RAID 2009
12th International Symposium
On Recent Advances In Intrusion Detection
Saint-Malo, Brittany, France | September 23-25, 2009

CERN Investigation of Network Behaviour and Anomaly Detection

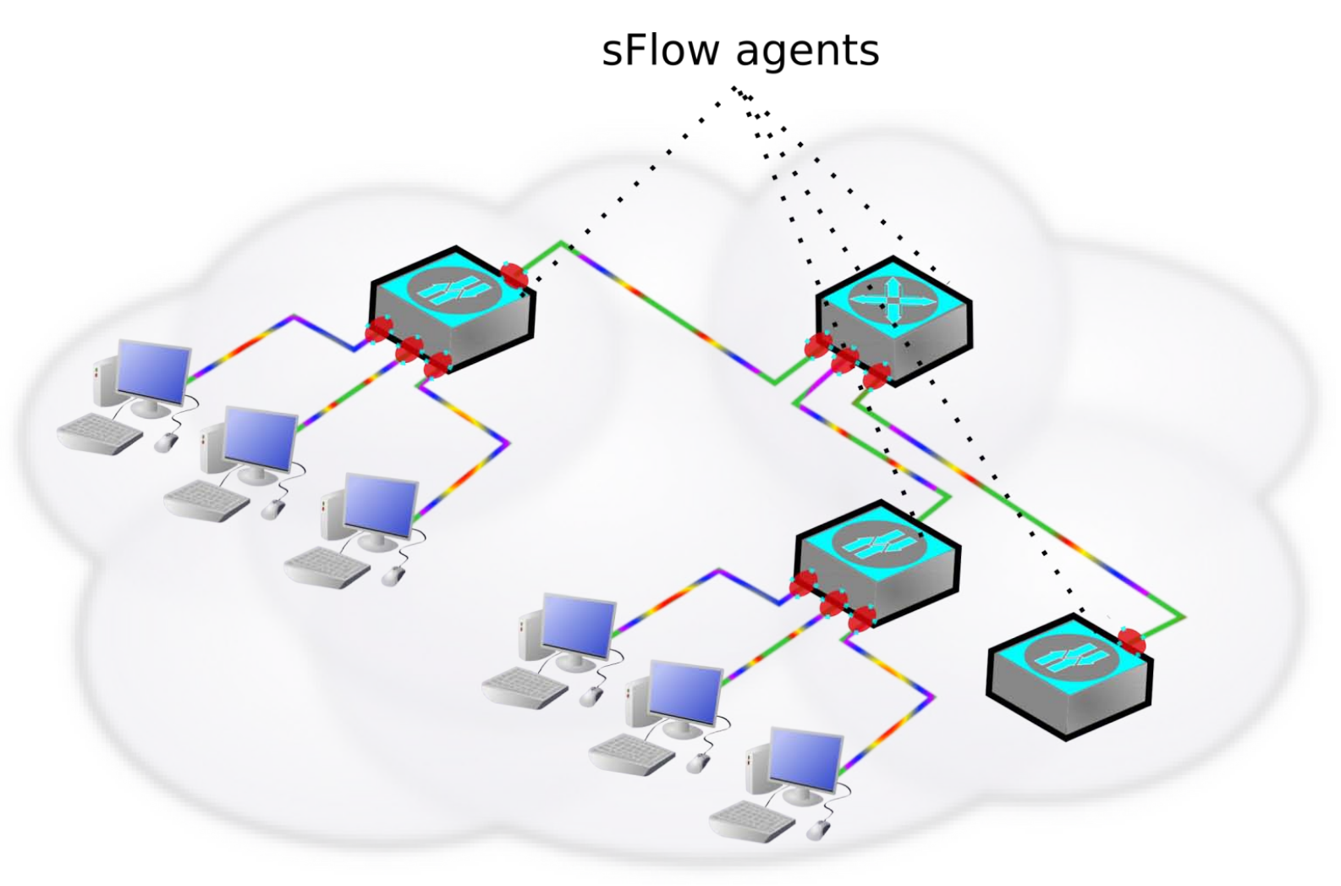
Milosz Marian Hulboj and Ryszard Erazm Jurga
{mhulboj,rjurga}@cern.ch
CERN – HP Procurve openlab project

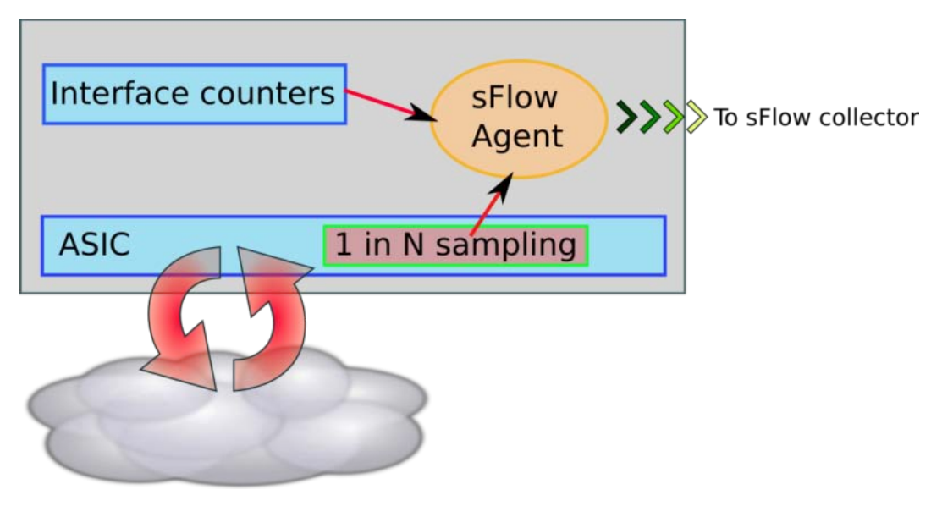
The CINBAD (CERN Investigation of Network Behaviour and Anomaly Detection) project was launched in 2007 in collaboration with ProCurve Networking by HP. The project mission is to understand the behaviour of large computer networks in the context of high performance computing and large campus installations such as at CERN, whose network today counts roughly 70,000 Gigabit user ports. The goals of the project are to be able to detect traffic anomalies in such systems, perform trend analysis, automatically take counter measures and provide post-mortem analysis facilities.

http://cern.ch/openlab-cinbad

With the modern high-speed networks it is impossible to monitor all the packets traversing the links. sFlow is the industry standard for monitoring high-speed switched networks overcomes this issue by providing randomly sampled packets (first 128 bytes) from the network traffic.

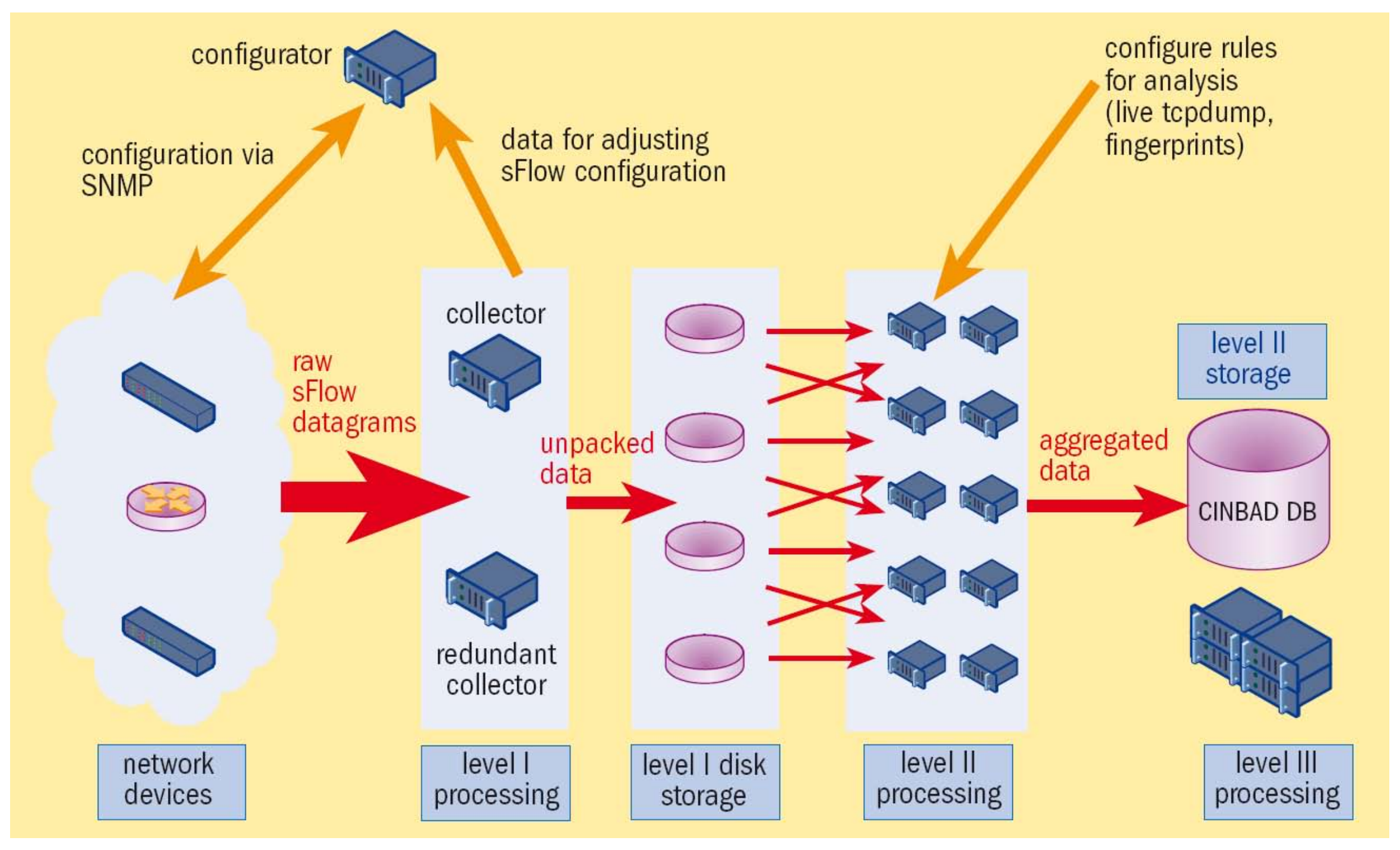
sFlow is scalable and can monitor links of all speeds without impacting the performance of the network devices. It is a low cost solution that is supported by a wide range of vendors.





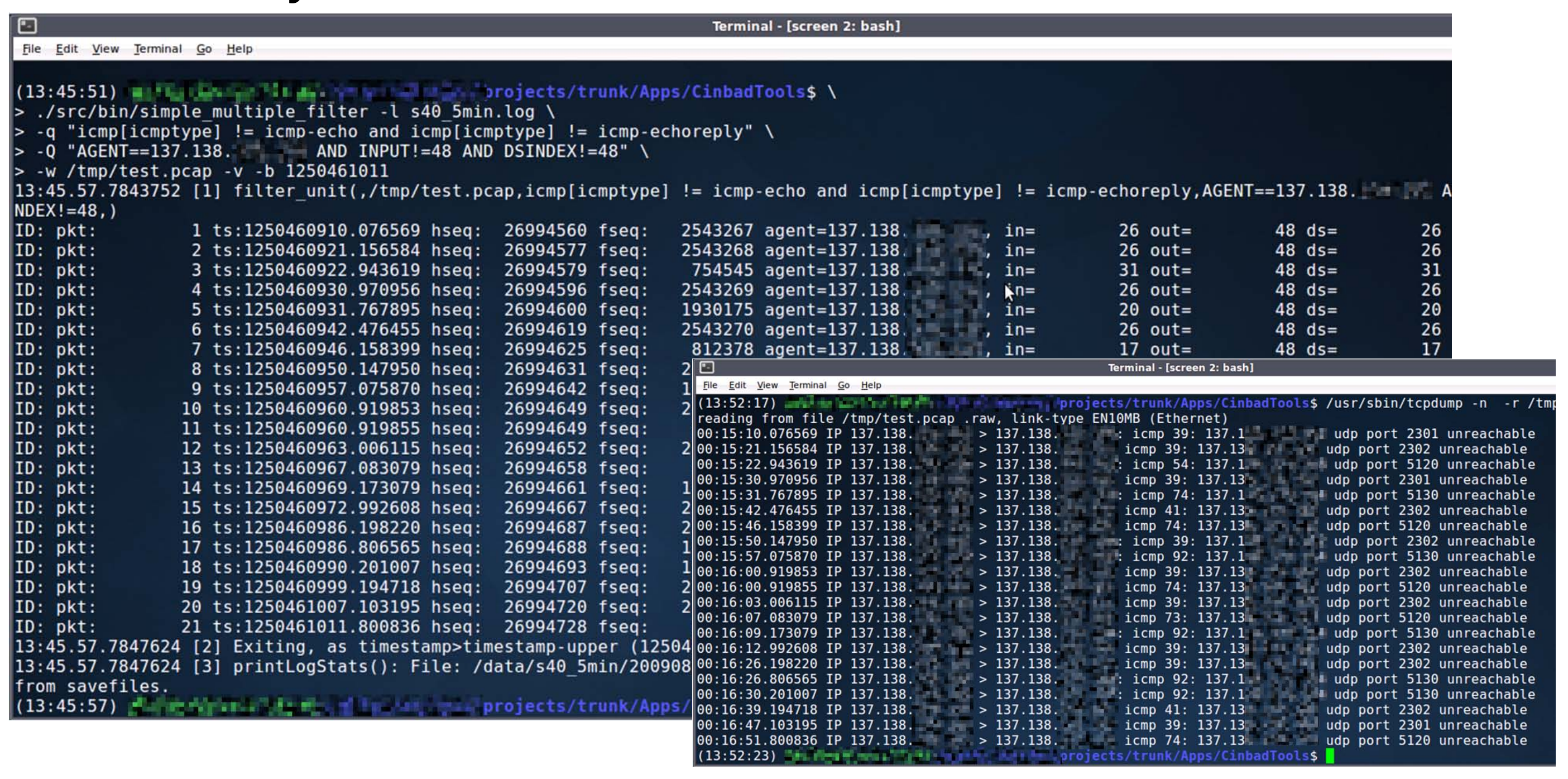
At CERN we collect the sFlow data from more than 1000 switches and routers. Per month we gather more than 3TB of data.

Initial bytes of data obtained by sFlow provide a centralised network-wide view of the network activity. To give an illustration, one could compare it to network-wide tcpdump that collects randomly sampled packet headers from the whole network providing additional metadata at the same time.



Credit to IOP Publishing

Apart from providing a useful debugging information for the network engineers, collected packets are crucial for the analysis conducted by the CINBAD team.



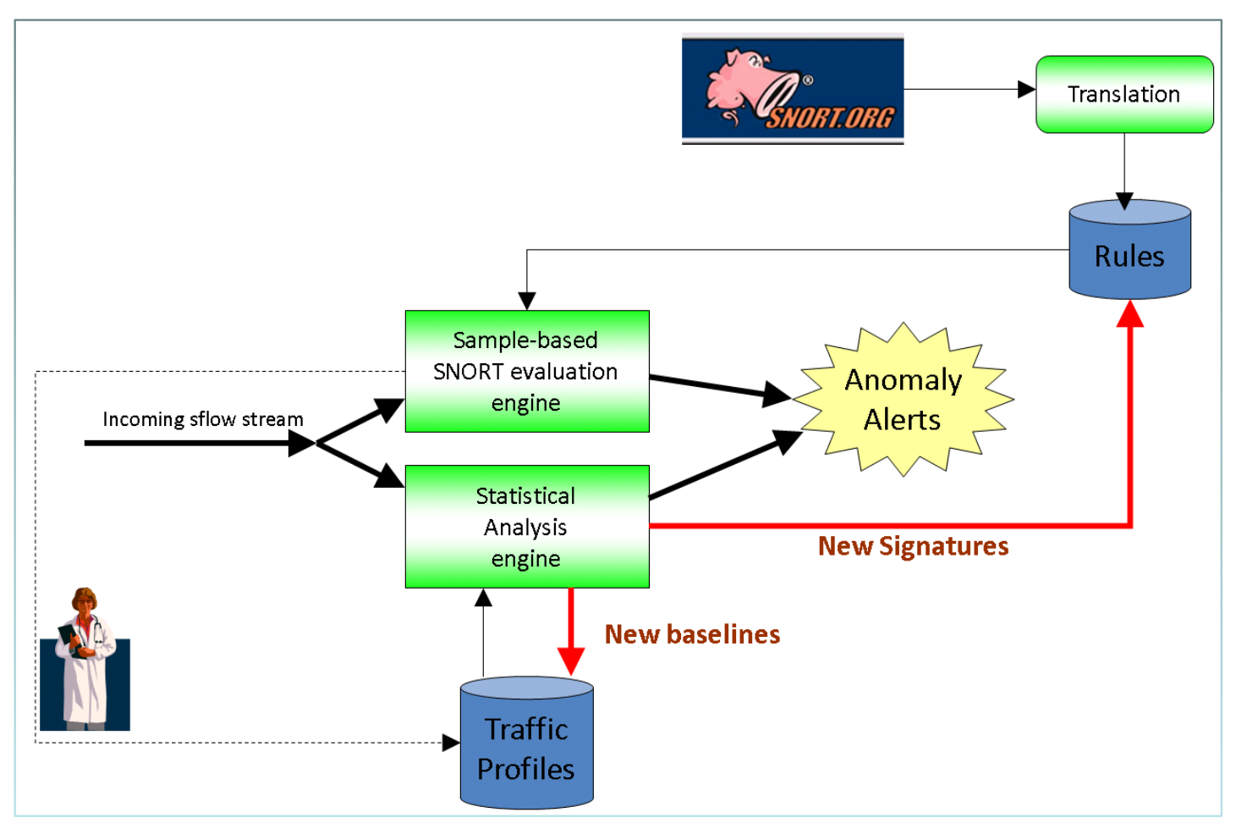
```
Terminal - [screen 2: bash]
File  Edit  View  Terminal  Go  Help
(13:45:51) [redacted]projects/trunk/Apps/CinbadTools$ \
> ./src/bin/simple_multiple_filter -l s40_5min.log \
> -q "icmp[icmptype] != icmp-echo and icmp[icmptype] != icmp-echoreply" \
> -Q "AGENT==137.138.[redacted] AND INPUT!=48 AND DSINDEX!=48" \
> -w /tmp/test.pcap -v -b 1250461011
13:45.57.7843752 [1] filter_unit(,/tmp/test.pcap,icmp[icmptype] != icmp-echo and icmp[icmptype] != icmp-echoreply,AGENT==137.138.[redacted] A
NDEX!=48,)
ID: pkt:        1 ts:1250460910.076569 hseq:  26994560 fseq:   2543267 agent=137.138.[redacted], in=       26 out=       48 ds=       26
ID: pkt:        2 ts:1250460921.156584 hseq:  26994577 fseq:   2543268 agent=137.138.[redacted], in=       26 out=       48 ds=       26
ID: pkt:        3 ts:1250460922.943619 hseq:  26994579 fseq:    754545 agent=137.138.[redacted], in=       31 out=       48 ds=       31
ID: pkt:        4 ts:1250460930.970956 hseq:  26994596 fseq:   2543269 agent=137.138.[redacted], in=       26 out=       48 ds=       26
ID: pkt:        5 ts:1250460931.767895 hseq:  26994600 fseq:   1930175 agent=137.138.[redacted], in=       20 out=       48 ds=       20
ID: pkt:        6 ts:1250460942.476455 hseq:  26994619 fseq:   2543270 agent=137.138.[redacted], in=       26 out=       48 ds=       26
ID: pkt:        7 ts:1250460946.158399 hseq:  26994625 fseq:    812378 agent=137.138.[redacted], in=       17 out=       48 ds=       17
ID: pkt:        8 ts:1250460950.147950 hseq:  26994631 fseq:   2
ID: pkt:        9 ts:1250460957.075870 hseq:  26994642 fseq:   1
ID: pkt:       10 ts:1250460960.919853 hseq:  26994649 fseq:   2
ID: pkt:       11 ts:1250460960.919855 hseq:  26994649 fseq:
ID: pkt:       12 ts:1250460963.006115 hseq:  26994652 fseq:   2
ID: pkt:       13 ts:1250460967.083079 hseq:  26994658 fseq:
ID: pkt:       14 ts:1250460969.173079 hseq:  26994661 fseq:   1
ID: pkt:       15 ts:1250460972.992608 hseq:  26994667 fseq:   2
ID: pkt:       16 ts:1250460986.198220 hseq:  26994687 fseq:   2
ID: pkt:       17 ts:1250460986.806565 hseq:  26994688 fseq:   1
ID: pkt:       18 ts:1250460990.201007 hseq:  26994693 fseq:   1
ID: pkt:       19 ts:1250460999.194718 hseq:  26994707 fseq:   2
ID: pkt:       20 ts:1250461007.103195 hseq:  26994720 fseq:   2
ID: pkt:       21 ts:1250461011.800836 hseq:  26994728 fseq:
13:45.57.7847624 [2] Exiting, as timestamp>timestamp-upper (12504
13:45.57.7847624 [3] printLogStats(): File: /data/s40_5min/200908
from savefiles.
(13:45:57) [redacted]projects/trunk/Apps/
```

```
Terminal - [screen 2: bash]
File  Edit  View  Terminal  Go  Help
(13:52:17) [redacted]projects/trunk/Apps/CinbadTools$ /usr/sbin/tcpdump -n  -r /tmp
reading from file /tmp/test.pcap .raw, link-type EN10MB (Ethernet)
00:15:10.076569 IP 137.138.[redacted] > 137.138.[redacted]: icmp 39: 137.1[redacted] udp port 2301 unreachable
00:15:21.156584 IP 137.138.[redacted] > 137.138.[redacted] icmp 39: 137.13[redacted] udp port 2302 unreachable
00:15:22.943619 IP 137.138.[redacted] > 137.138.[redacted]: icmp 54: 137.1[redacted] udp port 5120 unreachable
00:15:30.970956 IP 137.138.[redacted] > 137.138.[redacted] icmp 39: 137.13[redacted] udp port 2301 unreachable
00:15:31.767895 IP 137.138.[redacted] > 137.138.[redacted]: icmp 74: 137.1[redacted] udp port 5130 unreachable
00:15:42.476455 IP 137.138.[redacted] > 137.138.[redacted] icmp 41: 137.13[redacted] udp port 2302 unreachable
00:15:46.158399 IP 137.138.[redacted] > 137.138.[redacted] icmp 74: 137.13[redacted] udp port 5120 unreachable
00:15:50.147950 IP 137.138.[redacted] > 137.138.[redacted]: icmp 39: 137.1[redacted] udp port 2302 unreachable
00:15:57.075870 IP 137.138.[redacted] > 137.138.[redacted]: icmp 92: 137.1[redacted] udp port 5130 unreachable
00:16:00.919853 IP 137.138.[redacted] > 137.138.[redacted] icmp 39: 137.13[redacted] udp port 2302 unreachable
00:16:00.919855 IP 137.138.[redacted] > 137.138.[redacted] icmp 74: 137.13[redacted] udp port 5120 unreachable
00:16:03.006115 IP 137.138.[redacted] > 137.138.[redacted] icmp 39: 137.13[redacted] udp port 2302 unreachable
00:16:07.083079 IP 137.138.[redacted] > 137.138.[redacted] icmp 73: 137.13[redacted] udp port 5120 unreachable
00:16:09.173079 IP 137.138.[redacted] > 137.138.[redacted]: icmp 92: 137.1[redacted] udp port 5130 unreachable
00:16:12.992608 IP 137.138.[redacted] > 137.138.[redacted] icmp 39: 137.13[redacted] udp port 2302 unreachable
00:16:26.198220 IP 137.138.[redacted] > 137.138.[redacted] icmp 39: 137.13[redacted] udp port 2302 unreachable
00:16:26.806565 IP 137.138.[redacted] > 137.138.[redacted]: icmp 92: 137.1[redacted] udp port 5130 unreachable
00:16:30.201007 IP 137.138.[redacted] > 137.138.[redacted]: icmp 92: 137.1[redacted] udp port 5130 unreachable
00:16:39.194718 IP 137.138.[redacted] > 137.138.[redacted] icmp 41: 137.13[redacted] udp port 2302 unreachable
00:16:47.103195 IP 137.138.[redacted] > 137.138.[redacted] icmp 39: 137.13[redacted] udp port 2301 unreachable
00:16:51.800836 IP 137.138.[redacted] > 137.138.[redacted] icmp 74: 137.13[redacted] udp port 5120 unreachable
(13:52:23) [redacted]projects/trunk/Apps/CinbadTools$
```

We have been investigating various data analysis approaches that could be categorised mainly into the two domains: statistical and signature based analysis. The former depends on detecting deviations from normal network behaviour while the latter uses existing problem signatures and matches them against the current state of the network. The signature based approach has numerous practical applications with SNORT being a prominent example.

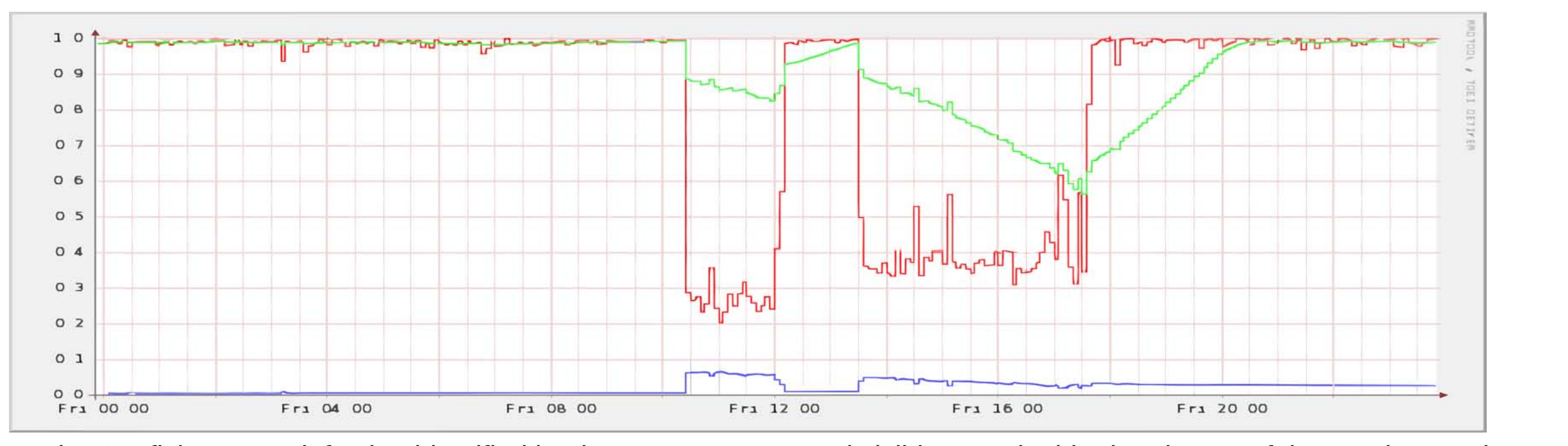
The CINBAD team has successfully ported SNORT and adapted various rules in order to work with sampled data. It seems to perform well, and provides a low false positive rate. However, the system is blind and can yield false negatives in case of unknown anomalies.

Fully automatic detection of novel worms and unsupervised signature generation is the unattainable *Holy Grail*. By combining the statistical analysis of the data from the protocol headers and by analysing the payload we attempt to minimise the necessary supervision.



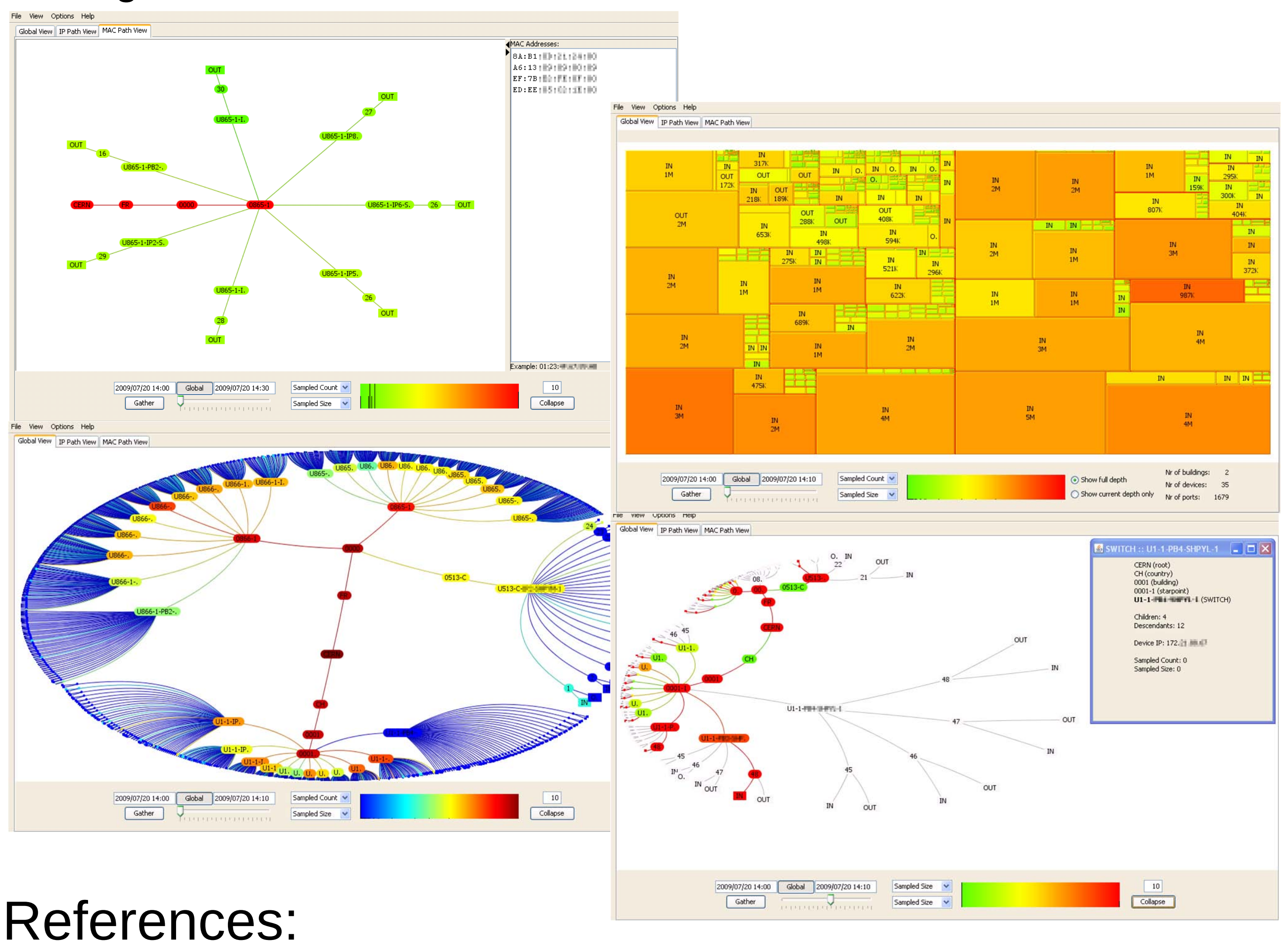
The behaviour of each host is described by its profile. A set of all host profiles is considered as a network profile. This is the most natural way to represent the network, since each host partially influences the network behaviour and we can directly point out dominant hosts in case of an anomaly.

Each profile consists of features originating from packet headers (e.g. number of distinct TCP ports contacted in Δt, ratio of egress/ingress traffic of a given type) or packet payload (e.g. entropy of the payload, *n*-gram distribution, hashes of the partitioned payload). For each profile we set up a baseline using historical data and then compare those baselines against latest ones. Different distance metrics are employed, most notably divergence, standarized Euclidean distance and Mahalanobis distance.



The Conficker worm infection identified by the CINBAD team and visible as a double drop in one of the metrics used

CINBAD team had also developed (with the help of student Vlad Petre) several visualisation tools for CERN network engineers to help them with daily work. Network-wide visibility proves to be extremely important for maintenance and problem solving.



References:

[1] Jurga, R., Hulboj, M., *Technical Report Packet Sampling for Network Monitoring*, CERN, 2008.
[2] Kim H., Karp B., *Autograph: Toward Automated, Distributed Worm Signature Detection*, USENIX, 2004.
[3] Wang K., Stolfo J., *Anomalous Payload-Based Worm Detection and Signature Generation*, RAID 2005.