

INFORMATION SOCIETY TECHNOLOGIES (IST) PROGRAMME



A Scaleable Monitoring Platform for the Internet
Contract No. IST-2001-32404

D4.2 “Report on Measurement and Monitoring BoF”

Contractual Date of Delivery	31 October 2003
Report Date	30 July 2003
Deliverable Security Class	Public
Editor	Kevin Meynell
Contributors	Herbert Bos, Luca Deri, Kevin Meynell & Manolis Petsagourakis

The SCAMPI Consortium consists of:

TERENA	Coordinator	The Netherlands
IMEC	Principal Contractor	Belgium
FORTH	Principal Contractor	Greece
LIACS	Principal Contractor	The Netherlands
NETikos	Principal Contractor	Italy
Uninett	Principal Contractor	Norway
CESNET	Principal Contractor	Czech Republic
FORTHnet	Principal Contractor	Greece
4Plus	Principal Contractor	Greece
Siemens	Principal Contractor	Germany



Introduction

The SCAMPI Monitoring and Measurements BoF was held on 21 May 2003 in conjunction with the TERENA Networking Conference (TNC-CNC 2003) in Zagreb, Croatia. The objective was to publicise the SCAMPI project, as well as discuss specific aspects of the work that are of wider interest to the monitoring community. It also provided an opportunity for feedback from the European research networking community.

The BoF was run as a parallel session of the conference from 16.00 to 17.30. It attracted a total of 36 participants.

Presentations were as follows:

- The SCAMPI Project – *Kevin Meynell, TERENA*
- Denial-of-Service and Anomaly Detection – *Vasilios Siris, FORTH*
- Monitoring Networks at Gigabit Speeds – *Luca Deri, NETikos*
- The OKE Corral – *Herbert Bos, Leiden University*

The full proceedings of the BoF can be found on the SCAMPI website at:

<http://www.ist-scampi.org/events/bof-2003/>

The SCAMPI Project

Kevin Meynell, TERENA

SCAMPI is a thirty-month IST project to develop scalable monitoring platform for the Internet, and promote the use of monitoring tools for improving services and technology. It started in April 2003 and comprises ten partners from the commercial, research and academic sectors.

The project is developing a high-performance intelligent monitoring adapter, at speeds of up to 10 Gbps, which can utilise standard PC architectures and offer simple installation. The aim is also to establish an open and extensible architecture for network monitoring through the introduction of a common API (known as the MAPI) against which applications may be developed.

In addition, a number of monitoring and measurement tools are being developed. This includes applications for denial-of-service detection, SLS auditing, and billing/accounting, amongst others. Strategies and methodologies for monitoring systems operating at speeds higher than 10 Gbps are also being considered.

More information about the project can be found on the SCAMPI website (<http://ist-scampi.org/>).



Denial-of-Service and Anomaly Detection

Vasilios Siris, FORTH

The Internet is faced with an increasing number of denial-of-service (DoS) attacks which prevent users from receiving service, or greatly degrade the performance of their service. These attacks consume bandwidth, memory, and processor time by flooding networks with unnecessary traffic, eventually making them unusable. Recent surveys have shown that 40% of all network incidents are DoS attacks, with 90% of these being TCP-based. They have affected many large organisations, and the cost of dealing with the effects is estimated at several billions of dollars.

One possible technique for preventing or reducing DoS attacks, is to develop algorithms that can provide early and reliable detection of them. If normal traffic patterns can be analysed over time, it should be easier to spot deviations when they occur. Fixed threshold tests are not suitable for this as traffic patterns naturally vary, so an adaptive threshold is required.

Different algorithms are being investigated that take into account aggregate traffic volume, traffic volume per flow, number of requests, inter-arrival time of requests, duration of requests and packet size, in order to develop an efficient alarm mechanism when suspicious levels of traffic are observed.

The results achieved to-date, show that simple and relatively straightforward procedures are effective for detecting intense attacks. However, more detailed procedures are required to achieve effective and robust detection behaviour over a wide range of attacks with varying characteristics, including low intensity attacks. The application of advanced statistical techniques has shown encouraging results, and ongoing work includes investigating the impact of the various parameters involved. Such knowledge will enable the effective tuning of the parameters to the various environments where the procedures will be applied.


Future work will focus on further improving the algorithms and how to combine various alarm states. In addition, there will be investigations into how to apply such techniques to QoS measurement.

Monitoring Networks at Gigabit Speeds

Luca Deri, NETikos

It is already possible to monitor low-speed networks (e.g. 100 Mbps) with common tools such as libpcap, but monitoring higher speed networks presents greater challenge. The PCI bus that is commonly used in PCs is limited to 533 Mbps, which either means expensive custom hardware needs to be utilised, or some pre-processing needs to be undertaken by the network devices.

One approach is to utilise intelligent routers (e.g. Juniper M-series) in conjunction with standard x86-based PCs and Gigabit NICs. The aim is to passively monitor networks at gigabit speeds with little or no packet loss, generate traffic information in a standard format

	Deliverable 4.2	IST-2001-32404
---	-----------------	----------------

(e.g. NetFlow/nFlow), have the ability to monitor both IPv4 and IPv6, and provide accounting and performance information.

The problem is that NetFlow implementations consume significant resources on the router/switch, whilst most operating systems and NICs have not been designed to handle thousands of packets per second. However, it is possible to utilise the traffic filter and mirroring capabilities of a Juniper router, and forward the traffic of interest to a PC for processing.

Various methods of processing the traffic received by the PC have been investigated, in order to find the most efficient way of processing it. The use of libpcap was too costly in terms of system resources and resulted in severe packet loss, whilst kernel packet capture only improved matters by about 10%. The best technique would appear to be kernel packet classification, which passes flows rather than packets to user applications and therefore makes more efficient use of system resources. The main limitation would appear to be the speed of the CPU, but it is sufficient to provide in-kernel generation of NetFlow statistics and pass them to user-space accounting applications.

The OKE Corral


Herbert Bos, Leiden University

The Open Kernel Environment (OKE) allows third-party programming at low-levels, with safety enforced by software-based isolation. This is targeted at environments with little support for isolating applications, such as kernels or network processors. It allows devices to be safely manipulated by regulating access on a per-user basis to various resources (e.g. processor, heap and stack).

OKE combines trust management with a compiler and code loader. The code loader accepts code and authenticates the submitted credentials, before sending it to the compiler with the appropriate level of access. Policies are established through the Environment Set-up Code (ESC) that is automatically prepended to user code. This defines the level of run-time support, explicitly declares the API the code can use, and removes the ability to perform unsafe operations.

The OKE programming language avoids special-purpose languages by using Cyclone, a strongly-typed, pointer-protected dialect of C that provides region-based memory protection. This provides for different user classes with different restrictions, whilst interfacing with low-level features.

The advantages of OKE are that it forms a basis for resource control, even when there are multiple mistrusting partners, whilst only incurring overhead when in use. It also offers authorisation procedures that can be applied throughout SCAMPI. On the downside, the resource control comes at a runtime cost, and writing ESC can be complex.

	Deliverable 4.2	IST-2001-32404
---	-----------------	----------------

Finally, a variant of OKE known as Diet OKE has also been developed for controlling network processors. This allows multiple applications to access microengines of the type used on Intel IXP cards, and can potentially be used in intelligent traffic monitoring adapters.