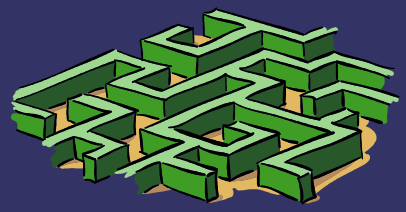
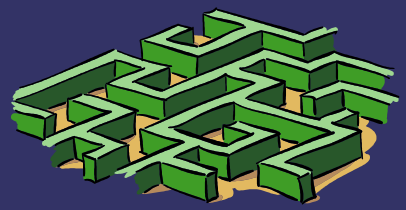
NCAP

Paul Vixie, ISC
with
Duane Wessels, Measurement Factory
July, 2007



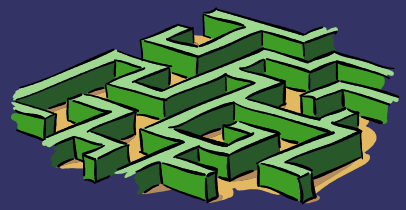
PCAP Pros and Cons

- PCAP works at the packet (interface) level
  - lots of useless link-level drivel in *.pcap files
  - every PCAP application has a big switch()
  - new link formats are not backward compatible
- Impedance mismatch w/ WAN analysis:
  - need IP (UDP) reassembly of large EDNS
  - need TCP reassembly (ordering, duplicates)



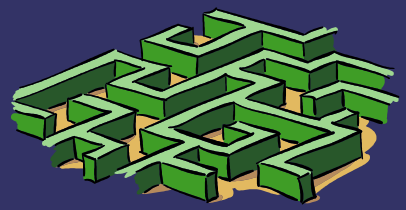
IP (UDP) Reassembly

- Challenges:
  - IP fragments only include UDP headers in first fragment
  - each user-mode reassembler has to ask BPF for *all* frags and then do flow isolation
- Dangers:
  - bad guys wanting to evade trivial analysis and detection can force large-EDNS
  - all known existing tools won't reassemble IP



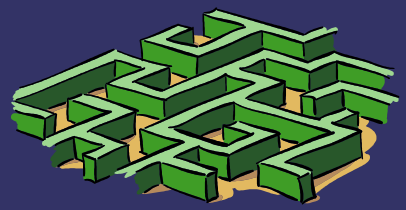
TCP Reassembly

- Challenges:
  - TCP segments can arrive out of order and/or be duplicated
- Dangers:
  - any transaction that falls back to TCP will probably be missed by all existing analysis tools



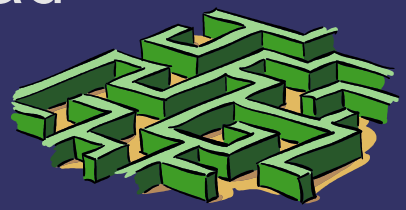
PCAP File Format Problems

- Link level information
  - When analyzing WAN data, the link level information is almost never interesting
  - Most PCAP analyzers skip it altogether
  - When combining PCAP streams, a unified output linktype has to be chosen
- Packet size problems
  - snaplen and packet size are 16-bit fields
  - DNS message length is also 16 bits
  - DNS+UDP+IP can be larger than 16 bits



NCAP File Format Proposal

- Universal format, no interface-specific data
  - All fields are 32-bit network byte order
  - Original wire-format headers (IP, UDP, TCP, etc) are simply discarded
- Enumerations
  - network: IP, IP6
  - transport: UDP, TCP
- Payloads
  - UDP: sport, dport, length, payload
  - TCP: sport, dport, length, offset, payload

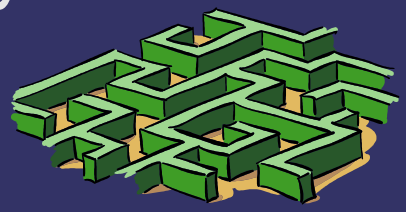


NCAP Library Proposal

- libncap would appear similar to libpcap
  - some evolution, fewer rough edges in the API
- libncap would probably be implemented using libpcap
  - pass the filters across to the kernel
  - grab the output from the kernel
  - reassemble packets/segments, strip headers
  - make and deliver ncap-format messages
- Challenges:
  - event handling is still nonportable, so merging input from many interfaces will be tricky

NCAP Kernel Module Proposal

- The kernel already has efficient packet and segment reassemblers
- An NCAP kernel module to tap into data as it passes into/outof the transport would be more efficient than doing it in user mode
- This would only work for locally sourced and locally destined traffic
- Use of the kernel module would be optional, and hidden behind the NCAP library API
- Not very compelling at the moment



Implications on Passive DNS

- For OARC passive DNS, it's nec'y to be able to rebroadcast centrally the data that is collected in many places
- Raw IP/UDP and IP/TCP isn't a good format for this – fragments, segments, etc
- Raw PCAP isn't a good format either
- We need messages (NCAP) not packets (PCAP) for this