pfSense - Bug #3404

DHCP Server Fails to Start on Interfaces that are Slow to Come Online During Boot

01/22/2014 10:12 AM - Jason Crowley

Status:	New	**Start date:**	01/22/2014
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	50%
Category:	DHCP Server	**Estimated time:**	0.00 hour
Target version:			
Affected Version:	All	**Affected Architecture:**	All

Description

When the services_dhcpd_configure() function is called during boot, it will skip interfaces that are not fully online. If all dhcpd-enabled interfaces are not online, dhcpd will fail to start. If only some of the interfaces are online, it will start but not serve dhcp on the slow-to-start interfaces.

The place where we've been able to reproduce this consistently is OpenVPN interfaces that have dhcpd enabled.

Background Information

OpenVPN's native IP-address allocation system does not work with dnsmasq to register clients' IP addresses in DNS. To work around this limitation, we build an OpenVPN tunnel that allows us to obtain IP addresses from pfSense's DHCP server. The dhcpd instance will then go through the normal process to ensure that the client's IP is registered with dnsmasq.

Platform Affected

2.1-RELEASE We're using amd64, but I expect it affects all processor architectures.

Steps to Reproduce

1. Configure an OpenVPN Server in tap mode. Ensure you've set the following parameters.
   - Device Mode: tap
   - IPv4 Tunnel Network: <blank>
     - *We want dhcpd, not openvpn assigning IP addresses.*
   - Advanced configuration: server-bridge
     - *This enables the DHCP broadcast traffic to traverse the tunnel to the dhcpd instance on the pfSense OpenVPN interface*
2. Configure the OpenVPN interface with a static IP address.
3. Configure and enable a DHCP server on the OpenVPN interface.
4. Reboot.
5. Log in via SSH and execute the following command.

```
# ps -axww | grep dhcpd
...
46118  ??  Ss     0:00.09 /usr/local/sbin/dhcpd -user dhcpd -group _dhcp -chroot /var/dhcpd -c
f /etc/dhcpd.conf -pf /var/run/dhcpd.pid em0 em2
```

6. Note that the last two arguments in the dhcpd command line above are the interfaces for dhcpd to listen on. There is not an OpenVPN interface (ovpns1) there. If you try to acquire a DHCP lease over an OpenVPN connection, you will get no response.
7. Restart dhcpd via the web gui Services page.

```
# ps -axww | grep dhcpd
...
69734  ??  Ss     0:00.00 /usr/local/sbin/dhcpd -user dhcpd -group _dhcp -chroot /var/dhcpd -c
f /etc/dhcpd.conf -pf /var/run/dhcpd.pid em0 em2 ovpns1
```

8. Note that the ovpns1 interface is now in the command line. You can now acquire a DHCP lease through your OpenVPN tunnel.

Recommended Solution

One of my coworkers (Micah Mitchell) is working on a simple solution that will add about 10 lines of code to the services_dhcpd_configure() function. This code will check each interface configured with a DHCP server to see that it is up before starting dhcpd. If an interface is not up, it will sleep for 1 second and loop for up to 10 seconds before moving on.

Our initial testing shows this code resolves the problem on the pfSense instances we've tested it on. During boot, we see a two-second delay while services_dhcpd_configure() waits for interfaces to come online prior to launching dhcpd. Expect the code to be submitted within the next day or two.

History

#1 - 01/22/2014 10:46 AM - Ermal Luçi

The proper solution for this is to bounce the dhcpd when the openvpn link comes up.
Check rc.newwanip[v6] script on this and there is the proper solution to be added.

From what i see dhcpd is not bumped from this script with the frame that dhcpd cannot run on dynamic interfaces.
Though some more checks can be added there to aid the need.

#2 - 01/22/2014 12:43 PM - Micah Mitchell

- File services.inc.patch added

I have attached a patch file for /etc/inc/services.inc

This will have the services_dhcpd_configure() function check the config for enabled interfaces with enabled dhcp servers and then give the interface up to 10 seconds to come up before continuing on. This is based on the same logic that used with checking if ppp(oe) interfaces have come online.

This code only waits if both the interface and dhcp server are enabled and the interface is not up.

I have tested this on the gateway that Jason was referencing above and it did fix our issue. It only added a 2 second delay during boot to allow the OpenVPN interface to come up. If no OpenVPN interfaces are enabled, there is no delay during boot.

#3 - 01/22/2014 02:30 PM - Ermal Luçi

Please read my comment on the proper solution.
This is a workaround/hack for your local installation.

#4 - 01/24/2014 08:07 AM - Jason Crowley

- File openvpn.inc.patch added

Thanks for the help Ermal. When I try to bounce dhcpd in the rc.newwanip script, I run into a problem where it appears that multiple instances of rc.newwanip are running simultaneously. Because of this, they try to write the dhcpd.conf file and start dhcpd on top of each other. I get output like this.

> rc.newwanip: The command '/usr/local/sbin/dhcpd -user dhcpd -group _dhcp -chroot /var/dhcpd -cf /etc/dhcpd.conf -pf /var/run/dhcpd.pid em0 em2' returned exit code '1', the output was 'Internet Systems Consortium DHCP Server 4.2.5-P1 Copyright 2004-2013 Internet Systems Consortium. All rights reserved. For info, please visit https://www.isc.org/software/dhcp/ Wrote 4 leases to leases file. Listening on BPF/em2/1e:07:96:ec:8e:4e/172.28.68.224/27 Sending on BPF/em2/1e:07:96:ec:8e:4e/172.28.68.224/27 Listening on BPF/em0/ca:32:32:49:19:1d/172.28.68.0/25 Sending on BPF/em0/ca:32:32:49:19:1d/172.28.68.0/25 Can't bind to dhcp address: Address already in use Please make sure there is no other dhcp server running and that there's no entry for dhcp or bootp in /etc/inetd.conf.

Also, I can see that rc.newwanip is getting called prior to the OpenVPN interface coming completely online.

When I run this command string:

> kill `cat /var/run/openvpn_server1.pid` ; /usr/local/sbin/openvpn --config /var/etc/openvpn/server1.conf ; date ; ifconfig ovpns1 ; sleep 1 ; date ; ifconfig ovpns1 ; sleep 1 ; date ; ifconfig ovpns1 ; sleep 1 ; date ; ifconfig ovpns1 ; sleep 1 ; date ; ifconfig ovpns1 ; sleep 1 ; date ; ifconfig ovpns1 ; sleep 1 ; date ; ifconfig ovpns1

I get this output:

```
OK
Fri Jan 24 07:39:03 CST 2014
ovpns1: flags=8842<BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80000<LINKSTATE>
        ether 00:bd:4f:07:00:01
        Opened by PID 71188
Fri Jan 24 07:39:04 CST 2014
ovpns1: flags=8842<BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80000<LINKSTATE>
        ether 00:bd:4f:07:00:01
        Opened by PID 71188
Fri Jan 24 07:39:05 CST 2014
ovpns1: flags=8842<BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80000<LINKSTATE>
        ether 00:bd:4f:07:00:01
        Opened by PID 71188
Fri Jan 24 07:39:06 CST 2014
ovpns1: flags=8842<BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80000<LINKSTATE>
        ether 00:bd:4f:07:00:01
        Opened by PID 71188
Fri Jan 24 07:39:07 CST 2014
ovpns1: flags=8842<BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80000<LINKSTATE>
        ether 00:bd:4f:07:00:01
        Opened by PID 71188
Fri Jan 24 07:39:08 CST 2014
ovpns1: flags=8842<BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80000<LINKSTATE>
        ether 00:bd:4f:07:00:01
        Opened by PID 71188
Fri Jan 24 07:39:09 CST 2014
ovpns1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80000<LINKSTATE>
        ether 00:bd:4f:07:00:01
        inet 172.28.68.193 netmask 0xfffffffe0 broadcast 172.28.68.223
        inet6 fe80::2bd:4fff:fe07:1%ovpns1 prefixlen 64 tentative scopeid 0x8
        nd6 options=1<PERFORMNUD>
        Opened by PID 71188
Fri Jan 24 07:39:10 CST 2014
ovpns1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80000<LINKSTATE>
        ether 00:bd:4f:07:00:01
        inet 172.28.68.193 netmask 0xfffffffe0 broadcast 172.28.68.223
        inet6 fe80::2bd:4fff:fe07:1%ovpns1 prefixlen 64 scopeid 0x8
        nd6 options=1<PERFORMNUD>
        Opened by PID 71188
```

And I find this in system.log:

```
Jan 24 07:39:03 nfkcgw01 kernel: ovpns1: link state changed to DOWN
Jan 24 07:39:03 nfkcgw01 kernel: ovpns1: link state changed to UP
Jan 24 07:39:03 nfkcgw01 check_reload_status: rc.newwanip starting ovpns1
Jan 24 07:39:03 nfkcgw01 check_reload_status: Reloading filter
Jan 24 07:39:06 nfkcgw01 php: rc.newwanip: rc.newwanip: Informational is starting ovpns1.
```

```
Jan 24 07:39:06 nfkcgw01 php: rc.newwanip: rc.newwanip: on (IP address: ) (interface: opt2) (real interface: o
vpns1)
.
Jan 24 07:39:06 nfkcgw01 php: rc.newwanip: rc.newwanip: Failed to update opt2 IP, restarting...
Jan 24 07:39:06 nfkcgw01 check_reload_status: Configuring interface opt2
Jan 24 07:39:07 nfkcgw01 php: rc.filter_configure_sync: Could not find IPv4 gateway for interface (opt2).
Jan 24 07:39:09 nfkcgw01 php: rc.interfaces_wan_configure: Deny router advertisements for interface opt2
```

Note that rc.newwanip is getting called six seconds before the OpenVPN interface actually has an IP.

I've added the code below to the openvpn_restart() function to wait for the OpenVPN interface to come up before returning. I've also attached this code in a patch file.

```
/* look for interface to come up before continuing */
$i = 1;
while(mwexec("/sbin/ifconfig $devname | grep '[<,]UP[,>]'", true)) {
        if($g['debug'])
                log_error("Sleeping 1 second waiting for openvpn interface to come up attempt: $i of 10.\n")
;
        sleep(1);
        $i++;
        if ($i > 10) {
                log_error("Timeout waiting for openvpn interface $devname to come up.\n");
                break;
        }
}
```

That alleviates our problem but causes any call of that function to take 6-7 seconds on our test pfSense instance (1 virtual CPU: AMD @ 2.8 GHz).

Would it be better to put the loop in rc.newwanip[v6]? Do you think a 10-second time out will be enough? Do you have ideas for a way to do this without a timeout loop?

#5 - 01/30/2014 11:19 PM - Jason Crowley

I performed more testing with different configurations and locations for the wait loop today. My plan was to find where rc.newwanip was executed in the process of launching the openvpn process, and make sure it didn't start until the link was actually up. It appears that rc.newwanip is being called when the following code is executed from /usr/local/sbin/ovpn-linkup.

```
/usr/local/sbin/pfSctl -c "interface newip $1"
```

The ovpn-linkup script is called by the openvpn process when it first establishes the link. Inserting the wait loop in that script was useless since the ovpns1 interface doesn't actually come online until after that script has completed and the openvpn process daemonizes.

It appears to me that my patch for openvpn.inc is adequate to resolve this bug. Ermal and others, would you agree? Do you feel there is anything else I need to do prior to merging this into the code base for further testing? Thanks!

#6 - 02/11/2014 05:09 PM - Chris Buechler

- Target version deleted (2.1.1)

- Affected Version changed from 2.1 to All

Files

services.inc.patch	945 Bytes	01/22/2014	Micah Mitchell
openvpn.inc.patch	981 Bytes	01/24/2014	Jason Crowley