Kerberos and Authentication

Fabrizio Grossi
FabrizioG@icubed.onmicrosoft.com
Fabrizio.grossi@stcon.eu

Overview

- Kerberos Overview
- How Kerberos Authenticates: Tickets to Paradise
- Making Sure You Use Kerberos… Not NTLM
- Cranking Up Kerberos: Encryption Methods
- Watching Kerberos
- Keeping Kerberos Trim: Dealing with Token Bloat
- Understanding Kerberos Delegation
- SPN City: Kerberos's Service Names

Why Are We Here at 8:30 AM?

- Most of the time, Kerberos is "just there"
- So why this talk? Three reasons
  - There are some "edge conditions" that it can run up against that can be confounding to troubleshoot
  - Some networking arrangements need a bit of special tweaking
  - Windows 6/7 offer us some new options in shoring up our authentication and heck, if you've already paid for it, then why not use those tools?

KERBEROS OVERVIEW

Kerberos Overview

- It's an authentication protocol – so it's not involved with authorization
- Standards-based
- Used for authentication in AD forests

It's a Matchmaker

- Kerberos sees users (which are usually the client) as UPNs and services as SPNs
- Your AD logon name – the one that looks like an email address (e.g., mark@bigfirm.com) – is your UPN
- SPNs are a mite uglier, and I've got a section on them later
- Kerberos "introduces" UPNs to SPNs by giving a UPN a "ticket" to the SPN's service

HOW KERBEROS AUTHENTICATES

Authentication Overview/Review

Tom

Tom wants to talk to his local print server, \\PS:

To do that, Tom needs a bit of data called a *service ticket* to PS

ADMIT ONE ST-PS

Tom gets that service ticket by asking his local Key Distribution Center (KDC) for it

(You and I would call the KDC an AD domain controller)

Kerberos In Pictures

to accomplish that…

KDC

Tom needs something that gives him the right to talk to those servers

Tom's DCs create both kinds of tickets

That "something" is called a *ticket*; there are two kinds

ADMIT ONE

ADMIT ONE ST-TMS

ADMIT ONE ST-PS

ADMIT ONE TGT

Service tickets get Tom access to services, like the "workstation" service on TOMSPC, or the print server service on PS

Ticket Granting Tickets give Tom the right to ask the DC to issue him service tickets

Slide 9

How Tickets Work

- Goal: a KDC first generates a random cryptographic key and then it gets that key to the user and the service
- Why? Well, once they've got that key, they have a shared secret and can
  - authenticate
  - (optionally) encrypt/sign communications
- Key is short-lived, ten hours in Windows

The Sequence: First, a TGT

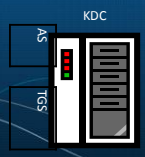
- First, Tom authenticates himself to the KDC, using his (Tom's) password hash as a crypto key
- Once he's proved who he is, the AS creates a key that Tom can use to talk to the KDC (for just ten hours) – let's call it "Tom's 'today' password" or "Tom's 'today' key"
- The KDC wraps this new temporary key into a ticket called the "ticket-granting ticket"
- This TGT is created by something called the Authentication Service or AS… time for a sidebar

Two Keys, Two Services

- Initial logon:
  - Get a ticket-granting ticket (TGT) from the "Authentication Service" (AS)
- Afterwards, when you want access to a service, ask the Ticket Granting Service (TGS) for a Service Ticket
- Use the TGT to re-authenticate yourself to the TGS

How Does This Fit In a DC?

- Key Distribution Center = Authentication Service + Ticket Granting Service
- KDC=AS+TGS
- The role of KDC, AS, TGS are just part of what an AD DC does
- You can't, however, see AS vs TGS etc in Task Manager; it's all in LSASS
- Back to the sequence…

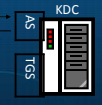


Slide 13

Kerberos In Pictures

First, Tom needs a Ticket Granting Ticket

Tom requests a TGT from the AS and proves he's Tom by encrypting the date as "YYYYMMDDHHMMSSZ" using Tom's password as the encryption key (this is called the "preauthenticator")



The AS decrypts the time/date stamp with Tom's password and checks that it's close to the actual time; if it is, then the AS gives Tom a TGT

(Oh, and how close is "close" vis-à-vis time/date as far as AS is concerned? You get to configure that, but the default is 5 minutes.)

Slide 14

So Far…

- Tom has pressed ctrl-alt-del, punched in a name and password, and gotten a TGT
- But he's still not logged onto PS
- He'll get print server
- So that's the next thing he needs to ask for, from the Ticket Granting Service

Slide 15

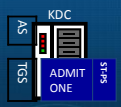
Tell Me Again… *Why* Two Tickets?

- Answer: to protect Tom's password.
- The original logon preauthenticator was encrypted with the user's password, which doesn't change much and so shouldn't be exposed unless necessary
- The TGT gives the user a "password for the day"

Kerberos in Pictures

next, Tom gets a Service Ticket to his PC

Tom shows the TGS his TGT and asks for an ST to PS's print server service



The TGS validates the TGT and creates a Service Ticket identifying Tom to PS; Tom then presents that to PS

PS

This identifies – "authenticates" – Tom to PS; PS then looks at Tom's personal and group SIDs and, if there's a permissions match, okays Tom.

Now Tom is logged onto his local print server!

Slide 17

So Far…

- Tom's logged onto his workstation
- He's got a TGT that he'll use to request Service Tickets from the Ticket Granting Service
- He's got a Service Ticket to his workstation
- These "tickets" are really just data in TOMSPC's RAM
- Tom can see them with klist, from the 2003 Resource Kit
- Klist is now built into Server 2008 and 2008 R2

Slide 18

Sample klist Output: TGT

```
Client: TOM @ BIGFIRM.COM
Server: krbtgt/BIGFIRM.COM @ BIGFIRM.COM
KerbTicket Encryption Type: AES-256-CTS-HMAC-
  SHA1-96
Ticket Flags 0x60a00000 -> forwardable
  forwarded renewable pre_authent
Start Time: 5/11/2009 13:16:53 (local)
End Time:   5/11/2009 23:16:49 (local)
Renew Time: 5/18/2009 13:16:49 (local)
Session Key Type: AES-256-CTS-HMAC-SHA1-96
```

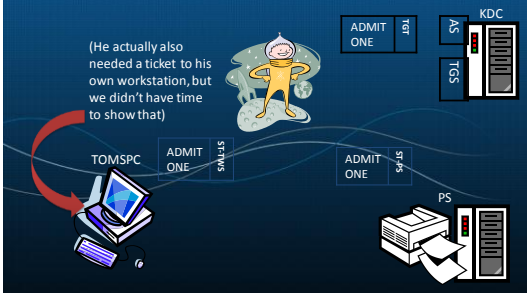
Slide 19

Sample klist Output: Service Ticket

```
Client: TOM @ BIGFIRM.COM
Server: cifs/PS.bigfirm.com @ BIGFIRM.COM
KerbTicket Encryption Type: AES-256-CTS-HMAC-
  SHA1-96
Ticket Flags 0x40a00000 -> forwardable
  renewable pre_authent
Start Time: 5/11/2009 13:32:59 (local)
End Time:   5/11/2009 23:16:49 (local)
Renew Time: 5/18/2009 13:16:49 (local)
Session Key Type: AES-256-CTS-HMAC-SHA1-96
```

Slide 20

Tom and His Tickets
(the more complete list)



Slide 21

What's in a Ticket, Anyway?

- Tickets contain
  - time start
  - total lifetime
  - your security token
  - optional IP address info

Securing Tickets: the Keys

- Kerberos's crypto depends on certain keys for its strength:
  - The user's password is the key for the authentication to the AS
  - An internal account called "krbtgt" has a password that the KDC uses to ensure that you didn't fake a TGT
  - When the KDC generates a service ticket, it encrypts it using the password of the service (the password of PS's AD machine account, in Tom's case)
- Remember those… we'll use them again

KERBEROS: ENCRYPTION METHODS

Kerberos and Crypto

- Needs good encryption algorithm; original used DES, which was nice years ago but sorta weak now
- Initial AD created in a regime where exporting 57+ bit encryption was unlawful
- Result: RC4-HMAC, RFC 4757
- Good dev info at http://blogs.technet.com/authentication/
- If we're going to the trouble to ensure that we're using Kerberos, we may as well control the quality of encryption that we use

Windows 6/7 Crypto Changes

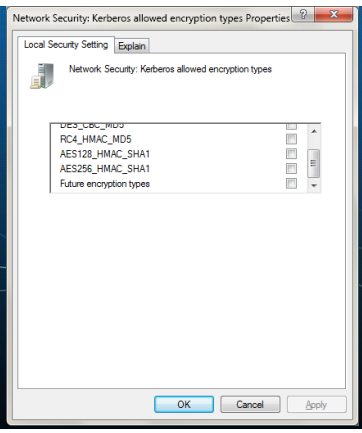
- Windows Vista/2008 adds AES
  - Needs 2008 DFL
  - Happens automatically upon shift to 2008 DFL
  - Server 2003 R2 and earlier DCs cannot employ AES
  - Side-effect: win 6/7 systems will always fail in their first logon attempt (as they "forget" to include a pre-authenticator), that is *normal*!

Supported Kerberos Crypto

- DES/CRC and DES/MD5
  - For compatibility with non-Windows systems only, never used in an AD
- RC4 HMAC/MD5
  - Crypto used in all pre-Vista AD members and DCs
- AES128/AES256 and SHA1
  - Used between Vista+ clients, 2008+ DCs
- At some point – perhaps now for some organizations – you'll want to force a minimum level of crypto

Windows 7 / 2008 R2 Group Policies

- Can restrict crypto types in Kerberos through group policy setting that only Win 7 and R2 can use
- In Security Options, "Network Security: Configure encryption types allowed for Kerberos"
- You can either set this on clients, servers, or both



Using a Trace or the Logs
common Kerberos issues

- AD troubleshooting rule #1:
  - It's probably DNS.
- AD troubleshooting rule #2:
  - If rule 1 doesn't apply, then it's probably DNS.
- The time service really, really, really has to be working, or Kerberos will fail
- If it's a three-tier sort of authentication, like a Web application with a SQL back-end, then understand delegation and SPNs, which we're going to get to soon
- Traces can help with all of that… use them!

Forcing Kerberos to Use TCP

speaking of stuff you find when sniffing a trace…

- Kerberos chooses either TCP *or* UDP
- Kerberos UDP and VPNs don't mix well
- In W2K, XP, 2003 then Kerberos goes UDP if the packet is < 1465 (2003) or < 2000 bytes (XP, 2000)
- Answer: set the minimum to 1 byte, so it's always TCP – see KB 244474, *but…*
- Win2K systems may need patch at KB 320903
- What about Vista and later? They always use TCP

Kerberos Logs

- Most Kerberos-related event log entries are actually pretty clear – some in System, some in Security
- For more, look to "Troubleshooting Kerberos Errors" white paper – search the Web for "Troubleshooting Kerberos Errors in Windows" to find the download from the Microsoft site

Extra Logging

- You can set Kerberos to log its activity to the System log
- Go to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Parameters
- Add new REG_DWORD value LogLevel, set to 1
- Reboot for it to take effect
- Details in Kerb Troubleshooting white paper

Slide 33

Creating Kerberos Logs

- In Parameters, REG_DWORD LogToFile =1 to create an ASCII log file
- KerbDebugLevel (REG_DWORD) sets the verbosity of the log
- Look to the white paper for specific error code-to-English translation

Slide 34

MAKING SURE YOU USE KERBEROS… NOT NTLM

Kerb's Not the Only Auth Protocol

- You probably know that pre-Windows 2000 systems only used LM, NTLM, NTLMv2 authentication protocols
- They were good for their time, but in modern networks they're attack targets
- What many don't know is that they're surprisingly persistent
- Part of the 2009 IT pro's job is to eradicate them as much as is possible

Why Kerberos Rather than NTLM?

- Stronger encryption than NTLM/LM auth algorithms
- Mutual authentication
- Time stamps, signing, man-in-the-middle much more difficult
- NTLM is a 4LA, Kerberos is an 8LA
- Exposes user hash much less frequently than NTLM does

What? *When* Do I Not Use Kerb?

- Even in an AD-centric network, you may not get Kerberos
  - NET USE to an IP address
  - Connect to a workgroup system on Windows of any version
  - Connect to a pre-2000 system
  - Failover from a busy DC
  - Badly-written apps
  - Intranet site not added to "local intranet" zone

Kerberos Logon vs NTLM Logon

- How you know you're NTLM-ing:
  - Can't join machines to domains
  - Don't get group policies
  - Netmon traces show NTLM, not Kerberos traffic
  - Klist shows no tickets, yet you're logged in
- Tracking this stuff down by hand is a pain, so Windows 7 and Server 2008 R2 offer some new group policies

NTLM Restriction Policies

- Essentially these new policies let you first track and then block NTLM logons
- There are basically three policies, each with an "audit" and a "block" option:
  - Incoming NTLM traffic (server tracking)
  - Outgoing NTLM traffic (client tracking)
  - Domain traffic (DC tracking)
- They create new event log entries of source "NTLM," numbers 8001, 8002, 8003, 8004

NTLM Restrictions

- In Computer Config / Windows Settings / Security Settings / Local Policies / Security Options
- All start with "Network security: Restrict NTLM:
- Log entries go to the log in Applications and Services Log / Microsoft / Windows / NTLM
- Some sources say that the log will be named "NTLMBlock" in the final release
- These only work on Win 7 and 2008 R2

"Incoming NTLM Traffic"

- Systems acting as servers can audit/block NTLM logons
  - from accounts in the local domain
  - from accounts in any domain in the forest(s)
- Two policy settings:
  - Audit incoming NTLM Traffic (audits only)
  - Incoming NTLM Traffic (blocks and logs)

"NTLM authentication in this domain"

- Tracks only intra-domain logins, options
  - domain accounts to domain servers
  - domain accounts
  - domain servers
  - all
- Again, two policies, one that just audits, another that blocks: "Audit NAITD" and "NAITD"
- Can exempt servers with "Add server exceptions for NTLM authentication in this domain"

"Outgoing NTLM traffic to remote servers"

- Attempts by client software on this computer to use NTLM logons to other systems audited/tracked
- One policy does both audit and block; options
  - "allow all"
  - "audit all"
  - "deny all"
- Can exempt servers with "Add remote server exceptions for NTLM authentication"

Simple Illustrative Example

- Set up an R2 DC that is also a file server
- Create group policies to track NTLM activity
- Join a Windows 7 system as a domain member
- Log onto the member with a *local* account
- NET USE to the share with the /u: option to present domain credentials

Result

- On the server, you'll see events 8002, NTLM incoming traffic that would be blocked
- On the client, you'll see events 8001, NTLM outgoing traffic that would be blocked
- Another example:
  - Log into the W7 box as a domain member
  - NET USE to the file server using \\ followed by the IP address

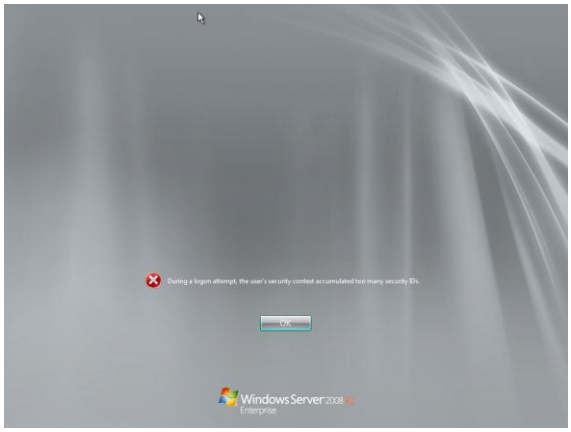
KEEPING KERBEROS TRIM: DEALING WITH TOKEN BLOAT

What is Token Bloat?

- Kerberos tickets only set aside a certain amount of space for their data
- Part of what goes in that data is your token
- If the token's too large, it can't fit, and the Kerberos logon fails, and at that point either you're not logged on (in which case you notice the problem) or you log on under NTLM (in which case you don't notice a problem until you note that there are a few important things you can't do)
- KB 275266 discusses one very clear case

Token Bloat Symptoms

- Logon failure dialog box referring to "too many SIDs"
- Logged on via NTLM, not Kerberos (as discussed previously)
- Event Kerberos ID 6 (some process has failed due to oversize token)
- Event LSASRV ID 6035 (DCs can't replicate) [bad one to see]
- Event "Directory Service," ID 1308: KCC failure [ditto]



What Causes Bloat

- Lots of group memberships
- (Nested memberships count!)
- SID histories
- About 1024 SIDs in a token is the limit

Testing for Bloat: NTDSUTIL

- Tool to help enumerate all group memberships:
  - ntdsutil
  - group membership evaluation
  - run bigfirm.com mark
- Produces a tab-delimited file
- Lists groups and nested groups
- Look at WhenChanged parm first, as it's the most recent changes; also look at large "Depth From User" to smoke out heavy nesting
- Look also at group owner and groups that have changed from distribution to security group

Example Group Evaluation

- We have a
  - domain: bigfirm.com
  - user: mark
  - Universal group named "Really Great Guys"
  - Contains global group named "Great Guys"
  - Mark's in "Great Guys"
- Run evaluation in ntdsutil
  - group membership evaluation
  - run bigfirm.com mark

Fixing It

- Basically remove groups and/or SID histories
- May have to boot in Safe Mode to get to the accounts
- Ditto DCs that have been locked out because they're a member of too many groups (ntdsutil works for DCs as well)

Want To See the Blight of Bloat?

- You can, with the BloatMaster 5000™
- Open Notepad, type this in, save as "makebloat.cmd:"

```
net user joe Big$Pw22 /add
for /l %%a in (1 1 1200) do (net group
  junkgroup%%a /add & net group
  junkgroup%%a joe /add)
```

- Run this on a DC
- You'll now have a domain user named "Joe" who's bloated (and you'll have 1200 new global groups), and who can *not* log in

Cleanup: "The Anorexiomatic"

- Don't want to keep those groups?
- Run this from an elevated command prompt on a domain controller:
- `for /l %a in (1 1 1200) do (net group junkgroup%a /delete)`
- (This assumes that you don't already have global groups with names like "junkgroup47" in your domain)

Enlarging TOken

- kb 263693

Testing for Bloat: tokensz

- Get at www.microsoft.com/downloads
- Calculates token size for a user account
- Notice that this means it'll tell you the size of your token correctly even if you've just joined a group and haven't logged off/on
- Basic syntax:
- tokensz /compute_tokensize
- add /user:*username* to compute user other than the currently-logged-in one
- (Blows up in some oversize token cases)

Example Tokensz Run

- tokensz /compute_tokensize /user:mark

```
Name: Kerberos Comment: Microsoft Kerberos V1.0
Current PackageInfo->MaxToken: 12000
QueryKeyInfo:
Signature algorithm = HMAC-SHA1-96
Encrypt algorithm = Kerberos AES256-CTS-HMAC-SHA1-96
KeySize = 256
Flags = 2083e
Signature Algorithm = 16
Encrypt Algorithm = 18
  Start:5/6/2009 2:05:25
 Expiry:5/6/2009 12:05:25
Current Time: 5/6/2009 2:05:25
MaxToken (complete context) 1387
```

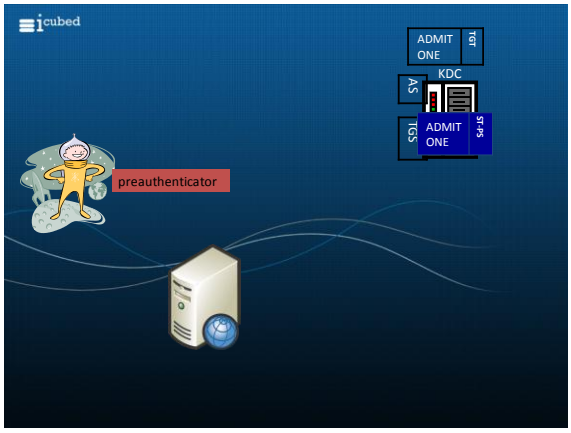
Token Size Formula (Reference)

- This is what's inside tokensz, basically
- Token size = ticket overhead (estimate at 1200 bytes) + 40 x (# domain local users you're a member of + #universal group memberships in UGs outside your domain + #groups in your SID history) + 8 x (# global groups you belong to + #universal groups you belong to)

KERBEROS DELEGATION

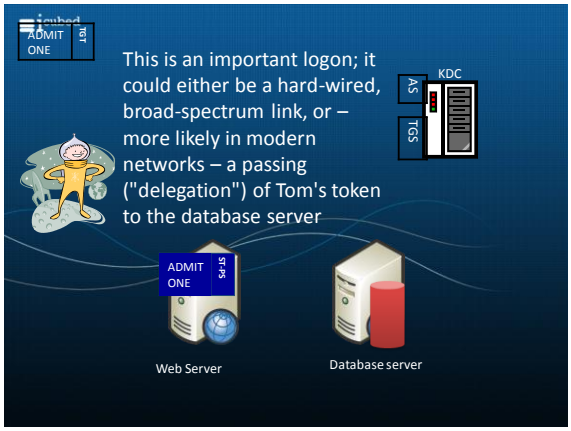
What's Kerberos Delegation?

- We've talked about a three-player model before
- If Tom wants to access a domain-secured Web server, it'll look like this:



Now Let's Add a Fourth Player

- What if the Web application we ran needed data to respond to our requests, data that lived on a database server?
- Well, when the Web server queries the database server, what credentials does it offer?



Delegation Specifics

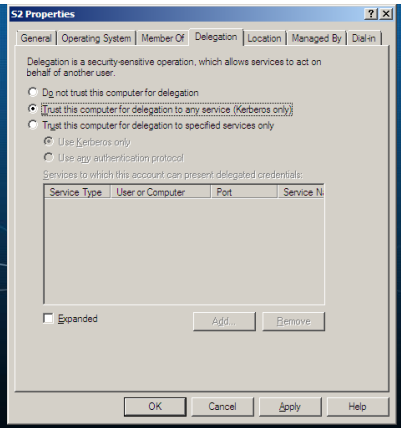
- How's it done? Depends on the app
- Why not use a hard-wired "LocalSystem to LocalSystem" connection?
- Simple… least privilege
- But what about when we want to take things further?
- That's where *constrained* delegation takes place

Definitions

- Tom hands his ST to the service running the Web server
- The Web server then needs to present credentials to the database server
- … why is it okay?
- Because Tom handed the Web server a ticket that could be *delegated*
- Think of the word in the Exchange context and it'll make more sense
- There's just one problem…

It won't work.

You see, this whole idea that some server service could grab one of your tickets and run around town ~~shopping with your credit card~~ doing things in your name is downright frightening, so it's disabled by default.



Delegation and Win2x

- Windows 2003 offered more granularity in controlling delegation
- Called "constrained delegation," it let you say, "you can only pass my token along to service type X in domain Y, etc"
- Again, usually set up by the installing app… or you may have to work with the devs to hand-craft a delegation
- But if we're going to start naming services that we trust, it's time to turn to SPNs…

SPN CITY: KERBEROS'S SERVICE NAMES

Service Principal Names

- Kerberos needs an account for every instance of every service running in the domain – particular services running on particular servers, application pools in IIS and the like
- Remember that Kerberos encrypts service tickets to a given service with that service's password hash
- In most cases, that means the machine account's password – but sometimes we want to run a service under a user account, and then things get interesting

Service Principal Names

- Because you might have some services on a given server running under the context of that server's machine account and other services running under some other set of AD accounts, Kerberos includes a sort of label for the service that abstracts the service from the machine
- It's also useful for services that offer many equally-good systems, like picking a DC
- The label is called a Service Principal Name or SPN

A Simple Example

- In a Kerberized application, a client (UPN, "mark@bigfirm.com") requests a ticket to a given SPN, like TERMSRV/R2S1.bigfirm.com
- In this simple example, Kerberos must find the user account corresponding to the UPN and the machine corresponding to the SPN
- (It's obvious in this case, but not in all)
- Kerberos looks up the UPN and SPNs in the global catalog

Service Principal Names structure

- SPNs look like
- **ldap/R2S2.bigfirm.com:6000/bigfirm.com**
- <type of service>/ <hostname> [:<port>] [/servicename]
- Type of service = description of the server, like "CIFS" for file server, "LDAP" for DC, etc
- Hostname = obvious, but for two things:
  - As we'll see, it's a good idea to create SPNs using both the host name and the FQDN
  - "Hostname" really means host name – CNAMES will not work (which can lead to some Sharepoint problems)

Service Principal Names structure

- Port's obvious, only include if non-standard
- "Servicename" is either
  - DNS service name of a "replicable" service – like a replicated SQL database or an AD instance – available throughout the domain; represented as the domain's name, like "bigfirm.com"
  - A DN/GUID in AD, like cn=mysvc,dc=bigfirm,dc=com
  - The DNS name of an SRV or MX record

Service Principal Names structure

- Everyone has a "host" spn:
  - HOST/CLASSVISTA
  - HOST/Classvista.bigfirm.com
- DCs have LDAP spns:
  - ldap/R2S2/BIGFIRM
  - ldap/8a53de22-291a-44b9-b112-4fa5d87b16a9._msdcs.bigfirm.com
  - ldap/R2S2.bigfirm.com/BIGFIRM
  - ldap/R2S2
  - ldap/R2S2.bigfirm.com
  - ldap/R2S2.bigfirm.com/bigfirm.com

Service Principals examples

- DCs also have FRS or DFS-R spns:
  - Dfsr-12F9A27C-BF97-4787-9364-D31B6C55EB04/R2S1.bigfirm.com
- Anyone with Remote Desktop/Services:
  - TERMSRV/R2S1
  - TERMSRV/R2S1.bigfirm.com

SPN Lookup

- A given instance of a service has an SPN identifier
- The account that holds that SPN stores it in AD in the servicePrincipalName attribute
- servicePrincipalName can hold any number of SPNs
- Thus, given an SPN, AD must search those values across the forest to find the user/machine account associated with the SPN – the GCs keep a list

Why You Care

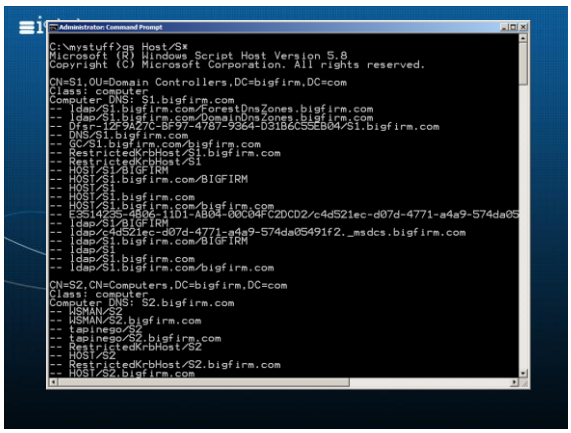
- Reason #1: if Kerberos looks up a SPN and finds that more than one account contains that SPN, the music stops
- Reason #2: if Kerberos looks up a SPN and no account contains that SPN, then again the melody ceases
- So we need some SPN tools

Finding SPNs: Listing on a system

- Find all SPNs associated with an account with "–L"
- setspn –l s2
- Use the hostname, not the FQDN
- As it's just pointed at one account, it's just listing the value of servicePrincipalName in the AD account

Finding SPNs: Querying AD

- spn_query.vbs queries GC for a SPN
- Find it on Technet – search on "spn_query" and "Craig Wiand"
- Takes wild cards, so the searching is easier



Adding a SPN

- setpsn –a lets you add a new SPN to a given AD account
- To make that work, you've got to supply
  - Service name (HTTP, CIFS, etc)
  - The name of the computer that the service resides upon
  - The port used to access the service

Adding a SPN

- Often done in secured Web apps (i.e. Sharepoint) or database servers
- Usually there's some automatic help in creating and installing SPNs
- But some home-grown systems may require a SPN
- Install one with –a
- My favorite way to screw it up: accidentally put a given SPN on more than one system

setspn -a

- To assert that there's a service called "prophecy" running on port 20201 on the physical system s1.bigfirm.com BUT that the service runs under a user account named "joe" in bigfirm.com, we'd type
- setspn –a prophecy/s1.bigfirm.com:20201 joe
- IMPORTANT: you should always also register just the hostname
- setspn –a prophecy/S1:20201 joe

SPNs and 2008

- Small but important change to -a option on setspn
- In Server 2008 and later, setspn first searches AD to see if a proposed SPN already exists
- (Yay!)
- And if you hate managing SPNs a lot, take a look at 2008 R2's Managed Service Accounts (MSAs)

Managed Service Accounts
background: what problem does this solve?

- Services must run under an account, and LocalSystem/LocalService/NetworkService can't always do the job
- IIS, Exchange, SQL are some common examples
- In that case, techies need to create accounts to act as service accounts
- That works fine, except for the issue of passwords: they need regular changing or services stop working

Managed Service Accounts
background: what problem does this solve?

- Basically, it's a pain to manage passwords for the user accounts that we happen to use for services
- Also, introducing new user accounts into services means having to develop expertise with setspn
- Additionally, you've got to be a domain admin to modify SPNs… MSAs let you delegate this to others

Managed Service Accounts
answer: managed service accounts

- New class of accounts
- Sorta user accounts, sorta machine accounts (new icon)
- Need one account / member

Managed Service Accounts
installation steps (overview)

- Create one on the domain
- "Install" it on the member server
- Configure the svchost or the IIS application pool so that it logs on as that account, and from there password updates etc are automatic

91

Managed Service Accounts
password details

- 240-character passwords created
- MSAs ignore group policies about passwords and ignore fine-grained password policies
- Automatically handle password changes every 30 days

92

Managed Service Accounts
SPN management

- As mentioned, you can control who can administer SPNs rather than needing to be a domain admin
- If you rename a machine account, the SPN gets fixed automatically
- If you change a DNS host name, the SPN gets fixed automatically
- SPN management requires 2008 R2 DFL

93

Managed Service Accounts
requirements/details

- Requires at least one 2008 R2 DC (which means a 2008 R2 schema on the forest)
- Requires AD Powershell (and therefore AD Web Service) to create accounts
- Live in their own new folder (not an OU) called "Managed Service Accounts"
- Servers hosting services that use the accounts must be R2/Win 7
- Need R2 DFL to get the automatic SPN management

94

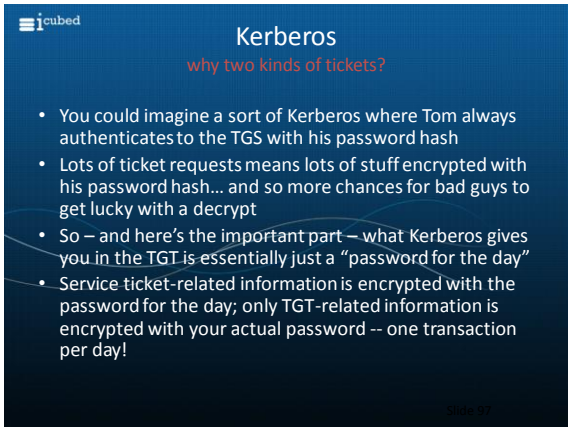
Two Tickets, Two Services

- First you introduce yourself to the KDC by logging on; you only want to have to do this once a day and so you ask the KDC for a "ticket to the KDC"… that's the Ticket-Granting Ticket
- That is granted by a *piece* of the KDC called the "Authentication Service" or AS
- Once you've got a TGT, then you can show the TGT to the KDC and say "remember me? Now I need a Service Ticket to such-and-such service"
- Service tickets are issued by a *different* part of the KDC called the Ticket Granting Service or TGS

Slide 95

The Sequence: Next, a Service Tkt

- Now that Tom's got a "ticket to the KDC," he uses it to re-introduce himself to the KDC, but this time, to the Ticket Granting Service (TGS) rather than the AS
- Tom authenticated himself to the KDC the first time with something encrypted using his password hash as the key
- This time, Tom again authenticates himself to ask for a service ticket, but this time he'll use something encrypted with his "today" key


icubed
Kerberos
why two kinds of tickets?
• You could imagine a sort of Kerberos where Tom always authenticates to the TGS with his password hash
• Lots of ticket requests means lots of stuff encrypted with his password hash… and so more chances for bad guys to get lucky with a decrypt
• So – and here's the important part – what Kerberos gives you in the TGT is essentially just a "password for the day"
• Service ticket-related information is encrypted with the password for the day; only TGT-related information is encrypted with your actual password -- one transaction per day!
Slide 37