Tranalyzer – Netflow extension

```c
#else L2PROTO == L2_L2TPV3
    if (newPacket.snapIpVersion < sizeof(l2tpv2Header_t)) return;
    if (((l2tpv2Header_t*)newPacket.layer2Header->type != 0) return;
    newPacket.layer2Type = L2_L2TPV3;
#endif

    for (i = 0; i < tranalyzer_plugins->num_of_func_pluginClaimLayer2Information; i++) {
        tranalyzer_plugins->pluginClaimLayer2Information[i](&newPacket);
    }
#endif ENABLE_VLAN_SCAN == 1

    switch (ntohs(((ethernetHeader_t*)newPacket.layer2Header->ether_type)) {
        case ETHERTYPE_IP:
            newPacket.layer3Header = (const 13Header_t*)((u_char*)packet + sizeof(ethernetHeader_t));
            newPacket.layer3Type = ETHERTYPE_IP;
            disassembleIPPacket(&newPacket);
            break;
        default:
            shape = (uint16_t*)newPacket.layer2Type;
            while (ntohs(shape->type) == 0)
                shape = (uint16_t*)((u_char*)newPacket.layer2Type + 10);

            newPacket.version = shape->version;
            newPacket.ipLength = shape->ipLength;
            newPacket.ipAddress = shape->ipAddress;
            newPacket.protocol = shape->protocol;
            newPacket.ttl = shape->ttl;
            return;
    }
#endif IGNORE_VLAN == 0
    else {
        return;
    }
#else
    if (ntohs(((ethernetHeader_t*)newPacket.layer2Header->ether_type) == ETHERTYPE_IP) {
        newPacket.layer3Header = (const 13Header_t*)((u_char*)packet + sizeof(ethernetHeader_t));
        newPacket.layer3Type = ETHERTYPE_IP;
        disassembleIPPacket(&newPacket);
    } else {
        return;
    }
#endif
```
“It's the network – go fix it!”
Features

- Command-line based → GUI: Traviz
- Extendable by plugins
- Fast and simple
- Practitioners: Anomaly and security related flags
- Researchers: Full Statistical and Packet Signal Analysis support
- Interfaces: Matlab, GnuPlot, SPSS, Excel etc.
For the Practitioners

- Known Netflow information (L2/L3/L4 information + VLAN, direction, time, number of packets or bytes, etc.)
- Min/max statistics of L3 and L4, packet and byte stream asymmetry
- Full TCP state-machine including malicious packet detection and flag aggregation with anomaly support
- ICMP aggregated type and code bitfields
- Number of distinct connections to neighbors
- Number of traffic channels between two hosts
Applications for practitioners

- Machine load indication by IPIID differences
- Flow quality: via TCP window size signal behavior
- IP and TCP aggregated option information
- Routing anomalies: via TTL
- Transmitted/Received bytes via TCP sequence and acknowledge number differences
Applications for practitioners

- Detect bottlenecks by finding top talkers
  - Helping to improve load balancing
- Detect packet flow asymmetries (Traffic loops)
- Detect network misconfiguration, such as packet filtering
For the Researchers

- Min/Max packet length, Mean packet length
- Lower quartile/Median/Upper quartile of packet lengths
- Inter quartile distance
- Packet length standard deviation/Robust standard deviation
- Packet length skewness and excess
- Min/Max/Mean inter arrival times
- Inter arrival times standard deviation/Robust standard deviation
- N-first packet statistics
- Packet size inter arrival time two-dimensional statistics
Applications for Researchers

- n-first packet byte length signal:
  - Quick application profiling
  - State machine reverse engineering
Packet size inter arrival time two-dimensional statistics

TCP P2P Skype VOIP and File transfer via proxy
User profiling

- Identify abnormal User: Warez (0.8% of users, 42% Traffic)
Questions?

Want to contribute?

http://tranalyzer.sourceforge.net

stefan.burschka@swisscom.com
torben.ruehl@swisscom.com
florian.buehlmann@swisscom.com