Retrospective on upstreaming VPN in AOSP

Sam Protsenko

04 April 2019

Linaro connect
Bangkok 2019

Agenda

1. Adopting Upstream VPN in AOSP (Retrospective)
2. Fixing L2TP in Kernel

Adopting Upstream VPN in AOSP

Previous Presentation

- ”Using VPN implementation from upstream kernel in Android”
  - https://connect.linaro.org/resources/yvr18/sessions/yvr18-501/
  - https://s3.amazonaws.com/connect.linaro.org/yvr18/videos/yvr18-501.mp4
  - https://s3.amazonaws.com/connect.linaro.org/yvr18/presentations/yvr18-501.pdf
- In this presentation we will focus more on recent L2TP work in kernel

The problem

Protocol	Android kernel	Upstream kernel
PPTP	drivers/net/ppp/pppopns.c (PX_PROTO_OPNS)	drivers/net/ppp/pptp.c (PX_PROTO_PPTP)
L2TP	drivers/net/ppp/pppolac.c (PX_PROTO_OLAC)	net/l2tp/l2tp_ppp.c (PX_PROTO_OL2TP)

Now that upstream kernel implementation exists, we can adopt it.

Retrospective Timeline

02 Feb 2017	Initial patches submitted
26 Jul 2018	L2TP patches are merged
30 Aug 2018	PPTP patches are merged
12 Sep 2018	Kernel config patch is merged
03 Oct 2018	L2TP regression found by Google
15 Dec 2018	First submission of the fix to upstream
21 Dec 2018	PFC patches are merged to net-next
24 Jan 2018	Backported in Android kernels

Patches: external/ppp

- pppd: Remove obsolete way of receiving args from mtpd
- pppd: Enable plugin support in pppd
- pppd: Convert Android.mk to Android.bp
- pppd: Add pppol2tp-android plugin
- pppd: Fix pppol2tp-android.so build
- pppd: Add rules for building the pppol2tp-android plugin
- pppd: Add pppopptp-android plugin

Patches: external/mtpd

- mtpd: Remove obsolete way of passing args to pppd
- mtpd: l2tp: Fix endianness issues in log prints
- mtpd: Use L2TP implementation from mainline kernel
- mtpd: pptp: Fix endianness issues in log prints
- mtpd: Use PPTP implementation from upstream kernel

Patches: kernel/configs

- Enable L2TP and PPTP from upstream kernel:

```
- CONFIG_PPPOLAC=y
- CONFIG_PPPOPNS=y
+ CONFIG_PPPOL2TP=y
+ CONFIG_PPTP=y
```

Android kernel delta

```
drivers/net/ppp/Kconfig       |  17 ---
drivers/net/ppp/Makefile      |   2 -
drivers/net/ppp/pppolac.c     | 450 ----------------
drivers/net/ppp/pppopns.c     | 429 ----------------
include/linux/if_pppolac.h    |  33 -----
include/linux/if_pppopns.h    |  32 -----
include/linux/if_pppox.h      |  21 -----
include/uapi/linux/if_pppox.h |   6 +-

8 files changed, 1 insertion(+), 989 deletions(-)
```

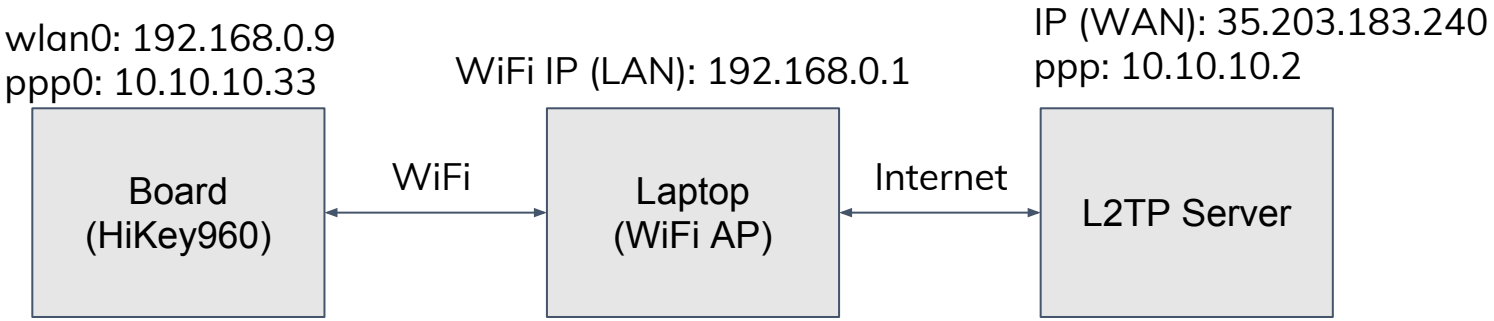
Fixing L2TP in Kernel

What’s wrong

- Testing in Google lab revealed the issue:
- ping within the VPN network fails
- Happens only on particular server configuration
  - Why it doesn’t happen with my server?

- Let’s try to sniff that “ping” with Wireshark:
  - ...using Google testing VPN server
  - Collect dump with `tcpdump` on device
  - Open that dump on PC with Wireshark
  - Decrypt ESP packets in Wireshark

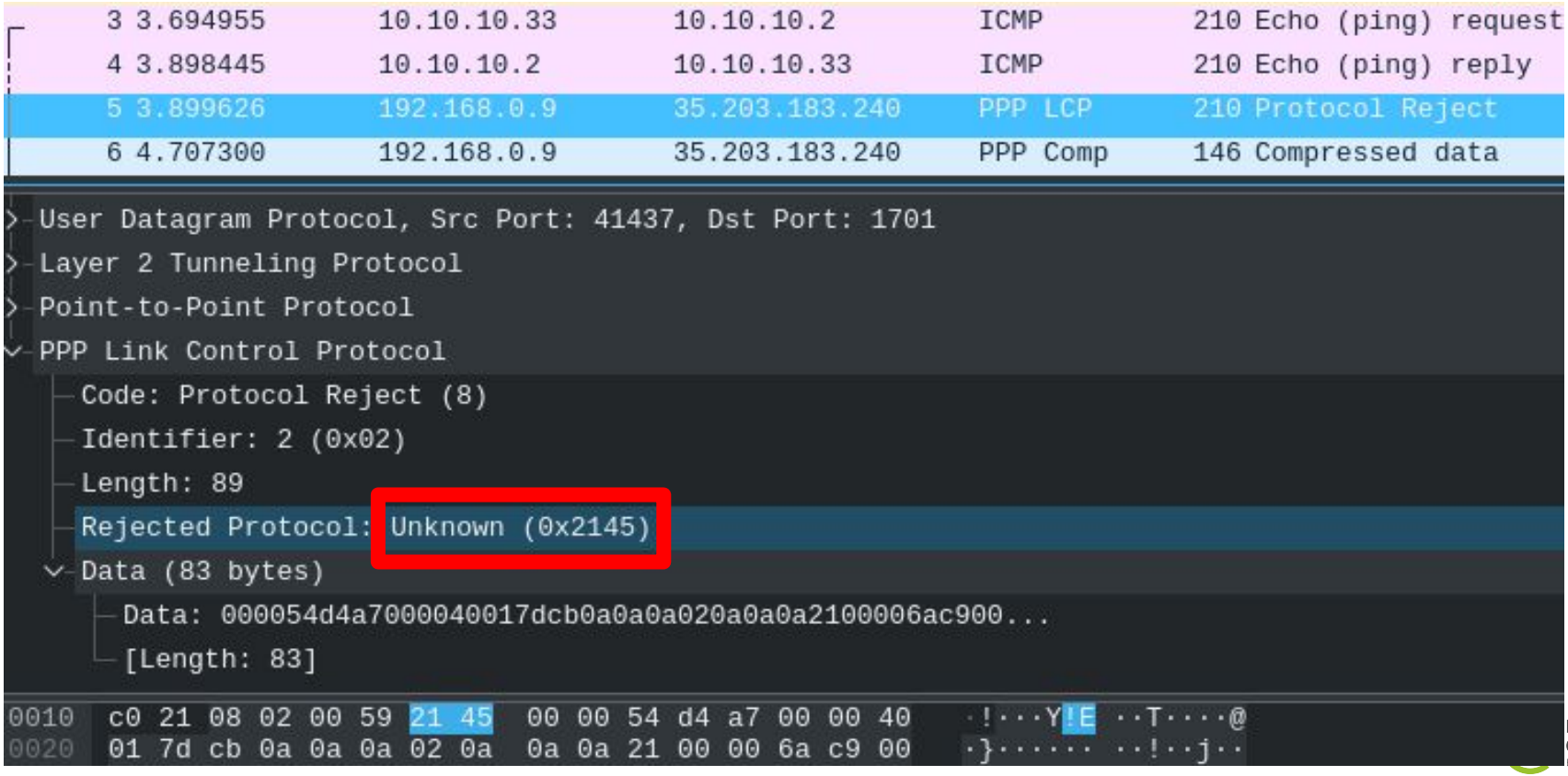
Investigating



- Collect ping dump (from device), once VPN session is established:

```
# tcpdump -i wlan0 -s 65535 -w ping_dump.pcap &
# ping -I ppp0 10.10.10.2                    // doesn't work
^C
```

Decrypted Sniffed Packets on Ping



Why is that?

Issue:

- This 0x2145 code is actually malformed
- It should be 0x0021 for IPv4 protocol
- And "45" part actually belongs to next field

Root cause:

- That’s because PFC (Protocol Field Compression) is enabled on Google server
- Turns out Linux kernel L2TP driver wasn’t able to handle PFC
- ...hackish way to “fix” it is to provide “`nopcomp`” option to pppd
- **Proper fix**: add PFC support to L2TP driver and upstream it.

Patches

1. **l2tp: Add protocol field decompression**
   - Seems like PFC is implemented for PPTP, but not implemented for L2TP
   - Do the same for L2TP (if PFC is enabled)

```
static void pppol2tp_recv(...)
{
    ...
+   /* Decompress protocol field if PFC is enabled */
+   if ((*skb->data) & 0x1)
+       *(u8 *)skb_push(skb, 1) = 0;
```

Patches (cont'd)

2. **ppp: Move PFC decompression to PPP generic layer**
   - ...where it actually belongs
   - As a consequence, it also enables PFC in PPPoE, etc

Remove decompression in:

- net/l2tp/l2tp_ppp.c
- drivers/net/ppp/pptp.c
- drivers/net/ppp/ppp_synctty.c
- drivers/net/ppp/ppp_async.c

Add decompression in:

+ drivers/net/ppp/ppp_generic.c

Backporting to stable/Android kernels

- No need to send to linux-stable (feature vs bugfix)
- Sent patches to Android kernels: 4.9, 4.14, 4.19 (Android-Q)
  - UPSTREAM: ppp: Move PFC decompression to PPP generic layer
  - UPSTREAM: l2tp: Add protocol field decompression

Some notes on patches in Android kernel:

- Tags in patch subject:
  - “UPSTREAM” - clean cherry-pick
  - “BACKPORT” - if conflicts were resolved

Kernel Networking Unit Tests

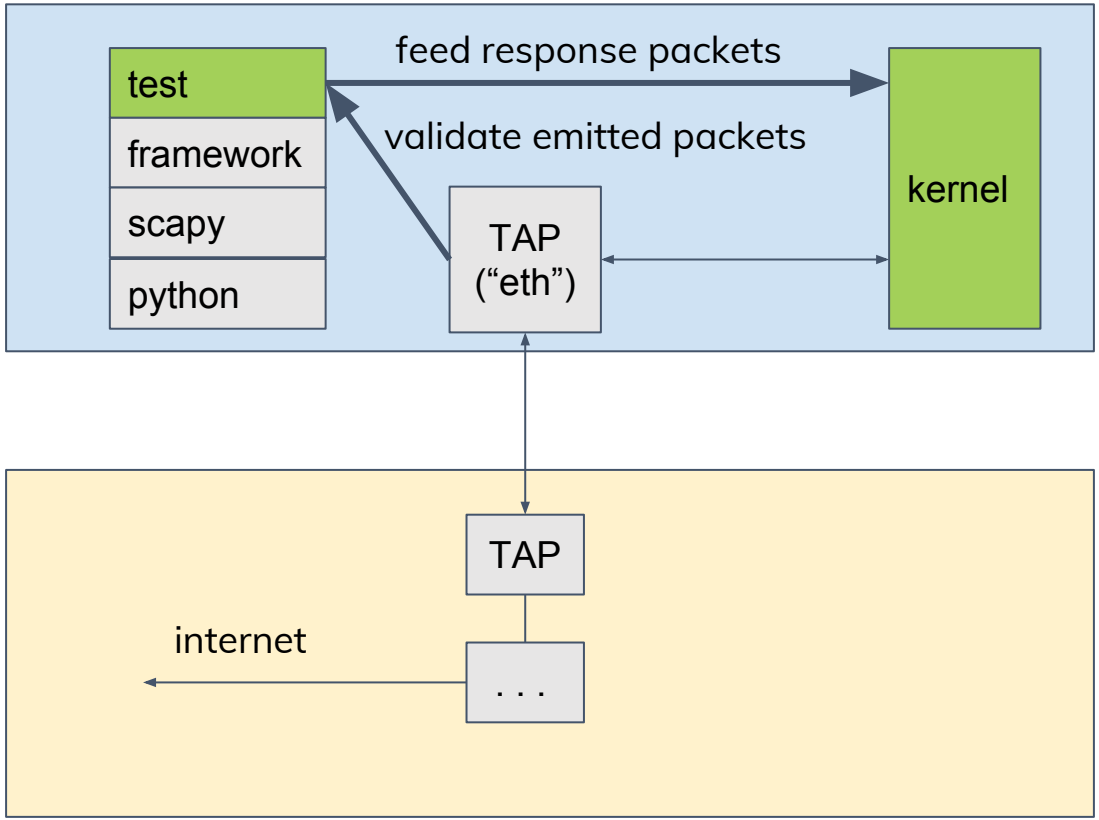
- Network patches need test covering in Android kernels 4.19+
- “net_test” framework exists in AOSP:
  - kernel/tests/net/test/
  - Documentation: https://source.android.com/devices/architecture/kernel/network_tests
  - README: https://android.googlesource.com/kernel/tests/+/master/net/test/README
- Those tests run both:
  - in a VM (for rapid development)
  - and in VTS on devices for conformance tests
- The tests use Python and scapy and make it very easy to validate network packets emitted by the kernel and to feed it responses.
- Testable kernel is being run in User-Mode Linux (UML)

Kernel Networking Unit Tests (cont'd)

```
./run_net_test.sh <test>
```

1. It compiles your kernel to UML binary
2. Starts it on your host, like VM (kernel + Debian rootfs)
3. Test will be mounted from host
4. Starts specified `<test>` from init process

Kernel Networking Unit Tests (cont'd)



Thank you!

semen.protsenko@linaro.org

Linaro
connect
Bangkok 2019