Foreman - Bug #1430

Adding a new host in a subdomain that has no SOA or NS record fails

01/05/2012 04:43 AM - Andreas Ntaflos

Status:	Duplicate		
Priority:	Normal		
Assignee:			
Category:	DNS		
Target version:			
Difficulty:		**Fixed in Releases:**	
Triaged:		**Found in Releases:**	
Bugzilla link:		**Red Hat JIRA:**	
Pull request:			

Description

DNS subdomains usually are defined in their own zone but this is not mandatory. A zone may perfectly legally comprise multiple subdomains, in Bind implemented by using $ORIGIN statements throughout the zone definition. This is sometimes called a *virtual subdomain* (see http://www.zytrax.com/books/dns/ch9/subdomain.html or Pro DNS and Bind) and is often easier to implement than fully delegating each subdomain to its own zone. For example a zone "example.com" can contain subdomains "us.example.com" and "ny.us.example.com", and those subdomains don't have a SOA or NS record.

Foreman tries to be smart about DNS and doesn't consult the local resolver (configured in /etc/resolv.conf). Instead it tries to do everything by itself and looks up SOA or NS records for subdomains. This avoids issues with stale local DNS caches, but leads to the following problem:

When a new host is added in Foreman that is part of such a virtual subdomain (ny.us.example.com, host web01.ny.us.example.com) Foreman tries to look up a SOA or NS record but fails, because there doesn't exist one. Adding the host then fails with the following errors:

```
Errno::ECONNREFUSED
Connection refused - recvfrom(2)
/usr/lib/ruby/1.8/resolv.rb:703:in `recv'
/usr/lib/ruby/1.8/resolv.rb:703:in `recv_reply'
/usr/lib/ruby/1.8/resolv.rb:618:in `request'
/usr/lib/ruby/1.8/resolv.rb:489:in `each_resource'
/usr/lib/ruby/1.8/resolv.rb:939:in `resolv'
/usr/lib/ruby/1.8/resolv.rb:937:in `each'
/usr/lib/ruby/1.8/resolv.rb:937:in `resolv'
/usr/lib/ruby/1.8/resolv.rb:936:in `each'
/usr/lib/ruby/1.8/resolv.rb:936:in `resolv'
/usr/lib/ruby/1.8/resolv.rb:934:in `each'
/usr/lib/ruby/1.8/resolv.rb:934:in `resolv'
/usr/lib/ruby/1.8/resolv.rb:481:in `each_resource'
/usr/lib/ruby/1.8/resolv.rb:386:in `each_address'
/usr/lib/ruby/1.8/resolv.rb:362:in `getaddress'
/usr/share/foreman/lib/net/dns.rb:21:in `lookup'
/usr/share/foreman/lib/net/dns.rb:15:in `lookup'
/usr/share/foreman/lib/net/dns.rb:65:in `dns_lookup'
/usr/share/foreman/lib/net/dns/a_record.rb:24:in `conflicts'
/usr/share/foreman/lib/net.rb:22:in `conflicting?'
/usr/share/foreman/app/models/orchestration/dns.rb:47:in `validate_dns'
/usr/share/foreman/app/models/orchestration.rb:55:in `valid?'
/usr/share/foreman/app/controllers/hosts_controller.rb:93:in `create'
```

Apparently (according to strace) when no records are returned for the subdomain Foreman tries to query 0.0.0.0:53 for information but can't, because nothing listens on 0.0.0.0:53.

The attached patch by Ohad Levy extends the resolver method of Foreman's Domain class to query the local resolver (/etc/resolv.conf) when Foreman's internal DNS lookup doesn't return anything, as in this case.

Related issues:

Is duplicate of Foreman - Bug #1426: Foreman should not fail if dns domain ha...	**Closed**	**01/03/2012**

History

#1 - 01/05/2012 08:44 AM - Ohad Levy

- Status changed from New to Duplicate

Files

domain.rb.patch	430 Bytes	01/05/2012	Andreas Ntaflos