Not A Security "Boundary"

Breaking  Trusts

Red teamer and offensive engineer

SpecterOps.io

I write lots of code :)

Red teamer, hunter, researcher

SpecterOps.io

I love shiny things :)

Forests as Security Boundaries

Each forest is a single instance of the directory, the top-level Active Directory container, and a security boundary for all objects that are located in the forest. This security boundary defines the scope of authority of the administrators. In general, a security boundary is defined by the top-level container for which no administrator external to the container can take control away from administrators within the container. 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.

Microsoft's "What Are Domains and Forests?" documentation

What If...

A computer in ForestA authenticates* to a computer ForestB

Full ticket-granting-tickets could move across the trust from ForestA to ForestB

We had a way to easily extract and reuse these TGTs while in ForestB?

*bonus if we can force the authentication step :)

Kerberos in 60 seconds

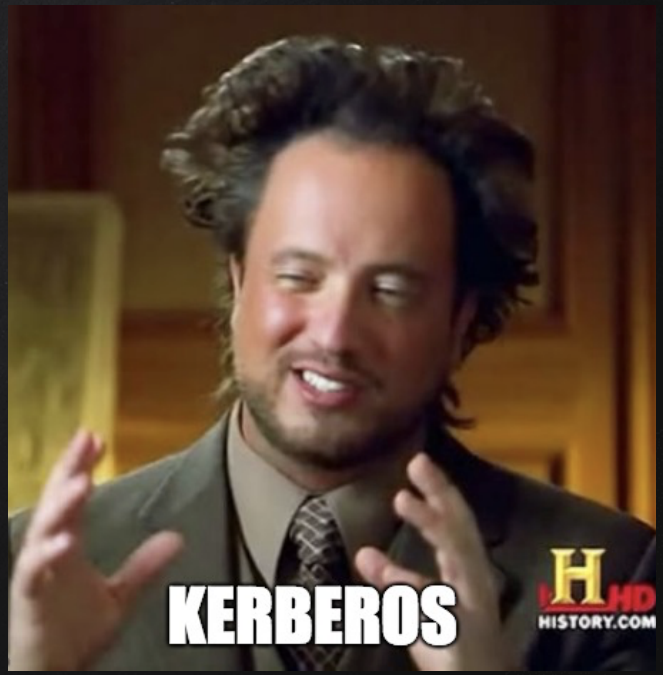
OK maybe a few minutes ;)

Active Directory

Kerberos TL;DR

- ✘ Kerberos is dense, and we don't have time to explain the entire protocol
  - Instead we'll focus on a few key terms and points that are necessary to understand the nuances of the trust attack

- ✘ For a more in depth explanation, see Sean Metcalf's post at https://bit.ly/2JhbAXl



Kerberos TL;DR

1. An account authenticates to a domain controller (DC/KDC) by encrypting some data with with a key derived from the user's password (e.g. RC4_HMAC(NTLM hash) or AES128/256_HMAC keys)

2. If auth is successful, the DC returns a ticket granting ticket (TGT) to the user that contains a privileged attribute certificate (PAC)
   - The PAC is encrypted with the hash of the `krbtgt` (Kerberos ticket-granting-ticket service) account and contains auth information like the user's SID and groups they're in.

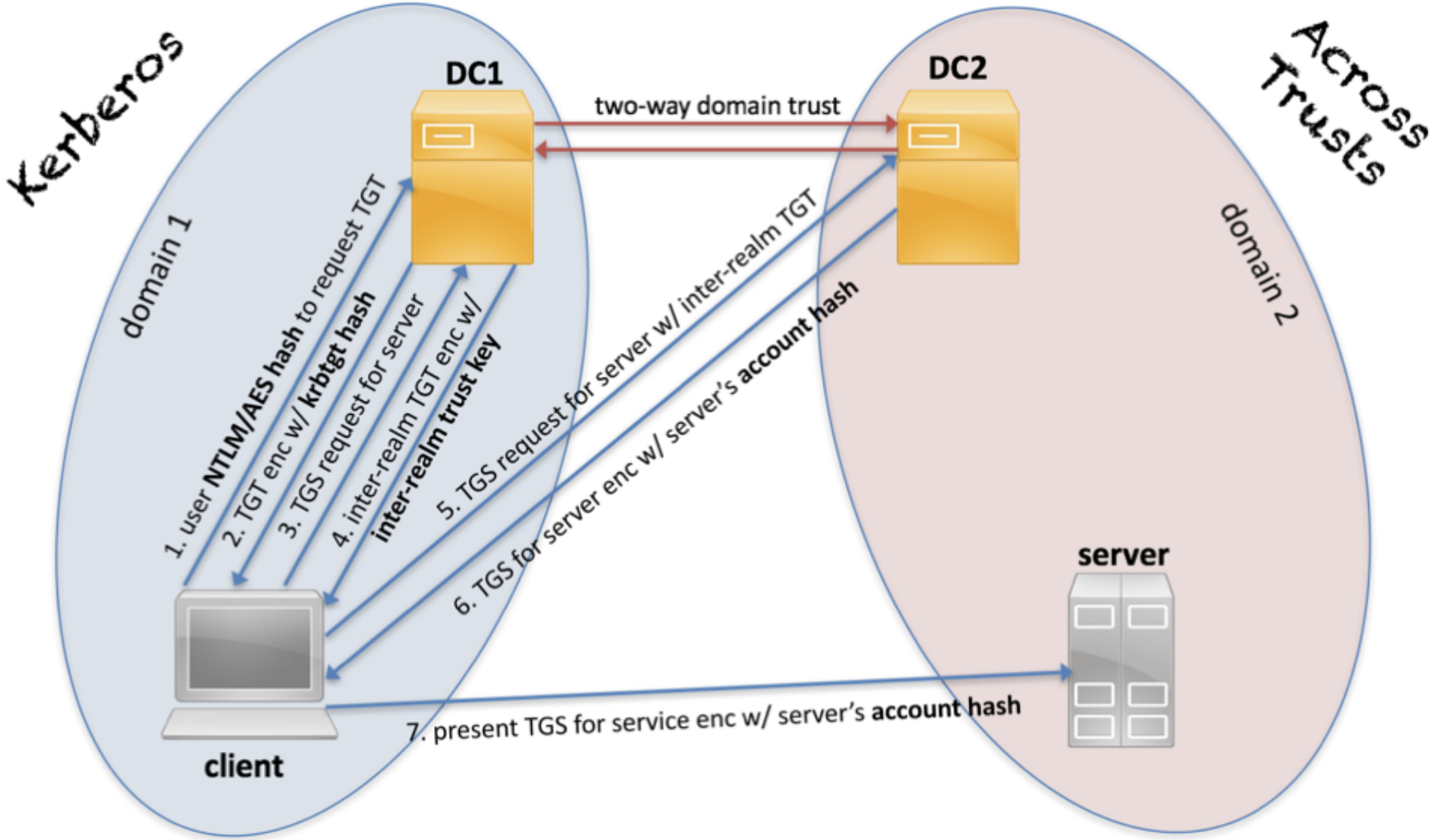
Active Directory

Kerberos tl;dr

3. The account requests a service ticket to a particular service principal name (SPN) by presenting the TGT to the domain controller

4. The DC returns a service ticket with the same auth info as the TGT.

5. The user sends the service ticket to the target service/machine, which decides whether to grant the user access.


Kerberos
Across Trusts
DC1
DC2
two-way domain trust
domain 1
domain 2
1. user **NTLM/AES hash** to request TGT
2. TGT enc w/ **krbtgt hash**
3. TGS request for server
4. inter-realm TGT enc w/ **inter-realm trust key**
5. TGS request for server w/ inter-realm TGT
6. TGS for server enc w/ server's **account hash**
7. present TGS for service enc w/ server's **account hash**
server
client

Delegation

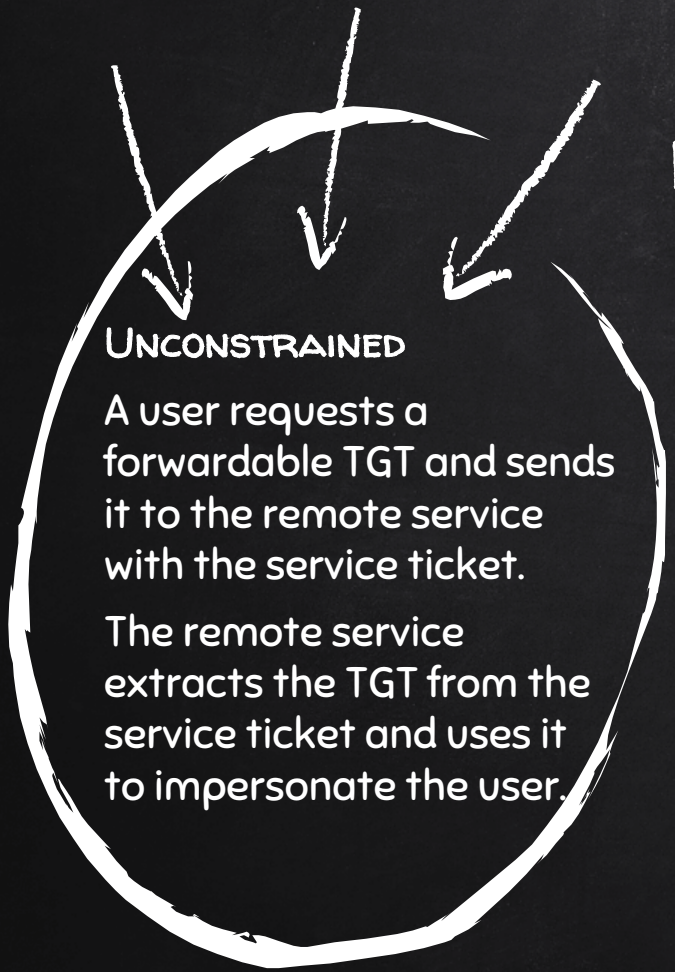
When You Need to be Someone Else

The Reason Delegation is Needed

User

Front End

Back End



Active Directory

Delegation Types

Unconstrained

A user requests a forwardable TGT and sends it to the remote service with the service ticket.

The remote service extracts the TGT from the service ticket and uses it to impersonate the user.

Traditional Constrained

The service requests a ticket to itself as another user (S4U2self)

The service uses this ticket to request a service ticket to another service as that user (S4U2proxy).

Service must be specified in msDS-AllowedToDelegateTo

Resource-based Constrained

ACL in a field (msDS-AllowedToActOnBehalfOfOtherIdentity) on the target *resource* that dictates who can perform S4U2proxy to the resource

Active Directory

Unconstrained Madness

- ✘ UNCONSTRAINED DELEGATION IS DANGEROUS!
- ✘ If an attacker can compromise a server with unconstrained delegation, they can obtain the TGT for any (non-protected) user who authenticates to that server
- ✘ In modern domains, only domain controllers are configured for unconstrained delegation by default
  - But we often see "misconfigurations" in the field :)


```
distinguishedname   : CN=PRIMARY,OU=Domain Controllers,DC=testlab,DC=local
useraccountcontrol  : SERVER_TRUST_ACCOUNT
                      TRUSTED_FOR_DELEGATION
samaccountname      : PRIMARY$
```

```
C:\WINDOWS\system32>whoami
testlab\da

C:\WINDOWS\system32>dir \\primary.testlab.local\C$
 Volume in drive \\primary.testlab.local\C$ has no label.
 Volume Serial Number is A48B-4D68

 Directory of \\primary.testlab.local\C$
```

```
[*] Action: List Kerberos Tickets (All Users)

  UserName                 : da
  Domain                   : TESTLAB
  LogonId                  : 0x1de4e6
  UserSID                  : S-1-5-21-883232822-274137685-41
  AuthenticationPackage    : Kerberos
  LogonType                : Network
  LogonTime                : 3/15/2019 9:54:13 PM
  LogonServer              :
  LogonServerDNSDomain     : TESTLAB.LOCAL
  UserPrincipalName        :

    [0] - 0x12 - aes256_cts_hmac_sha1
      Start/End/MaxRenew: 3/15/2[illegible] 2:54:13 PM ; 3/15/2019 7
      Server Name       : krbtgt/TESTLAB.LOCAL @ TESTLAB.LOC
      Client Name       : da @ TESTLAB.LOCAL
      Flags             : name_canonicalize, pre_authent, re
```

Domain Trusts: Crash Course

Active Directory

Trusts 101

- Trusts link up the authentication systems of two domains
  - This allows authentication traffic to flow between them

- This is done by each domain negotiating an "inter-realm trust key" that's used to encrypt Kerberos referral tickets

- Access is passed around with via these referrals and "inter-realm ticket granting tickets"

Active Directory

Trusts 201

- ✘ Trust directions/transitivity:
  - One-way – one domain trusts the other
  - Two-way – both domains trust each other (2x one-way trusts)
  - Transitive – A trusts B and B trusts C, so A trusts C

- ✘ The main trust type categories we care about:
  - Intra-forest – parent/child, cross-link (all transitive)
  - Inter-domain – forest (transitive), external (non-transitive)

Privilege Attribute Certificates (PAC)

- ✘ Recall: When you first authenticate, you receive a TGT
  - Inside each TGT is a PAC
  - A TGT's PAC contains the user/group SIDs that identify the user

Ticket Granting Ticket

Privilege Attribute Certificate (PAC)	
User	S-1-5-21-2532535433-4733566781-1284343941-1001
Groups	S-1-5-21-2532535433-4733566781-1284343941-1353 S-1-5-21-2532535433-4733566781-1284343941-2604
ExtraSids	S-1-5-21-3416895347-7456555532-9337766299-519

CORP\itadmin

CORP\FileShareAccess

CORP\HelpDesk

ACME\Enterprise Admins

SID Filtering

- ✘ During auth to another domain, the remote domain (the "trusting domain") analyzes the SIDs in the TGT's PAC
- ✘ Depending on the trust type, the remote domain removes ("filters") SIDs under various circumstances (see MS-PAC section 4.1.2.2)
- ✘ E.g. when authenticating from ForestA to ForestB, the PAC from ForestA should not contain SIDs for a default set of *privileged* groups in ForestB.
  - ✘ Other cross-domain/forest group memberships can be exploited

Intent of SID filtering

Stop a compromised *trusted* domain/forest from compromising a *trusting* domain/forest.

How well does this work in practice? Let's find out...

Why The Domain != A Security Boundary

ForestSpecific	The ForestSpecific rule is for those SIDs that are never allowed in a PAC that originates from out of the forest or from a domain that has been marked as QuarantinedWithinForest, unless it belongs to that domain. SIDs in this category is filtered out for QuarantinedWithinForest, CrossForest, External, and QuarantinedExternal trust boundaries.

S-1-5-21-<Domain>-519	Enterprise Admins	ForestSpecific*

Why The Domain != A Security Boundary

- ✘ The SID for "Enterprise Admins" is NOT filtered out by default for inter-realm tickets *if both domains are within the same forest*
  - So if you can set your sidHistory to be "Enterprise Admins" (i.e. ExtraSids in the PAC), you can escalate from a child domain to the forest root domain!

Forests == A Security Boundary? 🤔

- ✘ SID filtering of sensitive groups DOES protect across Forest boundaries
  - Hence, people have assumed that Forests were a security boundary :)

Forests as Security Boundaries

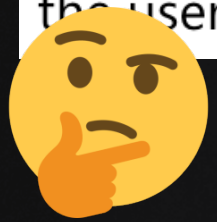
Each forest is a single instance of the directory, the top-level Active Directory container, and a security boundary for all objects that are located in the forest. This security boundary defines the scope of authority of the administrators. In general, a security boundary is defined by the top-level container for which no administrator external to the container can take control away from administrators within the container. As shown in the following figure, no

Active Directory

Sidenote: "Authenticated Users" in Referrals

The Problem of Authenticating Users from a Trusted Forest

When users authenticate from a trusted forest, they receive the Authenticated Users SID in their token. Many of the default rights for users in a forest are granted through the Authenticated Users SID. Because the Authenticated Users group is a computed group and its SID is added on the server to which the user authenticates, you cannot change the membership of the group.

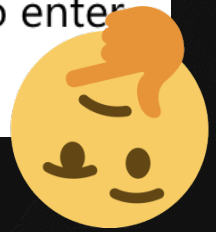


Active Directory

Sidenote: Delegation and Trusts

Enforcement for forest boundary for Kerberos full delegation

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.



So what? Let's review

- ✘ Delegated TGTs (like some TGTs found on unconstrained servers) are usable across Forests boundaries.
- ✘ A compromised unconstrained delegation server means an an attacker can extract TGTs of users who auth to that machine, *even if that user connects from another forest!*
- ✘ Hmm……can we coerce accounts to authenticate to an unconstrained delegation server?


YES WE CAN!

The "Printer Bug"

Our Final Ingredient

Active Directory

Printer Bug Overview

- ✘ Abuses the old enabled-by-default Print System Remote Protocol (MS-RPRN).
- ✘ RPC Methods: RpcRemoteFindFirstPrinterChangeNotification(Ex)
  - Purpose: "<ComputerA>, please send <ComputerB> a notification when ____ happens" (e.g. when there's a new print job)
  - When invoked, ComputerA will authenticate to ComputerB
- ✘ This a way to coerce authentication. There are others and likely more to come.

Active Directory

Reference: Printer Bug Details

- ✘ Print System Remote Protocol (MS-RPRN)
  - SMB-RPC (TCP 445)
  - Named Pipe: \pipe\spoolss
  - RPC UUID: 12345678-1234-ABCD-EF00-0123456789AB
  - Opnum 62 – RpcRemoteFindFirstPrinterChangeNotification
  - Opnum 65 – RpcRemoteFindFirstPrinterChangeNotificationEx
- ✘ The RPC server is accessible by "Authenticated Users" on Windows >= 8 if the Spooler service is started (Server & Workstations have it enabled by default).
  - Supposedly this will change in the future....
  - On Windows < 8 , seems possible if hosts have shared a printer.
  - Independently discovered by Elad Shamir (@elad_shamir)

Weaponization

- SpoolSample – https://github.com/leechristensen/SpoolSample

```
PS C:\> hostname
WIN10
PS C:\> whoami
corpwest\marketer
PS C:\> .\SpoolSample.exe LabDC01 WIN10
[+] Converted DLL to shellcode
[+] Executing RDI
[+] Calling exported function
TargetServer: \\LabDC01, CaptureServer: \\WIN10
RpcRemoteFindFirstPrinterChangeNotificationEx failed.Error Code 1722 - The RPC serve

PS C:\> .\Seatbelt.exe -q logonevents


=== 4624 Account Logon Events ===

TimeCreated,TargetUser,LogonType,IpAddress,SubjectUsername,AuthenticationPackageName
3/17/2019 9:26:04 PM [redacted]\-,Kerberos,,
3/17/2019 9:26:04 PM CORPWEST.LOCAL\LABDC01$,Network,192.168.230.100 -\-,Kerberos,,
```

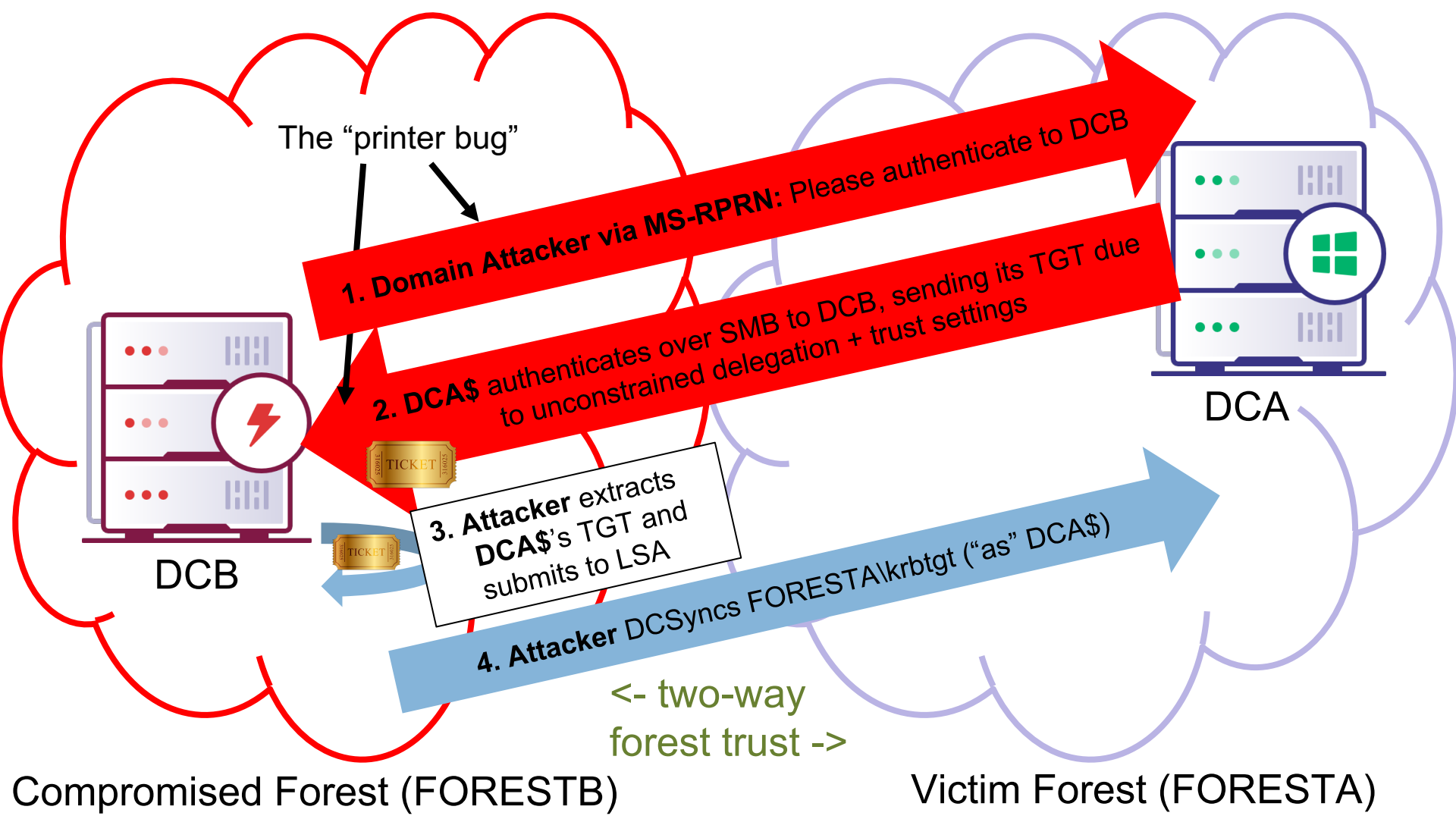
LABDC01$ authenticated to WIN10

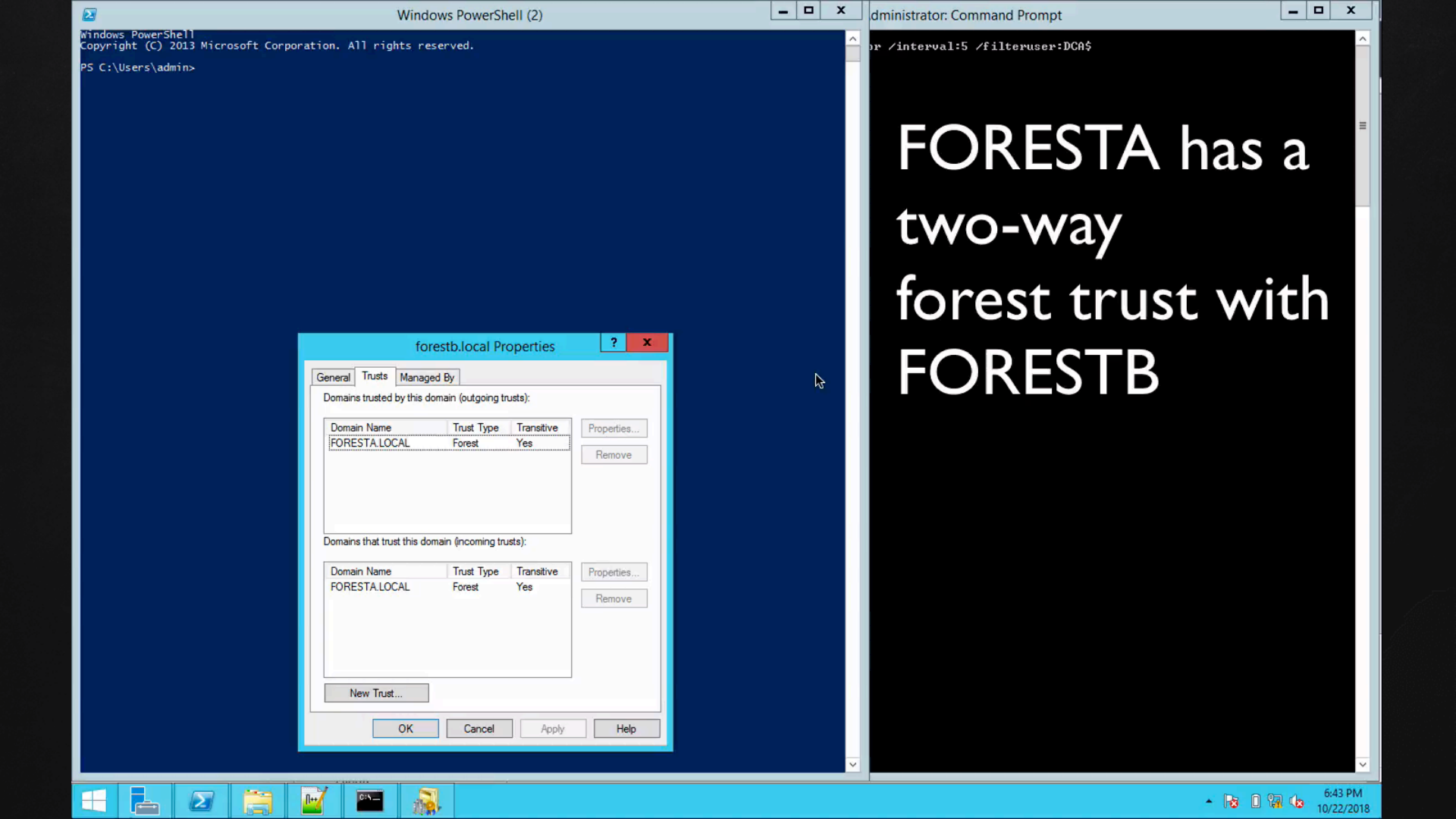
Breaking Forest Trusts

Smash Smash Smash

Scenario

- ✘ An attacker completely compromises ForestB
  - This includes ForestB's DC with unconstrained delegation ;)
- ✘ ForestB shares a two way *forest* trust with ForestA
- ✘ Tools Used:
  - Rubeus — TGT monitoring/extraction
  - SpoolSample — coerced authentication (the "printer bug")
  - Mimikatz — DCSync \m/


The “printer bug”
1. Domain Attacker via MS-RPRN: Please authenticate to DCB
2. DCA$ authenticates over SMB to DCB, sending its TGT due to unconstrained delegation + trust settings
DCA
DCB
TICKET
TICKET
3. Attacker extracts DCA$’s TGT and submits to LSA
4. Attacker DCSyncs FORESTA\krbtgt (“as” DCA$)
<- two-way forest trust ->
Compromised Forest (FORESTB)
Victim Forest (FORESTA)


Windows PowerShell (2)
Windows PowerShell
Copyright (C) 2013 Microsoft Corporation. All rights reserved.
PS C:\Users\admin>
forestb.local Properties
General
Trusts
Managed By
Domains trusted by this domain (outgoing trusts):
Domain Name
Trust Type
Transitive
FORESTA.LOCAL
Forest
Yes
Properties...
Remove
Domains that trust this domain (incoming trusts):
Domain Name
Trust Type
Transitive
FORESTA.LOCAL
Forest
Yes
Properties...
Remove
New Trust...
OK
Cancel
Apply
Help
dministrator: Command Prompt
or /interval:5 /filteruser:DCA$
FORESTA has a two-way forest trust with FORESTB
6:43 PM
10/22/2018

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!*

Public Reaction? (mostly good, but...)

"Still a security boundary as long as it is not a two way trust AD forest/domain"

"I just don't think I've heard anyone claim a boundary still exists when a 2-way trust is in place."

"Step 1: Have forest root domain admin credentials.

Step 2: Have things be grossly misconfigured."

Active Directory

Why This Matters

- ✘ This attack works with default, modern configurations for Active Directory forests as long as a two-way forest trust is in place.

- ✘ The security of ForestA is now completely dependent on the security of ForestB (think acquisitions…)
  - Even if ForestA has *near perfect* security it can be completely compromised by the takeover of a single unconstrained delegation server in ForestB!

Active Directory

Microsoft: A Great Example

- ✘ We reported this to MSRC in the fall of last year
  - The associated Microsoft engineering teams determined that it wasn't a vulnerability they would patch, but it would be something they might harden in the future (i.e. v.Next)
- ✘ After we published details (and defensive guidance) they decided this decision was a mistake, and soon released an advisory (and eventually a patch!)
- ✘ We applaud Microsoft/MSRC at admitting their error and handling the resulting situation in the best way possible!

CVE-2019-0683 | Active Directory Elevation of Privilege Vulnerability

Security Vulnerability

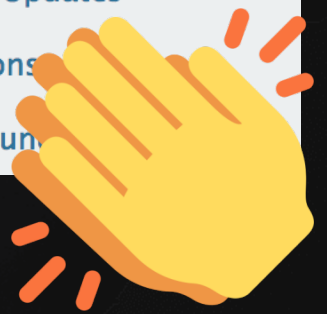
Published: 03/12/2019
MITRE CVE-2019-0683

An elevation of privilege vulnerability exists in Active Directory Forest trusts due to a default setting that lets an attacker in the trusting forest request delegation of a TGT for an identity from the trusted forest. To exploit this vulnerability, an attacker would first need to compromise an Active Directory forest.

An attacker who successfully exploited this vulnerability could request delegation of a TGT for an identity from the trusted forest.

This update addresses the vulnerability by ensuring Active Directory Forest trusts disable TGT delegation by default.

On this page



Defenses

Preventing and Detection

Active Directory

Selective Authentication

- ✘ "*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.*"

- ✘ However, this is focused on administrative *users*
  - Domain controller objects often need the "Allowed to authenticate" right on foreign domain controllers in order for the system to work correctly

Disabling Kerberos Full Delegation Across Trusts

- ✘ To prevent ForestA from accepting delegated TGTs from ForestB:
  - netdom trust foresta.local /domain:forestb.local /EnableTGTDelegation:no
- ✘ This flips the TRUST_ATTRIBUTE_CROSS_ORGANIZATION_NO_TGT_DELEGATION bit to prevent delegated TGTs from transiting the forest boundary
- ✘ *This has to be done on each end of the trust(s)!*

CVE-2019-0683 Details

- ✘ As mentioned, Microsoft recently recognized this issue as CVE-2019-0683
- ✘ A patch is being slowly rolled out (see next slide) that disables TGT delegation across forest trust boundaries by default
- ✘ On the roadmap (July 2019)
  - "*...adding a new safe default configuration for unconstrained Kerberos delegation across Active Directory forest trusts.*"

Active Directory

CVE-2019-0683 Timeline

- ✘ March 12, 2019
  - Kerberos full delegation block is backported to Server 2008[R2]
- ✘ May 14, 2019
  - A new trust flag will be introduced in case you need full delegation across trusts
  - "EnableTGTDelegation" set to "no" for all new trusts
- ✘ July 9, 2019
  - Start of enforcement of new trust flag
  - "EnableTGTDelegation" ignored from this point forward

THANKS!

Any questions?

[will | lee]@specterops.io

@harmj0y | @tifkin_

Active Directory

References

- ✘ Breaking Forests Trusts (Red) – https://bit.ly/2Ck1HlW
- ✘ Breaking Forests Trusts (Blue) – https://bit.ly/2Y2Etd5
- ✘ Attack Demo Video – https://bit.ly/2ULitRW
- ✘ Microsoft Advisory – https://bit.ly/2ObHAv2
- ✘ Microsoft Updates Timeline – https://bit.ly/2ugBZdM
- ✘ CVE-2019-0683 details – https://bit.ly/2CopVLR