European Sixth Framework Network of Excellence FP6-2004-IST-026854-NoE

Deliverable D2.5

Virtual Laboratory Integration Report

The EMANICS Consortium

Caisse des Dépôts et Consignations, CDC, France
Institut National de Recherche en Informatique et Automatique, INRIA, France
University of Twente, UT, The Netherlands
Imperial College, IC, UK
Jacobs University Bremen, IUB, Germany
KTH Royal Institute of Technology, KTH, Sweden
Oslo University College, HIO, Norway
Universitat Politecnica de Catalunya, UPC, Spain
University of Federal Armed Forces Munich, CETIM, Germany
Poznan Supercomputing and Networking Center, PSNC, Poland
University of Zürich, UniZH, Switzerland
Ludwig-Maximilian University Munich, LMU, Germany
University College London, UCL, UK
University of Pitesti, UniP, Romania

For more information on this document or the EMANICS Project, please contact:

Dr. Olivier Festor
Technopole de Nancy-Brabois - Campus scientifique
615, rue de Jardin Botanique - B.P. 101
F-54600 Villers Les Nancy Cedex
France
Phone: +33 383 59 30 66
Fax: +33 383 41 30 79
E-mail: <olivier.festor@loria.fr>

Document Control

Title:	Virtual Laboratory Integration Report
Type:	Public
Editor(s):	Jürgen Schönwälder, Ha Manh Tran
E-mail:	j.schoenwaelder@jacobs-university.de
Author(s):	WP2 Partners
Doc ID:	D2.5

AMENDMENT HISTORY

Version	Date	Author	Description/Comments
0.1	2008-11-25	J. Schönwälder	Initial version of a LaTeX template
0.2	2008-11-28	D. Hausheer	Added EmanicsLab 2.0 details
0.3	2008-12-09	R. Sadre	Added trace collection and labeling details
0.4	2008-12-09	J. Schönwälder	Updated the tracelabel text
0.5	2008-12-15	R. Sadre	Updated the trace collection and labeling text
0.6	2008-12-19	D. Hausheer	Updated the EmanicsLab 2.0 text
0.7	2008-12-30	O. Festor	Updated the trace collection and labeling text
0.8	2008-12-31	D. Hausheer	Updated the EmanicsLab 2.0 text
0.9	2009-01-08	J. Schönwälder, H.M. Tran	Finalized the deliverable

Contents

1 Executive Summary

This fifth "Virtual Laboratory Integration Report" presents the activities of the virtual laboratory and common test-beds work package (WP2) in the second phase of the EMANICS project. This phase started in June 2007 with an open call for the first nine-month period covering the time June 2007 to March 2008. The results were reported in deliverable D2.4. This report covers the period April 2008 to December 2008. An open call for this period issued in March 2008 led to the selection of the following two projects:

1. The EmanicsLab 2.0 project led by UniZH focuses on integrating new partners into the EmanicsLab testbed and upgrading the EmanicsLab infrastructure. The improvement of the capacity of EmanicsLab and its geographic spread better serves the demands coming from research projects in terms of distributed computation and storage. The project integrates ten partners (UniZH, INRIA, UT, UPC, IUB, UniBwM, LMU, UCL, PSNC, UPI) and one user UniS. The testbed can be used by non-EMANICS partners. A charter has been established regulating EmanicsLab membership, EmanicsLab usage policy, and conflict resolution.

2. The network trace collection and labeling project led by UT aims at collecting and labeling network traces from various operational networks. It is important to maintain this activity and encourage new partners to join the project in order to provide trace data for trace analysis purposes. This activity also includes collaborative coordinated trace data collection, integrating all seven partners (UT, INRIA, IUB, UniZH, PSNC, LMU and UPI).

The extension of the EmanicsLab 2.0 has improved the collaboration of partners and provides a common shared infrastructure for research and education. This report briefly describes 15 different activities that have used EmanicsLab in the reporting period. The summer school on Network and Service Management hosted by the University of Zurich in June 2008 provided a tutorial on the testbed and its features for PhD students and non-EMANICS partners. This activity will likely yield more collaborative activities between EMANICS partners and non-EMANICS partners in the future.

The continuation of the trace collection and labeling project continuous the fruitful collaboration of partners who have worked together on this from the very beginning of the work package. The partners of the project met at a two-day workshop in July 2008 in Enschede and during the NETFLOW/IPFIX workshop in October 2008 in Munich to share experiences with trace data collection and trace data storage. A collaborative activity of coordinated trace data collection was planned and implemented in December.

Overall, the two funded projects have performed well during the reporting period and they have obtained their defined objectives.

2 Introduction

The second phase of the EMANICS project lasts eighteen months starting from July 2007 to December 2008. In this phase, the work package has three objectives: the establishment of collaboration environments, the development of supporting tools, and the creation and maintenance of trace repositories for research and educational purposes. The work package description defines two deliverables. Deliverable D2.4 documents the work done in the period July 2007 until March 2008. This report (deliverable D2.5) documents the work carried out in the period April 2008 until December 2008.

An open call for this period for the second nine-month period was issued in March 2008 has resulted in two supported projects:

- The EmanicsLab 2.0 project integrates ten partners: UniZH, INRIA, UT, UPC, IUB, UniBwM, LMU, UCL, PSNC and UPI. This project is a continuation of the EmanicsLab testbed established in the first nine months period of the second phase of EMANICS. The EmanicsLab is a small version of PlanetLab providing distributed computing and storage resources to EmanicsLab users. Compared to PlanetLab, EmanicsLab has a much smaller user base and can provide much richer resources to research projects. During the reporting period, new partners have been integrated into EmanicsLab and the testbed software has been updated and extended. Finally, as recommended by the evaluators, it has been investigated how the testbed can be opened to non-EMANICS users and a charter has been established regulating EmanicsLab membership, EmanicsLab usage policy, and conflict resolution.
- The network trace collection and labeling project integrates seven partners: UT, INRIA, IUB, UniZH, PSNC, LMU and UPI. This project has been funding network trace data collection and labeling activities such as the collection and labeling of NETFLOW data sets, the collection and labeling of full packet traces, or the collection and labeling of network management traffic traces. This project is a continuation of trace collection activities that started at the beginning of WP2.

The two projects integrate several partners and they provide support for other work packages. EmanicsLab provides a collaborative infrastructure for partners for research and education. This project builds up not only collaborations among partners but also supports collaboration between work packages. The EmanicsLab infrastructure has been used by 15 research and educational activities during the reporting period linked to three research work packages WP7, WP8, and WP9.

The trace collection and labelling project is linked research carried out in work package seven (WP7). It provides support by providing network traces captured by different operators and at different locations. At the end of the reporting period, a coordinated trace collection activity has been started aiming at the collection of traces during the same time period at different location. This collaborative activity will support intrusion detection research in WP7.

The rest of the deliverables is structured as follows. Section 3 documents the evolution of the EmanicsLab to what is called EmanicsLab 2.0. Section 4 reports the traffic trace collection and labeling work performed in the second activity. Section 5 reports collaboration issues in the work package before the deliverable concludes in Section 6.

3 EmanicsLab 2.0

The main purpose of the EmanicsLab 2.0 activity was a continuation of the EmanicsLab testbed [1] which has been successfully setup and widely used in the previous phase. In particular, 3 new partners, namely PSNC, UCL, and UPI, were integrated into EmanicsLab and the testbed has been updated and extended. The upgrade to the newest MyPLC version [2, 3] includes a new Linux kernel and distribution. The new kernel is required to support the newer hardware brought in by the new partners integrated into EmanicsLab, while the newer distribution benefits all EmanicsLab users. In addition, the newest MyPLC version corrects several bugs of the older version and includes enhancements of the management interface. Table 1 shows the differences of EmanicsLab 2.0 compared to the previous version.

Component	**EmanicsLab 1.0**	**EmanicsLab 2.0**
MyPLC	0.5	4.2
Linux kernel	2.6.12	2.6.22
Linux distribution	Fedora Core 4	Fedora 8

Table 1: Difference Between Old and New EmanicsLab Version

Meanwhile, all existing EmanicsLab partners provided continuous maintenance of their local nodes such as reboots after power failures. UPC had a meeting with their network administrators and managed to get the full port range opened for experiments on EmanicsLab after their nodes were moved into a different network. LMU was able to adapt their monitoring policy, which solved previous problems with experiments on their nodes. Many EMANICS members are continuously using EmanicsLab in their research and teaching activities.

Additionally, it has been discussed how the testbed can be opened, i.e. under what circumstances new participants may join EmanicsLab. To this end, a set of rules have been defined based on which non-EMANICS partners will be able to join. Moreover, the specification of an EmanicsLab usage policy has been started, which defines what are acceptable uses of the testbed. This was required by UCL to join the testbed.

In order to make a continuation of EmanicsLab on the long-term beyond the life-time of the EMANICS project possible, potential peering solutions with other testbeds will be investigated in the next phase.

3.1 EmanicsLab Overview

An overview on the different sites and nodes after integration of the new partners in EmanicsLab is provided in Figure 1. At this stage, the EmanicsLab research network includes 11 sites with 20 nodes. The different sites are outlined in Table 2. Every partner site (except UniS which is only a user of the testbed) provides two nodes as outlined in Table 3. Furthermore, for every site a principal investigator (pi) and a technical contact (tech) had to be determined. Additionally, UniZH serves as EmanicsLab Administrator (admin). The key users and their roles are outlined in Table 4.

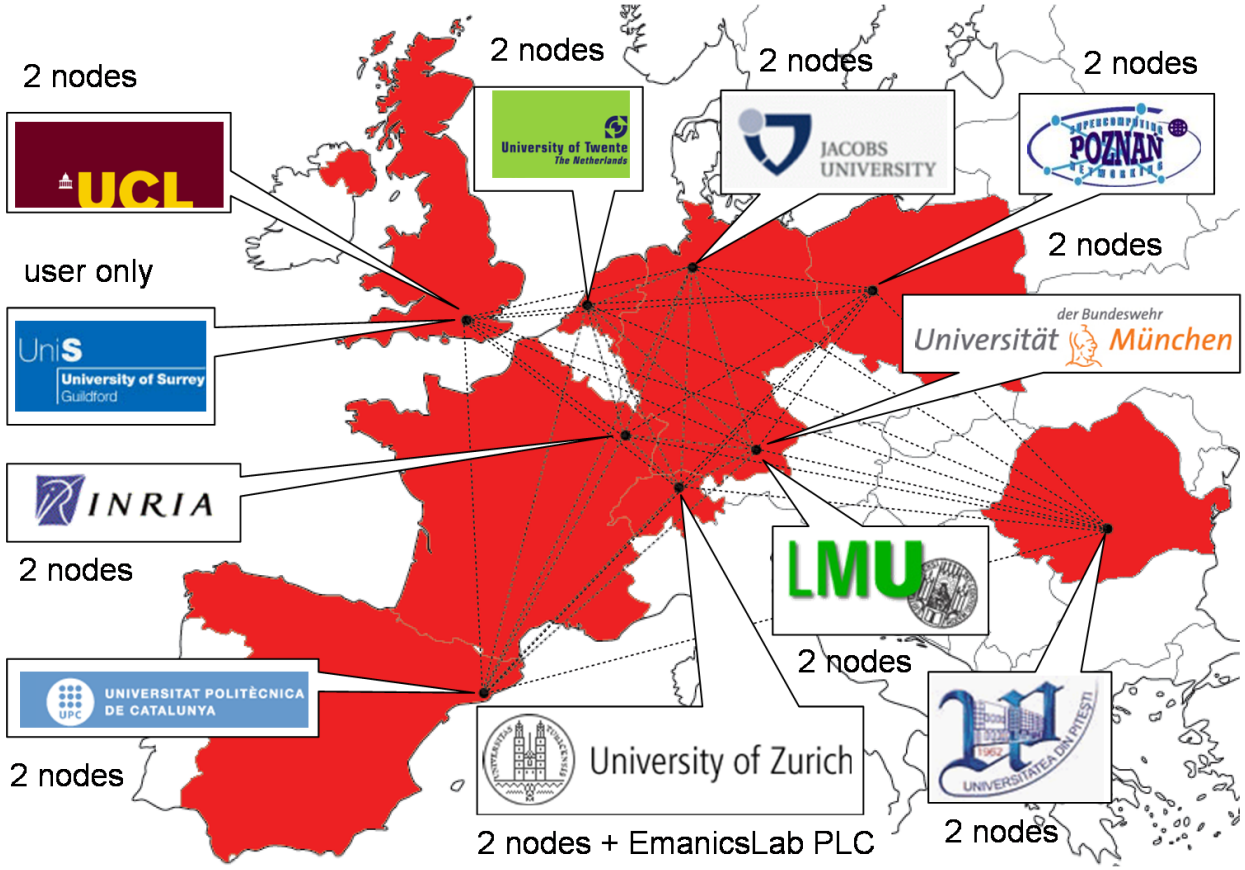

Figure 1: EmanicsLab Research Network

Abbreviated Name	Name	Login Base
EmanicsLab	EmanicsLab Central	pl
INRIA	Institut National de Recherche en Informatique et Automatique	inria
IUB	Jacobs University Bremen	iub
LMU	Ludwig-Maximilian University Munich	lmu
PSNC	Poznan Supercomputing and Networking Center	psnc
UCL	University College London	ucl
UPC	Universidat Politechnica de Catalunya	upc
UniBw	University of Federal Armed Forces Munich	unibw
UniS	University of Surrey	unis
UPI	University of Pitesti	upi
UT	University of Twente	ut
UZH	University of Zurich	uzh

Table 2: EmanicsLab Sites

Site	Hostname	Model
INRIA	host1-plb.loria.fr	Dell Precision 360, Pentium 4, 3.0 GHz, 2 GB RAM, 750 GB HDD
INRIA	host2-plb.loria.fr	Dell Precision 390, Core 2 X6800, 2.93 GHz, 3 GB RAM, 750 GB HDD
IUB	emanicslab1.eecs.jacobs-university.de	Dell OptiPlex GX620, Pentium D, 2.8 GHz, 1 GB RAM, 80 GB HDD
IUB	emanicslab2.eecs.jacobs-university.de	Dell OptiPlex GX620, Pentium D, 2.8 GHz, 1 GB RAM, 80 GB HDD
LMU	emanicslab1.lab.ifi.lmu.de	Fujitsu Siemens Scenic, Pentium 4, 3.0 GHz, 1 GB RAM, 400+400 GB HDD
LMU	emanicslab2.lab.ifi.lmu.de	HP DC7700, Core 2 6400, 2.13 GHz, 3.6 GB RAM, 80+500+500 GB HDD
PSNC	emanicslab1.man.poznan.pl	HP ProLiant DL320R05p, Xeon (Dual Core), 2.66GHz, 1GB RAM, 500 GB HDD
PSNC	emanicslab2.man.poznan.pl	HP ProLiant DL320R05p, Xeon (Dual Core), 2.66GHz, 1GB RAM, 500 GB HDD
UCL	emanics1.ee.ucl.ac.uk	Sun Ultra 24, Intel Core2 Quad, 2.5Ghz, 4 GB RAM, 750 GB HDD
UCL	emanics2.ee.ucl.ac.uk	Sun Ultra 24, Intel Core2 Quad, 2.5Ghz, 4 GB RAM, 750 GB HDD
UPC	muro.upc.es	Custom Model, Core 2 6400, 2.13 GHz, 1 GB RAM, 250 GB HDD
UPC	moscu.upc.es	Custom Model, Athlon XP 1600+, 1.4 GHz, 1 GB RAM, 200 GB HDD
UniBw	emanicslab1.informatik.unibw-muenchen.de	Dell PowerEdge 2850, Xeon, 3.0 GHz, 2 GB RAM, 150 GB HDD
UniBw	emanicslab2.informatik.unibw-muenchen.de	Dell PowerEdge 2850, Xeon, 3.0 GHz, 2 GB RAM, 150 GB HDD
UPI	emanicslab1.upit.ro	Intel DQ35MP, Core 2 Duo E6750, 2.66 Ghz, 4 GB RAM, 1 TB HDD
UPI	emanicslab2.upit.ro	Intel DQ35MP, Core 2 Duo E6750, 2.66 Ghz, 4 GB RAM, 1 TB HDD
UT	emanicslab1.ewi.utwente.nl	Dell PowerEdge 860, Dual Core Xeon 3070, 2.66 GHz, 4 GB RAM, 2 TB HDD
UT	emanicslab2.ewi.utwente.nl	Dell PowerEdge 860, Dual Core Xeon 3070, 2.66 GHz, 4 GB RAM, 2 TB HDD
UZH	emanicslab1.csg.uzh.ch	Dell PowerEdge 850, Pentium 4, 3.6 GHz, 1 GB RAM, 500 GB HDD
UZH	emanicslab2.csg.uzh.ch	Dell PowerEdge 850, Pentium 4, 3.6 GHz, 1 GB RAM, 500 GB HDD

Table 3: EmanicsLab Nodes

Site	Firstname	Lastname	Email	Roles
INRIA	Emmanuel	Nataf	nataf@loria.fr	pi, tech
IUB	Jürgen	Schönwälder	j.schoenwaelder@jacobs-university.de	pi, tech
LMU	Feng	Liu	liufeng@mnm-team.org	pi, tech
PSNC	Krzysztof	Nowak	krzysztof.nowak@man.poznan.pl	pi, tech
UCL	Marinos	Charalambides	m.charalambides@ee.ucl.ac.uk	pi, tech
UPC	Pau	Valles	pvalles@nmg.upc.edu	pi, tech
UniBw	Frank	Eyermann	frank.eyermann@unibw.de	pi, tech
UniS	Stylianos	Georgoulas	s.georgoulas@surrey.ac.uk	pi, tech
UPI	Florin	Smaranda	florin@upit.ro	pi, tech
UT	Ramin	Sadre	sadrer@ewi.utwente.nl	pi, tech
UZH	Cristian	Morariu	morariu@ifi.uzh.ch	admin, pi, tech
UZH	David	Hausheer	hausheer@ifi.uzh.ch	admin, pi, tech
UZH	Thomas	Bocek	bocek@ifi.uzh.ch	pi, tech

Table 4: EmanicsLab Key Users and their Roles

3.2 Testbed Usage

In the reporting period, 15 activities have used EmanicsLab for education and research (cf. Table 5).

Slice Name	Leading Researcher
uzh_psh	Thomas Bocek
uzh_p2pcost	Dalibor Peric
uzh_datta	Cristian Morariu
uzh_smoothit	Fabio Hecht
uzh_iploc	Martin Waldburger
uzh_voip	Gregor Schaffrath
uzh_issnsm	Cristian Morariu
ut_snid	Ramin Sadre
upc_sblomars	Pau Valles
iub_anl	Jürgen Schönwälder
iub_buglook	Jürgen Schönwälder
inria_p2psip	Olivier Festor
inria_p2prevocation	Thibault Cholez
inria_p2psybilattack	Thibault Cholez
unibw_asam	Frank Eyermann

Table 5: EmanicsLab Slices

For each of those activities a slice has been created on EmanicsLab. The purpose of each slice is described in the following.

3.2.1 PSH (uzh_psh)

The uzh_psh slice has been used for testing and simulating the private shared history incentive mechanism (PSH) in a distributed environment. For that purpose, 12 nodes were used to run peer clients and simulate a file sharing scenario. Every node ran up to 100 peer clients. In this scenario, an automated script generated file sharing traffic. From this traffic, figures were generated to show and compare the performance of PSH to other incentive mechanisms. Furthermore, PSH2, which is an enhancement to PSH, also uses this slice and measurements for PSH2 are ongoing.

3.2.2 P2PCOST (uzh_p2pcost)

The EmanicsLab testbed was used in the uzh_p2pcost slice to test and evaluate the Distributed Reliable File System (DRFS) regarding its scalability (with respect to the number of the nodes in the system) and data availability in case of node failures. DRFS is intended to provide a cooperative peer-to-peer file system service. In such a cooperative environment, the presence of malicious nodes is not expected, and therefore the security is not a

main concern. However, participating peers can still fail: to improve the availability, a data object is replicated and stored in different nodes.

By using EmanicsLab's testbed, it was possible to compare different-sized DRFS systems, consisting of a dozen nodes up to hundreds of nodes, and to monitor the message exchange between them. The results showed that the number of messages exchanged remained constant, which proved DRFS' scalability. Another test consisted of measuring data availability in case of node failures. Results showed that the system provides a 99.9% data availability even when 26% of the nodes are down.

3.2.3 DATTA (uzh_datta)

The key aim of this project is the design and prototypical implementation of a distributed NETFLOW collection platform integrated within EmanicsLab. The flow records collected by each partner are stored locally on a separate host within the premise of the partner. The platform offers a client interface for requesting flow records from any of the participating partners whereas the EmanicsLab nodes act as brokers between clients and storage repositories and additionally have the task to control access. Moreover, the platform offers an API that allows developers to build their own applications on top of the provided library.

3.2.4 SmoothIT (uzh_smoothit)

This activity is using EmanicsLab to evaluate proposed ETM mechanisms for the ICT-SmoothIT project. The testbed is being used to simulate a peer-to-peer network for live video streaming. Such a bandwidth and CPU-intensive activity is difficult to perform on PlanetLab [4], because many nodes are constantly overloaded with other processes. Topologies of ISPs are simulated within the testbed to investigate optimization potential involving direct communication between ISPs and peer-to-peer applications. Results suggest that, achieving localization of traffic, a peer-to-peer live streaming application can profit from information provided by the ISP to increase its quality of experience for end users and reduce cost for the ISP. Other possibilities are being investigated within this scope.

3.2.5 IPLoc (uzh_iploc)

IPloc is an IP address query tool that performs lookups at all five Regional Internet Registries (RIR) to determine a client's location with an information granularity of country or state. EmanicsLab was used to conduct functional and performance evaluation. In a local deployment within the UniZH network, it had been found that the average amount of achievable queries per second varied significantly from RIR to RIR. RIPE NCC provided very fast lookup times with over 14 queries per second. Queries for LACNIC and APNIC performed much slower at 0.77 to 0.83 queries per second. These differences were supposed to result from various geographical distances between the query machine and a RIR's whois server as well as its service capacities in terms of processing power or network link speed. In order to verify location-dependent query times, all evaluation tests that

had run locally were run again from distributed sites using the EmanicsLab and PlanetLab platforms. Query hosts located geographically closer to a specific RIR were found to achieve significantly higher query rates for this RIR and lower rates for other whois servers. For example, a query host located in Brazil almost doubled its queries per second and at the same time achieved about 10 times less queries per second for RIPE NCC's whois server located in the Netherlands. Query hosts located in Japan achieved higher query rates for APNIC and again few queries per second for whois servers further away. Thus, it can be concluded that a query host's location shows a significant impact on lookup performance. However, there is still a notable difference in the whois server's processing power to be considered.

3.2.6 VoIP (uzh_voip)

This slice has been allocated for SIP4EMANICS related experiments. It served the following four purposes: (1) Feasibility evaluation to run actual asterisk instances inside of EmanicsLab, (2) prototypical implementation of an experimental VoIP environment as proof of concept of the defined SIP4EMANICS VoIP architecture design, (3) evaluation of potential issues with respect to specific asterisk components in the EmanicsLab environment, and (4) field for initial experimentation and sandbox for potential SIP4EMANICS research users to allow familiarization with the environment and concepts before the actual project draft and start.

3.2.7 ISSNSM (uzh_issnsm)

The uzh_issnsm_* slices were used during the EMANICS summer school 2008 [5] at which a tutorial about EmanicsLab was given. This tutorial introduced the basic concepts of virtual distributed test-labs like PlanetLab or EmanicsLab and gave a hands-on training about how to use them for research activities. In particular, the monitoring and management capabilities of EmanicsLab were shown and a set of practical exercises were carried out based on a simple service which was deployed on EmanicsLab.

3.2.8 SNID (ut_snid)

The SNID slice has been created to support research activities in the area of large scale network analysis and intrusion detection (originally in the context of WP7 activity SNID, now SNAID). NETFLOW traffic traces from different sources have been collected in this context. Stemming from high-speed networks, the traces pose a challenge to the evaluation: a two-day trace collected at the UT comprises around one billion records. To improve the efficiency of the analysis, the traces are stored in relational SQL databases running in this slice. The collected data has been used to develop intrusion detection algorithms for different kind of well-known attack types (scans, spamming, etc.) and for the characterization of anomalies by analyzing time series of basic flow features.

3.2.9 SBLOMARS (upc_sblomars)

The purpose of this slice is to deploy SBLOMARS, a distributed network monitoring system to track the usage of network node resources as well as end to end network transmission parameters. The system is installed in each network node and after booting it instantiates as many monitoring agents as needed to monitor the different resources existing in the node (one agent per resource). All these agents run as independent software threats. Data in different time windows is stored in the local node, ready to be exported to any other system. When used in cooperation with a scheduling system, the data collected by the monitoring agents will be exported to the scheduler. Nevertheless, scenarios with a P2P data exchange mechanism could also be considered. Initially conceived for large-scale grids, the system has been adapted to be deployed in EmanicsLab. Potential use of this monitoring system is to help the Lab administrator in his role but also guide users when they plan to deploy their experiments. In its current version, the system is providing per node memory and CPU usage only. The intention is to make the appropriate adaptation of the system in order to be able to monitor at a per slice basis. Data is provided through a web interface that shows the nodes usage snapshots updated every minute.

3.2.10 ANL (iub_anl)

The iub_anl_* slices were used during the Advanced Networking Lab course [6] at Jacobs University. Students used the slices to create a distributed VoIP installation, that is a network of Asterisk servers communicating via IAX and providing access to phones via SIP. In general, the feedback of the students was quite positive; the only downside being that EmanicsLab does not provide true virtual machines so that things such port number usage had to be coordinated.

3.2.11 Buglook (iub_buglook)

This slice was used for the testing and evaluation of buglook, a web crawler for bug tracking systems. The buglook system is indexing bug trackers and storing information in a unified bug tracking data model. Feature vectors and semantic vectors are computed in order to search in the extracted data base. The data was accessible through a simple web frontend and there was an interface to a distributed case-based reasoning system. The slice was linked to research in WP9 of the EMANICS project.

3.2.12 P2P-SIP (inria_p2psip)

P2PSIP can be considered as an extension work of SIP, which addresses the use of distributed resource discovery and management instead of the centralized architecture present in the traditional SIP network. This approach is mainly beneficial in terms of scalability and reliability when compared to single point failure in centralized network. As part of the SIP4Emanics activity, INRIA did extend the architecture to support P2PSIP while staying compatible with standard SIP as well as with the other SIP4Emanics initiatives. This is

described in deliverable 2.4. EmanicsLab was successfully used to easily deploy and test the P2PSIP environment and its interoperability with the remainder of SIP4Emanics.

3.2.13 P2P Revocation (inria_p2prevocation)

The slice aims to evaluate the performance of a distributed revocation mechanism on the widely deployed P2P network KAD. Modified aMule clients have been launched on the slice. They implement the revocation mechanism and monitoring functions. In particular, they manage a new kind of information, replicated on several peers, telling who has to be revoked. In this context, we study how the number of peers storing the information affects the performance of the mechanism in a real P2P network. EmanicsLab gives us the possibility to run clients on different nodes being directly connected to the P2P network.

3.2.14 P2P Sybil Attack (inria_p2psybilattack)

The purpose of the slice is to execute modified aMule clients to evaluate if the indexation mechanism of KAD can be cheated by a very localised attack despite the new protection mechanisms. In fact, the last version of KAD clients (eMule 0.49x and aMule 2.2.x) implement new protection mechanisms to mitigate the Sybil attack. To our knowledge, large Sybil attacks from a single source are effectively removed but it should still be possible to take locally the control over the P2P network with a distributed attack, as long as the KADID can be freely chosen and the Sybils placed very close to the target in the address space.

3.2.15 ASAM (unibw_asam)

This slice is used for experiments within the ASAM activity. The objective of ASAM (Auditing of SLOs Across Multiple Provider Domains) is to develop an architecture and a new protocol for metering and auditing of network performance across multiple, co-operating network providers. In this context the auditing of Service Level Agreements (SLA) defines the process of monitoring whether a service provider delivers agreed upon service levels or not. While frameworks exist to monitor application level SLAs, the end-to-end monitoring of IP-carrying SLAs, especially in a multi-domain environment like the Internet, is still an open issue. Measurements from within EmanicsLab will be used to parametrize ASAM simulation scenarios.

3.3 Testbed Monitoring

Measurement about the resource usage in the EmanicsLab testbed have been taken using Ganglia and Sblomars, as presented in the following.

3.3.1 Ganglia

The purpose and installation of Ganglia in EmanicsLab has been presented in detail in the previous deliverable D2.4.

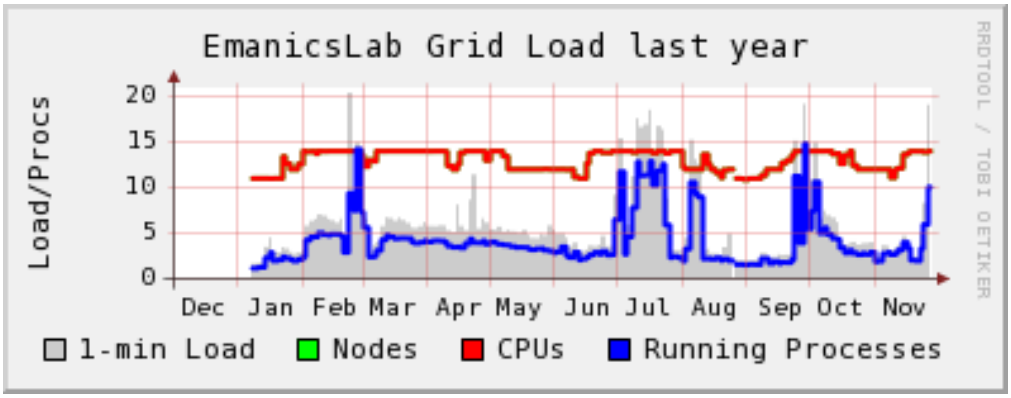

Figure 2: Resource Usage in EmanicsLab

Figure 2 is an updated Ganglia graph showing the resource usage in EmanicsLab in the reporting period. Additional figures can be found on the Ganglia website [7].

3.3.2 Sblomars

UPC is using EmanicsLab for the development of Sblomars, a software package intended for monitoring large scale distributed systems. More specifically, they have tuned this software to be applicable to the monitoring of computational resources in the EmanicsLab environment. The application works installing an image in each computational node of EmanicsLab. From the point of view of EmanicsLab, Sblomars is another application working in a virtual machine, one among the different slices that a node can support. Once installed, Sblomars detects how many threads it has to deploy in order to monitor the different computational resources existing in the node. These resources are, memory usage, CPU usage, disc usage and eventually it can also monitor end-to-end delay and jitter of links involving the particular node. All these results are periodically and automatically updated in a web page that constitutes the interface through which any user can access activity data of the virtual laboratory.

Figure 3 and Figure 4 are two screenshots of Sblomars showing the CPU and Memory used in the two EmanicsLab nodes of UPC and UT. The URL to the Sblomars web interface will be made available by UPC to all EmanicsLab users soon.

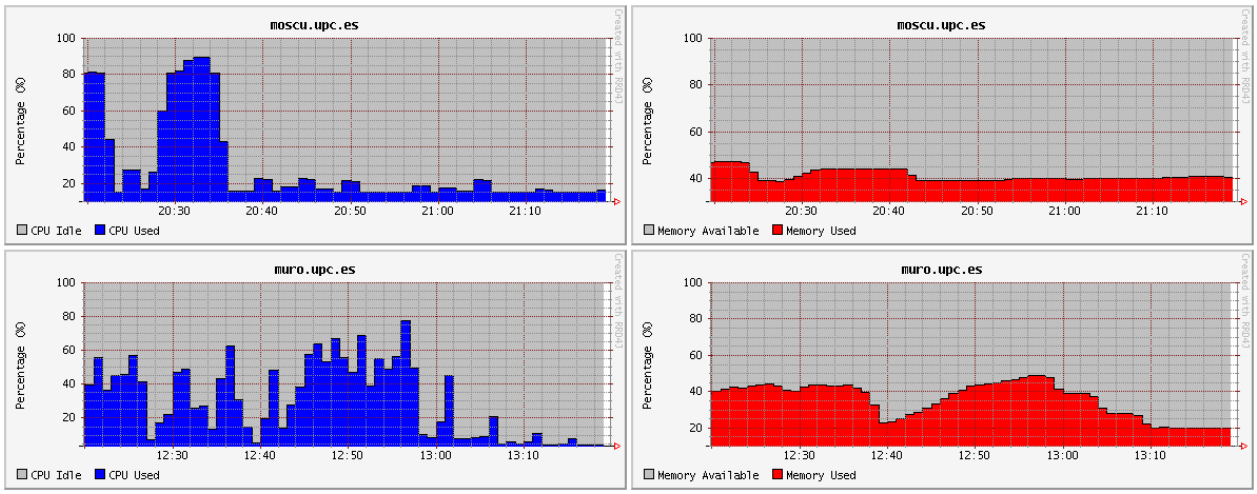

Figure 3: Resource Usage in UPC's EmanicsLab nodes measured by Sblomars

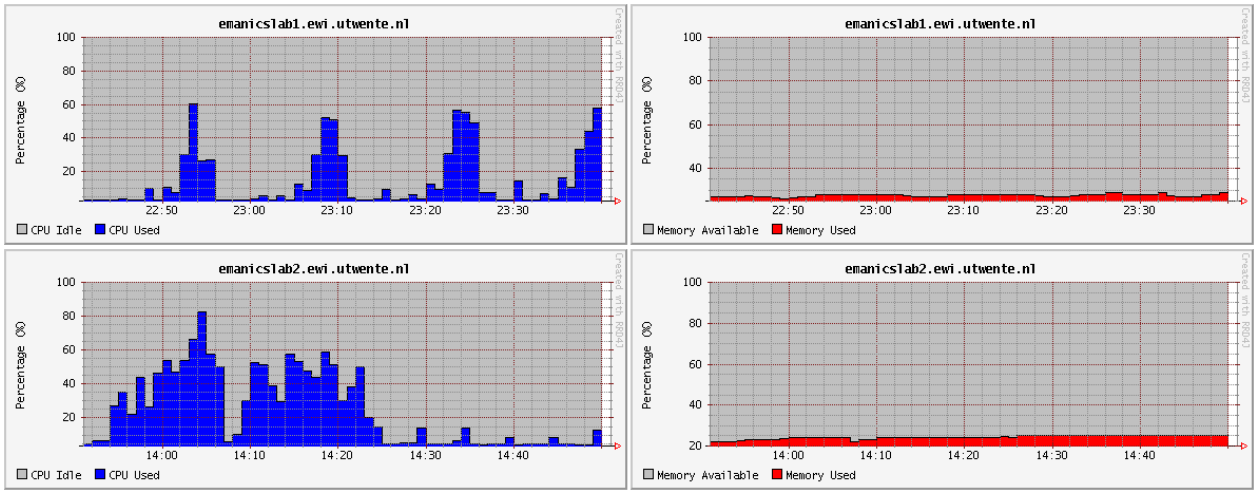

Figure 4: Resource Usage in UT's EmanicsLab nodes measured by Sblomars

3.4 EmanicsLab Charter

An EmanicsLab Charter has been defined. This was required by UCL to join the testbed. The charter includes a membership policy and a usage policy and defines how problems shall be resolved.

3.4.1 Membership Policy

The EmanicsLab membership policy governs the addition of EmanicsLab members.

- In general, only EMANICS partners can join EmanicsLab.
- Non-EMANICS partners can join EmanicsLab if there is a justified relationship to EMANICS.
- The decision about new partners has to be taken by the EmanicsLab ”steering committee” (who later informs the EMANICS Execom) with unanimous ”consensus”. Until there is an EmanicsLab ”steering committee”, the EMANICS Execom can perform this task.
- The principal investigator (PI) of a site creates new slices on EmanicsLab and thus decides which specific activities are allowed to make use of EmanicsLab.
- The usage charter defines what are acceptable uses of EmanicsLab. The PI of a site is responsible that each slice of its site complies with the usage charter. Failure to do so may result in the exclusion of the site from EmanicsLab.

3.4.2 Usage Policy

The EmanicsLab usage policy governs the use of EmanicsLab similar to the “PlanetLab Acceptable Use Policy” [8].

Acceptable uses of EmanicsLab include:

- Research activities or production services of an EmanicsLab partner, with direct or indirect relation to EMANICS

Unacceptable uses of EmanicsLab include:

- Denial of Service (DoS) attacks against 3rd party nodes
- Password attacks against 3rd party nodes
- Publicly accessible repository of copyrighted content
- Repository of illegal content
- Open email relays or web proxies without corresponding protection measures
- Any other illegal activities in any of the countries

3.4.3 Problem Resolution

The first contact point when a problem arises (e.g., a process on a node that sends or receives some strange traffic) is the PI of the respective site.

The PI decides how urgent the problem is. If it is really urgent (e.g., a DoS attack) he can block certain ports or even shutdown the node and will inform the EmanicsLab admins about it. Then they resolve the problem together.

If the problem is not so urgent, the PI contacts the researchers responsible for the respective slice from which the problem originates (with a copy to the EmanicsLab admins). Usually, the problem turns out to be not severe and can be resolved quickly, between the PI and the slice user. If not, the PI can ask the EmanicsLab admin to take the necessary actions to resolve the problem (e.g., remove a slice).

4 Network Trace Collection and Labeling

The goal of this activity was to perform network trace data collection and labeling activities such as the collection and labeling of NETFLOW data sets, the collection and labeling of full packet traces, or the collection and labeling of network management traffic traces.

4.1 Network Trace Collection

A new traffic trace has been recorded at UT in September. For that trace, the NETFLOW data from three different networks (UT, SURFnet, Geant) has been collected. A similar collection was already done in the previous COLLECT activity in WP2. For the new trace, UT has tried to solve, respectively, work around, several problems and limitations that were identified after analyzing the results of the previous collection activity.

In addition to the NETFLOW data, UT has collected at the same time several other data sources, namely: (i) the packet headers of the IP packets in a subnetwork of the UT, (ii) the log files of UT's honeypot and of other services (for example, the DNS server), (iii) log files from a honeypot set up by us at the UT. This additional data is used to identify and label relevant (security) incidents in the collected NETFLOW data (see section 4.2).

The SNMP Traffic Measurement document edited by Jacobs University has been progressed through IRSG review and IRSG and IESG approval. Some review comments led to a new version of the document, which went through the editing process and got published as RFC 5345 in October 2008 [9].

A coordinated trace collection has started on December 1st. PSNC collects NETFLOW v5 traces (sampled 1:1000) from networks controlled by PSNC Network Operations Center. Traces are collected periodically and are available to partners upon request after signing proper Non Disclosure Agreement. UniZH collects NETFLOW v5 data from a router that connects their testbed (including PlanetLab nodes) to the SWITCH network. The data is available to EMANICS members upon request. IUB collects NETFLOW v5 data from their PlanetLab/EmanicsLab nodes, available to EMANICS members upon request. This coordinated collection of data at several network locations during the same period of time is especially useful for the monitoring and analysis of non-local incidents in the Internet. Vast and highly distributed scanning activities, for example initiated by botnets, worm spreads, etc. are usually hard to identify if data from only one measurement point is available. Using data from a distributed and coordinated collection allows to correlate the single incidents observed at different points in the network.

4.2 Network Trace Labeling

In general, labeled traffic traces where the characteristics are known and relevant incidents in the data are tagged are rare, mainly because privacy concerns prevent researchers to publish traffic data sets. In particular, such labeled (or annotated) data traces are needed in the field of intrusion detection: the performance of an intrusion detection system can only be evaluated by a test traffic trace if all attacks in the trace are known. The lack

of such labeled traces has forced the researchers to either use non-public, and, hence, non-comparable, traces or to rely on the rather outdated but public DARPA data set from 1999.

The purpose of this activity is twofold. First, we want to create and publish a traffic trace where attacks and other malicious activities are tagged. Secondly, we want to gain experience in (i) manually/semi-automatically labeling traffic traces and (ii) storing the labeling information so that it can be used by other researchers.

In order to collect the trace, we set up a honeypot inside the UT network, representing a typical server in a company hosting web, ftp, a database and ssh. Of course, the data obtained from such a honeypot is limited because all traffic to the honeypot can be considered as harmful. However, this restriction simplifies the labeling process and reduces privacy concerns in the case that the data is shared. In addition, since the measurement setup emulates a typical company server, our data can be combined with real traffic traces from similar setups to create more complete traces.

4.2.1 Honeypot Setup

In the chosen setup, the honeypot was installed on a virtual machine running on a Citrix Xen server. The decision for virtualizing the honeypot are due to the following reasons:

- easy to configure and install
- easy to extend to multiple honeypot-hosts (for this experiment we limited ourself to one virtual host)
- the virtualization offer us the possibility to be always able to manage the virtual host also in case it is badly compromised or unreachable (possibility to stop the virtual machine)
- it is possible to save the virtual machine for later analysis.

The virtual machine was running a Debian Etch 4.0 operating system. In a previous attempt, the logging capabilities of honeypot software like Honeyd and Nepenthes appeared to be not suitable for our necessities. So we decided not to rely on existing honeypot software but to configure the host ourself. In particular, the following services have been installed:

- SSH (sshd) [10]: The Openssh service running on Debian has been patched in order to log each session. For each login, the transcript of the session and the timing are recorded. We consider this patch particularly important since it allows us to keep track of active hacking activities, such as download of malicious software, starting of malicious activities, or maybe only to observe the hackers behaviour.
- FTP (proftpd): The FTP service has logging capabilities for attempted and successful connections.
- APACHE + MYSQL: A simple webpage with a subscription form has been deployed. Apache logs the content of HTTP connections.

Column	Type	Description
id	unsigned int	primary key
srcip, dstip	unsigned int	source and destination address
srcport, dstport	unsigned short	source and destination port
starttime, endtime	unsigned long	start and end time of the flow (UNIX timestamp, millisecond resolution)
packets, octets	unsigned int	number of packets and bytes in the flow
tcpflags	unsigned byte	TCP flags
prot	unsigned byte	IP protocol number

Figure 5: Flow table structure

The virtual machine did run for one week. During the first three days, it collected almost no traffic. In the last three days, the honeypot was the origin of multiple SSH scans, providing us with enough data to be analyzed. We decided to restrict ourself to this period of three days. During the collection period, we collected the tcpdump file of all the traffic reaching and leaving the honeypot (24GB, around 155 million packets).

4.2.2 Labeling

For the labeling, we have only considered UDP and TCP traffic. The labeling has been performed on flow-level, i.e., labels are assigned to entire groups (flows) of IP packets

- with same source, resp., destination IP address and port number, and
- separated by periods of time not longer than specific time-outs. In the case of TCP traffic, a packet with a set FIN flag is considered as the last packet of a flow.

The flows have been created from the tcpdump file using a modified version of Softflowd[1]. This decision is due to the fact that relying on an external probe (such as for example a flow exporter on the university router) would introduce timing problems that would make the alert/flow correlation unfeasible (clock skew). The processing of the tcpdump file resulted in around 14 million flows that we have stored in a MySQL database. Figure 5 shows the structure of the table.

For each incident found in the trace an alert entry is created in the database. Figure 6 shows the structure of the alert entries. The alerts are either created by analyzing the packet dump files and flow information for malicious activities or by analyzing the log files of the various services running on the honeypot, such as the log files of the web server, the typescripts of the SSH sessions, etc. If the alert entry has been created from the information found in a log file, the timestamp information in the log file has to be used to assign the alert to one or more flows. The connection between a flow and an alert is then stored in a separate table (see Figure 7). This helps to keep the labeling information separated from the flow information.

Alerts can also be grouped to clusters (Figure 8). For example, the single scan attempts of a port scan are grouped to one alert representing the whole scan run. Attacks that

[1] `http://www.mindrot.org/projects/softflowd/`

Column	Type	Description
id	unsigned int	primary key
srcip, dstip	unsigned int	source and destination address
srcport	unsigned short	source port
timestamp	unsigned int	time of the attack (UNIX timestamp)
type	int	type of the attack (SSH scan, HTTP scan,...)
automated	boolean	the attack was automated (scripted)
succeeded	boolean	the attack succeeded
description	text	description

Figure 6: Alert table structure

Column	Type	Description
flow id	unsigned int	id of flow
alert id	unsigned int	id of alert

Figure 7: Flow↔Alert table structure

are interdependent can be structured into hierarchies using the table shown in Figure 9. For example, the attack to gain access to the honeypot is regarded as the ancestor of subsequent attacks performed from the honeypot to other hosts.

The labeling of the trace is still in process. Preliminary results show that most of the attacks toward the honeypot are automated SSH scans or scripted attempts to exploit vulnerabilities in the web server and installed web applications. We have also recorded around 50 non-automated SSH sessions where the attackers installed malicious software on the honeypot and used this software to attack other hosts outside the UT. In particular, the honeypot was used for further SSH scans. Around 83% of all recorded flows are stemming from those scans.

4.3 Security Attacks Network Traces and Flow Labelling

The High Security Laboratory (LHS) at INRIA is designed to enable researchers to perform certain experiments under a legal umbrella, with the possibility to publish results and data. The experiments considered are the deployment of attack and defense systems against malicious programs, the usage of viral technologies to develop new technologies, vulnerabilities detection, security audit and systems certification. The LHS is composed of two projects closely related:

- A network telescope, which role is to collect malware together with network traces in order to analyze them;

Column	Type	Description
cluster id	unsigned int	id of cluster alert
child id	unsigned int	id of child alert

Figure 8: Cluster table structure

Column	Type	Description
parent id	unsigned int	id of parent alert
child id	unsigned int	id of child alert

Figure 9: Hierarchy table structure

- Pro-active defense against known malware.

The Madynes team was in charge of the first project. The objectives are large scale malicious code capture, the collection of network traces, in vitro analysis of the malicious code and the building up of an experimentation platform for other projects.

The idea with this telescope is to have information and to collect data as close as possible to what home users can encounter in the Internet. To do so, we chose to use classical public DSL offers. For the telescope, we are addressing two kinds of scenarios:

- Home users with two Home DSL Lines with 1 public IP per ISP;
- A small enterprise with two Professional DSL lines, one with only one public IP, the other one with a dedicated /24 IP prefix.

The collect environment is in charge of capturing malicious code and collecting network traces. It is composed of 7 servers using Xen virtualization in order to run as many probes as possible. This environment will emulate vulnerabilities and capture the malwares trying to exploit them. Emulation ensures that the malicious code can not gain control of the system and spread any further.This is done thanks to low interaction honeypots (Nepenthes and Argos). At the moment, 86 Virtual Machines are deployed, 80 of them running Nepenthes and 6 running Argos. The captured malicious code is sent for “in vitro” analysis to sandboxes (Norman sandbox and CWSandbox). On all probes, PCAP traces of the traffic are collected with tcpdump. These dumps are stored on the storage environment and give information about spreading mechanisms of the malwares. Attacks, captured malwares and traces are stored on a storage server in the storage environment. This environment is isolated from the other ones thanks to a dedicated Cisco ASA firewall to ensure its integrity. All attacks are logged in a postgresql database and are displayed on a WEB interface. These components are part of the logserver from the SurfNet IDS project. The logserver also retrieves the sandboxes reports, and identifies the malwares captures by running anti-virus checking on them (ClamAV, BitDefender and Kaspersky).

The platform is also composed of an analysis environment that permits to run experiments linked to the analysis of the malicious code captured, or experiments related to other project that can benefit from the infrastructure.

The telescope already permits to collect complementary information about malicious programs, from the binaries to the network traces. It permits to understand their behavior, from the system point of view thanks to the “in vitro” analysis in sandboxes, and from the network point of view via the PCAP traces. It also enables the detection of 0-day attacks. However, collecting network traces as flows and extracting attack patterns based on the properties of these flows is an interesting feature to add. In the scope of the EMANICS trace collection project, we decided to deploy a NETFLOW infrastructure on the telescope.

On each server of the collect environment, we deployed a NETFLOW probe (fprobe). This probe captures all the incoming and outgoing traffic of the honeypots, and exports it to the storage environment in the NETFLOW format. On the storage server, several instances of a NETFLOW collector (nfdump) capture these flows and store them locally. All the flows are marked with timestamps which eases their analysis.

In order to display the information in a human readable way, we deployed on the storage server a visualization interface for NETFLOW called Nfsen. It permits to view all the flows in graphs, to put filters or zoom on a given period of a flow. Figure 10 shows an example of a graph generated by Nfsen.

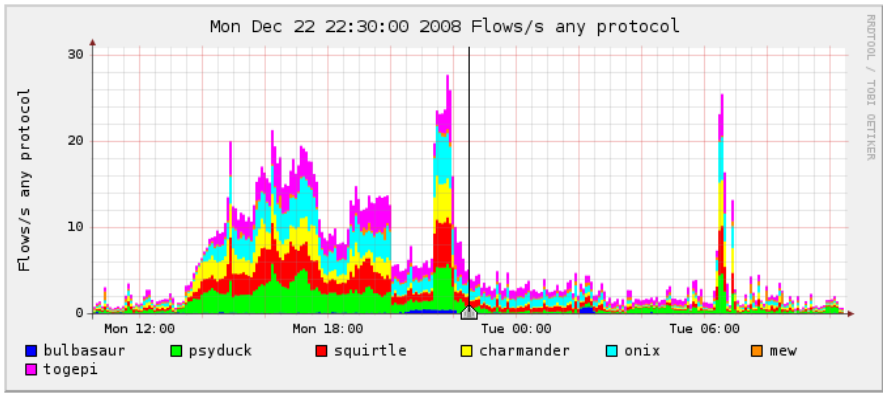

Figure 10: NETFLOW Traces on the High Security Lab at INRIA

These flows collected will be analyzed and used to investigate protection and defense mechanisms against malware based on network flows. We will extract attack patterns and stop the attacks at the network level before they harm the systems. Since the 9th of September, which is the date considered as the end of the deployment phase, and to the 21st of December 2008, the telescope underwent 3,421,353 attacks. These attacks are composed of 3,243,162 possible attacks, which are in fact abnormal traffic (port scans...) that did not lead to vulnerabilities exploits, 178,191 malicious attacks that offered 177,820 malwares, and 133,078 malwares were successfully downloaded over this period resulting in 26,169 unique binaries. On a daily basis, the telescope sees 32,571 attacks, composed of 30,875 possible attacks, 1,696 malicious attacks resulting in 1,267 malwares downloaded.

5 Collaboration

The EmanicsLab 2.0 project, a continuation of the EmanicsLab project, has extended the integration from 8 partners to 11 partners. The establishment and operation of EmanicsLab requires collaboration and coordination. This concerns not only technical aspects but also policy issues. To this end, a first charter for EmanicsLab has been established in the reporting period. The collaboration on technical matters is organized and driven by UniZH since they provide the central infrastructure. Short-term operational changes are usually announced on the WP2 mailing list so that people can take appropriate actions. Overall, EmanicsLab has been a great success and now integrates 11 out of the 13 technical project partners. Furthermore, EmanicsLab is actively used by other work packages, several of them again being collaborative efforts themselves. The efforts of making the testbed accessible to non-EMANICS partners will allow the collaboration between EMANICS partners and non-EMANICS partners in the future.

The trace data collection and labeling project has a smaller degree of collaboration than the EmanicsLab 2.0 project due to the nature of the trace data collection activity. Every partner involved usually collects trace data at its own operational networks or at networks the partner has access to. However, through this activity, partners share their data sets and the trust relationships that have evolved between partners make it easier to deal with the legal aspects of accessing traces collected by a different partner. One partner (INRIA) has joined the project during this reporting period, resulting in seven partners engaged in trace collection and labeling activities. During December 2008, a coordinated trace collection activity was started, which presents collaborative work among the partners involved.

The work package has also acquired better collaboration through several face-to-face meetings and exchanges:

- Partners engaged in the trace data collection and labeling project organized and attended the NETFLOW/IPFIX workshop held in October 2008 in Munich.
- The International Summer School on Network and Service Management (ISSNSM 2008) hosted by the University of Zurich in June 2008 included an tutorial on the EmanicsLab testbed and its features targeted to PhD students. Since the summer school was also well attended by non-EMANICS PhD students, the tutorial might stimulate collaborative research with non-EMANICS partners.

Since these two funded projects are well established, most partners mainly communicate together through one-to-one meetings or through email exchanges. More general discussions usually take place next to general assemblies or EMANICS sponsored events such as the AIMS conference, the ISSNSM summer school, or specific workshops.

6 Conclusion

The fifth "Virtual Laboratory Integration Report" documents the achievements of the two projects sponsored by WP2 during the last nine-month period (March 2008 - December 2008). The achievements of the two funded projects can be summarized as follows:

- The EmanicsLab has been upgraded and extended to include eleven partners. The number of research and education activities running on the testbed has increased during this period. Several research activities are experiments belonging to collaborative projects undertaken in other work packages. Work has also started in this reporting interval to formalize a charter for EmanicsLab and to pave the way to keep the EmanicsLab infrastructure running past the end of the EMANICS project.
- Trace data collection and labeling activities have been continued by the existing partners and a new partner INRIA. All partners collect not only trace data at their operational networks (or other networks they have access to). In addition, an effort was started to collect trace data at multiple locations at the same time (coordinated trace data collection). Such coordinated trace collection activities are useful for understanding the distribution of security attacks on the Internet.

For the year 2009, the third phase of the EMANICS project, WP2 will continue with the successful open call process. In fact, an open call for projects to fund in 2009 has already been issued in December 2008 and discussions are underway between EMANICS partners to develop strong joint proposals. It is essential for the continuing or new projects in 2009 to improve the usage of the EmanicsLab testbed and the trace data for collaborative research and education purposes. In addition, it is necessary to establish an organizational framework allowing the continued use of the infrastructure and data sets past the end of the EMANICS project in December 2009.

7 Abbreviations

DNS	Domain Name System
DoS	Denial of Service
DSL	Digital Subscriber Line
IAX	Inter-Asterisk eXchange
ISP	Internet Service Provider
KAD	Kademlia P2P overlay protocol
NoE	Network of Excellence
P2P	Peer-to-Peer
PCAP	Packet Capture
PI	Principal Investigator
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
SQL	Structured Query Language
SSH	Secure Shell
URL	Uniform Resource Locator
VoIP	Voice over Internet Protocol

8 Acknowledgement

This deliverable was made possible due to the large and open help of the WP2 Partners of the EMANICS NoE. Many thanks to all of them.

References

[1] EmanicsLab Homepage. Available at https://emanicslab.csg.uzh.ch/; Last accessed October 2008.

[2] MyPLC: A complete PlanetLab Central (PLC) portable installation. Available at http://www.planet-lab.org/doc/myplc; Last accessed June 2008.

[3] Newest MyPLC installation version. Available at http://svn.planet-lab.org/wiki/MyPLC; Last accessed June 2008.

[4] PlanetLab: An open platform for developing, deploying, and accessing planetary-scale services. Available at http://www.planet-lab.org/; Last accessed June 2008.

[5] The 2nd International Summer School on Network and Service Management. University of Zurich, Switzerland. Available at http://www.aims-conference.org/issnsm-2008/; Last accessed December 2008.

[6] The Advanced Networking Lab Course. Jacobs University Bremen, Germany. Available at http://www.faculty.jacobs-university.de/jschoenwae/anl-2008/; Last accessed December 2008.

[7] Ganglia: EmanicsLab Node Usage Monitor. Available at http://emanicslab.csg.uzh.ch/ganglia/; Last accessed June 2008.

[8] PlanetLab Acceptable Use Policy. Available at http://www.planet-lab.org/aup; Last accessed June 2008.

[9] J. Schönwälder. Simple Network Management Protocol (SNMP): Traffic Measurements and Trace Exchange Formats. RFC 5345, Jacobs University Bremen, October 2008.

[10] T. Ylonen and C. Lonvick. The Secure Shell (SSH) Protocol Architecture. RFC 4251, SSH Communications Security Corp, Cisco Systems, December 2006.