DNS Scavenging

DNS Scavenging is a great answer to a problem that has been nagging everyone since RFC 2136 came out way back in 1997. Despite many clever methods of ensuring that clients and DHCP servers that perform dynamic updates clean up after themselves sometimes DNS can get messy. Remember that old test server that you built two years ago that caught fire before it could be used? Probably not. DNS still remembers it though. There are two big issues with DNS scavenging that seem to come up a lot:

"I'm hitting this 'scavenge now' button like a snare drum and nothing is happening. Why?"

or

"I woke up this morning, my DNS zones are nearly empty and Active Directory is sitting in a corner rocking back and forth crying. What happened?"

This post should help us figure out when the first issue will happen and completely avoid the second. We'll go through how scavenging is setup then I'll give you my best practices.

Scavenging setup

Scavenging will help you clean up old unused records in DNS. Since "clean up" really means "delete stuff" a good understanding of what you are doing and a healthy respect for "delete stuff" will keep you out of the hot grease. Because deletion is involved there are quite a few safety valves built into scavenging that take a long time to pop. When enabling scavenging patience is required. It will work just fine, but not today!

Note: For purposes of this discussion we are going to concentrate on the most common Windows DNS scenario: Windows Server 2003 DNS servers hosting AD integrated zones.

Scavenging is set in three places on a Windows Server:

1. On the individual resource record to be scavenged.
2. On a zone to be scavenged.
3. At one or more servers performing scavenging.

It must be set in all three places or nothing happens.

Scavenging settings on a Resource Record

To see the scavenging setting on a record hit View | Advanced in the DNS MMC then bring up properties on a record.

image

Image not found or type unknown

Scavenging gets set on a resource record in one of three ways. The first is by someone coming in here, checking the "Delete this record when it becomes stale" checkbox and hitting apply. When you hit apply the time of day will be rounded down to the nearest hour and applied as the timestamp on the record. Static records have a timestamp of 0 indicating do not scavenge.

The second is when a record gets created by a client machine registering using dynamic DNS. Windows clients will attempt to dynamically update DNS every 24 hours. All DDNS records get set to scavenge. When a record is first created by a client that has no existing record it is considered an "Update" and the timestamp is set. If the client has an existing host record and changes the IP of the host record this is also considered an "Update" and the timestamp is set. If the client has an existing host record with the same IP address then this is considered a "Refresh" and the timestamp may or may not get changed depending on zone settings. More on this later.

The third way to set scavenging on records is by using DNScmd.exe with the /ageallrecords switch. Let's pause here for a few moments to consider a few important words: All, Records, Delete, Stuff. If you actually run this command against a zone it will truly set scavenging and a timestamp on all records in the zone including static records that you never want to be scavenged. Because of the time it takes scavenging to do it's thing people find this command and get tempted to give it a try. Do not. It will delete stuff. Have patience instead.

Once a timestamp is set on a record it will replicate around to all servers that host the zone. There is one caveat to this. If scavenging is not enabled on the zone that hosts the record then it will never scavenge so the timestamp is essentially irrelevant. The timestamp may get updated on the server where the client dynamically registers but it will not replicate around to the other servers in the zone.

Scavenging Settings at the Zone

Before a server will even look at a record to see if it will be scavenged the zone must have scavenging enabled. To access the scavenging settings for a zone right click the zone, select properties then on the general tab hit the "Aging" button. This screen is universal for the zone. If you view it on any DNS server where this zone is replicated it will be the same.

image
Image not found or type unknown
When you first set scavenging on a zone the timestamp seen at the bottom (reload zone if you don't see it) will be set to the current time of day rounded down to the nearest hour plus the Refresh interval. This also gets reset any time the zone is loaded or any time dynamic updates get enabled on the zone.

The "zone can be scavenged after" timestamp is the first of your safety valves. It gives clients time to get their record timestamp updated before the big axe swings. Since new record timestamps are not replicated while zone scavenging is disabled this also gives replication time to get things in order.

Refresh and No-Refresh intervals

The next safety valves are the Refresh and No-refresh intervals. Both of these must elapse before a record can be deleted.

The No-refresh interval is a period of time during which a resource record cannot be refreshed. Recall from earlier that a refresh is a dynamic update where we are not changing the host/IP of a resource record, just touching the timestamp. If a client changes the IP of a host record this is considered an "update" and is exempt from the No-refresh interval. The purpose of a No-refresh interval is simply to reduce replication traffic. A change to a record means a change that must be replicated.

After the (Record Timestamp) + (No-refresh interval) elapses we enter the Refresh interval. The refresh interval is the time when refreshes to the timestamp are allowed. This is the time when good things must happen. The client is allowed to come in and update it's timestamp. This timestamp will be replicated around and the No-refresh interval begins again. If for some reason the client fails to update it's record during the refresh interval it becomes eligible to be scavenged. Will it disappear immediately? Probably not but it is certainly possible.

Note: When setting Refresh and No-Refresh intervals be sure to allow enough time for clients to get several registration attempts during a Refresh interval. Failure to do so could allow a record to become eligible for scavenging simply from a failed refresh attempt.

One last thing before we leave the zone setting behind. If you right click on your server you will see the option to "Set Aging/Scavenging for All Zones...". Selecting this will take you to a screen similar to the one above. What does this do? This sets the default settings that will be used if a new zone is created by this server. Unless you check the subsequent box "Apply these settings to the existing Active Directory-integrated zones" it will not touch existing zones.

Scavenging settings on the Server

So you now have a resource resource record set to scavenge and a zone set to scavenge. All that is left is for somebody to come along, check all the timestamps and delete some stuff. This is done by any server that hosts the AD integrated zone.

Setting scavenging on the server is done by right clicking the server in the MMC, selecting properties, going to the advanced tab and checking the "Enable automatic scavenging of stale records" checkbox.

image
Image not found or type unknown
The Scavenging Period is how often this particular server will attempt to scavenge. When a server scavenges it will log a DNS event 2501 to indicate how many records were scavenged. An event 2502 will be logged if no records were scavenged. Only one server is required to scavenge since the zone data is replicated to all servers hosting the zone.

Tip: You can tell exactly when a server will attempt to scavenge by taking the timestamp on the most recent 2501/2502 event and adding the Scavenging period to it.

Although you can set every server hosting the zone to scavenge I recommend just having one. The logic for this is simple: If the one server fails to scavenge the world won't end. You'll have one place to look for the culprit and one set of logs to check. If on the other hand you have many servers set to scavenge you have many logs to check if scavenging fails. Worse yet, if things start disappearing unexpectedly you don't want to go hopping from server to server looking for 2501 events.

To facilitate strict control over which server is scavenging for a zone you can use DNSCmd.exe to specify exactly which servers may scavenge. For example the following command will make it so that only 192.168.1.1 and 192.168.1.2 DNS servers are allowed to scavenge on the contoso.com zone:

DNSCmd . /ZoneResetScavengeServers contoso.com 192.168.1.1 192.168.1.2

With the server now scavenging, zones enabled for scavenging, and resources records set what actually happens when the server does it's thing?

The scavenging process and final safety valves

When the last 2501/2052 event + the server scavenging period comes around the server is going to make a scavenging attempt. You can also manually initiate an attempt by right clicking the server and selecting "Scavenge Stale Resource Records". Note that manually making an attempt in no way bypasses the safety valves. These are the final safety valves before we "delete stuff":

- Is scavenging enabled on the zone? Pretty self explanatory.
- Is dynamic update enabled on the zone? If it's not there is a good chance timestamps will be old enough that mass deletions can occur.
- Is the scavenging server listed as one of the "Scavenge Servers" for the zone?
- Are we past the "zone can be scavenged after" timestamp on the zone? This gives the clients and AD replication to get things squared away before we start.
- Has it been longer than a refresh interval since this zone was last replicated in Active Directory? If scavenging gets enabled on a server that has replication issues this will prevent it from tombstoning a bunch of records that may be perfectly fine on other servers.

If all of the above checks are good then the zone is ready to be scavenged. At this point the scavenging server checks the timestamp on each individual resource record. If the current date/time is greater than the timestamp + No-refresh + Refresh then the record is deleted.

My best practices

Here is how I set scavenging up on a preexisting zone. This procedure is designed for maximum safety. Using default settings this process can take as long as 4-5 weeks (2 weeks Sanity phase, 2-3 weeks for Enable phase)

Setup phase

1. Turn off scavenging on all servers. To confirm scavenging won't inadvertently run use the DNSCmd /ZoneResetScavengeServers to confine scavenging to a single server then ensure this server has scavenging disabled.
2. Turn on scavenging on the zones you wish to scavenge. Set the refresh and No-refresh intervals as desired. If you want things to scavenge more aggressively I would recommend lowering the No-refresh interval at the cost of some replication traffic. Leave the refresh at the default.
3. Add today's date plus the Refresh and No-Refresh intervals. Come back in a few weeks when this time has elapsed. Seriously you can't rush this.

Sanity check phase

Sift through your DNS records looking for any records older than the Refresh + No-Refresh interval. If you see any then something has gone wrong with the dynamic registration process and it must be corrected before proceeding. A thorough check at this point is the most important step in setup

Things to check if you find old records:

- Does an IPConfig /registerdns work?
- Who is the owner of the record (see security tab in the record properties)?
- Was the record statically created by an admin then later enabled for scavenging? If so you may need to delete the record to clear ownership and run an IPConfig /registerdns to get it updated.
- Is the server replicating OK with AD?

Do not proceed unless you can explain any outdated records. In the next phase they will be deleted.

Enable phase

The final step is to actually enable scavenging. Enable scavenging on the single server you used the /ZoneResetScavengServers command on.

Once enabled create a new test record and enable it for scavenging. Then map out the point in time when this record will disappear. Here is how:

1. Start with the timestamp on the record
2. Add the refresh interval
3. Add the no refresh interval
4. The result will be your "eligible to scavenge" time. The record will not disappear at this time though. It's just eligible.
5. Check your DNS event logs for 2501 and 2502 events to find what hour the DNS server is doing a scavenging run.
6. Take your "eligible to scavenge" time, find the most recent 2501/2502 event and add the server's Scavenging Period (from server properties | advanced tab) to it. This is the point in time when the test record you just created will disappear.

Lets look at an example with the following assumptions:

- Zone is set to a 3 day Refresh and a 3 day No-Refresh interval
- Server Scavenging period is set to 3 days
- Last DNS Event id 2501 or 2502 occurred at 6am on 1/1/2008
- We have a record with a timestamp of 1/1/2008 at 12:00 noon

Given these assumptions you can rub your temples for a bit and predict that the record will be deleted at approximately 6am on 1/10/2008.

image
Image not found or type unknown
Once scavenging is enabled you can check back periodically to look for the 2501 and 2502 events to see how things are going. You can also come back at the predicted date and time and see if your test record disappeared.

That's it!

From <http://blogs.technet.com/b/networking/archive/2008/03/19/don-t-be-afraid-of-dns-scavenging-just-be-patient.aspx>

Revision #1
Created 23 December 2023 00:24:47 by ColtM
Updated 7 August 2024 23:24:39 by ColtM