

Foreman - Bug #15527

api/v2/hosts is slow to load permissions for non-admins

06/27/2016 11:49 AM - Daniel Lobato Garcia

Status:	Closed	
Priority:	Normal	
Assignee:	Daniel Lobato Garcia	
Category:	Performance	
Target version:	1.11.4	
Difficulty:		Fixed in Releases:
Triaged:		Found in Releases:
Bugzilla link:	1422304	Red Hat JIRA:
Pull request:	https://github.com/theforeman/foreman/pull/3618	

Description

After <http://projects.theforeman.org/issues/13639>, the show response of api/v2/hosts (used in GET/PUT/POST/DELETE requests) checks if certain permissions are true for a pair 'user/host'.

The response looks like

```
"permissions": {  
  "view_hosts": true,  
  "create_hosts": true,  
  "edit_hosts": true,  
  "destroy_hosts": true,  
  "build_hosts": true,  
  "power_hosts": true,  
  "console_hosts": true,  
  "ipmi_boot": true,  
  "puppetrun_hosts": true,  
  "view_discovered_hosts": true,  
  "provision_discovered_hosts": true,  
  "edit_discovered_hosts": true,  
  "destroy_discovered_hosts": true,  
  "submit_discovered_hosts": true,  
  "auto_provision_discovered_hosts": true  
}
```

To do this, it calls `.can?` on all permissions:

<https://github.com/theforeman/foreman/blob/develop/app/views/api/v2/hosts/show.json.rabl#L38>

`.can?` finds all `Host::Managed` to which a permission can be applied to, then calls `.include?(host)` on the collection to see whether the permission action is applicable to the host in question:

<https://github.com/theforeman/foreman/blob/develop/app/services/authorizer.rb#L15>

This means any API call will have to call

```
find_collection
```

at least 9 times (more depending on plugins).

```
find_collection
```

is an expensive query to make, for a user that has a lot of permissions it can take seconds.

```
irb(main):129:0> Permission.where(:resource_type => "Host").all.inject({}) do |hash, permission|  
irb(main):130:1* puts permission.name
```

```
irb(main):131:1> puts Benchmark.measure {
irb(main):132:2* authorizer.can?(permission.name, host)
irb(main):133:2> }
irb(main):134:1> end
view_hosts
 1.560000  0.220000  1.780000 ( 1.878100)
create_hosts
 2.270000  0.190000  2.460000 ( 2.562798)
edit_hosts
 1.700000  0.170000  1.870000 ( 1.972199)
destroy_hosts
 1.650000  0.210000  1.860000 ( 2.002876)
build_hosts
 2.150000  0.180000  2.330000 ( 2.452150)
power_hosts
 1.630000  0.200000  1.830000 ( 1.928230)
console_hosts
 1.610000  0.200000  1.810000 ( 1.913544)
ipmi_boot
 1.650000  0.210000  1.860000 ( 1.957224)
puppetrun_hosts
 2.610000  0.200000  2.810000 ( 2.913967)
=> nil
```

find_collection is also considerably slower when the user has a lot of roles. For example `.can?` calls on an user with 15 roles took ~2 seconds, while calls on an user with 7 roles took ~0.5 seconds. This is in an environment without organizations and locations.

We had reports on #theforeman of requests that took 10 seconds to run on this endpoint. Thanks to Steve Traylen & Nacho Barrientos from CERN for their help debugging this issue on IRC.

One quick "fix" we should implement is to start using the Rails cache in Authorizer, so that it survives multiple requests, unlike the current cache in memory. Obviously we should also make find_collection faster, maybe making the 'WHERE' query at the same time we call find_collection would make it faster.

Related issues:

Related to Foreman - Feature #13639: show permissions for hosts as part of sh...

Closed

02/09/2016

Associated revisions

Revision 1ff7e259 - 07/18/2016 05:21 AM - Daniel Lobato Garcia

Fixes #15527 - api/v2/hosts is slow loading permissions

The `.eager_load` call when called with `[]` generates a SELECT query that selects all host attributes. This can cause it to take a lot longer when there are large number of hosts involved. These attributes are irrelevant in this context (when they are relevant, they're put in the `scope_components` hash)

With 5700 hosts, it takes about 50% less time to run this than to run it with `eager_load`.

Chaining the call with `.where` makes the call near-instant. It selects the host from the list of hosts we can see using the db directly.

Revision 7fabb50a - 07/22/2016 10:03 AM - Daniel Lobato Garcia

Fixes #15527 - api/v2/hosts is slow loading permissions

The `.eager_load` call when called with `[]` generates a SELECT query that selects all host attributes. This can cause it to take a lot longer when there are large number of hosts involved. These attributes are irrelevant in this context (when they are relevant, they're put in the `scope_components` hash)

With 5700 hosts, it takes about 50% less time to run this than to run it with `eager_load`.

Chaining the call with `.where` makes the call near-instant. It selects the host from the list of hosts we can see using the db

directly.

(cherry picked from commit 1ff7e25908f81912458a0fd85c421fb5feeb90a1)

Revision 905f42b0 - 07/22/2016 10:05 AM - Daniel Lobato Garcia

Fixes #15527 - api/v2/hosts is slow loading permissions

The `.eager_load` call when called with `[]` generates a `SELECT` query that selects all host attributes. This can cause it to take a lot longer when there are large number of hosts involved. These attributes are irrelevant in this context (when they are relevant, they're put in the `scope_components` hash)

With 5700 hosts, it takes about 50% less time to run this than to run it with `eager_load`.

Chaining the call with `.where` makes the call near-instant. It selects the host from the list of hosts we can see using the db directly.

(cherry picked from commit 1ff7e25908f81912458a0fd85c421fb5feeb90a1)

History

#1 - 06/28/2016 03:02 AM - Dominic Cleal

- Related to Feature #13639: show permissions for hosts as part of show api added

#2 - 07/01/2016 03:34 AM - Steve Traylen

The worst and simplest case we found that was slow was that of our global reader account.

It has role - 'View all hosts and hostgroups' and then we 25,000 hosts. Not sure if the hosts need anything special associated with them.

#3 - 07/01/2016 04:12 AM - Daniel Lobato Garcia

OK - I see it now:

```
authorizer = Authorizer.new(User.current)
host = Host.last
puts Benchmark.measure {
  Permission.where(:resource_type => "Host").all.inject({}) do |hash, permission|
    puts permission
    puts Benchmark.measure { authorizer.can?(permission.name, host) }
  end
  puts 'total time'
}
```

This should be good enough to reproduce it. With 5672 hosts the total running time is around 9 seconds, 0.5 on each permission.

```
view_hosts
  0.430000  0.060000  0.490000 ( 0.509471)
create_hosts
  0.360000  0.050000  0.410000 ( 0.434622)
edit_hosts
  1.310000  0.050000  1.360000 ( 1.386734)
destroy_hosts
  0.340000  0.050000  0.390000 ( 0.408201)
build_hosts
  0.680000  0.050000  0.730000 ( 0.744966)
power_hosts
  0.370000  0.040000  0.410000 ( 0.430918)
console_hosts
  0.410000  0.040000  0.450000 ( 0.479227)
ipmi_boot
  0.440000  0.060000  0.500000 ( 0.507271)
puppetrun_hosts
  0.410000  0.040000  0.450000 ( 0.475734)
view_discovered_hosts
  0.450000  0.050000  0.500000 ( 0.518474)
provision_discovered_hosts
  0.350000  0.050000  0.400000 ( 0.420052)
edit_discovered_hosts
  0.440000  0.050000  0.490000 ( 0.507206)
```

```

destroy_discovered_hosts
  0.340000  0.050000  0.390000 ( 0.401846)
saltrun_hosts
  0.760000  0.060000  0.820000 ( 0.848326)
submit_discovered_hosts
  0.320000  0.040000  0.360000 ( 0.379871)
auto_provision_discovered_hosts
  0.830000  0.050000  0.880000 ( 0.897404)
total time
  8.240000  0.790000  9.030000 ( 9.356212)

```

#4 - 07/01/2016 04:54 AM - Daniel Lobato Garcia

A bit more info - the `.include?` seems to be the slowest portion of the equation always, but I'm afraid it's just Rails delaying actually running the query until the very end.

```

view_hosts
-- get resource permissions
  0.010000  0.000000  0.010000 ( 0.005155)
-- get base permissions
  0.000000  0.000000  0.000000 ( 0.000047)
-- build filtered scope components
  0.000000  0.000000  0.000000 ( 0.000067)
-- eager load and join scope
  0.000000  0.000000  0.000000 ( 0.000432)
-- call .where on scope components
  0.000000  0.000000  0.000000 ( 0.000008)
-- call include host
  0.340000  0.070000  0.410000 ( 0.424963)
can? view_hosts total time
  0.360000  0.070000  0.430000 ( 0.454942)
create_hosts
-- get resource permissions
  0.000000  0.000000  0.000000 ( 0.000801)
-- get base permissions
  0.000000  0.000000  0.000000 ( 0.000087)
-- build filtered scope components
  0.000000  0.000000  0.000000 ( 0.000044)
-- eager load and join scope
  0.000000  0.000000  0.000000 ( 0.000355)
-- call .where on scope components
  0.000000  0.000000  0.000000 ( 0.000005)
-- call include host
  0.350000  0.070000  0.420000 ( 0.440600)
can? create_hosts total time
  0.350000  0.070000  0.420000 ( 0.450844)
edit_hosts
-- get resource permissions
  0.000000  0.000000  0.000000 ( 0.000529)
-- get base permissions
  0.000000  0.000000  0.000000 ( 0.000039)
-- build filtered scope components
  0.000000  0.000000  0.000000 ( 0.000041)
-- eager load and join scope
  0.000000  0.000000  0.000000 ( 0.000341)
-- call .where on scope components
  0.000000  0.000000  0.000000 ( 0.000004)
-- call include host
  0.430000  0.050000  0.480000 ( 0.501443)
can? edit_hosts total time
  0.440000  0.050000  0.490000 ( 0.511189)
destroy_hosts
-- get resource permissions
  0.000000  0.000000  0.000000 ( 0.000519)
-- get base permissions
  0.000000  0.000000  0.000000 ( 0.000048)
-- build filtered scope components
  0.000000  0.000000  0.000000 ( 0.000047)
-- eager load and join scope
  0.000000  0.000000  0.000000 ( 0.000329)
-- call .where on scope components
  0.000000  0.000000  0.000000 ( 0.000005)
-- call include host
  0.390000  0.080000  0.470000 ( 0.466731)
can? destroy_hosts total time

```

```

0.400000 0.080000 0.480000 ( 0.476783)
build_hosts
-- get resource permissions
0.000000 0.000000 0.000000 ( 0.000535)
-- get base permissions
0.000000 0.000000 0.000000 ( 0.000064)
-- build filtered scope components
0.000000 0.000000 0.000000 ( 0.000035)
-- eager load and join scope
0.000000 0.000000 0.000000 ( 0.000556)
-- call .where on scope components
0.000000 0.000000 0.000000 ( 0.000004)
-- call include host
0.530000 0.060000 0.590000 ( 0.603243)
can? build_hosts total time
0.540000 0.060000 0.600000 ( 0.615611)
power_hosts
-- get resource permissions
0.000000 0.000000 0.000000 ( 0.000546)
-- get base permissions
0.000000 0.000000 0.000000 ( 0.000042)
-- build filtered scope components
0.000000 0.000000 0.000000 ( 0.000041)
-- eager load and join scope
0.000000 0.000000 0.000000 ( 0.000351)
-- call .where on scope components
0.000000 0.000000 0.000000 ( 0.000007)
-- call include host
0.550000 0.070000 0.620000 ( 0.631016)
can? power_hosts total time
0.550000 0.070000 0.620000 ( 0.643122)
console_hosts
-- get resource permissions
0.000000 0.000000 0.000000 ( 0.000535)
-- get base permissions
0.000000 0.000000 0.000000 ( 0.000043)
-- build filtered scope components
0.000000 0.000000 0.000000 ( 0.000038)
-- eager load and join scope
0.000000 0.000000 0.000000 ( 0.000306)
-- call .where on scope components
0.000000 0.000000 0.000000 ( 0.000006)
-- call include host
0.360000 0.060000 0.420000 ( 0.436719)
can? console_hosts total time
0.370000 0.060000 0.430000 ( 0.446748)
ipmi_boot
-- get resource permissions
0.000000 0.000000 0.000000 ( 0.000432)
-- get base permissions
0.000000 0.000000 0.000000 ( 0.000047)
-- build filtered scope components
0.000000 0.000000 0.000000 ( 0.000036)
-- eager load and join scope
0.000000 0.000000 0.000000 ( 0.000273)
-- call .where on scope components
0.000000 0.000000 0.000000 ( 0.000006)
-- call include host
0.450000 0.060000 0.510000 ( 0.510587)
can? ipmi_boot total time
0.450000 0.060000 0.510000 ( 0.522243)
puppetrun_hosts
-- get resource permissions
0.000000 0.000000 0.000000 ( 0.000698)
-- get base permissions
0.000000 0.000000 0.000000 ( 0.000085)
-- build filtered scope components
0.000000 0.000000 0.000000 ( 0.000040)
-- eager load and join scope
0.000000 0.000000 0.000000 ( 0.000365)
-- call .where on scope components
0.000000 0.000000 0.000000 ( 0.000006)
-- call include host
0.630000 0.060000 0.690000 ( 0.708434)
can? puppetrun_hosts total time

```

```

0.630000 0.060000 0.690000 ( 0.712979)
view_discovered_hosts
-- get resource permissions
0.000000 0.000000 0.000000 ( 0.000776)
-- get base permissions
0.000000 0.000000 0.000000 ( 0.000075)
-- build filtered scope components
0.000000 0.000000 0.000000 ( 0.000049)
-- eager load and join scope
0.000000 0.000000 0.000000 ( 0.000446)
-- call .where on scope components
0.000000 0.000000 0.000000 ( 0.000008)
-- call include host
0.330000 0.060000 0.390000 ( 0.406091)
can? view_discovered_hosts total time
0.330000 0.060000 0.390000 ( 0.409876)
provision_discovered_hosts
-- get resource permissions
0.000000 0.000000 0.000000 ( 0.000680)
-- get base permissions
0.000000 0.000000 0.000000 ( 0.000075)
-- build filtered scope components
0.000000 0.000000 0.000000 ( 0.000045)
-- eager load and join scope
0.000000 0.000000 0.000000 ( 0.000401)
-- call .where on scope components
0.000000 0.000000 0.000000 ( 0.000016)
-- call include host
0.420000 0.070000 0.490000 ( 0.499028)
can? provision_discovered_hosts total time
0.420000 0.070000 0.490000 ( 0.503011)
edit_discovered_hosts
-- get resource permissions
0.000000 0.000000 0.000000 ( 0.000854)
-- get base permissions
0.000000 0.000000 0.000000 ( 0.000071)
-- build filtered scope components
0.000000 0.000000 0.000000 ( 0.000062)
-- eager load and join scope
0.000000 0.000000 0.000000 ( 0.000513)
-- call .where on scope components
0.000000 0.000000 0.000000 ( 0.000010)
-- call include host
0.380000 0.050000 0.430000 ( 0.455582)
can? edit_discovered_hosts total time
0.390000 0.050000 0.440000 ( 0.460197)
destroy_discovered_hosts
-- get resource permissions
0.000000 0.000000 0.000000 ( 0.000776)
-- get base permissions
0.000000 0.000000 0.000000 ( 0.000057)
-- build filtered scope components
0.000000 0.000000 0.000000 ( 0.000039)
-- eager load and join scope
0.000000 0.000000 0.000000 ( 0.000337)
-- call .where on scope components
0.000000 0.000000 0.000000 ( 0.000005)
-- call include host
0.400000 0.050000 0.450000 ( 0.467105)
can? destroy_discovered_hosts total time
0.400000 0.050000 0.450000 ( 0.470384)
saltrun_hosts
-- get resource permissions
0.000000 0.000000 0.000000 ( 0.000470)
-- get base permissions
0.000000 0.000000 0.000000 ( 0.000045)
-- build filtered scope components
0.000000 0.000000 0.000000 ( 0.000034)
-- eager load and join scope
0.000000 0.000000 0.000000 ( 0.000278)
-- call .where on scope components
0.000000 0.000000 0.000000 ( 0.000004)
-- call include host
1.120000 0.040000 1.160000 ( 1.171215)
can? saltrun_hosts total time

```

```

1.120000 0.040000 1.160000 ( 1.174038)
submit_discovered_hosts
-- get resource permissions
0.000000 0.000000 0.000000 ( 0.000712)
-- get base permissions
0.000000 0.000000 0.000000 ( 0.000109)
-- build filtered scope components
0.000000 0.000000 0.000000 ( 0.000033)
-- eager load and join scope
0.000000 0.000000 0.000000 ( 0.000300)
-- call .where on scope components
0.000000 0.000000 0.000000 ( 0.000008)
-- call include host
0.320000 0.060000 0.380000 ( 0.390384)
can? submit_discovered_hosts total time
0.320000 0.060000 0.380000 ( 0.393552)
auto_provision_discovered_hosts
-- get resource permissions
0.000000 0.000000 0.000000 ( 0.000629)
-- get base permissions
0.000000 0.000000 0.000000 ( 0.000040)
-- build filtered scope components
0.000000 0.000000 0.000000 ( 0.000035)
-- eager load and join scope
0.000000 0.000000 0.000000 ( 0.000307)
-- call .where on scope components
0.000000 0.000000 0.000000 ( 0.000004)
-- call include host
0.370000 0.070000 0.440000 ( 0.458432)
can? auto_provision_discovered_hosts total time
0.370000 0.070000 0.440000 ( 0.461458)
total time
7.440000 0.980000 8.420000 ( 8.710428)

```

#5 - 07/01/2016 05:53 AM - Daniel Lobato Garcia

And here are the 2 main queries that are run on find_collection. We're creating a ton of auxiliary tables for no reason I think -

```

"SELECT `filters`.* FROM `filters` INNER JOIN `filterings` ON `filterings`.`filter_id` = `filters`.`id` INNER JOIN `permissions` ON `permissions`.`id` =
`filterings`.`permission_id` INNER JOIN `roles` ON `filters`.`role_id` = `roles`.`id` INNER JOIN `cached_user_roles` ON `roles`.`id` =
`cached_user_roles`.`role_id` WHERE `cached_user_roles`.`user_id` = 134 AND (permissions.resource_type = 'Host') AND (permissions.name =
'view_hosts') ORDER BY filters.role_id, filters.id"

```

```

"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`.`salt_proxy_id` AS t0_r38, `hosts`.`salt_environment_id` AS
t0_r39, `hosts`.`discovery_rule_id` AS t0_r40, `hosts`.`global_status` AS t0_r41, `hosts`.`lookup_value_matcher` AS t0_r42, `hosts`.`chef_proxy_id`
AS t0_r43 FROM `hosts` WHERE `hosts`.`type` IN ('Host::Managed')"

```

#6 - 07/01/2016 12:39 PM - The Foreman Bot

- Status changed from New to Ready For Testing
- Assignee set to Daniel Lobato Garcia
- Pull request <https://github.com/foreman/foreman/pull/3618> added

#7 - 07/14/2016 04:38 AM - Marek Hulán

We're creating a ton of auxiliary tables for no reason I think

what do you mean by auxiliary tables?

#8 - 07/18/2016 06:02 AM - Daniel Lobato Garcia

- Status changed from Ready For Testing to Closed
- % Done changed from 0 to 100

Applied in changeset [1ff7e25908f81912458a0fd85c421fb5feeb90a1](#).

#9 - 07/18/2016 08:37 AM - Dominic Cleal

- *translation missing: en.field_release set to 169*

#10 - 02/14/2017 07:52 PM - Chris Duryee

- *Bugzilla link set to 1422304*

#11 - 02/21/2017 09:53 AM - Daniel Lobato Garcia

- *Target version set to 1.11.0*