

Smart Proxy - Bug #17301

ISC DHCP known reservations/leases not updated over NFS

11/09/2016 05:49 PM - Eli Landon

Status: Closed	
Priority: Normal	
Assignee:	
Category: DHCP	
Target version:	
Difficulty:	Fixed in Releases:
Triaged:	Found in Releases:
Bugzilla link: 1397702	Red Hat JIRA:
Pull request:	
Description	
After upgrading to 1.13.1 the DHCP behavior is not consistent -	
<ul style="list-style-type: none">• Adding a host successfully retrieves an available IP and inserts a reservation in the leases file• Updating a hosts interface values causes a DHCP exception on submit, in the prior version I received an overwrite prompt• Deleting a host works without error but the DHCP reservation is not removed from the leases file.	
Environment:	
ISC-DHCP-Server version 4.3.3 on Ubuntu 16.04 LTS server Smart-Proxy version 1.13.1 on separate Ubuntu 14.04 LTS server	
Entry from leases file persists after host is deleted:	
<pre>host joni-cantrell.one.den.net { dynamic; hardware ethernet 0b:78:6d:44:33:c8; fixed-address 192.168.1.10 supersede server.filename = "pxelinux.0"; supersede server.next-server = 0a:75:1e:1e; supersede host-name = "joni-cantrell.one.den.net"; }</pre>	
From Smart-Proxy Log:	
<pre>D, [2016-11-09T15:00:30.248930 #2407] DEBUG -- : Loading subnet data for 192.168.1.0/255.255.255.0 D, [2016-11-09T15:00:30.253857 #2407] DEBUG -- : omshell: executed - set name = "joni-cantrell.one.den.net" D, [2016-11-09T15:00:30.254083 #2407] DEBUG -- : nil D, [2016-11-09T15:00:30.254196 #2407] DEBUG -- : omshell: executed - set ip-address = 192.168.1.10 D, [2016-11-09T15:00:30.254289 #2407] DEBUG -- : nil D, [2016-11-09T15:00:30.254392 #2407] DEBUG -- : omshell: executed - set hardware-address = 0b:78:6d:44:33:c8 D, [2016-11-09T15:00:30.254482 #2407] DEBUG -- : nil D, [2016-11-09T15:00:30.254583 #2407] DEBUG -- : omshell: executed - set hardware-type = 1 D, [2016-11-09T15:00:30.254672 #2407] DEBUG -- : nil D, [2016-11-09T15:00:30.254874 #2407] DEBUG -- : omshell: executed - set statements = "filename = \"pxelinux.0\"; next-server = 0a:75:1e:1e; option host-name = \"joni-cantrell.one.den.net\";" D, [2016-11-09T15:00:30.254989 #2407] DEBUG -- : nil D, [2016-11-09T15:00:30.255097 #2407] DEBUG -- : omshell: executed - create D, [2016-11-09T15:00:30.255194 #2407] DEBUG -- : nil E, [2016-11-09T15:00:30.259507 #2407] ERROR -- : Omshell failed: obj: <null> , > obj: host</pre>	

```

, > obj: host
, name = "joni-cantrell.one.den.net"
, > obj: host
, name = "joni-cantrell.one.den.net"
, ip-address = 0a:75:4f:12
, > obj: host
, name = "joni-cantrell.one.den.net"
, ip-address = 0a:75:4f:12
, hardware-address = 0b:78:6d:44:33:c8
, > obj: host
, name = "joni-cantrell.one.den.net"
, ip-address = 0a:75:4f:12
, hardware-address = 0b:78:6d:44:33:c8
, hardware-type = 1
, > obj: host
, name = "joni-cantrell.one.den.net"
, ip-address = 0a:75:4f:12
, hardware-address = 0b:78:6d:44:33:c8
, hardware-type = 1
, statements = "filename = "pxelinux.0"; next-server = 0a:75:1e:1e; option host-name = "joni-cantrell.one.den.net";"
, > can't open object: already exists
, obj: host
, name = "joni-cantrell.one.den.net"
, ip-address = 0a:75:4f:12
, hardware-address = 0b:78:6d:44:33:c8
, hardware-type = 1
, statements = "filename = "pxelinux.0"; next-server = 0a:75:1e:1e; option host-name = "joni-cantrell.one.den.net";"
, >
E, [2016-11-09T15:00:30.259827 #2407] ERROR -- : Failed to add DHCP reservation for joni-cantrell.one.den.net (192.168.1.10 / 0b:78:6d:44:33:c8): Entry already exists
D, [2016-11-09T15:00:30.259916 #2407] DEBUG -- : Failed to add DHCP reservation for joni-cantrell.one.den.net (192.168.1.10 / 0b:78:6d:44:33:c8): Entry already exists (Proxy::DHCP::Error)
/usr/share/foreman-proxy/modules/dhcp_isc/dhcp_isc_main.rb:99:in `report'
/usr/share/foreman-proxy/modules/dhcp_isc/dhcp_isc_main.rb:82:in `om_disconnect'
/usr/share/foreman-proxy/modules/dhcp_isc/dhcp_isc_main.rb:56:in `om_add_record'
/usr/share/foreman-proxy/modules/dhcp_isc/dhcp_isc_main.rb:32:in `add_record'
/usr/share/foreman-proxy/modules/dhcp/dhcp_api.rb:88:in `block in <class:DhcpApi>'
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:1541:in `call'
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:1541:in `block in compile!'
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:950:in `[]'
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:950:in `block (3 levels) in route!'
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:966:in `route_eval'
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:950:in `block (2 levels) in route!'
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:987:in `block in process_route'
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:985:in `catch'
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:985:in `process_route'
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:948:in `block in route!'
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:947:in `each'
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:947:in `route!'
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:1059:in `block in dispatch!'
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:1041:in `block in invoke'
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:1041:in `catch'
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:1041:in `invoke'
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:1056:in `dispatch!'
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:882:in `block in call!'
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:1041:in `block in invoke'
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:1041:in `catch'
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:1041:in `invoke'
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:882:in `call!'
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:870:in `call'
/usr/lib/ruby/vendor_ruby/rack/methodoverride.rb:21:in `call'
/usr/lib/ruby/vendor_ruby/rack/commonlogger.rb:33:in `call'
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:212:in `call'
/usr/share/foreman-proxy/lib/proxy/log.rb:63:in `call'
/usr/lib/ruby/vendor_ruby/rack/protection/xss_header.rb:18:in `call'
/usr/lib/ruby/vendor_ruby/rack/protection/path_traversal.rb:16:in `call'
/usr/lib/ruby/vendor_ruby/rack/protection/json_csrf.rb:18:in `call'

```

```
/usr/lib/ruby/vendor_ruby/rack/protection/base.rb:50:in `call'  
/usr/lib/ruby/vendor_ruby/rack/protection/base.rb:50:in `call'  
/usr/lib/ruby/vendor_ruby/rack/protection/frame_options.rb:31:in `call'  
/usr/lib/ruby/vendor_ruby/rack/nulllogger.rb:9:in `call'  
/usr/lib/ruby/vendor_ruby/rack/head.rb:11:in `call'  
/usr/lib/ruby/vendor_ruby/sinatra/showexceptions.rb:21:in `call'  
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:175:in `call'  
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:1949:in `call'  
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:1449:in `block in call'  
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:1726:in `synchronize'  
/usr/lib/ruby/vendor_ruby/sinatra/base.rb:1449:in `call'  
/usr/lib/ruby/vendor_ruby/rack/builder.rb:138:in `call'  
/usr/lib/ruby/vendor_ruby/rack/urlmap.rb:65:in `block in call'  
/usr/lib/ruby/vendor_ruby/rack/urlmap.rb:50:in `each'  
/usr/lib/ruby/vendor_ruby/rack/urlmap.rb:50:in `call'  
/usr/lib/ruby/vendor_ruby/rack/builder.rb:138:in `call'  
/usr/lib/ruby/vendor_ruby/rack/handler/webrick.rb:60:in `service'  
/usr/lib/ruby/1.9.1/webrick/httpserver.rb:138:in `service'  
/usr/lib/ruby/1.9.1/webrick/httpserver.rb:94:in `run'  
/usr/lib/ruby/1.9.1/webrick/server.rb:191:in `block in start_thread'
```

Related issues:

Related to Smart Proxy - Bug #2687: Performance issues with large ISC dataset...

Closed

06/20/2013

History

#1 - 11/10/2016 03:36 AM - Dominic Cleal

- Project changed from Plugins to Smart Proxy

- Category set to DHCP

1) Are you using NFS here?

ISC-DHCP-Server version 4.3.3 on Ubuntu 16.04 LTS server
Smart-Proxy version 1.13.1 on separate Ubuntu 14.04 LTS server

2) Does restarting the smart proxy between creating the first DHCP reservation and deleting the host work? Does it now delete the DHCP reservation?

#2 - 11/10/2016 03:55 AM - Anonymous

Environment:
ISC-DHCP-Server version 4.3.3 on Ubuntu 16.04 LTS server
Smart-Proxy version 1.13.1 on separate Ubuntu 14.04 LTS server

Smart-proxy must be located on the same server as isc dhcp server. NFS-mounted partitions will not work anymore.

#3 - 11/10/2016 04:11 AM - Ohad Levy

does it make sense to check if its mounted as NFS ?
also, should we provide a fall back option?

#4 - 11/10/2016 04:12 AM - Ohad Levy

I think NFS support is important, as I know multiple users who uses HA solution for dhcp proxy using shared NFS storage (e.g. using pacemaker)

#5 - 11/10/2016 04:20 AM - Anonymous

Accessing leases file over NFS will absolutely destroy the performance: we no longer will be able to track updates and will have to parse the leases on every request. On top of that, accessing leases over NFS adds additional latency. If HA is required, we'll need to address this via different architecture, and not by relying on NFS.

I feel we shouldn't support NFS at all -- it's a corner case that hugely impacts performance.

#6 - 11/10/2016 09:59 AM - Eli Landon

Yes, we're mounting over NFS. The smart-proxy and DHCP server are in the same cluster and traffic is layer 2 so latency should be very low. I can run perf tests if needed. I'll test with the DHCP server and smart-proxy on the same host.

Restarting the smart-proxy after deleting the host does not remove the entry from the leases file.

This may be unrelated but when provisioning a new host the TFTP smart-proxy adds entries to the UEFI grub2 folder even though the PXE loader is set as PXELinux and we're using PXELinux provisioning templates. Will see if issue persists when running the DHCP server and smart-proxy on the same box.

Thanks you

#7 - 11/10/2016 10:04 AM - Dominic Cleal

- Related to Bug #2687: Performance issues with large ISC dataset (DHCP smart proxy) added

#8 - 11/10/2016 10:04 AM - Dominic Cleal

- Subject changed from ISC DHCP Addresses Not Updating to ISC DHCP known reservations/leases not updated over NFS

Restarting the smart-proxy after deleting the host does not remove the entry from the leases file.

Restart it before deleting the host, it'll re-read the file at startup.

#9 - 11/10/2016 10:08 AM - Anonymous

Please see the description of the problem above. Accessing leases file over NFS isn't going to work with this implementation.

#10 - 11/20/2016 04:18 PM - Konstantin Orekhov

While not ideal, NFS is still a very important part of HA strategies, mainly because of its simplicity. As we discussing in other thread, most of the DHCP smart-proxy performance already handled by your recent changes, Dmitri, so may be it makes sense to re-test the performance over NFS with those?

In terms of the latency - any of the file-sync'ing solutions for HA (cluster FS, rsync, inotify, etc.) will always have some sort of a delay or latency associated with that. I think the application should be able to handle that.

Just my 2 cents since this dropping NFS support in 1.13 just as performance issues were resolved is huge bummer for me - seems like I just can't get all the features I need, something is always missing :(

#11 - 11/20/2016 05:22 PM - Anonymous

While not ideal, NFS is still a very important part of HA strategies, mainly because of its simplicity. As we discussing in other thread, most of the DHCP smart-proxy performance already handled by your recent changes, Dmitri, so may be it makes sense to re-test the performance over NFS with those?

In your environment you'll be looking at 30+ seconds processing time: each request would require fetching leases file over the network and then parsing it. Additionally one web-server thread will be tied up for the duration of the request. I don't think the way forward is to rely on NFS as it sacrifices too much of performance, but to have a smart-proxy based HA.

#12 - 11/23/2016 03:46 AM - Lukas Zapletal

- Bugzilla link set to 1397702

Dmitri, external DHCP setup is fully supported feature of Satellite 6.1+ and documented here:

<https://access.redhat.com/documentation/en/red-hat-satellite/6.2/paged/installation-guide/chapter-5-configuring-external-services>

While there is some performance degradation over NFS, it's apparently not that horrible as customers are using this in production. Consider fixing this regression, my personal opinion on this is to have the inotify framework optional via settings, so in case of NFS setup users can turn this off. We can do warning in documentation informing about performance degradation.

I am associating Satellite 6.3 BZ with this issue to make triage team talks about this.

#13 - 11/23/2016 04:43 AM - Anonymous

I don't think we ought to discuss degrees of horribleness when we know we can do better.

To reiterate: If we were to use NFS, we'd have to retrieve and parse leases file on **each** request, which doesn't scale at all. I don't want to support and maintain a solution that only works for some installations. If HA setups are a must, we need a solution that supports them without degradation in performance compared to baseline, and not an NFS-based workaround.

#14 - 03/07/2017 12:10 PM - Imri Zvik

Dmitri Dolguikh wrote:

I don't think we ought to discuss degrees of horribleness when we know we can do better.

To reiterate: If we were to use NFS, we'd have to retrieve and parse leases file on **each** request, which doesn't scale at all. I don't want to support and maintain a solution that only works for some installations. If HA setups are a must, we need a solution that supports them without degradation in performance compared to baseline, and not an NFS-based workaround.

Just my two cents - I just encountered this my self.

My workaround was to patch the proxy to start a thread which computes checksum in a loop and call `observer.leases_recreated` if the digest changed.

This works perfectly, and it seems there is no noticeable performance hit.

If that is a valid course of action, I can open a PR for this.

#15 - 03/09/2017 04:07 PM - Konstantin Orekhov

Imri, would you mind sharing your patch? I don't know how large is your environment, mine is very large (several dozen updates to leases in a second) and I could test your approach from performance point of view.

#16 - 03/09/2017 04:14 PM - Konstantin Orekhov

Dmitri, if the above approach does not for whatever reason, I was wondering if you've given a thought of putting local DHCP cache that SmartProxy creates into memcached?

This way it is replicated across all memcached cluster nodes w/o extra efforts from user or smart-proxy perspective to allow multi-node access - I'm sure I don't really have explain to you the advantages of this product.

Performance may be a concern, of course, but I can offer my assistance in testing it out.

#17 - 04/14/2017 01:00 PM - Konstantin Orekhov

Dmitri, any thoughts on using memcached to keep DHCP leases and reservations in sync between multiple smart-proxy nodes?

#18 - 04/14/2017 01:47 PM - Anonymous

Dmitri, any thoughts on using memcached to keep DHCP leases and reservations in sync between multiple smart-proxy nodes?

I haven't thought about this; I suspect any distributed key-value store can be made to work (etcd would be my personal choice) at a cost of increased complexity and slightly increased response time. Probably a good candidate for a dhcp module provider.

PS> Not sure if you noticed but there's a provider that supports loading of leases file from remote filesystems: https://github.com/theforeman/smart_proxy_dhcp_remote_isc. In all likelihood it's too slow for your network though.

#19 - 05/29/2017 04:47 PM - Konstantin Orekhov

Hello, Dmitri!

Finally got a chance to try `dhcp_remote_isc` provider and ran into some issues, which I think maybe related to some missing documentation.

I followed your readme in a git repo and here are the steps I took:

- install a GEM:

```
1. gem list | grep dhcp
   smart_proxy_dhcp_remote_isc (0.0.2)
```

- add that GEM into `bundler.d` (BTW, you probably want to update your README as it references `smart_proxy_dhcp_infoblox` GEM instead of `smart_proxy_dhcp_remote_isc`)

```
cat /usr/share/foreman-proxy/bundler.d/dhcp_remote_isc.rb
gem 'smart_proxy_dhcp_remote_isc'
```

- based on your doc, i created a symlink from `/etc/foreman-proxy/settings.d/dhcp_isc.yml` to `/etc/foreman-proxy/settings.d/dhcp_remote_isc.yml`, otherwise I would get this:

```
I, [2017-05-29T12:52:28.768926 ] INFO -- : WEBrick::HTTPServer#start done.
W, [2017-05-29T12:58:14.333731 ] WARN -- : Couldn't find settings file /etc/foreman-proxy/settings.d/dhcp_remote_isc.yml. Using default settings.
E, [2017-05-29T12:58:14.337632 ] ERROR -- : Disabling all modules in the group ['dhcp_remote_isc', 'dhcp'] due to a failure in one of them: cannot load such file -- dhcp_common/isc/omapi_provider
```

- but now I'm stuck here - even though I believe I fulfilled all the requirements in a readme, I get this error:

```
I, [2017-05-29T13:30:34.861203 ] INFO -- : WEBRick::HTTPServer#start done.  
E, [2017-05-29T13:30:35.201942 ] ERROR -- : Disabling all modules in the group ['dhcp_remote_isc', 'dhcp'] due to a failure in one of them: cannot load such file -- dhcp_common/isc/oma_pi_provider  
I, [2017-05-29T13:30:35.579158 ] INFO -- : Successfully initialized 'dynflow'  
I, [2017-05-29T13:30:35.581172 ] INFO -- : Successfully initialized 'ssh'  
I, [2017-05-29T13:30:35.581282 ] INFO -- : Successfully initialized 'foreman_proxy'  
I, [2017-05-29T13:30:35.581350 ] INFO -- : Successfully initialized 'tftp'  
I, [2017-05-29T13:30:35.583296 ] INFO -- : Successfully initialized 'puppet_proxy_legacy'  
I, [2017-05-29T13:30:35.583431 ] INFO -- : Successfully initialized 'puppet'  
I, [2017-05-29T13:30:35.583517 ] INFO -- : Successfully initialized 'bmc'  
I, [2017-05-29T13:30:35.583655 ] INFO -- : Successfully initialized 'logs'
```

- if I update dhcp.yml back to use dhcp_isc provider, SmP starts just fine:

```
I, [2017-05-29T13:43:41.862860 ] INFO -- : going to shutdown ...  
I, [2017-05-29T13:43:41.863003 ] INFO -- : WEBRick::HTTPServer#start done.  
I, [2017-05-29T13:43:42.575187 ] INFO -- : Successfully initialized 'dynflow'  
I, [2017-05-29T13:43:42.577613 ] INFO -- : Successfully initialized 'ssh'  
I, [2017-05-29T13:43:42.577704 ] INFO -- : Successfully initialized 'foreman_proxy'  
I, [2017-05-29T13:43:42.577771 ] INFO -- : Successfully initialized 'tftp'  
I, [2017-05-29T13:43:42.609535 ] INFO -- : Successfully initialized 'dhcp_isc'  
I, [2017-05-29T13:43:42.609626 ] INFO -- : Successfully initialized 'dhcp'  
I, [2017-05-29T13:43:42.611096 ] INFO -- : Successfully initialized 'puppet_proxy_legacy'  
I, [2017-05-29T13:43:42.611178 ] INFO -- : Successfully initialized 'puppet'  
I, [2017-05-29T13:43:42.611222 ] INFO -- : Successfully initialized 'bmc'  
I, [2017-05-29T13:43:42.611267 ] INFO -- : Successfully initialized 'logs'
```

I must be missing something is a config somewhere, just not able to determine what that is so far.
If you have any suggestions, please let me know.

Thanks!

#20 - 05/30/2017 08:51 AM - Anonymous

Make sure you are using smart-proxy version 1.15. I'll update the provider to work with develop branch today.

#21 - 05/30/2017 10:53 AM - Anonymous

I released v. 0.0.3 of smart_proxy_dhcp_remote_isc gem that is required if smart-proxy develop branch is used.

#22 - 05/30/2017 03:25 PM - Konstantin Orekhov

Ah, I was on 1.14 actually, so will have to upgrade today while I have a chance.

#23 - 05/31/2017 04:18 PM - Konstantin Orekhov

And now I can't upgrade as I do need discovery working - https://groups.google.com/forum/#!topic/foreman-users/M_DcyFMZwxM

#24 - 05/31/2017 04:21 PM - Anonymous

Dhcp-related issues have been fixed in develop, with fixes available in the next stable release.

#25 - 11/20/2017 10:30 PM - Anonymous

- Status changed from New to Closed

I'm closing this issue, please open a new one if you experience problems with dhcp_isc_remote plugin.