TCPDUMP & LIBPCAP Change/Fix Log

F1: [libpcap] Remote Packet Capture Daemon (RPCAPD) Integer Overflow Leads to Heap Buffer Overflow

VERIFIED FIX. See: https://github.com/the-tcpdump-group/libpcap/commit/0b6a6fd8f347e298e18a02266f879c28f97199e9

F2: [tcpdump] Integer Arithmetic Error can Lead to Heap Buffer Overflow When Processing Large Files

UNVERIFIED FIX. Fix claimed, but no record of exactly which commit. See: https://github.com/the-tcpdump-group/libpcap/blob/bdf1cb551e589f0d219c054af2d146e6d5d219c6/CHANGES

F3: [tcpdump] Out of Bounds Read Processing BGPTYPE_MP_REACH_NLRI Packets

VERIFIED FIX. See: https://github.com/the-tcpdump-group/tcpdump/commit/13d52e9c0e7caf7e6325b0051bc90a49968be67f

F4: [tcpdump] Out of Bounds Read Processing IPv6 OSPF Packets

VERIFIED FIX. See: https://github.com/the-tcpdump-group/tcpdump/commit/e01c9bf76740802025c9328901b55ee4a0c49ed6

F5: [libpcap] Berkeley Packet Filter (BPF) Optimization Can Cause Stack Exhaustion

NO INFO

F6: [tcpdump] Out of Bounds Accesses in Server Message Block (SMB) Printer in print_trans2()

UNVERIFIED PARTIAL FIX. Self-described "Partial fix", no commit referenced: https://github.com/the-tcpdump-group/tcpdump/blob/tcpdump-4.9/CHANGES

F7: [tcpdump] Recursive Function Call Stack Exhaustion Processing SMB Packets in smb_fdata()

VERIFIED FIX. See: **https://github.com/the-tcpdump-group/tcpdump/commit/24182d959f661327525a20d9a94c98a8ec016778**

F8: [tcpdump] Unsafe Integer Arithmetic Can Lead to Heap Overflow in linkaddr_string()

NO FIX. New issue, no code path leads to exploitable condition, no CVE will be requested, future change planned regardless.

F9: [tcpdump] Out of Memory Crashes via Various Memory Leaks in addrtoname.c

NO FIX. Concurrent issue (https://github.com/the-tcpdump-group/tcpdump/issues/13), reproduced, no CVE yet, future fix planned.

F10: [tcpdump] Stack Exhaustion Processing BGPTYPE_ATTR_SET Packets

VERIFIED FIX. See:
https://github.com/the-tcpdump-group/tcpdump/commit/af2cf04a9394c1a56227c2289ae8da262828294a

F11: [libpcap] Remote Packet Capture Daemon Multiple Authentication Improvements

VERIFIED PARTIAL FIX. See:
https://github.com/the-tcpdump-group/libpcap/commit/484d60cbf7ca4ec758c3cbb8a82d68b244a78d58

For the first issue, the assessment team recommends utilizing Transport Layer Security (TLS) to encrypt the session end-to-end and prevent interception. The first of those has been addressed in the main branch by several commits by Cedric Cellier, adding TLS support for the rpcap control socket and for the data socket if TCP is being used (there's no DTLS support if UDP is used for the data socket). Those commits came from the pull request at: https://github.com/the-tcpdump-group/libpcap/pull/721

For the second issue, the assessment team recommends implementing mechanisms to hinder or prevent brute-force attacks against the authentication requests and having those mechanisms have a low-tolerance default threshold (perhaps five attempts) before initiating brute-force protection by increasing the time allowed between authentication attempts.

The second issue hasn't been addressed yet - it may require information to be saved in a file to handle attempts made to multiple processes.

F12: [libpcap] Remote Packet Capture Daemon Null Pointer Dereference Denial of Service

VERIFIED FIX. See:
https://github.com/the-tcpdump-group/libpcap/commit/437b273761adedcbd880f714bfa44afeec186a31

F13: [libpcap] Remote Packet Capture Daemon Allows Opening Capture URLs

VERIFIED FIX. See: **https://github.com/the-tcpdump-group/libpcap/commit/33834cb2a4d035b52aa2a26742f832a112e90a0a**

I8: [libpcap] Remote Packet Capture Daemon Parameter Reuse

VERIFIED FIX. See: https://github.com/the-tcpdump-group/libpcap/commit/617b12c0339db4891d117b661982126c495439ea