Realtime publishers

The Definitive Guide™ To

Active Directory Troubleshooting, Auditing, and Best Practices

2011 Edition

Don Jones

Copyright Statement

Chapter 8: Assorted Tips and Tricks for Active Directory Troubleshooting

We're at the end of this guide, and I find myself left with several things I wish I'd mentioned earlier—except that these things don't fit neatly into any of the topics we've already discussed. So in this chapter, I'll present these seemingly-random, yet completely-helpful, tips for troubleshooting various aspects of Active Directory (AD).

Troubleshooting FSMO Roles

Typically, there's no good "fix" for a broken Flexible Single Master Operation (FSMO) role—you're often left to nicely transfer the role to another domain controller or, in a worst-case scenario, seize the role on another domain controller. There are, however, some indications that tell you a FSMO role holder isn't working properly:

- If you can't add new domains, the Domain Naming Master is down. That FSMO can be down for ages without you realizing it because you probably don't often add domains.
- If users are changing their passwords but can't log on, the PDC Emulator is the likely cause. This FSMO role also plays a part in time synchronization.
- Failure of the PDC Emulator can also affect your ability to edit Group Policy Objects (GPOs) and prevent you from adding new domains to a forest.
- If you can't create new directory objects, you lost your RID Master—probably a while back, as domain controllers obtain RIDs in blocks and cache them.
- In a multi-domain environment, a failed Infrastructure Master can result in incomplete group memberships, meaning users may not be able to access all of their resources.
- Domain upgrades and schema extensions can rely on the Domain Naming Master and the Schema Master, depending on what work they're doing.

The PDC Emulator is the one role you'll probably miss the soonest if something goes wrong; many of my customers keep this role on a clustered domain controller for that exact reason.

Whatever you do, don't forcibly seize a FSMO role from a domain controller unless you're taking that domain controller completely offline, demoting it (removing AD), and planning to rebuild it before it's reconnected to the network. This is especially true of the Schema Master and Domain Naming Master: Under *no circumstances* must two servers believe they each hold one of those roles.

Checking your FSMOs is pretty easy: Use the DCDiag tool on a domain controller in each of your domains (it's not a bad idea to run it on several domain controllers, in different sites, to make sure you get the same results). It'll check your FSMOs and report back. The next step, if a FSMO appears to be broken, is to check DNS. Really, it seems like two-thirds of all AD problems can be traced back to a DNS issue. Make sure each FSMO role holder is properly registered in DNS, and you'll probably be fine.

Troubleshooting Domain Controllers in General

Domain controllers, by and large, "just work." Provided everything around them—replication, time sync, and so forth—is all working, you'll tend to have very little trouble with the AD database and services. When you *think* a domain controller is broken, start by going through a quick checklist on configuration and surrounding operations:

- Make sure the domain controller's site and subnet configuration is correct.
- Make sure time sync is working and that the domain controller's clock matches that of the domain's PDC Emulator (see the next section).
- Make sure replication is working. If a domain controller seems "broken," either replication, or some dependency like the network itself, is likely causing the problem.
- Make sure the domain controller is properly registered in DNS, and ensure that client computers and other domain controllers can properly resolve the domain controller's DNS records.
- Check the domain controller's event logs for any bad news, and deal with whatever you find.

Once you've eliminated those problems, you may in fact be looking at a broken domain controller. There are a number of things you can do to troubleshoot problems, rebuild the directory database, and so forth. Honestly, a lot of customers I work with will simply demote and re-promote the domain controller. That rebuilds everything from scratch. It's somewhat time consuming but not necessarily more so than a protracted troubleshooting-and-repair process that may result in a re-promotion anyway.

Troubleshooting Time Sync

Time synchronization is absolutely crucial in AD. By default, authentication traffic only allows for a 5-minute out-of-sync window; let any client or domain controller get further out-of-sync than 5 minutes, and authentication stops working. The solution to this problem is *not* to extend that time window; doing so creates a higher security risk because someone can more easily capture and "replay" authentication packets against your network. Instead, fix the time-sync problem.

Time sync is handled by a background service on all Windows computers, servers, and clients. Client computers and member servers sync time with the domain controller that authenticated them when they started; domain controllers sync with the domain controller holding the PDC Emulator FSMO role. The PDC Emulator should sync with an external, authoritative time source. The sync traffic occurs over UDP port 123, so your first step will be to make sure that port is open. Keep in mind that, by default, the PDC Emulator isn't configured to sync time, and it will repeatedly log messages to that effect until you do configure it.

The best troubleshooting tool you have is the W32tm tool, which must be run from the command line by an administrator. This tool cannot function if the Windows Time Service is running, so temporarily stop that service before running W32tm. Be sure to restart the service when you're done troubleshooting. Some specific tips—each of which must be completed by an Administrator:

- Run **net time /querysntp** to check time sync servers on domain controllers and workstations
- Run **w32tm /resync** to check sync with your domain controller
- Run **w32tm /monitor /domain:*domain_name*** to check the status of domain controller time sources.
- Run **net time /domain:*domain_name* /set /y** to try to synchronize with the local domain time source

The errors generated by those commands, if any, will tell you what needs to be fixed. Also note that the Time Service *won't* always *immediately* correct an out-of-sync local clock: If the local clock is faster than its time source but less than 3 minutes out of sync, the Time Service will merely slow the clock so that it *eventually* comes back into sync. When doing so, the Time Service will check the time about every 45 minutes until the clock is in-sync for three consecutive checks. The service then resumes its normal behavior of checking the clock every 8 hours.

> **Resource**
>
> You can find more step-by-step tips on troubleshooting time sync at http://cainmanor.com/tech/windows-time-troubleshooting/.

Troubleshooting Kerberos

Provided time sync is working, Kerberos will generally work as advertised. Try to avoid fiddling with Kerberos' configuration (which can be done through Group Policy), as tweaking Kerberos settings incorrectly can lead to problems. *Most* Kerberos issues stem from underlying DNS or network connectivity issues; start by assuming that a problem is with DNS or the network and resolve those problems first.

Specific symptoms of a possible Kerberos issue:

- Users or computers can't log on or can't access network resources, and Kerberos is the protocol in use. You do have to check this, as sometimes a different protocol can be used and troubleshooting Kerberos is just a waste of your time.
- The event log will show errors related to Kerberos Key Distribution Center (KDC), Local Security Authority Server (LsaSrv), or Net Logon (Netlogon) services.
- Failure events in the Security log will indicate which protocol is being used: Enable auditing of failed logons, if you haven't done so, to see if any of these audits are logged. Note that enabling this level of auditing can increase log volume significantly; be sure to turn off this setting if it isn't normally on in your environment.

To troubleshoot Kerberos:

- You'll need to be an Administrator on the computers involved.
- Obviously, make sure you're on the latest service pack, hotfixes, and whatnot. Restart the computer(s) affected.
- Make sure DNS is working and that the affected computer can resolve a domain controller via DNS.
- Make sure *all* domain controllers a client might use are accessible and can be resolved via DNS.
- Check time sync.

Install the Windows Support Tools (from the server installation DVD), including Ldifde, LDP, Setspn, and Tokensz. You should also enable logon failure auditing because those events can contain useful diagnostic information (see http://technet.microsoft.com/en-us/library/cc736727(WS.10).aspx for instructions on doing so).

Finally, start troubleshooting. Use the step-by-step guide at http://technet.microsoft.com/en-us/library/cc786325(WS.10).aspx to use the Windows Support Tools to resolve specific problem areas.

Troubleshooting RIDs

Relative Identifiers (RIDs) are used to ensure that a unique ID number can be assigned to each directory object created by a domain controller. The RID Master FSMO Role hands out unique RIDs in batches to domain controllers; the controllers cache those RIDs and use them when creating new objects. When a domain controller runs out of RIDs, it asks the RID Master for more. Earlier in this chapter, I mentioned that an inability to create new objects is a sign that the RID Master is either broken, offline, or inaccessible to domain controllers (inaccessibility is often a DNS issue or network infrastructure problem).

There are a number of event log entries you can watch for:

- 16642 indicates that the domain controller is out of RIDs. It should have requested more; check the RID Master and restart the domain controller.
- 16643 indicates that the domain controller hasn't gotten a pool of RIDs yet—often because the RID Master isn't accessible.
- 16644 tells you that the domain is out of RIDs. This is a Bad Situation and shouldn't normally occur, even in huge domains. The limit of RIDs is a bit over 1 billion (1,073,741,825, to be exact).
- 16645 says that the domain controller just assigned its last RID and couldn't get more. Again, check the availability of, and connectivity to, the RID Master.
- 16646 indicates a processing problem where a domain controller tried to use an invalid RID. Force the domain controller to invalidate its RID pool, which should force it to ask for a new one.
- 16647 means the domain controller is requesting a new RID pool. This is good.
- 16648 means a domain controller got a new RID pool—this is excellent news.
- 16651 means a RID pool request failed—Bad News. The domain controller will retry—look for another 16647 event.

Troubleshooting Object Deletion

It's important to understand how object deletion occurs in AD so that you can troubleshoot problems:

1. When you delete an object, it is actually just "marked as deleted," a process called *tombstoning.*
2. Like any other change to an object, the tombstone change is replicated, thus "deleting" the object on all other domain controllers.
3. The old default value for tombstone clean-up was 60 days; as of Windows Server 2003, it was set to 180 days. After this period, each domain controller permanently deletes tombstoned objects.

There are some consequences to this behavior:

- If you restore a domain controller from a backup that is older than the clean-up window, or connect a domain controller that has been offline longer than that, deleted objects *will come back* because the old domain controller (or its backup) will re-create the object.
- The "Active Directory Recycle Bin" feature introduced in Windows Server 2008 R2 actually *copies* deleted objects to a separate area of the directory rather than deleting them. Again, reviving a very old domain controller can thus make objects "reappear" in their original location.

Most object deletion issues can be prevented by simply never allowing an older domain controller, or a backup of one, to be reconnected to the network.

Troubleshooting Replication

Replication is probably the trickiest thing to troubleshoot in AD. Before you dive in, I have some recommendations that can make replication less prone to problems:

- Keep your sites and subnets up-to-date. This is really crucial, as replication relies on the topology of your sites and subnets. A *subnet* is a single IP subnet—Class A, Class B, Class C, whatever you use. A *site* is a collection of subnets that all exist in the same LAN-quality bandwidth—that is, all the subnets with a 100Gbps or better Ethernet connection.
- Make your site links reflect your physical WAN architecture, and avoid creating site bridge links unless you *absolutely must do so* in order to speed replication to far-flung sites. Allowing the directory to calculate its own replication topology based on your physical WAN is the best course of action.

Assuming you haven't dorked around with your site, subnet, and site link configuration, you'll need some tools to start troubleshooting things. Microsoft provides a good walkthrough at http://technet.microsoft.com/en-us/library/cc738415(WS.10).aspx; personally, I much prefer third-party tools that can help me *visualize* the replication topology and that can check it for me and even initiate fixes. Quest's Spotlight on Active Directory is one such tool I've used; search for "Active Directory replication tool" in your favorite search engine and you'll find others.

Troubleshooting DNS

DNS, as I've indicated elsewhere in this chapter, turns out to be the root cause for a *lot* of AD troubles. In fact, I counsel all of my customers to get a solid AD-specific DNS monitoring tool in place to continuously check DNS operations and proactively alert them if something goes wrong. Why "AD-specific?" Because of the way in which AD *uses* DNS. A tremendous number of DNS records get added by domain controllers, and a monitoring solution that's aware of those things can do a better job of monitoring the overall infrastructure.

For example, a solution can check the AD itself to see which domain controllers exist, then verify that each one has registered all the proper DNS records, and *then* verify that DNS is properly returning those records, and *then* verify that the computers are reachable using the data in those records—covering the entire loop of possible problems, essentially. Such monitoring tools are nearly always commercial, meaning you'll have to pay a bit for them.

There are some obvious first steps to making sure that DNS is working properly. Each of these, however, requires that you know what DNS *should* be doing. When sitting down at a client computer, for example, you need to know which domain controllers it should expect to see, what DNS records it should expect to receive from a query, and so forth. All you can do is *verify* that DNS is returning what you expect; if it doesn't, you've found your problem. If you don't know what *should* be happening, however, you'll never find the problem.

Those first steps:

- Clear the client DNS cache by running **ipconfig /flush**.
- Check the DNS cache to make sure you don't have any static records from a hosts file.
- Use Nslookup to perform the same queries a client computer would, and verify the results. What you query is going to depend on what situation you're trying to replicate, of course. http://technet.microsoft.com/en-us/library/bb726934.aspx has a great list of starting points, particularly with regard to improper DNS server configuration.

With those basics out of the way, you can start troubleshooting. DNS troubleshooting is a massive topic all by itself, and there are several entire books on the subject, so I can't go into a great deal of depth here. But http://technet.microsoft.com/en-us/library/cc787724(WS.10).aspx is a good guide to getting started and covers some of the most common problems.

Troubleshooting Permissions

Last up is the process of troubleshooting permissions. This is when someone *should* have permission to something in AD but they don't—or the opposite, when they *do* but shouldn't. Really, this isn't much different than troubleshooting the same problem in the Windows file system. Keep in mind the following facts:

- Permissions can be applied directly at an organizational unit (OU) or container, then inherited by objects.
- Permissions can be applied directly on an object.

A user's effective permissions are the combination of every inherited parent OU permission plus the permissions directly on the object. A "Deny" permission anywhere in that chain of inheritance will override an "Allow" that occurs anywhere else. You can minimize the complexity of troubleshooting by never applying permissions directly to objects and by minimizing the number of OUs you apply permissions to. That way, you have fewer places to look.

To troubleshoot permissions in Active Directory Users and Computers, you'll first need to enable Advanced Features from the View menu. Otherwise, objects' Security tabs aren't even visible. Tells you how much Microsoft thinks you should mess with this stuff!

Once on the Security tab for an object, click Advanced. Then use the Effective Permissions tab. This is probably the easiest way to resolve the inheritance of permissions and see the final, effective permissions a given user has over a given object or container. Just select the user you're troubleshooting, then review the permissions.

Thanks for Reading—and Good Luck

Thanks very much for reading this Definitive Guide. I hope you've found helpful tips and useful explanations and that you're ready to go the next time a problem strikes your AD infrastructure.

Download Additional eBooks from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this eBook to be informative, we encourage you to download more of our industry-leading technology eBooks and video guides at Realtime Nexus. Please visit http://nexus.realtimepublishers.com.