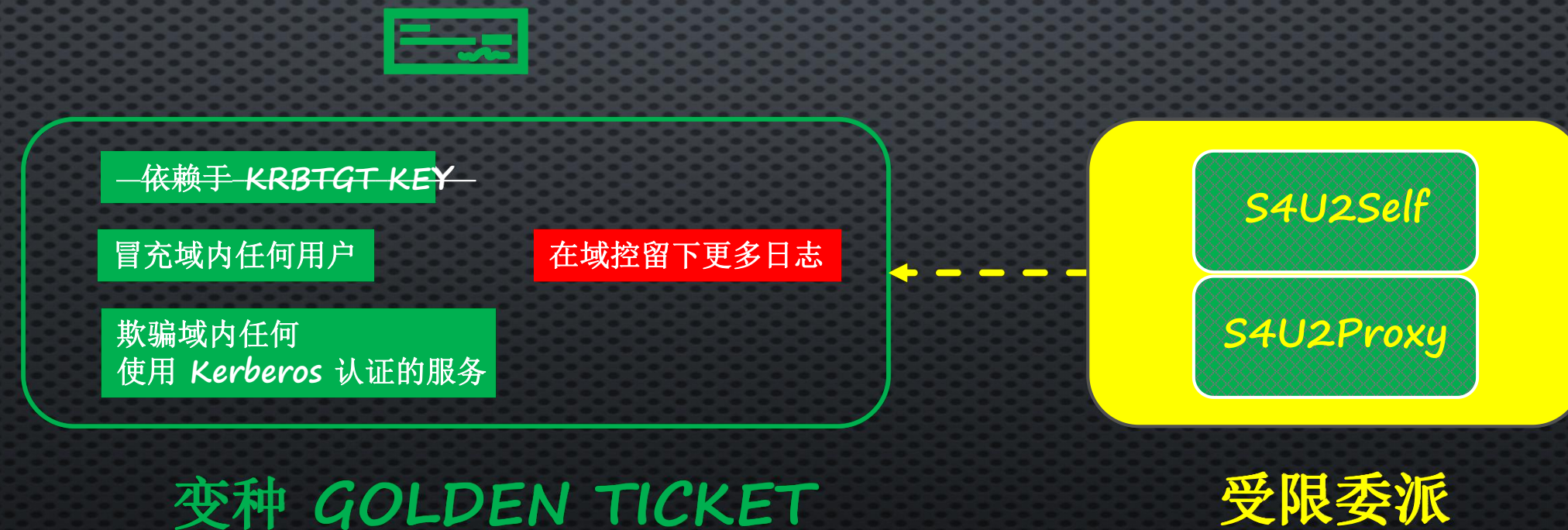


⚡ 拥有无限制委派的服务内将缓存有所有访客的 TGT

CONSTRAINED DELEGATION



受限委派与变种 *GOLDEN TICKET*



受限委派与变种 GOLDEN TICKET



ATEAM/TESTSERVICE

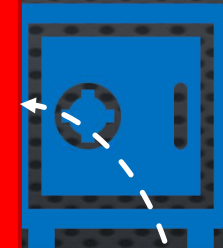
可以以任何用户的名义
申请一张访问自身的票

允许委派到 TGS 服务

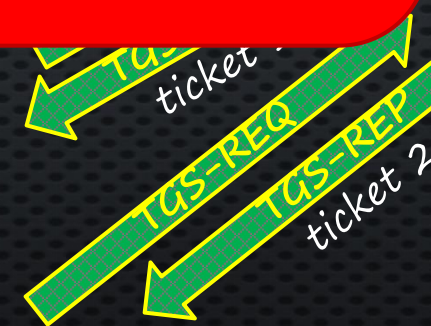
S4U2Self

S4U2Proxy

- 一张能以 **Administrator** 的名义访问 TGS 服务的票，就是 **Administrator** 的“TGT”
- 我们通过了普通账号 **USER01** 拿回了 **Administrator**（可以是域内任何用户）的 TGT。



KDC

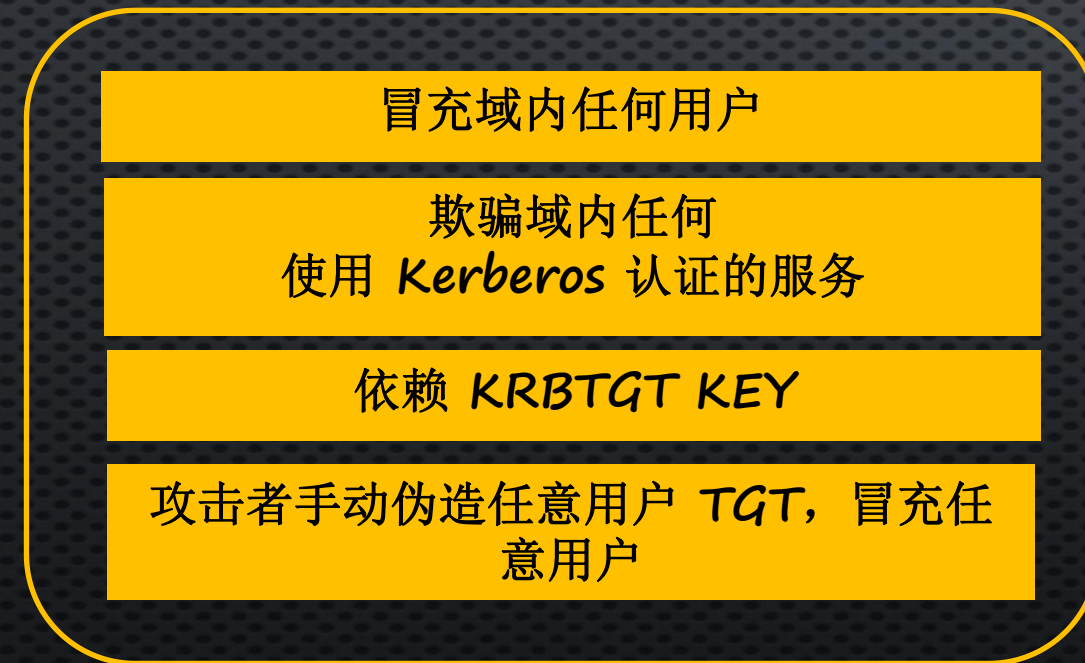


以 **USER01** 的身份
申请 TGT (*ticket 0*)

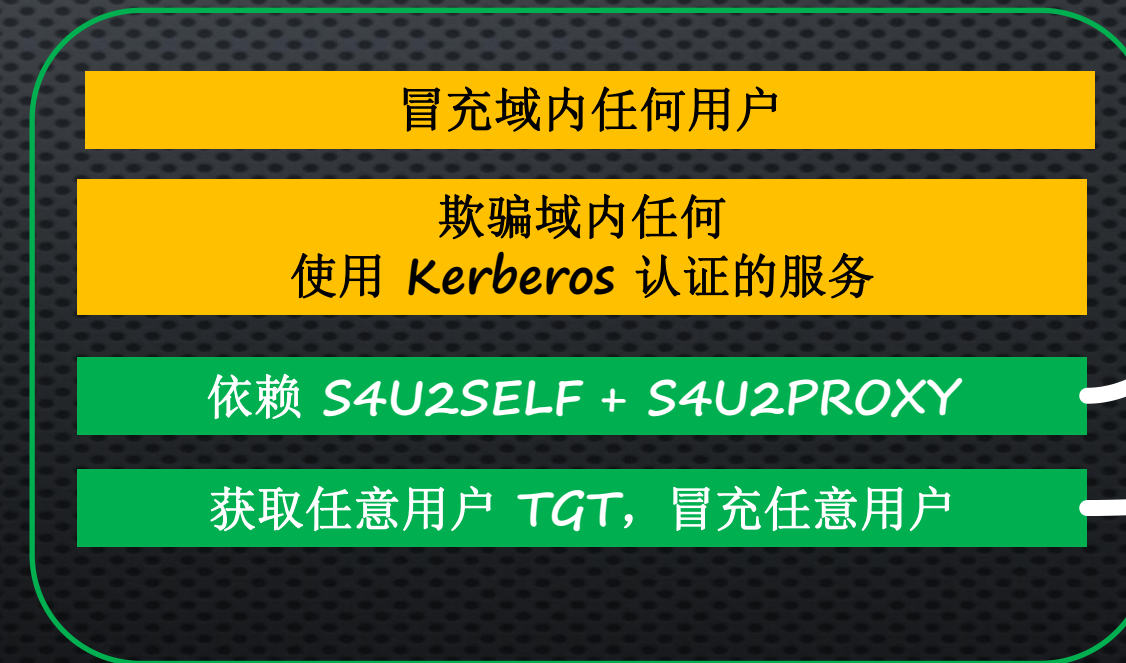
使用 **USER01** 的 TGT，通过 TGS-REQ 配合 S4U2Self 协议，以 **Administrator** 的名义申请一张访问 ATEAM/TESTSERVICE 的票 (*ticket 1*)

使用 **USER01** 的 TGT，通过 TGS-REQ 配合 S4U2Proxy 协议并带上 *ticket 1*，以 **Administrator** 名义申请一张委派到 TGS 服务的票 (*ticket 2*)

GOLDEN TICKET VS 变种 GOLDEN TICKET



GOLDEN TICKET



变种GOLDEN TICKET



KERBEROS 开发

- ASN.1
 - ASN1CRYPTO
 - PYASN1
- **IMPACKET**
- MINIKERBEROS



n1nty

@



360 A-TEAM

n1nty @ 360 A-TEAM

谢谢！

N1NTY @ 360 A-TEAM