pfSense - Bug #4317

firewall_edit_nat.php - memory exhaustion on 32 bit with VIP range

01/27/2015 01:28 PM - Mogamat Abrahams

Status:	Resolved	**Start date:**	01/27/2015
Priority:	High	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Web Interface	**Estimated time:**	0.00 hour
Target version:	2.2.1		
Plus Target Version:		**Affected Version:**	2.2
Release Notes:	Default	**Affected Architecture:**	i386

Description

Hi,

After upgrade to 2.2, experience memory limit errors even after increasing php memory_limit :

_Crash report begins. Anonymous machine information:

i386
10.1-RELEASE-p4
FreeBSD 10.1-RELEASE-p4 #0 36d7dec(releng/10.1)-dirty: Thu Jan 22 15:12:38 CST 2015
root@pfsense-22-i386-builder:/usr/obj.i386/usr/pfSensesrc/src/sys/pfSense_SMP.10

Crash report details:

PHP Errors:
[27-Jan-2015 21:17:45 Africa/Harare] PHP Fatal error: Allowed memory size of 536870912 bytes exhausted (tried to allocate 534773760 bytes) in /usr/local/www/firewall_nat_edit.php on line 692

If i comment out line 681 to 697, the foreach dealing with Virtual Ip's, I can now at least get into the NAT editing screen. I have no virtual ip's defined on the system.

History

#1 - 01/27/2015 02:18 PM - Chris Buechler

- Status changed from New to Feedback

not a replicable circumstance.

In order for the situation as described to occur, you have to have some kind of configuration under <virtualip> in your config, otherwise it'd completely skip the code whose removal you're saying fixes the issue. Open a backup of your config in a text editor and search for <virtualip>, what do you have from <virtualip> to </virtualip>?

#2 - 01/28/2015 03:03 AM - Mogamat Abrahams

You are right, must have missed it due to fatigue, although I do remember removing this before upgrading to 2.2.

<virtualip>
<vip>
<mode>proxyarp</mode>
<interface>opt2</interface>
<descr><![CDATA[NAT VIP]]></descr>
<type>network</type>
<subnet_bits>29</subnet_bits>
<subnet>197.xx.xx.131</subnet>
</vip>
</virtualip>

#3 - 01/28/2015 03:22 AM - Alejandro Olivan

I got exactly the same situation, so may I at least help consistently confirming the issue existence:

I have stopped upgrading deployed pfsense appliances after outbound NAT php error happens as described.
The issue has appeared on every upgraded router which had several virtual, public , IPs, assigned to the WAN interface and manual outbound rules where set to deal with outbound traffic.
I'm not sure whether this also occurs on appliances with ARP proxy alias on LAN interfaces, but at least I'm aware of the issue present on one appliance whit this setup.
The issue seems to exist on 100% upgraded appliances...but I have to investigate.
In every case the virtual IPs were part of the configuration, so no surprises with the xmls, they show <virtualip> fields as expected.

#4 - 01/28/2015 03:57 AM - Phillip Davis

Numbers like that work fine for me - e.g. subnet 197.1.2.131 subnet_bits 29
It build a correct list of 8 addresses.
I guess you did not really have "197.xx.xx.131" in the config, but have just obfuscated with "xx" some actual numbers of the public IP address block.
So I do not see how it was sent into a huge or infinite loop there.

#5 - 01/28/2015 04:20 AM - Alejandro Olivan

Here i paste relevant part of one upgraded router setup.
This particular one has a mixture of virtualIPs, may this help reproduce it.

```
&lt;virtualip&gt;
        &lt;vip&gt;
            &lt;mode&gt;proxyarp&lt;/mode&gt;
            &lt;interface&gt;lan&lt;/interface&gt;
            &lt;descr&gt;&lt;![CDATA[proxy ARP requests to SubnetXXX]]&gt;&lt;/descr&gt;
            &lt;type&gt;network&lt;/type&gt;
            &lt;subnet_bits&gt;24&lt;/subnet_bits&gt;
            &lt;subnet&gt;192.168.240.0&lt;/subnet&gt;
        &lt;/vip&gt;
        &lt;vip&gt;
            &lt;mode&gt;proxyarp&lt;/mode&gt;
            &lt;interface&gt;lan&lt;/interface&gt;
            &lt;descr&gt;&lt;![CDATA[catch local packets for somehost]]&gt;&lt;/descr&gt;
            &lt;type&gt;single&lt;/type&gt;
            &lt;subnet_bits&gt;32&lt;/subnet_bits&gt;
            &lt;subnet&gt;192.168.192.13&lt;/subnet&gt;
        &lt;/vip&gt;
        &lt;vip&gt;
            &lt;mode&gt;ipalias&lt;/mode&gt;
            &lt;interface&gt;wan&lt;/interface&gt;
            &lt;descr&gt;&lt;![CDATA[Catch IP traffic to example.net in WAN iface]]&gt;&lt;/descr&gt;
            &lt;type&gt;single&lt;/type&gt;
            &lt;subnet_bits&gt;32&lt;/subnet_bits&gt;
            &lt;subnet&gt;aaa.bbb.ccc.ddd&lt;/subnet&gt;
        &lt;/vip&gt;
    &lt;/virtualip&gt;
```

...Where obviously aaa.bbb.ccc.ddd appears as a real public IP address.

Best regards

#6 - 01/28/2015 07:07 AM - Jim Pingle

Are you all on i386?

I could see that loop going out of control due to #4318, source:usr/local/www/firewall_nat_edit.php#L687

#7 - 01/28/2015 08:06 AM - Mogamat Abrahams

Yep, on i386 Kernel.
Dont see a way to change architecture doing an auto upgrade and several machines are remote.

Any workarounds/Patches?

#8 - 01/28/2015 08:09 AM - Jim Pingle

A fix was just posted for #4318, apply that fix and try this again, I suspect it will work fine. If so, we can close this ticket out as a duplicate of #4318

#9 - 01/28/2015 08:10 AM - Alejandro Olivan

yes, correct, everything i386 on our side...

#10 - 01/28/2015 08:17 AM - Phillip Davis

Yes - I had been checking the code on a 64-bit system accidentally. Now I am at home with my Alix it all goes wrong:

```
$x1 = gen_subnet("197.1.2.131","29");
var_dump($x1);
$x2 = ip2long32($x1);
var_dump($x2);
$y1 = gen_subnet_max("197.1.2.131","29");
var_dump($y1);
$y2 = ip2long32($y1);
var_dump($y2);
$z2 = $y2 - $x1;
var_dump($z2);
```

```
string(11) "197.1.2.128"
int(-989789568)
string(15) "255.255.255.255"
int(-1)
float(-198.1)
```

So even converting and IP address like 197.1.2.128 to an int using ip2long32 will end up with a negative number. Any address in the 2nd half, from 128.0.0.0 onward is going to have this problem.

I will try with the fix now...

#11 - 01/28/2015 08:28 AM - Phillip Davis

Sorry - the last subtraction in my code above should have been "$z2 = $y2 - $x2" - so ignore the rubbish "float(-198.1)" that was there.
With fix from #4318 the output is:

string(11) "197.1.2.128"
int(-989789568)
string(11) "197.1.2.135"
int(-989789561)
int(7)

which is correct - the difference between the bottom and top address is 7, for a total of 8 addresses in the subnet. It works even though the (int) turn out to be negative here.

#12 - 01/28/2015 08:36 AM - Phillip Davis

Note:
$x2 = ip2long32("127.255.255.255");
var_dump($x2);
$y2 = ip2long32("128.0.0.0");
var_dump($y2);
$z2 = $y2 - $x2;
var_dump($z2);

int(2147483647)
int(-2147483648)
float(-4294967295)

If we convert 2 addresses either side of the 128.0.0.0 boundary into int32 and then try to subtract to find the difference, we get the wrong answer.
With subnets, the bottom and top address of a subnet are always in the same half of the address space, except for the "/0" case of the whole internet.
As long (pardon the pun) as comparison/subtraction is always done with addresses that are both in the same half of the address space, then ip2long32 is OK.

#13 - 02/20/2015 08:40 PM - Chris Buechler

- Subject changed from firewall_edit_nat.php - memory exhaustion and crash in php to firewall_edit_nat.php - memory exhaustion on 32 bit with VIP range

- Status changed from Feedback to Confirmed

#14 - 02/23/2015 03:47 PM - Ermal Luçi

- Status changed from Confirmed to Feedback

This seems to work now!

#15 - 02/23/2015 10:18 PM - Phillip Davis

Yes, agree with Ermal.
My comments were just to document/mention the negative numbers behavior on 32-bit systems. I am not sure what to do about it, if anything, but it is a separate theoretical issue that only very rarely would effect anything.

#16 - 02/27/2015 04:16 AM - Chris Buechler

- Status changed from Feedback to Resolved