

A large, stylized map of Europe is the central background element. It is composed of a grid of small squares in various shades of yellow and green, creating a pixelated or mosaic effect. The map is centered on the continent of Europe, with the British Isles to the west and the Mediterranean coast to the south. The overall aesthetic is modern and digital.

Access Control and Monitoring for Campus Computer Labs

Best Practice Document

Produced by the MARNET-led working group on network
security and monitoring

Author: V. Ajanovski (FCSE, UKIM)

April 2015

© GEANT Association, 2015. All rights reserved.

Document No: GN3plus-NA3-T2-CBPD MA1
Version / date: Version 1.0 / April 2015
Original language : English
Original title: "Access control and monitoring for campus computer labs"
Original version / date: Version 0.1 / November 2013
Contact: vangel.ajanovski@finki.ukim.mk

MARnet bears responsibility for the content of the document. The work has been carried out by a MARnet led working group on Network Security and Monitoring at FCSE (Faculty of Computer Science and Engineering – Skopje).

Parts of the report may be freely copied, unaltered, provided that the original source is acknowledged and copyright preserved.

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 605243, relating to the project 'Multi-Gigabit European Research and Education Network and Associated Services (GN3plus)'.

Table of Contents

Executive Summary	1
1 Introduction	2
2 Use-cases and priorities	3
3 Use-case descriptions	5
3.1 UC1: Admin – Configures the monitoring and access control services	5
3.2 UC2: Admin – Defines new computer labs and new destinations	5
3.3 UC3: Teacher – Blocks all network access except to a chosen destination	5
3.4 UC4: Teacher – Activates recording of screen snapshots and running user processes	6
3.5 UC5: Teacher – Monitors all the students' screens at once	6
3.6 UC6: Teacher – Blocks network access to a list of destinations	7
3.7 UC7: Teacher – Monitors all students' processes at once	7
3.8 UC8: Teacher – Saves a snapshot of the activities of a student	8
3.9 UC9: Teacher – Saves a comparative snapshot across two students' activities	8
3.10 UC10: Teacher – Prepares a set of computers for an exam	8
3.11 UC11: Teacher – Pulls all files from predefined locations from a list of computers	9
3.12 UC12: Teacher – Pushes a file to a predefined location on a list of computers	9
3.13 UC13: Teacher – Saves a video recording of student screen	10
4 Network architecture	11
4.1 Initial architecture	11
4.2 Proposed network architecture	12
5 Network access control	14
5.1 Computer labs firewall	14
5.2 FINKI-Firewall control application	14

5.3	Network access profiles	15
6	Auditing of access to various services	16
7	Monitoring lab activities	17
7.1	Recording services	17
7.1.1	Screenshot service	17
7.1.2	Info sending service	17
7.2	Monitoring application	18
7.2.1	Customising the application	19
8	Alternative solution for monitoring activities	21
	References	22
	Glossary	23

Table of Figures

Figure 2.1: Use-case diagram denoting all possibilities	3
Figure 4.1: Diagram of the usual network architecture employed per campuses	11
Figure 4.2: Diagram of the proposed network architecture to enable access control per computer lab	12
Figure 5.1: Screenshot of the FINKI-Firewall control application.	15
Figure 7.1: Screenshot of the Computer Lab Monitoring Application (*note that white places in the grid in the figure are computers that are not monitored.)	19

Executive Summary

Computer labs at the universities are used in three general situations: practical demonstrations, individual work by students on projects and conducting exams. Depending on the special use-cases for each situation, different access permissions are required, different network setup is required, access to online resources should be permitted/denied, and in most situations such adjustments should be performed by the teacher, without any network administration knowledge and equipment access.

This document should be considered as a reference and guide to possible simple solutions that can be used for such scenarios, based on many years of trials at computing departments within the Ss. Cyril and Methodius University, Skopje, Macedonia. The work is based on ideas from the current implementations at the Computer Labs of the Faculty of Computer Science and Engineering.

Ideally, the whole solution is organised as a fully automated integrated information and control system – building on top of several practices, tools and applications for network level access control and monitoring. In this document the design and organisational process development of such system is presented together with the tools that enable and ease the implementation of such process. Business-level use-cases are first presented, to understand the general level functional requirements that teachers and administrators place on the overall solution and further several non-functional requirements are discussed. Where the solution is not possible to be automated, a manual process is proposed.

1 Introduction

Problem statement regarding access control

The problem of ...	not being able to secure network access to specific destinations during classes and especially exams or to restrict network access fully.
affects ...	teachers in larger computer labs or in situations where the exams are being held online.
the impact of which is ...	playing games during classes and non-ethical conduct – exchange of solutions during exams, plagiarism, etc. also network cables suffer because teachers that are not able to control network access via software, plug-out the cables from the computers.
a successful solution would be ...	a system that enables the teacher to control and restrict the network access to specific destinations.

Problem statement regarding observation

The problem of ...	observing student activities during classes and exams.
affects ...	teachers in larger computer labs.
the impact of which is ...	increased amount of non-ethical conduct without being able to gather evidence to file a report.
a successful solution would be ...	a system that enables the teacher to monitor students' activities and react on non-ethical conduct in a timely manner.

2 Use-cases and priorities

Ideally, the solution would be completely automated and work as a part of an integrated information and control system, so this document is presented from the viewpoint of the design and development of such a system. Business-level use-cases are first presented, to understand the general level functional requirements and further several non-functional requirements are added. Where the solution cannot be automated, a manual process will be implemented for the same use-case scenario.

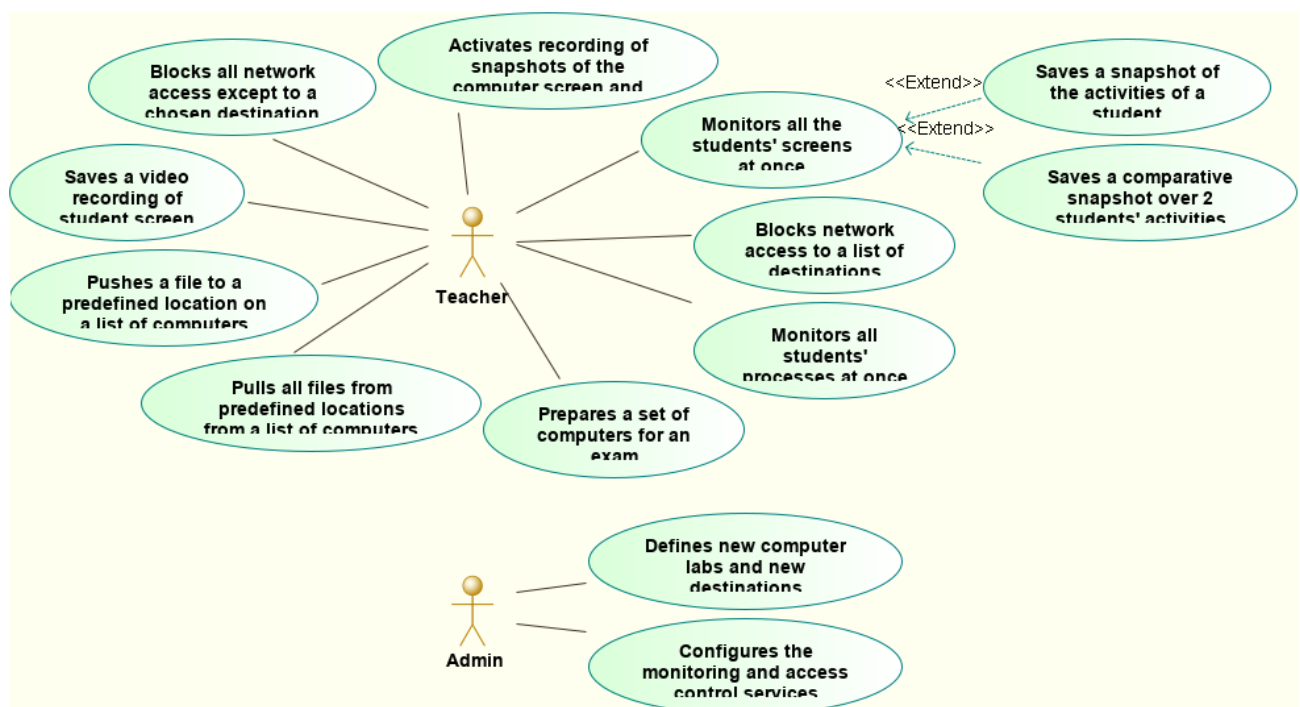


Figure 2.1: Use-case diagram denoting all possibilities

Must have:

- UC1: Admin – Configures the monitoring and access control services.
- UC2: Admin – Defines new computer labs and new destinations.
- UC3: Teacher – Blocks all network access except to a chosen destination.
- UC4: Teacher – Activates recording of snapshots of the computer screen and running user processes.
- UC5: Teacher – Monitors all the students' screens at once.

Should have:

- UC6: Teacher – Blocks network access to a list of destinations.
- UC7: Teacher – Monitors all students' processes at once.
- UC8: Teacher – Saves a snapshot of the activities of a student

Could have:

- UC9: Teacher – Saves a comparative snapshot across two students' activities.

Won't have (other use-cases):

- UC10: Teacher – Prepares a set of computers for an exam.
- UC11: Teacher – Pulls all files from predefined locations from a list of computers.
- UC12: Teacher – Pushes a file to a predefined location on a list of computers.
- UC13: Teacher – Saves a video recording of student screen.

3 Use-case descriptions

The functional requirements, written in the form of use-cases, are given in the following paragraphs.

3.1 UC1: Admin – Configures the monitoring and access control services

1. The admin access the monitoring service.
2. The admin configures the default timeouts for resetting the firewall rules in each computer lab.
3. The admin configures the scheduler for automatic invocation of the resetting process

3.2 UC2: Admin – Defines new computer labs and new destinations

At present, this use-case depends heavily on network configuration and connections, so the profiles necessary for new labs and new destinations cannot be created automatically. The administrator manually creates groups of firewall rules that can be applied in order to allow or block access, as needed by teachers. These rules are stored as pieces of source code and are included when choosing the destinations and type of access by the teacher.

3.3 UC3: Teacher – Blocks all network access except to a chosen destination

The teacher starts an exam or an activity in a lab, and depending on the exam requirements, allows access to only certain predefined network destinations (e.g. the course management server, e-testing server, code programming server, etc). All communication to other destinations originating from that computer lab are usually blocked.

1. The teacher access the labs access control system.
2. The teacher chooses the lab in question.
3. The teacher marks all the needed destinations.

4. The teacher sets a timeout period in minutes, after which the block is removed.
5. Firewall rules to allow access to each of the needed destinations are set up, and a follow-up rule to block all the other traffic originating from the chosen lab.
6. All teacher actions and choices are recorded in a log.

3.4 UC4: Teacher – Activates recording of screen snapshots and running user processes

The recorder is a hidden service running on each computer in a computer lab that records current activities and stores them on the file server, and it is setup to run during certain time periods. Whether it records activities and stores them locally, or stores activities on the file server, depends on the contents of a configuration file stored in a local path that is only accessible to administrators. When the teacher prepares a computer lab for exams/recording, this file is configured at each computer.

1. When scheduled, the recorder checks the contents of the local configuration file.
2. If instructed to not record, jump to the final step in this sequence.
3. If instructed to record, a screen snapshot is saved locally with a file listing the running processes.
4. If instructed to store locally, connection to the file server is checked and if still open it is closed. After that, jump to the final step.
5. If instructed to store on the file server, connection to the file server is checked and opened if necessary and the recorded files are copied on the server in a special location/filename for the specific lab and computer (for example: /snapshots/labname/computername-screen.png, /snapshots/labname/computername-processes.txt)
6. Reschedule the next execution based on the parameter in the configuration file.

3.5 UC5: Teacher – Monitors all the students' screens at once

The idea is to enable the teacher to view all the students' screens at once, in a layout that corresponds to the physical layout of the computers in the room. This helps in observing how the students perform their assignments in general and how a pair of students sitting at neighbouring workstations collaborate.

1. The teacher accesses the lab's monitoring system.

2. The teacher chooses the lab to monitor.
3. The system displays a map of the layout of the room, with a thumbnail of the last saved screenshot from each computer in the lab in its place in the layout.
4. (Optionally) The teacher can click on a certain screen and start UC8: to view details of a selected student's computer.
5. (Optionally) The teacher can mark two computers and invoke UC9: to view a comparison of activities.
6. The system refreshes the display (go to step 3) if no activity from the teacher is detected in a certain configurable period.

3.6 UC6: Teacher – Blocks network access to a list of destinations

The teacher starts an exam or an activity in a lab, and depending on assignment requirements may wish to stop access to certain predefined network destinations (file management server, ...). Communication to other destinations is allowed.

1. The teacher accesses the lab's access control system.
2. The teacher chooses the lab in question.
3. The teacher marks all the destinations to be blocked.
4. The teacher sets a timeout period in minutes, after which the block is removed.
5. Firewall rules to block access to each of the selected destinations are set up, and a follow up rule to allow all the other traffic originating from the chosen lab.
6. All teacher actions and choices are recorded in a log.

3.7 UC7: Teacher – Monitors all students' processes at once

1. The teacher accesses the lab's monitoring system.
2. The teacher chooses the lab to monitor and clicks on tab processes.
3. The system displays a map of the layout of the room, with a small frame for each computer in the lab in its place in the layout.
4. Inside each computer's frame, the system presents the last-saved list of the executed processes.

5. The system refreshes the display (go to step 3) if no activity from the teacher is detected in a certain configurable period.

3.8 UC8: Teacher – Saves a snapshot of the activities of a student

This is an extension of UC5.

1. When in UC5, the teacher decides to view the current activities of a certain student in more detail; the thumbnail screenshot of the student's computer is clicked on.
2. The system presents a full-size image of the last-saved screenshot of the student's computer.
3. The system presents a list of the last-saved user-activated processes under the image.
4. The system shows the exact time of the last-saved information.
5. The teacher can choose to save the presented information as an off-line document, to be used as a record of how the student performed the assignment.

3.9 UC9: Teacher – Saves a comparative snapshot across two students' activities

This is an extension of UC5.

1. When in UC5 the teacher decides to view activities in more detail and compare the current activities of a pair students, the checkmarks besides the screens are selected before selecting **Compare**.
2. The system splits the screen in two partitions.
3. The system invokes UC8 for each of the two selected computers, each in a separate part of the screen.

3.10 UC10: Teacher – Prepares a set of computers for an exam

1. The teacher accesses the monitoring system.
2. The teacher chooses the computer lab in question.

3. The teacher chooses the predefined location for the students work (for example, directory `"/exam"`).
4. The teacher configures the frequency for taking screen and process snapshots by the monitoring system.
5. The system traverses all computers in the lab and clears the predefined location.
6. The system configures the scheduler for taking screen and process snapshots at the configured frequency.

3.11 UC11: Teacher – Pulls all files from predefined locations from a list of computers

1. The teacher accesses the monitoring system.
2. The teacher chooses the computer lab in question.
3. The system traverses all computers in the selected lab, and packs all of the contents in a predefined configurable location at each computer (for example, the directory `"/exam"`).
4. The system puts all such packages in a specific location at the lab file server (for example, into the directory `"/teacherusername/yyyy-mm-dd/labshortname"`).
5. The system presents download links to the teacher.

3.12 UC12: Teacher – Pushes a file to a predefined location on a list of computers

1. The teacher uploads a file to the file server, and records the link.
2. The teacher accesses the monitoring system.
3. The teacher chooses the computer lab in question.
4. The system traverses all computers in the selected lab, and copies the file from the link to a predefined configurable location at each computer (for example, the directory `"/exam"`).

3.13 UC13: Teacher – Saves a video recording of student screen

This functionality can be useful on some occasions, but requires additional processing and recording power, so it is not discussed in the first version of the document and proposed solutions.

An alternative is to save snapshots every few seconds, so that a stop-motion video can be constructed.

4 Network architecture

4.1 Initial architecture

Usually campuses employ a network architecture similar to the one shown in Figure 4.1. On the left is the network block with all computer labs, on the right is the publicly accessible servers block. All computer labs usually have a separate L2 switch that might be located in the room or in a network concentration point/rack/closet/room. On occasions L3 switches are used. In order to save assigned public IP address space and protect from direct outside attacks, the labs are usually behind a small router/firewall that uses NAT/PAT, which then joins them into the campus network. Sometimes, campuses isolate the labs in separate VLANs, if they have support for this in the upstream network and the main lab's router/firewall.

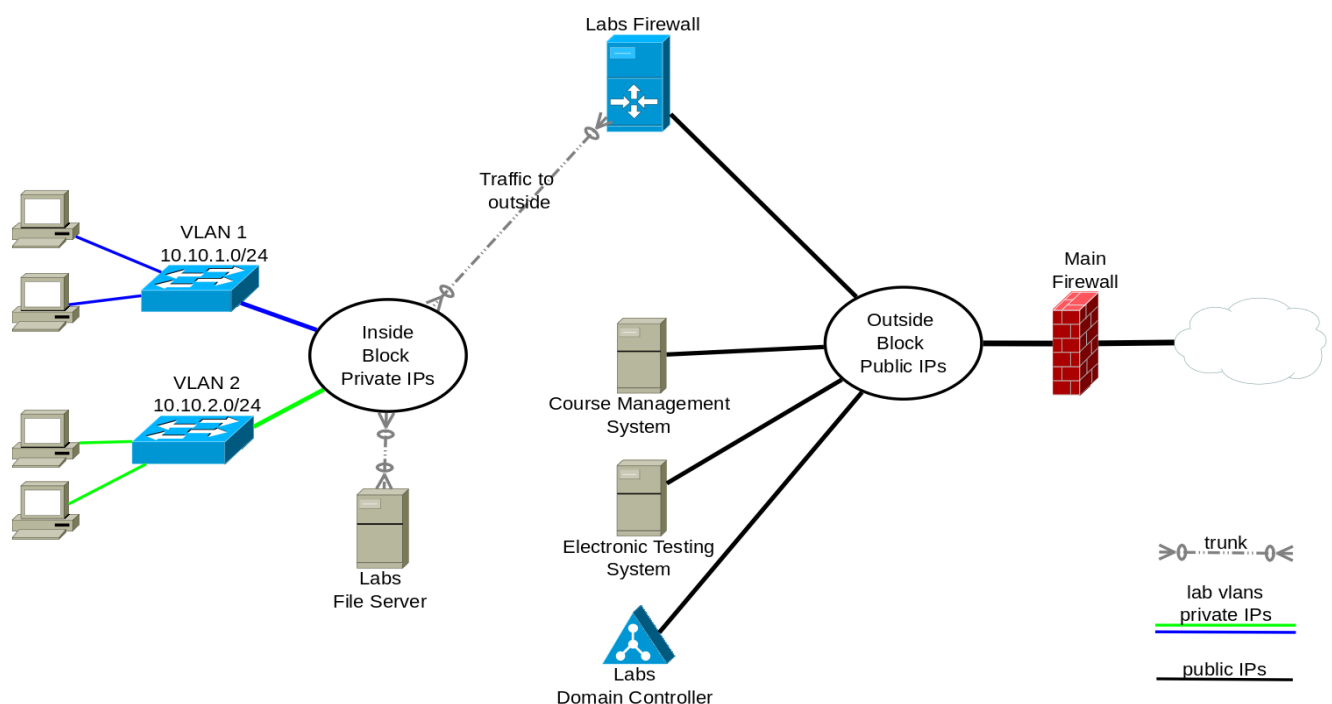


Figure 4.1: Diagram of the usual network architecture employed per campuses

There are several issues with the usual networking architecture, which prevent some of the use-cases we have as requirements:

- If VLANs are not used per lab, computers from different labs can communicate with each other, and in this way there is no isolation of one lab from another which is important when having computer exams in one lab and having the other lab open for all students to use. In such a situation a student in the open lab can communicate and help another student taking an exam in the other lab.
- When having some specialised classes – for example a course on Computer network design or a System services course – the students in one lab can activate a DHCP server or create loops in the network or open other conflicting services with the rest of the infrastructure.
- The Course management system / e-testing system does not know the exact IP addresses of the users accessing it from computer labs, which might be crucial when there are exams and deadlines.
- All access control is a fixed configuration usually implemented as static access lists on the routers. Teachers cannot activate or deactivate certain networking services or control access to network resources on the outside.

4.2 Proposed network architecture

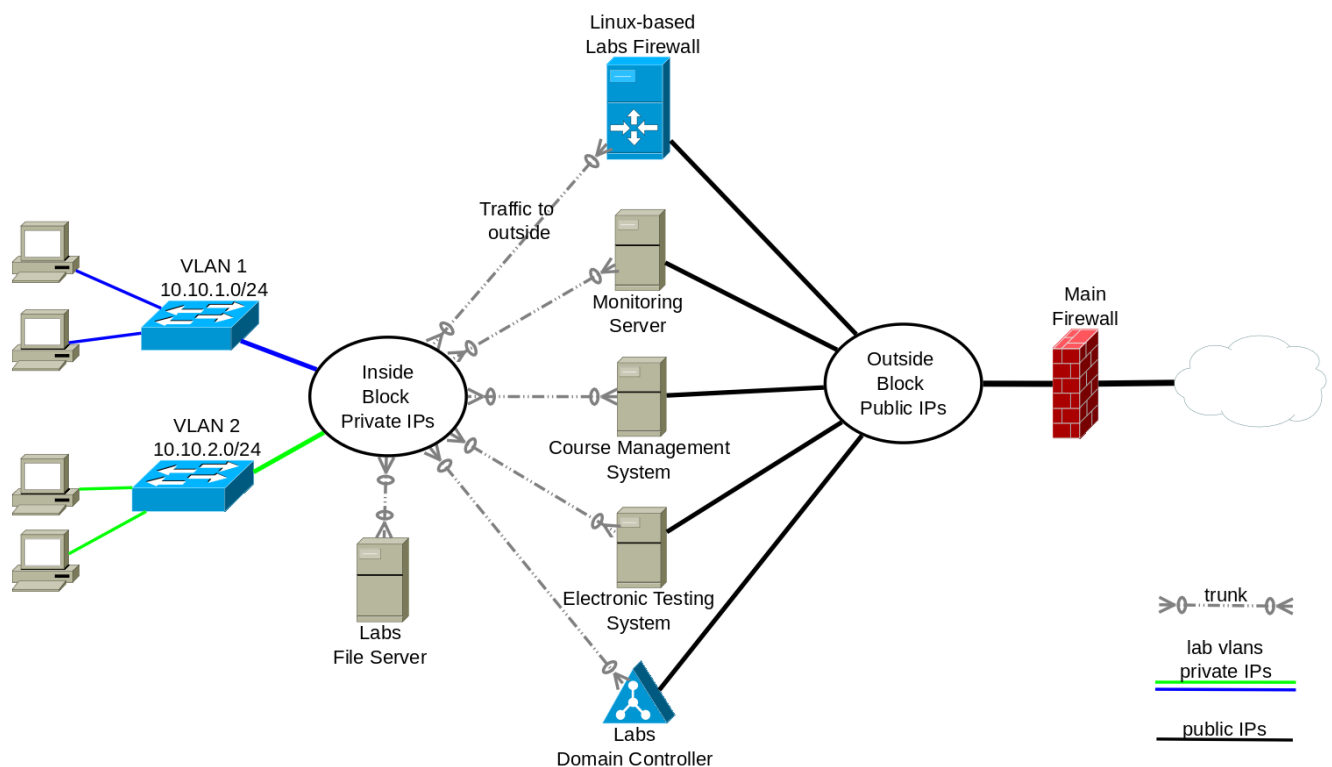


Figure 4.2: Diagram of the proposed network architecture to enable access control per computer lab

This architecture is based on the following premises:

- The network is again split in two blocks
 - Inside a privately-addressed computer labs block, with a separate VLAN and IP class for each computer lab (coloured lines).
 - Outside publicly-addressed servers block (black lines).
- The Labs firewall that is based on Linux has the following responsibilities:
 - it is a static router among different parts of the computer labs block.
 - NAT/PAT hiding the inside network from the public internet.
 - Hosts a custom software solution for switching on/off access to various network destinations chosen by teachers.
 - DNS server for resolving the server names present in the inside block, so that all computers in the labs will get only an internal IP for them.
- Computer labs have a presence only in the inside block; each lab is in a separate VLAN.
- Some servers can have presence in both blocks, the reason is:
 - Computers in the labs should be able to access such servers even if Internet access is disabled.
 - The servers should know the address of each lab computer that is accessing it.
- Access from the inside block to the outside block is only via the Linux-based labs firewall.

5 Network access control

5.1 Computer labs firewall

A Linux-based server is used for the purpose of a firewall, running the ClearOS distribution. It is setup on a virtual server, with many virtual interfaces connected to various parts of the network and the respective VLANs (as seen in Figure 4.2). This enables easier manipulation of the rules of the NAT and the router in order to achieve the described use-case scenarios, by modification of the ipchains tables.

Although such configurations can be prepared and loaded manually by the administrator, a special application called Finki-Firewall (<https://github.com/dragansah/finki-firewall>) can be used, to present an easy-to-use web interface accessible to all the teachers. With this application, teachers can introduce/remove some specific rules in the routing tables without any system administration skills or Linux knowledge.

5.2 FINKI-Firewall control application

FINKI-Firewall is a Tapestry5-based Java web application created by Dragan Sahpaski for use at the Faculty of Computer Science and Engineering in Skopje. This applicaiton is intended to be used by teachers, enabling them to control and block network traffic in the computer labs. The application was originally created with a focus mostly on UC1 and UC3.

Source: <https://github.com/dragansah/finki-firewall>

The application is under further development to integrate with the recording and monitoring services.

The idea is simple: ready-made snippets of ipchains rules are defined in configuration files by the system administrator and depending on the choice made by teachers these rules are injected into the ipchains list and as such, modify the firewall behaviour to block or pass traffic from some sources to some destinations. These rules are called profiles and they are presented as simple choices for the teacher in the control application.


Computer labs				Network Access Profiles			
No.	Code	Lab Name	Active profile	No.	Profile	Description	
1	Lab-200			1	Moodle	Access to local Moodle server	Lab 215
2	Lab-118			2	Oracle	Access to local Oracle server	
3	Lab-3			3	Home Network	Access restricted to local network services	
4	Lab-26			4	Internet	Unrestricted Internet Access	
5	Lab-215		Moodle 				

Figure 5.1: Screenshot of the FINKI-Firewall control application.

5.3 Network access profiles

The application for network access control uses predefined profiles, that are in fact iptables script snippets and are defined in a JSON file. The following example shows a simple profile:

```

{"name" : "Profile",
 "description" : "Access to SITE only",
 "iptables" : [
   "iptables -I FORWARD -s ${ipClass} -j DROP",
   "iptables -I FORWARD -s ${ipClass} -d SITE.ADDRESS -j ACCEPT",
   "iptables -I FORWARD -s ${ipClass} -d DOMAIN.CONTR.INTERN.ADDR -j ACCEPT",
   "iptables -I FORWARD -s ${ipClass} -d MONITOR.SERVER.INTERN.ADDR -j ACCEPT",
   "iptables -I FORWARD -s ${ipClass} -d FW.INTERNAL ADDR -j ACCEPT" ]},
    
```

`${ipClass}` is a variable in the script that the application changes to the class of IPs of the selected computer lab. The application is also customisable to support many computer labs, that are also configured inside a json file.

Many such rule-sets can be prepared by the network administrator and system administrator of the labs firewall, and can be configured inside the JSON configuration file. One of the profiles is also the set of rules for unrestricted internet access that will change.

When applying any profile, any previous rules are overwritten by new ones.

6 Auditing of access to various services

The DNS server in the lab's firewall resolved the names of all services that have a presence in the inside block, so that request for resolution of such servers will result in a private IP address from the inside block.

All servers are connected via trunk links, so that they have presence and IP address in all computer lab VLANs in the inside block. In that way it is ensured that:

- Access to any secured service from within the computer labs in the inside block, will be served by the server daemon (service) running on an internal IP address and will be logged by the relevant service as an access from an internal IP address of the exact lab computer, without hiding behind the NAT/PAT public address.
- Access to inside services will not get routed through the firewall, so will be faster.

7 Monitoring lab activities

7.1 Recording services

There are two services that have to be installed on each computer in the classroom:

- Screenshot service – it is started as each user is logged in, with the privileges of that user.
- Info sending service – running with higher privileges as a system service.

The reason for having two services are based on security restrictions. In general it is not allowed to directly access the screen of another operating system user. So, the service responsible for taking a screenshot has to run as the user itself. On the other hand, the information-sending service runs as a privileged user because it has to have configuration data for accessing remote services and has to have higher privileges to be able to extract some system information and send it to the monitoring server.

7.1.1 Screenshot service

Initial code: <http://develop.finki.ukim.mk/projects/fccapps>

It is a simple service, that when run by a desktop user, executes every 10 seconds and takes a screenshot and stores it to a file in a preconfigured location.

The necessity to run as a user complicates configuration, since there has to be a wrapper service that will do the checking and controlling when and how the service runs. Such a setup is maintained with the help of the wrapper service software YAJSW (Yet Another Java Service Wrapper) which is generally used for wrapping all types of services and doing it in a platform-independent way (since it is based on Java). In this special circumstance it helps to detect a user has logged in, and to run the wrapped service in the logged-in user session.

7.1.2 Info sending service

Initial code: <http://develop.finki.ukim.mk/projects/fccapps>

This service can be run as a regular service and is intended to run as a normal user, unless some special system information is gathered that requires elevated privileges

It runs every 10 seconds, records data about some system information (for example: logged-in username, cpu usage, temperature, list of processes) to a local file and uploads this file via SFTP to a

configured location. If the screen shot service is running it can also upload the file with the current screenshot via SFTP.

This service takes the local files with local names, but uploads them to the remote location renamed to the names <computername>.txt and <computername>.jpg where <computername> is the recorded local computer name or host name. This might create a conflict if the same computer name is used in several domains and all use a single monitoring server and a single location, but this is a very rare circumstance and in such an occasion the code should be customised. This practice gives the possibility for easy monitoring of the activities in the classrooms even without a special monitoring software, and plug-ins can easily be built for many of the existing monitoring platforms.

As best practice, we recommend the upload location to be inside the network (so as not to waste routing and firewall resources) and to be setup as a CHROOT-ed SFTP account. Otherwise, the network access profiles should always allow this traffic to pass through.

7.2 Monitoring application

A simple application that was created by members of the FCSE can be used to help with monitoring student activities while doing assignments/exams. The tool is a web-based JSP application, so it requires any Java web application server to operate (FCSE uses Apache Tomcat).

Initial code: <http://develop.finki.ukim.mk/projects/fccapps>

It is up to the web administrator what kind of access protection to employ, and to what extent. In general – access to the public should be prohibited. Only certified teachers should be able to access the application, and only for official purposes for teaching or exams. Otherwise, if misused and if the students are not properly informed, it might pose a major privacy concern.

At the FCSE, this is set up with a consent form for students when logging onto computers, and the access is controlled via JASIG CAS (a single-sign on service, see <http://www.jasig.org/cas>), where only specific teachers are allowed access to this application and view the students' activities. Because of privacy concerns, all the activities of the teacher are also logged for future inspection purposes.

The web application analyses the recorded files on the capturing server and presents the latest screenshots of all the workstations in one computer lab in a simple layout that corresponds to the physical plan in the room. This is the main interface that teachers can use to monitor students' activities. An example is shown in Figure 7.1 below. Computers that are not in use, or have been turned off for an extended period of time are marked with a special icon to indicate their status.

Lab 215

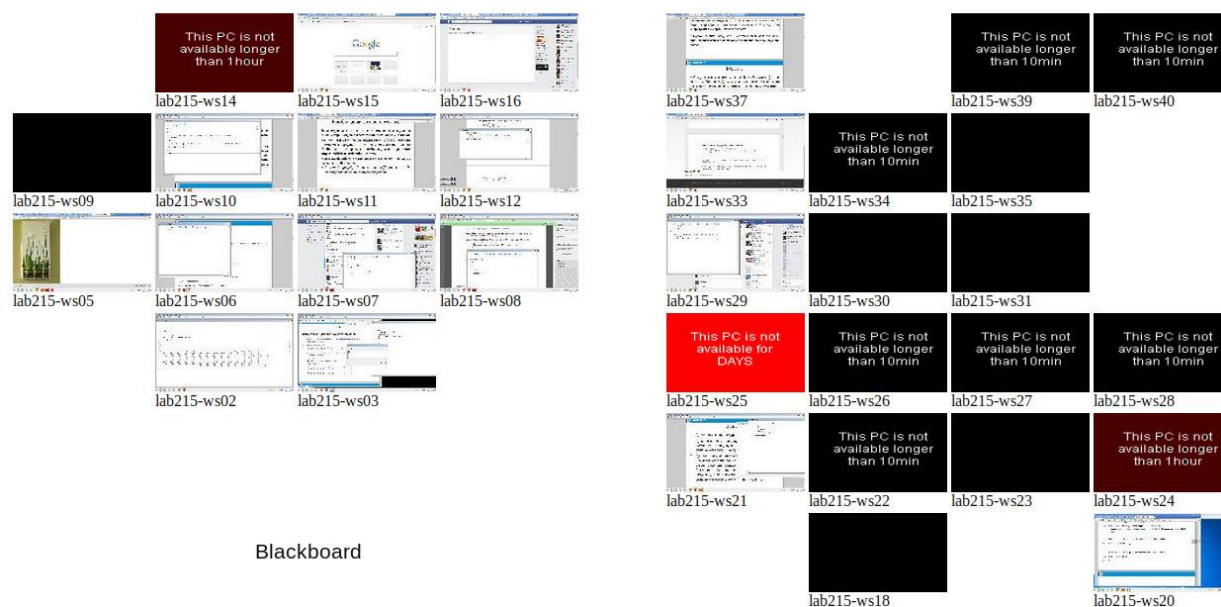


Figure 7.1: Screenshot of the Computer Lab Monitoring Application (*note that white places in the grid in the figure are computers that are not monitored.)

The teacher can click on any screen in order to view details, and can observe the full screenshot, details on logged on user, process list executed by the user, etc. The application only shows the latest records, but a special configuration can be made, for example to have a history of last 10 records, which is beneficial for monitoring the results of assessments that are not just static screens but dynamic in nature. Since records are created every 10 seconds, this can span a time period of a minute and a half.

7.2.1 Customising the application

The sources are given as templates, and being JSP files they can be easily modified even live in production, to tweak the layout to match the floor plan of any computer lab. This requires very basic HTML knowledge; for more advanced customisation, basic Java knowledge is required.

The part of the file within the `<!-- editable -->` block defines the layout of the table of workstations (see image above). It is all HTML code (tables) and uses only the JSP tag `<%=wks (COMPUTERNAME) %>` } to indicate that the application should generate a template for the computer workstation.

For example the following code creates a plan for this layout:

Workstation 1	Workstation 2	Workstation 3	Workstation 4
Workstation 5	Workstation 6	Workstation 7	Workstation 8

```
<table>
  <tr>
    <td><%=wks ("ComputerNameWks1") %></td>
    <td><%=wks ("ComputerNameWks2") %></td>
    <td><%=wks ("ComputerNameWks3") %></td>
    <td><%=wks ("ComputerNameWks4") %></td>
  </tr>
  <tr>
    <td><%=wks ("ComputerNameWks5") %></td>
    <td><%=wks ("ComputerNameWks6") %></td>
    <td><%=wks ("ComputerNameWks7") %></td>
    <td><%=wks ("ComputerNameWks8") %></td>
  </tr>
</table>
```

8 Alternative solution for monitoring activities

iTALC - Intelligent Teaching And Learning with Computers (<http://italc.sourceforge.net/>), is an open-source software that enables most of the monitoring and file-sharing scenarios described previously. On the other hand, it also has many more features that are superfluous to supporting the requested functionalities. There are many similar commercial and open-source solutions available, but this is one of the most advanced and open at the moment of writing.

Being open-source, it enables the implementers with coding experience to program special customisations. So in general there are two possibilities:

- Disregard iTALC and create a minimal solution for monitoring and file-sharing that will enable only the requested functionalities, and nothing more, but that will tightly integrate the monitoring and access control facilities that are required
- Use the advanced features of iTALC and modify its behaviour in order to support all the requested functionalities for monitoring and file-sharing, as proposed. This will better integrate with the requested scenarios and integrate better with the other solution for access control facilities

If the implementer has coding experience, both options are viable – certainly for a startup prototype and to evaluate the usefulness of the whole solution. In order to help new implementers, FCSE provides a sample solution suitable, in the first case with minimal source codes and templates are working in certain typical scenarios (as used at the time of writing at the FCSE). After evaluation of a minimal prototype solution that is both easy to understand and simple to install, the implementer can further decide on which route to take. At FCSE steps were taken using the first of the two possibilities.

References

- [FINKI-Firewall]** <https://github.com/dragansah/finki-firewall>
- [ClearOS]** ClearFoundation ClearOS
<http://www.clearfoundation.com>
- [Tapestry5]** Apache Tapestry 5 project
<http://tapestry.apache.org/>
- [FCCAPPS]** Faculty of Computer Science and Engineering Computing Center
Repository of open-source tools and small apps
<http://develop.finki.ukim.mk/projects/fccapps>
- [CAS]** Apereo Foundation Central Authentication Service (known as JASIG CAS in the past)
<http://www.apereo.org/cas>
- [YAJSW]** Yet Another Java Service Wrapper
<http://yajsw.sourceforge.net/>
- [iTALC]** Intelligent Teaching And Learning with Computers
<http://italc.sourceforge.net>
- [Tomcat]** Apache Tomcat
<https://tomcat.apache.org/>

Glossary

FCSE	Faculty of Computer Science and Engineering, University Ss Cyril and Methodius, Skopje, Macedonia
FINKI	Acronym of the FCSE official name in Macedonian language
UC	Use-case
VLAN	Virtual Local Area Network
DHCP	Dynamic Host Configuration Protocol
IP	Internet Protocol (used as IP address)
NAT	Network Address Translation
PAT	Protocol Address Translation
DNS	Domain Name System
JSON	JavaScript Object Notation
CHROOT	On the Unix operating systems, 'chroot' is an operation that changes the apparent root directory for the current running process and its children
HTML	Hyper Text Markup Language
JSP	Java Server Pages
CAS	Central Authentication Service, a single sign-on open-source software originally by JASIG, today Apereo.
JASIG	Java in Administration Special Interest Group

