strongSwan - Issue #964

Memory leak in charon - with libipsec encryption/decryption

22.05.2015 09:18 - Sriram Dharwadkar

Status:	Closed		
Priority:	Normal		
Assignee:	Tobias Brunner		
Category:	libipsec		
Affected version:	5.3.0	**Resolution:**	

Description

Hi,

I m using strongswan-5.3.0 for tunnel establishment & libipsec encryption/decryption.

In our lab I tested a scenario where I sent,

1. 20Mbps uplink traffic from the device where libipsec is running, to a remote server.
2. 80Mbps downlink traffic from the remote server to the device where libipsec is running.

These two traffics are sent simultaneously using iperf tool.
I see that charon's memory usage gradually shoots up, it goes upto 651MB before the device crashes with out of memory.

Top output

```
PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
3863 root      18  -2  651m 513m 1976 S  132 46.2   3:17.09 charon
4421 apps      20   0  126m  14m 4848 S   46  1.3   0:36.59 rrc_entity
4419 apps      20   0  138m 103m 1948 S   31  9.3   0:45.75 egtpuApp
523 apps      20   0 46496 3828 2928 R   29  0.3   0:44.82 fsmDevLogProxy
660 root      20   0 99.0m 4076 2300 S    8  0.4   0:22.47 PlatformOam
5979 root      20   0     0    0    0 S    6  0.0   0:06.23 kworker/0:2
609 apps      20   0  9508 2628 2152 S    4  0.2   0:13.04 fsmSwitchMgr
```

Attaching the file containing necessary information.
Please let me know if there is any information required to resolve this.

Regards,
Sriram.

History

#1 - 22.05.2015 12:25 - Tobias Brunner

- Description updated

- Status changed from New to Feedback

As far as I can tell there are no memory leaks in libipsec. But our userland IPsec implementation is currently not intended for tunnels with constant high traffic (or very large bursts), as it's not particularly optimized for performance.

Two threads handle encryption/decryption in parallel, but other than that processing is not parallelized. Lookups for SAs and policies are also not specially optimized. And the queue for in- and outbound packets is not limited, so if packets can't be processed fast enough they end up getting queued up, requiring more and more memory (which is probably the "memory leak" you are seeing). You could try the two patches in the *libipsec-queue* branch, which implement a "Controlled Delay" (CoDel) queue to drop packets if processing does not happen fast enough, but that won't increase throughput, and will still require a considerable amount of memory (packets are still copied to userland before they can be dropped). The actual maximum throughput may be determined by using TCP with iperf instead of UDP with a fixed bandwidth.

For high-throughput tunnels using the IPsec implementation in the Linux kernel is recommended, optionally with pcrypt enabled.

#2 - 08.07.2015 11:19 - Tobias Brunner

- Status changed from Feedback to Closed

- Assignee set to Tobias Brunner

Files

ipsec-info.txt	11.2 KB	22.05.2015	Sriram Dharwadkar