Suricata - Bug #2806

Parallel DNS queries dropped when using same socket

02/10/2019 05:08 AM - Rob Mosher

Status:	New		
Priority:	Normal		
Assignee:			
Category:			
Target version:			
Affected Versions:	4.0beta1, 4.0.0, 4.0.1, 3.2.5, 4.0.2/4.0.3, 4.0.4, 4.1beta1, 4.0.5, 4.1rc1, 4.1rc2, 4.0.6, 4.1, 4.1.1, 4.1.2, 4.0.7, 4.1.3	**Difficulty:**	
Effort:		**Label:**	

Description

Suricata is dropping DNS queries and not logging the drop when concurrent queries are performed using the same socket. This is standard behavior since glibc 2.10. This causes DNS resolution to be very slow.

The resolv.conf option 'single-request' can work around the issue, but that also delays DNS resolution slightly, though not nearly as bad.

The dns requests below are triggered using 'ping' which I believe calls gethostbyname

Initial queries sent concurrently using same socket:

```
 00:00:00.000000 IP 192.168.1.224.40887 > 192.168.1.1.53: 9647+ A? google.com. (28)
 00:00:00.000020 IP 192.168.1.224.40887 > 192.168.1.1.53: 52827+ AAAA? google.com. (28)
```

Only the A query receives a response. Glibc continues to wait for the AAAA response.

```
 00:00:00.001318 IP 192.168.1.1.53 > 192.168.1.224.40887: 9647 1/0/0 A 172.217.9.142 (44)
```

Attempt number 2 falls back to single request method, which seems to work just fine, but there's a 5 second delay before the retry.

```
 00:00:05.001888 IP 192.168.1.224.40887 > 192.168.1.1.53: 9647+ A? google.com. (28)
 00:00:05.003131 IP 192.168.1.1.53 > 192.168.1.224.40887: 9647 1/0/0 A 172.217.9.142 (44)
 00:00:05.003226 IP 192.168.1.224.40887 > 192.168.1.1.53: 52827+ AAAA? google.com. (28)
 00:00:05.004345 IP 192.168.1.1.53 > 192.168.1.224.40887: 52827 1/0/0 AAAA 2607:f8b0:4000:813::200e (56)
```

If we use the 'single-request' option in resolv.conf to not send multiple queries at the same time, there is minimal delay and no dropped query.

```
 00:00:00.000000 IP 192.168.1.224.53244 > 192.168.1.1.53: 11413+ A? google.com. (28)
 00:00:00.001201 IP 192.168.1.1.53 > 192.168.1.224.53244: 11413 1/0/0 A 172.217.9.142 (44)
 00:00:00.001295 IP 192.168.1.224.53244 > 192.168.1.1.53: 32038+ AAAA? google.com. (28)
 00:00:00.002325 IP 192.168.1.1.53 > 192.168.1.224.53244: 32038 1/0/0 AAAA 2607:f8b0:4000:813::200e (56)
```

Both concurrent queries are seen when logging from Suricata, but no AAAA reply, as the second query get's dropped, through Suricata definitely receives it and even logs it.

```
-- query
02/09/2019-23:57:09.942921 192.168.1.224 -> 192.168.1.1 A? google.com
-- query
02/09/2019-23:57:09.942927 192.168.1.224 -> 192.168.1.1 AAAA? google.com
-- query+answer
```

```
02/09/2019-23:57:09.943823 192.168.1.224 -> 192.168.1.1 A? google.com
02/09/2019-23:57:09.943823 192.168.1.1 -> 192.168.1.224 A: google.com: 216.58.194.110
```

5 seconds later on the single query retry...

```
-- query
02/09/2019-23:57:14.947594 192.168.1.224 -> 192.168.1.1 A? google.com
-- query+answer
02/09/2019-23:57:14.948331 192.168.1.224 -> 192.168.1.1 A? google.com
02/09/2019-23:57:14.948331 192.168.1.1 -> 192.168.1.224 A: google.com: 216.58.194.110
-- query
02/09/2019-23:57:14.948974 192.168.1.224 -> 192.168.1.1 AAAA? google.com
-- query+answer
02/09/2019-23:57:14.949637 192.168.1.224 -> 192.168.1.1 AAAA? google.com
02/09/2019-23:57:14.949637 192.168.1.1 -> 192.168.1.224 AAAA: google.com: 2607:f8b0:4002:0c06:0000
:0000:0000:0064
```

From a remote dns server's point of view, I have confirmed the second query gets dropped, and only the A query makes it through until the single query retry 5 seconds later.

```
 00:00:00.000000 IP 24.191.114.206.32749 > 209.51.175.25.53: 15540+ A? test.somedomain. (31)
 00:00:00.000141 IP 209.51.175.25.53 > 24.191.114.206.32749: 15540 NXDomain* 0/1/0 (96)
 00:00:05.005913 IP 24.191.114.206.50912 > 209.51.175.25.53: 57854+ A? test.somedomain. (31)
 00:00:05.006070 IP 209.51.175.25.53 > 24.191.114.206.50912: 57854 NXDomain* 0/1/0 (96)
 00:00:05.021736 IP 24.191.114.206.44855 > 209.51.175.25.53: 40938+ AAAA? test.somedomain. (31)
 00:00:05.021960 IP 209.51.175.25.53 > 24.191.114.206.44855: 40938 NXDomain* 0/1/0 (96)
```

No other changes made aside from stopping Suricata, this begins working normally as it no longer drops the second query

```
 00:00:00.000000 IP 192.168.1.224.34752 > 192.168.1.1.53: 5322+ A? google.com. (28)
 00:00:00.000019 IP 192.168.1.224.34752 > 192.168.1.1.53: 59535+ AAAA? google.com. (28)
 00:00:00.032286 IP 192.168.1.1.53 > 192.168.1.224.34752: 59535 1/4/4 AAAA 2607:f8b0:4000:813::200
e (240)
 00:00:00.033168 IP 192.168.1.1.53 > 192.168.1.224.34752: 5322 1/4/4 A 172.217.9.142 (228)
```

Related issues:

Related to Bug #2435: Suricata 4.0.3 in IPS mode seems to discard some DNS re...	**New**

History

#1 - 02/12/2019 12:23 PM - Victor Julien

How are you running Suricata? Does Suricata log anything?

#2 - 02/12/2019 06:33 PM - Rob Mosher

Victor Julien wrote:

> How are you running Suricata? Does Suricata log anything?

IPS mode using NFQ. There is Suricata log output detailed in the bug report. DNS logging was done via LUA.

#3 - 02/13/2019 09:54 AM - Victor Julien

What does your NFQ setup look like? Can you share the iptables/nftables rules as well as the Suricata nfq config?

#4 - 02/13/2019 08:43 PM - Rob Mosher

Victor Julien wrote:

> What does your NFQ setup look like? Can you share the iptables/nftables rules as well as the Suricata nfq config?

Just noting, this config works fine and has for some time with all other traffic. This DNS issue has been present for a very long time. I mentioned it a few times on IRC and email, but finally got around to creating a bug report for it.

These are the rules that match the DNS traffic destined to the router which is running Suricata.

```
iptables -t mangle -I INPUT -i br-lan -m mark ! --mark 0x1000/0x1000 \
-m connbytes ! --connbytes 1000000 --connbytes-dir both --connbytes-mode bytes \
-j NFQUEUE --queue-bypass

iptables -t mangle -I OUTPUT -o br-lan -m mark ! --mark 0x1000/0x1000 \
-m connbytes ! --connbytes 1000000 --connbytes-dir both --connbytes-mode bytes \
-j NFQUEUE --queue-bypass
```

general forwarding rule

```
iptables -t mangle -I FORWARD \
-m mark ! --mark 0x1000/0x1000 \
-m connbytes ! --connbytes 1000000 --connbytes-dir both --connbytes-mode bytes \
-j NFQUEUE --queue-bypass
```

This is the nfq config.

```
nfq:
  mode: repeat
  repeat-mark: 4096
  repeat-mask: 4096
```

The traffic is confirmed reaching Suricata based on log output, however Suricata is dropping the second dns query as it never makes it to the application (dnsmasq). It seems Suricata cannot handle multiple open queries on the same socket. Once one query is completed, it handles the second appropriately.

This is the output from dnsmasq with query logging enabled and Suricata running. You can see the delay here as the parallel AAAA query never makes it through.

```
parallel resolution.  There should be an A and AAAA query here.
Wed Feb 13 15:31:14 2019 daemon.info dnsmasq[2620]: 72 192.168.1.224/40269 query[A] google.com from 192.168.1.
224
Wed Feb 13 15:31:14 2019 daemon.info dnsmasq[2620]: 72 192.168.1.224/40269 forwarded google.com to 209.51.175.
25

5 seconds later, sequential resolution
Wed Feb 13 15:31:19 2019 daemon.info dnsmasq[2620]: 104 192.168.1.224/40269 query[A] google.com from 192.168.1
.224
Wed Feb 13 15:31:19 2019 daemon.info dnsmasq[2620]: 104 192.168.1.224/40269 forwarded google.com to 209.51.175
.25
Wed Feb 13 15:31:19 2019 daemon.info dnsmasq[2620]: 105 192.168.1.224/40269 query[AAAA] google.com from 192.16
8.1.224
Wed Feb 13 15:31:19 2019 daemon.info dnsmasq[2620]: 105 192.168.1.224/40269 forwarded google.com to 209.51.175
.25
Wed Feb 13 15:31:19 2019 daemon.info dnsmasq[2620]: 104 192.168.1.224/40269 reply google.com is 172.217.9.174
Wed Feb 13 15:31:19 2019 daemon.info dnsmasq[2620]: 105 192.168.1.224/40269 reply google.com is 2607:f8b0:4002
:c08::71
```

Simply stopping Suricata fixes this, and you can see the difference. The parallel AAAA query arrives as expected

```
Wed Feb 13 15:35:17 2019 daemon.info dnsmasq[2829]: 15 192.168.1.224/38629 query[A] google.com from 192.168.1.
224
Wed Feb 13 15:35:17 2019 daemon.info dnsmasq[2829]: 15 192.168.1.224/38629 forwarded google.com to 209.51.175.
25
Wed Feb 13 15:35:17 2019 daemon.info dnsmasq[2829]: 16 192.168.1.224/38629 query[AAAA] google.com from 192.168
.1.224
Wed Feb 13 15:35:17 2019 daemon.info dnsmasq[2829]: 16 192.168.1.224/38629 forwarded google.com to 209.51.175.
25
Wed Feb 13 15:35:17 2019 daemon.info dnsmasq[2829]: 15 192.168.1.224/38629 reply google.com is 172.217.12.206
Wed Feb 13 15:35:17 2019 daemon.info dnsmasq[2829]: 16 192.168.1.224/38629 reply google.com is 2607:f8b0:4002:
c08::71
```

#5 - 02/14/2019 01:31 PM - Eric Leblond

Is it possible to get a dump of the ruleset ? I'm wondering if you could have a CONNMARK rule somewhere that could mess thing up with the second packet.

#6 - 02/14/2019 09:02 PM - Rob Mosher

Eric Leblond wrote:

> Is it possible to get a dump of the ruleset ? I'm wondering if you could have a CONNMARK rule somewhere that could mess thing up with the second packet.

I commented out all the rule files except for one, which contains a single entry. The issue persists. Have you tried to reproduce? Should be fairly easy.

```
(DEBIAN)root@rooter:/etc/suricata# suricata -c suricata-rooter.yaml --dump-config | grep rule-files
rule-files = (null)
rule-files.0 = iprep.rules

(DEBIAN)root@rooter:/etc/suricata# cat rules/iprep.rules

# ----- Begin iprep Rules Category ----- #

# -- Begin GID:0 Based Rules -- #

alert ip any any -> any any (msg:"IPREP Blacklist Match"; metadata:policy balanced-ips alert; iprep:dst,Bad,=,
99; classtype:bad-unknown; sid:99; rev:1;)
```

#7 - 02/18/2019 11:14 AM - Victor Julien

- Related to Bug #2435: Suricata 4.0.3 in IPS mode seems to discard some DNS requests added

#8 - 02/18/2019 11:15 AM - Victor Julien

I think Eric was referring to the iptables rules.

Can you still reproduce this if Suricata+nfq is in accept mode instead of repeat mode?

#9 - 02/21/2019 05:07 AM - Rob Mosher

Victor Julien wrote:

> I think Eric was referring to the iptables rules.
>
> Can you still reproduce this if Suricata+nfq is in accept mode instead of repeat mode?

Same behavior in accept mode