Foreman - Bug #16219

Association named 'hostgroup' was not found on Nic::Base

08/22/2016 03:15 AM - Nacho Barrientos

Status:	Closed		
Priority:	Normal		
Assignee:	Daniel Lobato Garcia		
Category:	Users, Roles and Permissions		
Target version:	1.12.3		
Difficulty:		**Fixed in Releases:**	
Triaged:		**Found in Releases:**	
Bugzilla link:		**Red Hat JIRA:**	
Pull request:	https://github.com/theforeman/foreman/pull/3807		

Description

Hi,

We're running Foreman 1.11.4 on CentOS7 as installed directly from the RPM available via yum.theforeman.org and we're running into a very odd issue when querying the list of interfaces of a host via the API:

A GET request on *api/hosts/nacho1.cern.ch/interfaces* leads to a server crash:

```
< HTTP/1.1 500 Internal Server Error
  "error": {"message":"Association named 'hostgroup' was not found on Nic::Base; perhaps you misspelled it?"}
```

This is even happening with a pristine database (one non-admin user, one hostgroup, one host and one role assigned to the user allowing viewing hosts in the mentioned hostgroup). Other models queried via hosts/ like parameters/ are not affected.

Find attached the full stacktrace (production.log) and the full list of Gems installed (gemlist.txt). Please let us know if a dump of the test database would be useful.

We initially hit the bug when we backported 7a86dcfe6b36dd43cd6163ce70599e53f09cc217 (fix for CVE-2016-5390) to 1.11.2.

Thanks for looking into it and any question please don't hesitate to ask :)

Related issues:

Related to Foreman - Bug #15653: CVE-2016-5390 - access to API host interface...	**Closed**	**07/12/2016**

Associated revisions

Revision 3357cdaf - 09/12/2016 10:17 AM - Daniel Lobato Garcia

Fixes #16219 - Interfaces API works with scoped view_hosts

Before this commit, parent_scope called 'merge' on a scope that may contain conditions that do not make sense on 'resource_class'.

In this case, when a user with a filter :view_hosts and search 'hostgroup_fullname = foo' tried to view api/v2/hosts/somehost/interfaces, that would not work.

The SQL generated by `scope_for` contains `AND ((`hostgroups`.`title` = 'base'))`, which clearly cannot be merged with the Nic::Base scope, as 'hostgroups' isn't a field in Nic::Base.

To avoid this problem, we resolve the scope and call 'where' on the scope ids, instead of merging the parent and the child scope.

Notice this problem exists for any child/parent relationship where the user has a filter :view_parent , search 'parent_attribute = foo' if parent_attribute is not defined in the child.

Revision 108a5f7a - 09/19/2016 12:27 PM - Daniel Lobato Garcia

Fixes #16219 - Interfaces API works with scoped view_hosts

Before this commit, parent_scope called 'merge' on a scope that may
contain conditions that do not make sense on 'resource_class'.

In this case, when a user with a filter :view_hosts and search
'hostgroup_fullname = foo' tried to view
api/v2/hosts/somehost/interfaces, that would not work.

The SQL generated by `scope_for` contains
`AND ((`hostgroups`.`title` = 'base'))`, which clearly cannot be
merged with the Nic::Base scope, as 'hostgroups' isn't a field in
Nic::Base.

To avoid this problem, we resolve the scope and call 'where' on the
scope ids, instead of merging the parent and the child scope.

Notice this problem exists for any child/parent relationship where the
user has a filter :view_parent , search 'parent_attribute = foo' if
parent_attribute is not defined in the child.

(cherry picked from commit 3357cdaf190445fb38bd29de6f217c005dbc2e9e)

Revision 5f587b15 - 09/26/2016 07:38 AM - Daniel Lobato Garcia

Fixes #16219 - Interfaces API works with scoped view_hosts

Before this commit, parent_scope called 'merge' on a scope that may
contain conditions that do not make sense on 'resource_class'.

In this case, when a user with a filter :view_hosts and search
'hostgroup_fullname = foo' tried to view
api/v2/hosts/somehost/interfaces, that would not work.

The SQL generated by `scope_for` contains
`AND ((`hostgroups`.`title` = 'base'))`, which clearly cannot be
merged with the Nic::Base scope, as 'hostgroups' isn't a field in
Nic::Base.

To avoid this problem, we resolve the scope and call 'where' on the
scope ids, instead of merging the parent and the child scope.

Notice this problem exists for any child/parent relationship where the
user has a filter :view_parent , search 'parent_attribute = foo' if
parent_attribute is not defined in the child.

(cherry picked from commit 3357cdaf190445fb38bd29de6f217c005dbc2e9e)

History

#1 - 08/22/2016 03:18 AM - Dominic Cleal

- Category set to Users, Roles and Permissions

What filters do you have on the user precisely? I'm guessing you have view_hosts with hostgroup = Foo or similar?

#2 - 08/22/2016 03:18 AM - Dominic Cleal

- Related to Bug #15653: CVE-2016-5390 - access to API host interfaces, parameters etc. are not restricted by view_hosts filters added

#3 - 08/22/2016 03:55 AM - Nacho Barrientos

Yes, indeed, the user has a role called 'hgmanager_nacho' with "view_hosts, create_hosts, edit_hosts, destroy_hosts, build_hosts, power_hosts, console_hosts, ipmi_boot, puppetrun_hosts" as permissions and "hostgroup_fullname ~ nacho" as search query, plus the filters that come by default from the "Default user" role. I could slim it if necessary.

Btw, it's obvious but GET api/hosts/nacho1.cern.ch works fine with credentials for that user.

#4 - 08/23/2016 04:54 AM - Nacho Barrientos

In case it helped, got same outcome with a SQLite backend.

#5 - 08/30/2016 09:05 AM - Nacho Barrientos

Interesting. If I create a role with the same permissions as 'hgmanager_nacho' but with no filters (unlimited flag on) the bug is not triggered I can happily get the list of interfaces via the API :)

#6 - 08/30/2016 09:30 AM - Nacho Barrientos

Nacho Barrientos wrote:

> Interesting. If I create a role with the same permissions as 'hgmanager_nacho' but with no filters (unlimited flag on) the bug is not triggered I can happily get the list of interfaces via the API :)

s/I can/and I can

#7 - 09/01/2016 12:40 PM - Daniel Lobato Garcia

After a bit of testing I found that the SQL produced by `scope_for` is very different depending on whether resource_class is Host or Host::Managed. For Host::Managed (where it fails):

`
scope_for(parent_class, :permission => "#{parent_permission(action_permission)}_#{parent_name.pluralize}").to_sql
=> "SELECT `hosts`.`id` AS t0_r0, `hosts`.`name` AS t0_r1, `hosts`.`last_compile` AS t0_r2, `hosts`.`last_report` AS t0_r3, `hosts`.`updated_at` AS t0_r4, `hosts`.`created_at` AS t0_r5, `hosts`.`root_pass` AS t0_r6, `hosts`.`architecture_id` AS t0_r7, `hosts`.`operatingsystem_id` AS t0_r8, `hosts`.`environment_id` AS t0_r9, `hosts`.`ptable_id` AS t0_r10, `hosts`.`medium_id` AS t0_r11, `hosts`.`build` AS t0_r12, `hosts`.`comment` AS t0_r13, `hosts`.`disk` AS t0_r14, `hosts`.`installed_at` AS t0_r15, `hosts`.`model_id` AS t0_r16, `hosts`.`hostgroup_id` AS t0_r17, `hosts`.`owner_id` AS t0_r18, `hosts`.`owner_type` AS t0_r19, `hosts`.`enabled` AS t0_r20, `hosts`.`puppet_ca_proxy_id` AS t0_r21, `hosts`.`managed` AS t0_r22, `hosts`.`use_image` AS t0_r23, `hosts`.`image_file` AS t0_r24, `hosts`.`uuid` AS t0_r25, `hosts`.`compute_resource_id` AS t0_r26, `hosts`.`puppet_proxy_id` AS t0_r27, `hosts`.`certname` AS t0_r28, `hosts`.`image_id` AS t0_r29, `hosts`.`organization_id` AS t0_r30, `hosts`.`location_id` AS t0_r31, `hosts`.`type` AS t0_r32, `hosts`.`otp` AS t0_r33, `hosts`.`realm_id` AS t0_r34, `hosts`.`compute_profile_id` AS t0_r35, `hosts`.`provision_method` AS t0_r36, `hosts`.`grub_pass` AS t0_r37, `hosts`.`global_status` AS t0_r38, `hosts`.`lookup_value_matcher` AS t0_r39, `hosts`.`discovery_rule_id` AS t0_r40, `hosts`.`pxe_loader` AS t0_r41, `hostgroups`.`id` AS t1_r0, `hostgroups`.`name` AS t1_r1, `hostgroups`.`created_at` AS t1_r2, `hostgroups`.`updated_at` AS t1_r3, `hostgroups`.`environment_id` AS t1_r4, `hostgroups`.`operatingsystem_id` AS t1_r5, `hostgroups`.`architecture_id` AS t1_r6, `hostgroups`.`medium_id` AS t1_r7, `hostgroups`.`ptable_id` AS t1_r8, `hostgroups`.`root_pass` AS t1_r9, `hostgroups`.`puppet_ca_proxy_id` AS t1_r10, `hostgroups`.`use_image` AS t1_r11, `hostgroups`.`image_file` AS t1_r12, `hostgroups`.`ancestry` AS t1_r13, `hostgroups`.`vm_defaults` AS t1_r14, `hostgroups`.`subnet_id` AS t1_r15, `hostgroups`.`domain_id` AS t1_r16, `hostgroups`.`puppet_proxy_id` AS t1_r17, `hostgroups`.`title` AS t1_r18, `hostgroups`.`realm_id` AS t1_r19, `hostgroups`.`compute_profile_id` AS t1_r20, `hostgroups`.`grub_pass` AS t1_r21, `hostgroups`.`lookup_value_matcher` AS t1_r22, `hostgroups`.`subnet6_id` AS t1_r23, `hostgroups`.`pxe_loader` AS t1_r24 FROM `hosts` LEFT OUTER JOIN `hostgroups` ON `hostgroups`.`id` = `hosts`.`hostgroup_id` WHERE `hosts`.`type` IN ('Host::Managed') AND ((`hostgroups`.`title` = 'base'))"
`

For Host:

"SELECT `hosts`.* FROM `hosts` WHERE `hosts`.`type` IN ('Host::Managed')"

Notice the AND hostgroups.title = base. On the last line of the 'parent_scope' method of app/controllers/api/base_controller, we try to merge the parent scope (Host::Managed scope) with the children scope (Nic::Base) scope. Since Nic::Base cannot resolve that 'hostgroups.title = base', it'll fail. First thing that comes to mind is that we should resolve that scope before filtering Nic::Base with it.

#8 - 09/01/2016 12:43 PM - Daniel Lobato Garcia

Changing parent_scope to return: `resource_class.joins(association.name).where('host_id' => scope.map(&:id))` works :)

I'll write a test and submit a pull request

#9 - 09/01/2016 01:21 PM - The Foreman Bot

- Status changed from New to Ready For Testing

- Assignee set to Daniel Lobato Garcia

- Pull request https://github.com/theforeman/foreman/pull/3807 added

#10 - 09/06/2016 10:00 AM - Daniel Lobato Garcia

- Target version set to 1.6.2

#11 - 09/12/2016 10:19 AM - Dominic Cleal

- translation missing: en.field_release set to 181

#12 - 09/12/2016 11:02 AM - Daniel Lobato Garcia

- Status changed from Ready For Testing to Closed

- % Done changed from 0 to 100

Applied in changeset 3357cdaf190445fb38bd29de6f217c005dbc2e9e.

Files

production.log	24 KB	08/22/2016	Nacho Barrientos
gemlist.txt	1.82 KB	08/22/2016	Nacho Barrientos