### Foreman - Bug #27246

**IPAddr::InvalidAddressError on valid link-local address on Windows**

07/08/2019 12:03 PM - Martyn Smith

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>Ewoud Kohl van Wijngaarden</td>
</tr>
<tr>
<td>Category:</td>
<td>Facts</td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
</tr>
<tr>
<td>Difficulty:</td>
<td>Pull request:</td>
</tr>
<tr>
<td>Triaged:</td>
<td>No</td>
</tr>
<tr>
<td>Bugzilla link:</td>
<td></td>
</tr>
</tbody>
</table>

**Fixed in Releases:** 1.23.0

**Found in Releases:** 1.22.0

**Description**

Hi

There appears to be an issue with parsing valid link-local ipv6 addresses on Azure.

This address (assigned on Windows)

**Ethernet adapter Ethernet:**

- **Connection-specific DNS Suffix**: reddog.microsoft.com
  - **Description**: Microsoft Hyper-V Network Adapter
- **Physical Address**: 00-22-48-07-F2-71
- **DHCP Enabled**: Yes
- **Autoconfiguration Enabled**: Yes
- **Link-local IPv6 Address**: fe80::302f:51e0:1313:f50d%5(Preferred)
- **IPv4 Address**: XXX.XXX.XXX.XXX Preferred
- **Subnet Mask**: 255.255.255.0
- **Lease Obtained**: Friday, June 28, 2019 10:06:52 AM

is read by factor correctly as:

```
"ipaddress6": "fe80::302f:51e0:1313:f50d%5",
```

and then causes an exception

```
2019-07-08T03:44:22 [I|app|9449499f] Import facts for 'server.domain.tribal-int.com' completed. Added: 0, Updated: 0, Deleted 0 facts
IPAddr::InvalidAddressError: invalid address
/usr/share/foreman/app/models/subnet.rb:299:in `new'
/usr/share/foreman/app/models/subnet.rb:299:in `subnet_for'
/usr/share/foreman/app/models/host/base.rb:488:in `set_interface'
/usr/share/foreman/app/models/host/base.rb:222:in `block in set_interfaces'
/usr/share/foreman/app/models/host/base.rb:219:in `each'
/usr/share/foreman/app/models/host/base.rb:219:in `set_interfaces'
/usr/share/foreman/app/models/concerns/puppet_host_extensions.rb:3:in `populate_fields_from_facts'
/usr/share/foreman/app/models/host/managed.rb:412:in `populate_fields_from_facts'
/usr/share/foreman/app/models/concerns/puppet_host_extensions.rb:3:in `call_block_in_parse_facts'
```

---
This appears to be caused by the regex relating to ipv6 addresses not taking into account that a valid format can contain % followed by a number as per the thread below:

https://superuser.com/questions/99746/why-is-there-a-percent-sign-in-the-ipv6-address

Related issues:
Related to Foreman - Bug #26060: Fact parse error - IPAddress::InvalidAddressErr... Closed

Associated revisions
Revision 9e0afc5f - 07/15/2019 11:23 AM - Ewoud Kohl van Wijngaarden
Fixes #27246 - Handle device identifiers in IPs

On Windows Facter can return an IPv6 link local including the device identifier:

fe80::302f:51e0:1313:f50d%5

However, IPAddress doesn't handle this. This patch handles this case and cleans up the link local handling by using the underlying Ruby function which also detects ignores IPv4 link local.
The root cause of this was fixed in facter 3.11.0 / 3.13.4 / 3.14.3. However, we should carry support for this for some time. Having the original issue is easier to see if it can be removed.

History

#1 - 07/08/2019 12:05 PM - Martyn Smith
- Related to Bug #26060: Fact parse error - IPAddr::InvalidAddressError: invalid address added

#2 - 07/08/2019 12:14 PM - Martyn Smith
- Category set to Facts

#3 - 07/08/2019 02:06 PM - Martyn Smith
Looks like it relates to the RE_IPV6ADDRLIKE_FULL and RE_IPV6ADDRLIKE_COMPRESSED regexes from upstream Ruby code in ipaddr.rb

Neither of these recognise the link-local address.

#4 - 07/08/2019 04:27 PM - Martyn Smith
Ok - I've a hacky workaround that works but it's a change to the upstream ipaddr.rb

Documenting it here for those interested as this breaks default_hostgroup and subnets having the VMs that are provisioned being recognised as being on the subnet when the facts are uploaded into foreman.

Update the ruby/ipaddr.rb with the 2 extra lines in the regex as highlighted below.

```
RE_IPV6ADDRLIKE_COMPRESSED = %r{
  \A
  (?<:(?: [\da-f]{1,4} :)* [\da-f]{1,4} )? )
  (?<:(?: [\da-f]{1,4} :)* )
  (?<: [\da-f]{1,4} )
  | \d+
  \. \d+
  \. \d+
  \. \d+
  +
  [\\da-f]{1,4} \$[\\d]+$
  )? ? }
\z
```

Regards
Martyn
That's not really correct because in Linux the device identifiers (which are after the % sign) are the literal device identifier, e.g. eth0, enp0s31f6 or wlp58s0.

Perhaps we should catch it and do some fallback logic. Something like this is untested, but fairly defensive in its parsing.

def parse_ip(address)
    begin
        IPAddr.new(address)
    rescue IPAddr::InvalidAddressError
        # Discard the device identifier for IPv6 link local
        raise unless address.is_a?(String) && address.include?('%%')
        result = IPAddr.new(address.split('%%').first)
        raise unless result.ipv6? && result.link_local?
        result
    end
end

https://bugs.ruby-lang.org/issues/8464 looks like an upstream bug about this.

I'd definitely suggest we catch it as an exception on fact upload appears to stop other processes, which run after the fact upload, to run. Subnets assignment and default_hostgroup updates are processes I'm aware of but there are possibly others as well.

I'm happy that changing the ipaddr regex is the wrong approach and accept your suggestion.

Martyn

Fixed in Releases 1.23.0 added

- Status changed from Ready For Testing to Closed