

INFORMATION SOCIETY TECHNOLOGIES (IST) PROGRAMME



*Large Scale Monitoring of BroadBand Internet Infrastructure
Contract No. 004336*

Deliverable 4.2b: Second LOBSTER Workshop Proceedings and Summary

Abstract: This document provides a report of the 2nd LOBSTER Workshop held on 16 May 2006 in Catania, Italy.

Contractual Date of Delivery	30 June 2006
Actual Date of Delivery	23 October 2006
Deliverable Security Class	PU
Editor	Kevin Meynell
Contributors	Herbert Bos, Evangelos Markatos, Michalis Polychronakis, Vladimir Smotlacha & Arne Øslebø

The LOBSTER Consortium consists of:

FORTH-ICS	Coordinator	Greece
VU	Partner	Netherlands
CESNET	Partner	Czech Republic
UNINETT	Partner	Norway
ENDACE	Partner	United Kingdom
Alcatel	Partner	France
FORTHnet	Partner	Greece
TNO	Partner	Netherlands
TERENA	Partner	Netherlands



Table of Contents

Introduction	3
LOBSTER: Large-Scale Monitoring of Broadband Internet Infrastructure	3
Streamline: an Efficient Stream-Processing Framework	4
Ruler: Language-based Anonymisation	5
Hardware Anonymisation	6
Using passive measurements to debug and improve end-to-end network quality	7
Passive monitoring for security-related applications	7



Introduction

The 2nd Workshop on Large-Scale Monitoring of Broadband Infrastructures was held on 16 May 2006 in conjunction with the TERENA Networking Conference 2006 (TNC 2006) in Catania, Italy. The objective was to publicise the LOBSTER project activities and to provide an opportunity for feedback from the European research networking community.

The workshop presented the pilot Internet monitoring infrastructure being developed by the LOBSTER project. It showed how passive monitoring can be utilised on multi-gigabit links, outline anonymisation techniques, and demonstrated the new applications developed for characterising traffic. It also looked more widely at the challenges of high-speed passive monitoring, and the research possibilities in this area.

The workshop ran from 14.00 to 17.30 and was divided into two sessions. Presentations and attendance figures were as follows:

14.00-15.30 Chair: Kevin Meynell Attendance: 21

- LOBSTER: Large-Scale Monitoring of Broadband Internet Infrastructure – *Evangelos Markatos, FORTH*
- Streamline: an Efficient Stream-Processing Framework – *Herbert Bos, Vrije Universiteit*
- Ruler: Language-based Anonymisation – *Herbert Bos, Vrije Universiteit*

16.00-17.30 Chair: Kevin Meynell Attendance: 22

- Hardware Anonymisation – *Vladimir Smotlacha, CESNET*
- Using passive measurements to debug and improve end-to-end network quality – *Arne Øslebø, Uninett*
- Passive monitoring for security-related applications – *Michalis Polychronakis, FORTH*

The full proceedings of the workshop can be found on the LOBSTER website at:

<http://www.ist-lobster.org/events/workshop-2006/>

LOBSTER: Large-Scale Monitoring of Broadband Internet Infrastructure *Evangelos Markatos, FORTH*

LOBSTER is an IST project that aims to develop a network of passive monitoring sensors. Collection of traffic data is not only important for operational needs, but also for examining traffic behaviour, trends, and future network provisioning. Our understanding of the Internet needs to be improved, but the gap between we need to measure and what we can measure is large and getting larger. Better monitoring



therefore needs to be developed in order to cope with multi-Gigabit networks and help close this gap.

LOBSTER builds on the achievements of the former SCAMPI project which developed the Monitoring Application Programming Interface (MAPI). This implemented a standardized monitoring API on a variety of network interface cards (e.g. the COMBO cards developed by CESNET, Endace DAG cards, and regular NICs), and is being extended by the LOBSTER project to support multiple distributed sensors. Distributed MAPI (or DiMAPI) is important for intrusion detection and early-warning applications, and has also been utilized in conjunction with the honeypot network being developed by the NoAH project.

All that is required to host a LOBSTER sensor is a normal PC with regular network interface card, and traffic mirrored from a router that is to be monitored. This works well for line speeds of up to 100 Mbps, and can even monitor 1 Gbps links if minimal string searching is employed. MAPI is then installed on the PC, which can be interrogated by a variety of off-the-shelf applications such as pcap, or alternatively users can write their own specialist applications. A faster PC with a COMBO or DAG card is required for monitoring higher speeds (up to 10 Gbps), but MAPI is also able to support this hardware.

LOBSTER sensors can benefit a variety of users such as NRENs/ ISPs, network administrators, security personnel, as well as researchers trying to understand network behaviour. In order to allay privacy concerns about collected data, LOBSTER also offers an anonymisation framework that is able to remove any confidential or sensitive information.

At the present time, LOBSTER sensors have been installed by NRENs in Norway and the Czech Republic, and there are discussions for collaboration with China and Singapore. These are currently in the test phase, but the LOBSTER infrastructure will start accepting members from July 2006. More information about how to join is available at <http://www.ist-lobster.org/>

This presentation can be found on the web at:

<http://www.ist-lobster.org/events/workshop-2006/markatos.pdf>

Streamline: an Efficient Steam-Processing Framework

Herbert Bos, Vrije Universiteit

As networks increase in speed and complexity, it is increasingly difficult to develop monitoring systems that are able to perform all of the necessary tasks. One solution is therefore to better exploit the hardware that is available, rather than relying on generic software approaches.



Streamline has been developed as high-speed networking sub-system that improves performance by moving processing tasks to the most appropriate location. It can either undertake this within an operating system kernel, in dedicated hardware, or even by moving such tasks to remote machines. Unlike other specialised processing systems, it aims to integrate these methods without burdening application developers or end-users with complex APIs. This system grew out of the earlier Fairly Fast Packet Filter (FFPF) work, but significantly extends its functionality.

Streamline utilizes a single framework for packet processing that uses all levels in the processing hierarchy, is language neutral, and offers advanced processing in external hardware such as NICs. It supports both stateless and stateful filters and is backwardly compatible with pcap whilst also supporting other more powerful packet languages. For each application it constructs a tailored datastream at runtime that utilizes the most optimal datapath for high performance.

To date, this framework has been successful utilized with Intel IXP network processors for intrusion detection and anonymisation applications. Further work is ongoing to automatically generate FPGA code on-the-fly using the Filter Processing Language (FPL) framework, which will further extend the scope of monitoring applications.

This presentation can be found on the web at:

<http://www.ist-lobster.org/events/workshop-2006/bos-streamline.pdf>

Ruler: Language-based Anonymisation

Herbert Bos, Vrije Universiteit

One of the main issues with monitoring shared infrastructures is the reluctance of network operators to provide access to data belonging to others on privacy or commercial grounds. This ranges from a complete refusal to release any data, a willingness to release generalised aggregated statistics, through to the sharing of specific details to selected groups of people. Ruler has therefore been devised as a high-level language for rewriting packets, and is particularly designed for anonymisation of traffic traces.

It is a simple, but efficient rule-based language that aims to make implementation in hardware possible. The library contains several different functions for anonymising data produced by common network protocols, based on pattern matching and filtering. A number of definitions are included, but further definitions can be added as necessary. Different anonymisation rules can also be applied for different users, and credential checking is employed to determine what sort of access is allowed.

Ruler has been integrated with MAPI so that Ruler programs can be applied to packets of arbitrary flows. A snort2ruler compiler has also been developed that is able



to translate most of the rules written for the Snort intrusion detection program, although some may still need to be manually processed.

Preliminary tests indicate that Ruler is able to process packets at gigabit rates, although only a small number of rules have been tested. Obviously, a larger number of rules are likely to affect performance, although the impact can be minimised through improved compilation. Perhaps the most likely limitation is the amount of space available for the compiled rules on network processors, although even this may become less of an issue with more advanced versions.

This presentation can be found on the web at:

<http://www.ist-lobster.org/events/workshop-2006/bos-ruler.pdf>

Hardware Anonymisation

Vladimir Smotlacha, CESNET

One of the issues with passive monitoring is that it entails access to potentially sensitive data in packet headers and payloads. It's therefore necessary to display the collected data in such a manner that the privacy of individual users is ensured. The usual procedure is to encrypt the source and destination addresses of individual packets, and completely remove their payloads.

Anonymisation has traditionally been undertaken in software which provides full control over the processing of the collected data, but has the drawback that system operators still have access to the raw data. By contrast, hardware anonymisation improves trust as the raw data is processed as it arrives, whilst also improving processing performance by taking fewer CPU cycles.

The LOBSTER project has been developing hardware anonymisation techniques using the COMBO series of cards in conjunction with the SCAMPI monitoring firmware. This involves a programmable transformation unit (TU) that acts as a nano-processor and allows packets to be classified in up to 256 ways. Transformations can be applied to headers according to specified constants as follows: in a pseudorandom fashion, as a hash using map tables, as a prefix-preserving IP address mapping, or any combination of these.

The current implementation of the TU uses a 50 MHz clock and processes packets in 16-bit chunks. This permits the transformation of 1500-byte packets at speeds of up to 788 Mbps, and 64-byte packers at speeds of 595 Mbps. However, these speeds can likely be increased by utilising the improved classification unit in the COMBO6X version of the cards.

This presentation can be found on the web at:

<http://www.ist-lobster.org/events/workshop-2006/smotlacha.pdf>



Using passive measurements to debug and improve end-to-end network quality

Arne Øslebø, Uninett

Uninett has deployed a number of measurement probes across its network in order to undertake real-time analysis of network traffic, and to collect long-term statistics. The aim is to monitor traffic for security purposes, to characterise the types of traffic on the network, to show trends and provision for future traffic growth, and to improve end-to-end monitoring. There are often difficulties determining where the problems are on end-to-end connections over multiple domains, so establishing probes between the core and each campus network helps with debugging.

The probes collect their data through an optical splitter on each campus link. Information about TCP streams is either collected using the extended NetFlow or tcpdebug applications, both developed by Uninett. The extended NetFlow application uses the IPFIX function in the MAPI software that was originally developed by SCAMPI project, and makes it possible to obtain QoS information about all flows on the network. This can then be stored in the Stager application so that trends can be shown, and if quality deteriorates, it is possible to fix the problem before customers complain.

The tcpdebug application was developed for real-time debugging on ongoing TCP streams. Detailed statistics can be provided for each TCP stream, based on user-selectable intervals which make it easier to monitor those with a long duration. At the present time, this is managed with a command line tool, but a web-based interface is planned.

The tcpdebug software is still under development, but beta versions are available from <http://mapi.uninett.no/>. A stable release will be available soon.

This presentation can be found on the web at:

<http://www.ist-lobster.org/events/workshop-2006/oslebo.pdf>

Passive monitoring for security-related applications

Michalis Polychronakis, FORTH

Passive monitoring is able to examine network traffic in a non-intrusive manner and can be used for performance monitoring, troubleshooting, planning and characterisation purposes. However, it can also be used for security purposes by inspecting traffic and attempting to identify any malicious activity. Computers are vulnerable to cyber-attacks such as break-ins (exploits) and worms (self-replicating malicious programs). Unlike viruses which generally rely on user activation, they can compromise systems without users even being aware of it.



Network Intrusion Detection Systems (NIDS) aim to provide early warning of such attacks, even for poorly administered systems such as home computers. They work by observing network traffic, flagging suspicious behaviour, and possibly blocking suspect packets. This can be achieved through signature matching, whereby packets are scanned for known malicious code patterns.

Unfortunately, pattern matching is very processor intensive (occupying 31-81% of NIDS processing time) and becomes more difficult as both line speeds, and the number of signatures to scan for increase. In addition, exploits and worms cause problems very quickly, which means there is a need for automated signature generation.

All this requires improved pattern-matching algorithms, which can utilise known characteristics of most cyber-attacks. For example, most worm attacks utilise small, but similar payloads, and are targeted at multiple hosts. They also tend not to contain null characters as many exploit buffer overflow vulnerabilities, and use some form of random scanning to find targets. Furthermore, attack connections are usually chained, so it is possible to correlate inbound and outbound traffic.

However, cyber-attacks have employed polymorphism in recent years, to mutate malicious code and make fingerprinting very difficult. This requires much more involved static binary code analysis that disassembles incoming requests and searches through the majority of redundant code for valid instructions. Network-level emulation is therefore being developed to execute incoming network traffic as if were executable code, but in a protected environment where it is unable to cause any harm.

This presentation can be found on the web at:

<http://www.ist-lobster.org/events/workshop-2006/polychronakis.pdf>