# Vermont - A Versatile Monitoring Toolkit for IPFIX and PSAMP

Ronny T. Lampert*, Christoph Sommer*, Gerhard Münz† and Falko Dressler*
*Autonomic Networking Group, Dept. of Computer Science 7, University of Erlangen, Germany
†Computer Networks and Internet, Wilhelm Schickard Institute for Computer Science, University of Tübingen, Germany

***Abstract*— In this paper, we present Vermont, a flexible network monitoring toolkit for packet filtering and packet sampling, flow accounting, and flow aggregation. This toolkit supports the export and collection of IPFIX/PSAMP compliant monitoring data. Packet capturing is based on the well-known pcap library, which enables deployment on various hardware platforms and operating systems. Apart from an overview to Vermont's architecture, we present evaluation results with regard to performance, interoperability, and robustness. Furthermore, we compare Vermont to other open-source implementations of monitoring probes with respect to supported features and functionality.**

## I. Introduction

Network monitoring is a major building block for many domains in communication networks. Besides typical accounting mechanisms and the emerging area of charging in next generation networks, especially network security solutions rely on efficient means of monitoring.

In order to cope with the increasing amounts of monitoring data brought about by ever-growing network capacities, monitoring is commonly based on flow accounting and statistical packet sampling. In accordance with the terminology used in the literature, we use the term *flow* to designate a stream of packets sharing a set of common properties (called flow keys) like end point addresses or used protocol. Usually, a flow is defined by the IP-quintuple <proto, src_ip, dst_ip, src_port, dst_port>, but arbitrarily chosen flow keys are also allowed – even keys that depend on user-defined field types.

The techniques for flow accounting, as well as the transfer of observed monitoring data, are set out in the Cisco NetFlow.v9 protocol [1] and its successor, the IPFIX (IP Flow Information Export) protocol [2]. In contrast to IPFIX, that carries flow information, the PSAMP (Packet Sampling) protocol [3] was developed to satisfy the growing need for more detailed network monitoring. PSAMP gathers samples of individual packets and allows exporting of actual payload. Both the IPFIX protocol and the PSAMP protocol are being standardized by the IETF (Internet Engineering Task Force). Figure 1 illustrates the functional architecture of an IPFIX/PSAMP device consisting of Metering Processes (MP), Sampling Processes (SP), Aggregation Processes (AP), Collecting Processes (CP), and Exporting Processes (EP), that can be linked in various ways.

In this paper, we present the monitoring toolkit Vermont, which has been developed in the *History* project [4] and the European project *Diadem Firewall*. Vermont was designed for monitoring high-speed networks with link speeds of up to one gigabit per second using standard PC hardware. Furthermore, Vermont serves as a reference implementation of the aforementioned monitoring techniques, including the protocol extensions for flow aggregation [5].

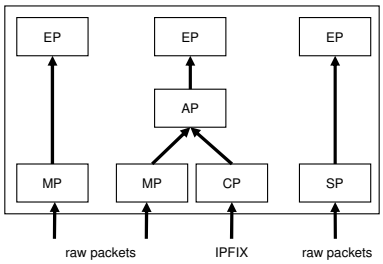


Fig. 1. IPFIX Device Architecture: Metering, collecting and sampling processes feed data to aggregation and exporting processes

The design principles of Vermont are:

- IPFIX/PSAMP compliant monitoring and data export
- Rule-based flow metering and aggregation
- Hardware-independent packet capturing
- Multiprocessor support
- High monitoring performance

The remainder of this paper is organized as follows. Section II briefly introduces other open-source implementations of monitoring probes and compares them to Vermont. Section III outlines Vermont's architecture. Application scenarios are provided in section IV and results obtained in performance, compatibility, and robustness evaluations are discussed in section V. Finally, section VI concludes the paper.

## II. Related Work

In this section, we provide a brief overview to open-source implementations of monitoring probes. There are several implementations supporting the NetFlow.v9 format, e.g. nprobe [6] and NetMate [7]. Currently, the authors of these tools are working on IPFIX compliant versions. Table I compares the supported features of these implementations and Vermont. Futhermore, there exist implementations from Cisco Systems and IBM [8], which are not available under an open-source compatible license and as such not listed here.

Vermont, being a reference implementation for both the IPFIX and the PSAMP standard, already supports IPFIX, PSAMP, and IPFIX aggregation schemes.

TABLE I
FEATURE COMPARISON

| | Vermont | nProbe | NetMate |
|---|---|---|---|
| IPFIX Support | yes | planned | |
| PSAMP Support | yes | | |
| IPFIX Aggregation | yes | planned | planned |
| Collector Functionality | yes | | |
| Save Data To Disk | | yes | |
| Save Data To SQL DB | yes | | |
| Remote Reconfigurability | yes | | yes |
| Plugin Architecture | | yes | yes |

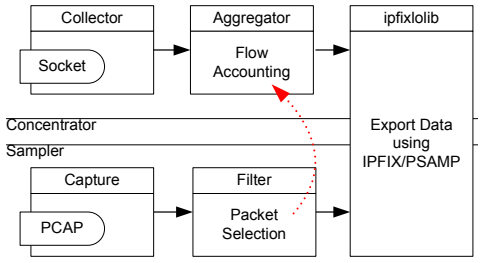


Fig. 2. Vermont: Architectural overview

## III. ARCHITECTURE

### A. Overview

Figure 2 depicts the architecture of Vermont. The functionality is divided into two main modules: A sampler module and a concentrator module. The sampler module implements all packet-based functions like packet capturing, filtering, sampling and PSAMP export. The functions for flow accounting and aggregation are provided by the concentrator module. Both main modules can run independently or in combination; in the second case, the sampled packets are passed from the sampler module to the flow metering and aggregation function of the concentrator module. A common exporter library called ipfixlolib realizes the export of monitoring data using the IPFIX/PSAMP protocol.

Both main modules are broken up into submodules, which are described in the following subsections. The main modules, as well as their submodules, have been designed for being instantiated more than once, giving the user a maximum degree of flexibility and caring for a wide range of monitoring needs. This modular approach also guarantees a high level of reusability and eases the incorporation of selected components into other software packages.

Vermont's modular design excels where customization is needed. Implementing the SQL database writer was a matter of simply writing an alternative export module and changing Vermont's configuration routines so it could be instantiated when configured to. A base class takes care of the details like allocating buffered queues and two methods have to be implemented which start and shutdown the submodule's operation. We use an URI-based scheme to indicate the export method used, i.e. exporting to the host `10.20.30.40` with destination port 6677 using the UDP protocol is expressed by `udp://10.20.30.40:6677`. This scheme can be arbitrarily extended.

Internally, submodules are contained within different threads. In order to avoid unnecessary copying of data, only references to the actual data are passed from one subsystem to the other using buffered queues. This design was chosen to optimally exploit the asynchronous processing capabilities of multi-processor systems.

Vermont's configuration is maintained in one consistent configuration file; additionally, a back-end for dynamic reconfiguration using Netconf protocol has been implemented [9], allowing for fast remote distribution of new configuration data.

Depending on the configuration, Vermont captures raw packets, performs packet sampling and optional flow accounting and exports the resulting monitoring data using the IPFIX/PSAMP protocol. Alternatively, Vermont can operate as an IPFIX concentrator that receives and aggregates data exported by other monitoring probes in order to reduce the overall data volume.

### B. Sampler Module

The sampler module captures raw packets from network interfaces, selects individual packets based on filters and sampling algorithms as specified in [10] and exports the resulting data. Filters and sampling algorithms are implemented as packet processors that can be executed in arbitrary order. Non-matching packets or packets that are sorted out by a sampling algorithm are immediately dropped and only packets that have passed all packet processors are exported. At any point in the packet processor chain, packets can also be injected into the concentrator module to be processed by the flow metering and aggregation function.

### C. Concentrator Module

Vermont's concentrator module consists of a NetFlow.v9/IPFIX collector, an aggregator and an IPFIX exporter, all interconnected using callbacks. The aggregator submodule implements the rule-based flow metering and aggregation approach specified in [5]. It can be configured to process flows received via the IPFIX collector and/or packets injected by the sampler module; this is shown in figure 2 by the dotted arrow. That way, the aggregator can be deployed for flow accounting of local traffic as well as for concentration of flow records received from other monitoring probes.

## IV. MONITORING SCENARIOS

Owing to the flexibility offered by its modular design, Vermont can be used in a wide range of scenarios, some of which are presented in this section. Data export can be done to one or more collecting stations.

### A. PSAMP Probe

With only the sampler module activated, Vermont acts as a PSAMP probe that captures packets from network interfaces, selects individual packets based on filters and sampling algorithms and exports the resulting packet-based monitoring data.

### B. IPFIX Probe

In this mode, Vermont performs rule-based flow accounting [5] on locally observed packets. Both the sampler module and the concentrator module are activated. The sampler module is configured to pass packets to the concentrator module, which performs flow accounting and exports the resulting flow records. For example, flow accounting can be used to generate byte or packet counters for the observed network. The sampler module's exporter, as well as the concentrator module's collector, are disabled.

### C. Concentrator

Only the concentrator module is activated. Flow records from other monitoring probes are received at the collector. Rule-based flow aggregation [5] is performed in order to reduce the amount of monitoring data. The resulting records are exported to higher-level concentrators or traffic analyzers.

### D. Specific accounting for SYN flood detection

In the context of *Diadem Firewall*, Vermont is used to count TCP SYN and SYN/ACK packets. The resulting counter values are used to detect SYN flood attacks on protected servers using the SYN-dog algorithm [11].

## V. Evaluation

Vermont and its modules have undergone extensive testing with respect to performance, interoperability and robustness.

### A. Performance

Figure 3 shows the results obtained from benchmarking the sampler module on a dual-processor system[1]. A specialized version [12] of the pcap library was used, which avoids multiple copying of captured data by mapping a user-space buffer into the kernel. The traffic was generated using a packet generator set to fixed packet generation rates running on a separate machine. To ensure the desired packet rates the UDP protocol was used and every two seconds captured as well as dropped packets were accounted using the counters supplied by the pcap library. Multiple tests have proven the data sample shown in figure 3 to be Vermont's typical behavior over time.

Until a packet rate of 250,000 packets per second (for 128 byte packets), the capture rate is 100%. 420,000 packets per second denotes the maximum packet generation rate and short-lived loss up to 45 percent can be observed; however, the loss is not of permanent nature and numerous measure points suggest the packet loss ratio being substantially smaller, averaging at 9.0 percent. At 300,000 packets per second the loss averages at 3.8 percent. The diagram shows a certain fluctuation regarding the packet capture ratio. This stems from Vermont's asynchronous design utilizing buffered queues between subsystems. If the input queue suffers from congestion incoming packets will no longer be forwarded, but immediately dropped instead. Capturing entire packets of 1500 bytes in size showed no loss at a maximum observed rate of 60,000 packets per second.

[1]Dual 3.06GHz Intel Xeon, 2GB RAM, 1GBit NIC, Linux Kernel 2.6.13, libpcap-mmap 0.9.20060417

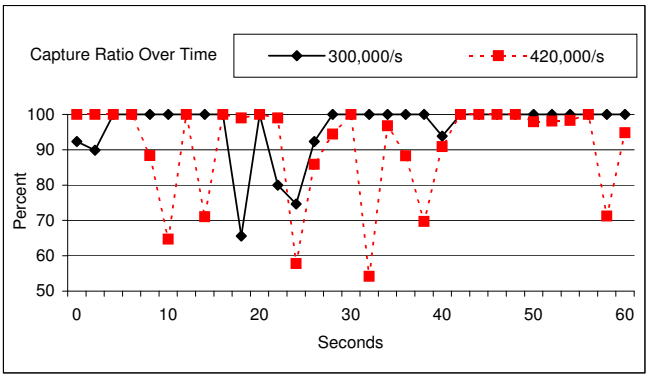


Fig. 3. Capture ratio for packet rates of 300,000 and 420,000 packets per second

```
keep protocolIdentifier
keep sourcetransportport
keep sourceipv4address
keep destinationtransportport
keep destinationipv4address
aggregate inpacketdeltacount
aggregate inoctetdeltacount
aggregate flowcreationtime
aggregate flowendtime
```

Fig. 4. Aggregation rule used by Concentrator module

The concentrator module's performance was tested in a real-world scenario. Vermont was configured to capture all packets at the access router of our university's network, perform flow accounting using the rules shown in figure 4 and export the resulting records. FIXME These rules create aggregated fields for packet and byte counters. During the tests, a maximum metering performance of 45,000 packets per second was achieved[2]. Repeating the test with different rule sets showed the maximum number of metered flows to be inversely proportional to the number of rules.

### B. Interoperability and Robustness

We participated in the IST MOME IPFIX Interoperability testing event [13] in July 2005 and tested Vermont's interoperability with several other IPFIX/PSAMP implementations. Vermont's collector and exporter successfully passed all applicable tests, including the handling of corrupt IPFIX data packets and showed excellent robustness and compatibility.

## VI. Conclusion

In this paper, we presented Vermont, a monitoring toolkit for IPFIX/PSAMP compliant flow monitoring and packet sampling. Vermont has fulfilled its design goal of providing a versatile high-speed monitoring toolkit consisting of a modular, reusable, and freely configurable architecture.The performance on state-of-the-art PC systems is satisfactory. Nevertheless, improvements are still needed if Vermont if used as a pure flow monitor. On the other hand, excellent compatibility and high

[2]Dual 2.0GHz Intel Xeon, 1GB RAM, 1GBit NIC, Linux Kernel 2.6.13, libpcap-mmap 1.0.20050129

robustness have been proven in interoperability tests. Vermont is available as an open-source package [14].

## ACKNOWLEDGMENT

We gratefully acknowledge support from the European project *Diadem Firewall* (FP6 IST-2002-002154). Furthermore, we would like to thank Jan Petranek, Michael Drüing, and Lothar Braun for contributing to the development of Vermont.

## REFERENCES

[1] B. Claise, "Cisco Systems NetFlow Services Export Version 9," RFC 3954, Oct. 2004.

[2] ——, "IPFIX Protocol Specification," Internet-Draft, work in progress, draft-ietf-ipfix-protocol-22.txt, June 2006.

[3] ——, "Packet Sampling (PSAMP) Protocol Specifications," Internet-Draft, work in progress, draft-ietf-psamp-protocol-06.txt, June 2006.

[4] F. Dressler and G. Carle, "HISTORY - High Speed Network Monitoring and Analysis," in *24th IEEE Conference on Computer Communications (IEEE INFOCOM 2005), Poster Session*, Miami, FL, USA, Mar. 2005.

[5] F. Dressler, C. Sommer, and G. Münz, "IPFIX Aggregation," Internet-Draft, work in progress, draft-dressler-ipfix-aggregation-03.txt, June 2006.

[6] L. Deri, "nProbe: an Open Source NetFlow Probe for Gigabit Networks," in *TERENA Networking Conference (TNC 2003)*, Zagreb, Croatia, May 2003.

[7] C. Schmoll and S. Zander, "NetMate - User and Developer Manual," Feb. 2004. [Online]. Available: http://www.ip-measurement.org/tools/netmate/

[8] IBM Zurich Research Laboratory, "Aurora - Network Traffic Analysis and Visualization," 2004. [Online]. Available: http://www.zurich.ibm.com/aurora/

[9] G. Münz, A. Antony, F. Dressler, and G. Carle, "Using Netconf for Configuring Monitoring Probes," in *IEEE/IFIP Network Operations & Management Symposium (IEEE/IFIP NOMS 2006), Poster Session*, Vancouver, Canada, Apr. 2006.

[10] T. Zseby, M. Molina, N. Duffield, S. Niccolini, and F. Raspall, "Sampling and Filtering Techniques for IP Packet Selection," Internet-Draft, work in progress, draft-ietf-psamp-sample-tech-07.txt, July 2005.

[11] H. Wang, D. Zhang, and K. Shin, "SYN-dog: Sniffing SYN Flooding Sources," in *22nd International Conference on Distributed Computing Systems (ICDCS'02)*, July 2002.

[12] P. Wood. A libpcap version which supports MMAP mode on Linux kernels. [Online]. Available: http://http://public.lanl.gov/cpw

[13] C. Schmoll, J. Quittek, S. Tartarelli, S. Niccolini, T. Dietz, A. Bulanza, and E. Boschi, "MOME Interoperability Testing Event," Deliverable D13 of IST MOME, Aug. 2005.

[14] Vermont - Versatile Monitoring Toolkit. [Online]. Available: http://vermont.berlios.de