

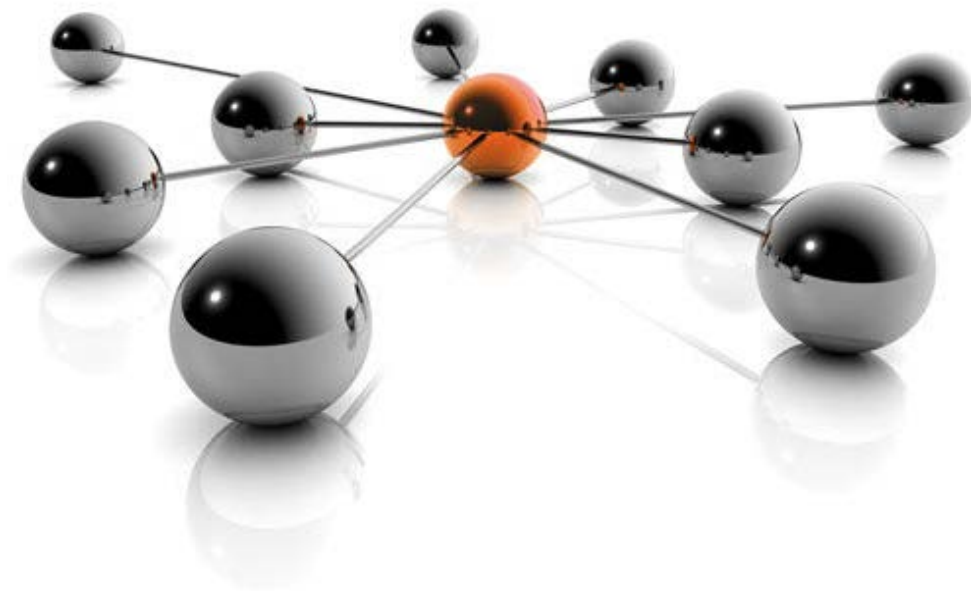
Class: 5AHITT  
Last modification: 10/9/2014  
Document Version: 1.0

## Software engineering

2014/2015

Bergler, Bobek, Janeczek, Mair, Özsoy

## Rock the net



## List of contents

Statement of task.....	1
Trained competencies .....	1
Basic tasks .....	1
Additional information .....	1
Advanced tasks (obligatory for grades better than C) .....	2
Teams .....	2
Grading .....	2
Submission .....	2
Interviews.....	3
Apportionment of work with effort estimation.....	4
Final time apportionment .....	4
Design consideration.....	5
User-Story .....	5
Technology description.....	7
MIB Browser .....	7
WinDump.....	8
Wireshark.....	9
SNMP Framework .....	9
SNMP4j .....	9
JFreeChart .....	12
JavaFx .....	12
Task execution .....	13
GUI Design.....	14
Mockup Design .....	14
JavaFX Design.....	15
SW-Design consideration.....	16
UML-Class diagram.....	16
Test report .....	17
Bibliography.....	18



## Statement of task

### Trained competencies

- Using APIs, Network programming
- Application programming: GUI-programming, parallel programming
- software engineering: buildsystems, testing with mock-objects, design patterns

### Basic tasks

Implement a simple-to-use application to monitor and configure a hardware firewall appliance “Juniper NetScreen 5GT “. The firewall allows read access over the SNMP-protocol (your app should be able to test if SNMPv3 is available and if not fallback on SNMPv2c) and write access over Telnet.

Your app should accomplish following tasks:

- List all configured firewall rules (policies) on the device, add the details of the mentioned services and zones as well.
- Allow refreshing of the list by clicking a button and by a configurable time-intervall. Your GUI should remain responsive even with short refresh-intervals!
- Visualize the thru-put for a highlighted firewall-rule (nice2have: multiple rows) in a line-chart (configurable refresh-interval, unit bytes/sec)
- Encapsulate the data retrieval for further reuse and easy expansion. An UML-model of your design will help you defend it at the review!
- Build a visual appealing and easy to use interface (there is more than Swing out there).

### Additional information

- Since there is only one firewall-appliance available, the time each team can test with the hardware will be strictly limited. Therefore it is essentially to use mock-objects to allow testing the app during times where the hardware is not available.
- An additional benefit of using mock-objects will be, that a CI-Server can use them for automated building and testing.
- You only need to consider firewall-rules for TCP and UDP connections in IPv4.
- You can find Information about the SNMP-Mibs special for the manufacturer of the used appliance here (maybe not all of the Mibs work with the used model):  
<http://www.oidview.com/mibs/3224/md-3224-1.html>
- For exploring the SNMP-Data coming from the appliance you can use tools like this:  
<http://ireasoning.com/mibbrowser.shtml>

## Advanced tasks (obligatory for grades better than C)

Additionally to the basic tasks your app should accomplish the following:

- Alarm the user visually and per email if the config of the firewall-rules changes. To avoid polling use the SNMP-trap mechanism.
- Allow managing of firewall-rules (CRUD). To accomplish this, you will have to send configuration commands via telnet or ssh. An admin-account is available per request.
- Use multicast-groups to build a simple transaction system to serialize administrative tasks on the firewall (for example pass an “admin token” to recognize the collaborator who is allowed to write to the firewall). This should also work in a heterogenous environment (different implementations, different OSes), so you have to coordinate with other teams.
- Make sure, that your interface to the firewall allows an easy change of the firewall-model (new releases, manufacturer, ...). It is not necessary to make this configurable in the GUI but must (explicitly) be considered in your software-design!

## Teams

Build teams with 3 to 5 participants (5 only if two or more members choose advanced level and at least one member chooses basic level). Each individual team-member has to implement, test and document code and is allowed to choose the level of difficulty he/she wants to achieve. For example: if you have a group of four students and two of them want to achieve advanced level, they can focus their implementation work on the advanced tasks. The other two team-members focus on the basic functionality. In any case there must be a working product, advanced tasks cannot stand for themselves.

## Grading

A team can apply for submission with a (mostly) functional product.

Each team-member will be graded separately, based on the documentation (and git-logs) which name him/her as author in all three main competencies as listed.

Advanced tasks will only be considered if the basic tasks are fulfilled for the most part in this team.

## Submission

- Every group must have its own design/solution! Meta-group solutions will end in massive loss of points!
- As for group work usual, a protocol with the UML-Design, the work-sharing, the timetable and test documentation is mandatory!
- Upload your solution as a ZIP file. Please submit only the sources of your solution and a build file (build.xml, pom.xml, Makefile etc.) not the compiled class files and only approved third-party libraries. Your submission must compile and run!
- Before the submission deadline, you can upload your solution as often as you like. Note that any existing submission will be replaced by uploading a new one.

## Interviews

- During the implementation there will be review interviews with the teams. Please be aware that the continuous implementation will be overseen and evaluated!
- After the submission deadline, there will be a mandatory interview.
- The interview will take place in the lesson. During the interview, every group member will be asked about the solution that everyone has uploaded (i.e., changes after the deadline will not be taken into account! There will be only extrapoints for nice and stable solutions!). In the interview you need to explain the code, design and architecture in detail.

## Apportionment of work with effort estimation

Competent person(s)	Task	Description	Estimated time in h
	SW-Design		6
	GUI-Design	Creating a GUI-Design via MockUp-Tool	2
	GUI-Implementation		6
	CRUD-Implementation		15
	Chart-Implementation		1
	SNMP-trap mechanism		5
	Transaction system		30
	Read Firewall-Configurations		4

### Estimated total time exposure

Person	Time exposure in h
Bergler	20
Bobek	25
Janeczek	22
Mair	26
Özsoy	24
<b>Sum:</b>	<b>117</b>

## Final time apportionment

Competent person(s)	Task	Estimated time in h	Actual time in h	Comment
Bergler, Bobek, Janeczek, Mair, Özsoy	Project-Meeting	0,5	0,5	--
	Creating the first version of the documentation and establishing a Google-Doc for the joint creation of the content	2	3	--
Janeczek	Finding an appropriate SNMP-Framework	3	2	--
Mair	Collecting information for SNMP and downloading the required tools for the connection to the router	4	4	--

Bobek	Writing down the user stories	2	2	--
Bobek, Mair, Özsoy	Project-Meeting	0,5	0,75	--

**Actual total time exposure**

Person	Time exposure in h
Bergler	0,5
Bobek	3,25
Janeczek	2,5
Mair	5,25
Özsoy	4,25
<b>Sum:</b>	<b>15,75</b>

**Design consideration****User-Story**

User	
<b>As a user I want to list all configured firewall rules (policies) and some details on the device.</b>	Basic Task
<b>As a user I want to refresh the list by clicking a button and by a configurable time-interval.</b>	Basic Task



<b>As a user I want to visualize the thru-put for a highlighted firewall-rule in a line-chart.</b>	Basic Task
<b>As a user I want to encapsulate the data retrieval for further reuse and easy expansion.</b>	Basic Task
<b>As a user, I want to be warned visually and per email, if the configuration of the firewall-rules changes to avoid polling use the SNMP-trap mechanism.</b>	Advanced Task

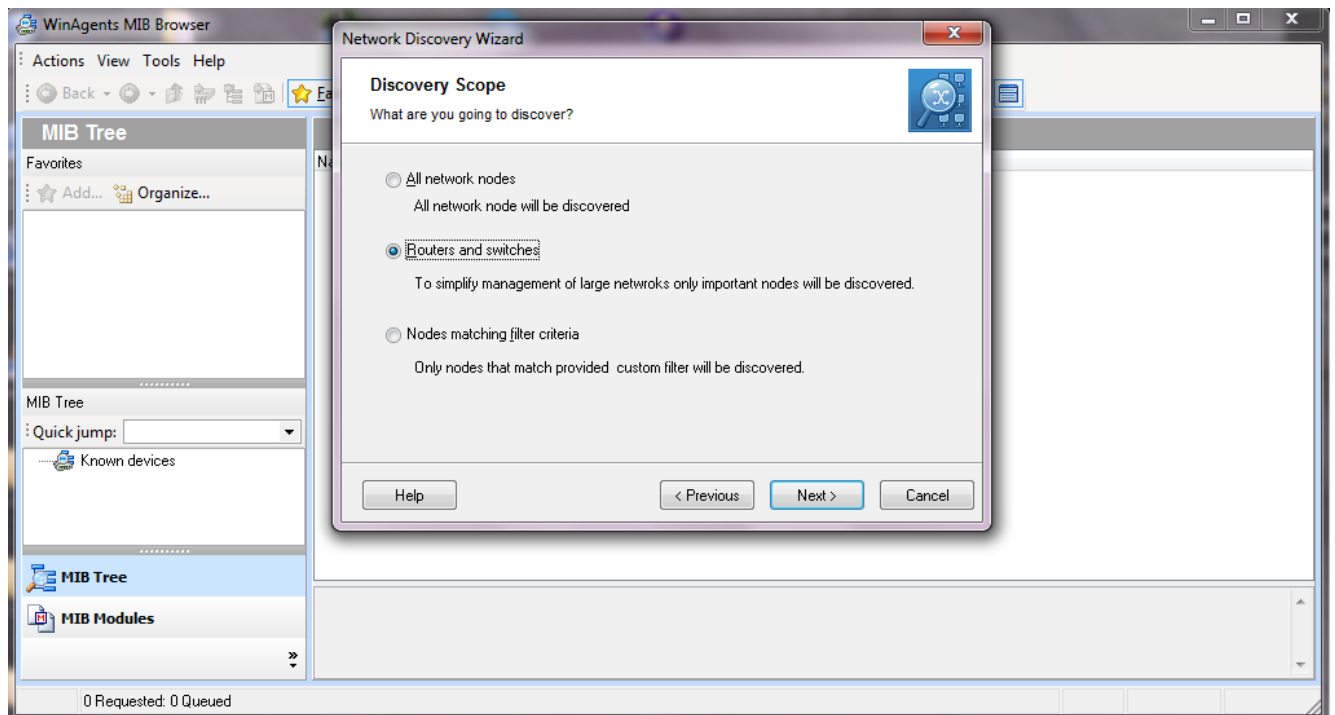
<b>Developer</b>	
<b>As a developer I want to build a visual appealing and easy to use interface.</b>	Basic Task
<b>As a developer I want to use multicast-groups to build a simple transaction system to serialize administrative tasks on the firewall.</b>	Advanced Task

<b>Administrator</b>	
<b>As an administrator I want to be able to manage the firewall-rules (CRUD).</b>	Advanced Task
<b>As an administrator I want to be able to change the firewall-model thru the interface.</b>	Advanced Task

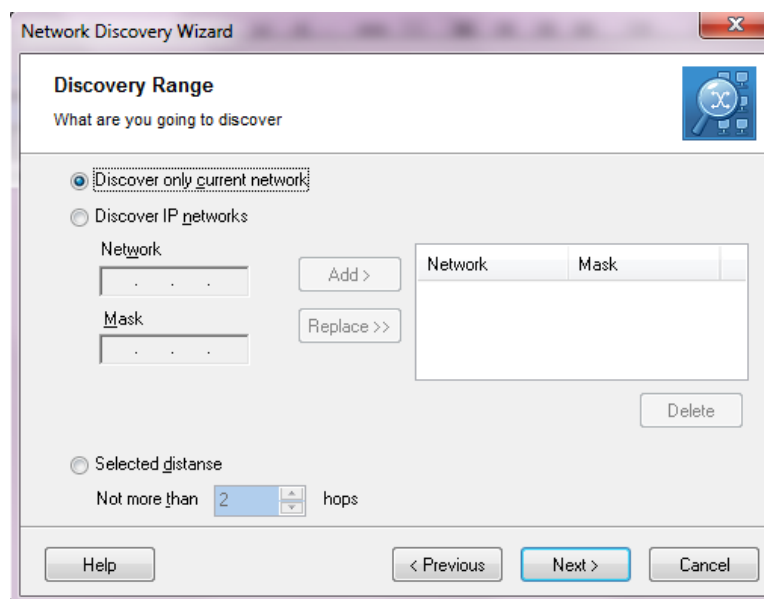
## Technology description

### MIB Browser

**MIB-Browser** provides a user interface that can be used for reading and modifying network packages. In addition it is possible to limit the received packages to router- and switch-packages. The major flaw about this program is that it is not for free. The picture below shows the 30-day-test-version.

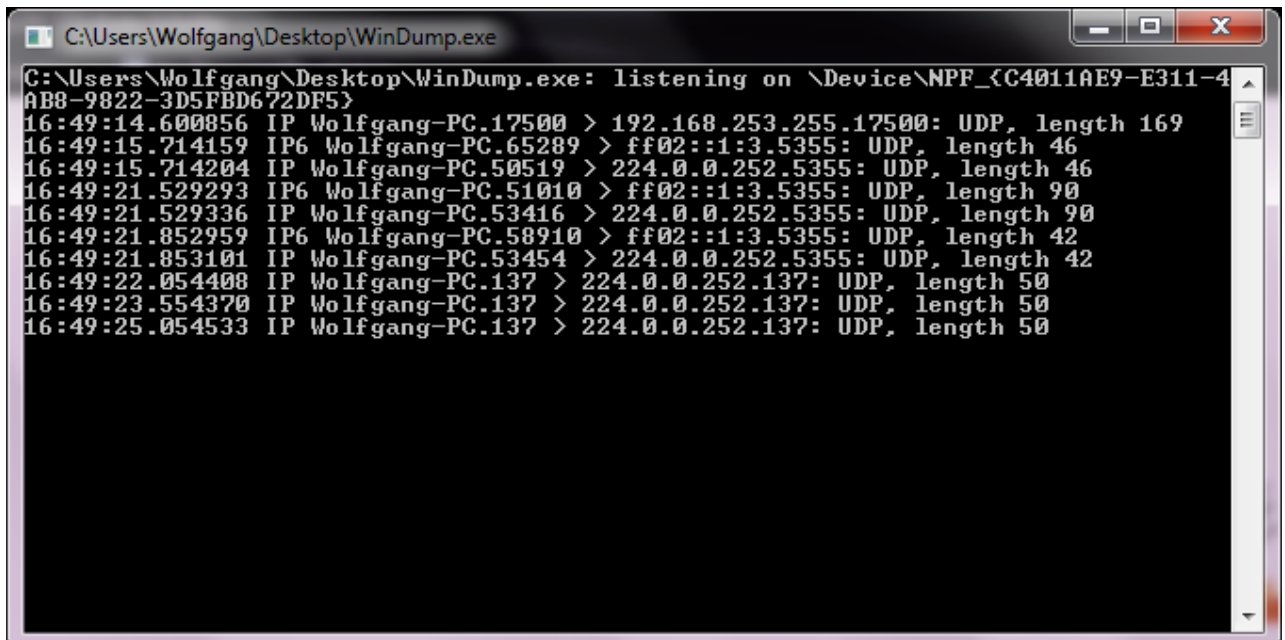


You can modify the discovery range (the range in which packages are searched) to IP-ranges and others.



## WinDump

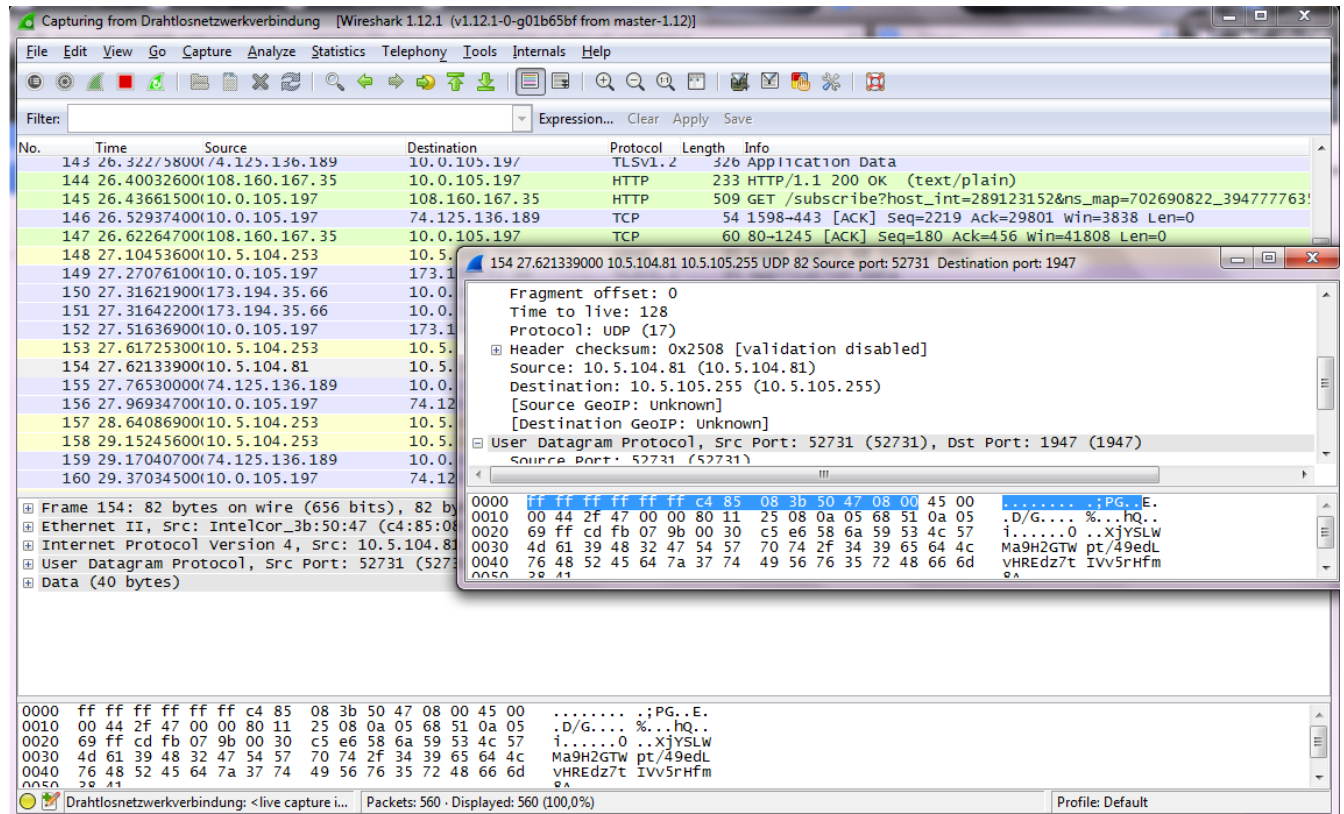
**WinDump** shows the raw packages that the networkcard receives before the operating system modifies them. Before you can use WinDump properly you have to install WinPcap. You can either download and install WinPcap on its own but it is also included with Wireshark (it asks you if you want to install WinPcap right when you install Wireshark)

A screenshot of the WinDump application window. The title bar shows the file path 'C:\Users\Wolfgang\Desktop\WinDump.exe'. The main window displays a list of captured network packets. The first line shows the interface name and MAC address: 'C:\Users\Wolfgang\Desktop\WinDump.exe: listening on \Device\NPF\_{C4011AE9-E311-4AB8-9822-3D5FBD672DF5}'. Subsequent lines show packet details including timestamp, IP addresses, and protocol (UDP).

```
C:\Users\Wolfgang\Desktop\WinDump.exe: listening on \Device\NPF_{C4011AE9-E311-4AB8-9822-3D5FBD672DF5}
16:49:14.600856 IP Wolfgang-PC.17500 > 192.168.253.255.17500: UDP, length 169
16:49:15.714159 IP6 Wolfgang-PC.65289 > ff02::1:3.5355: UDP, length 46
16:49:15.714204 IP Wolfgang-PC.50519 > 224.0.0.252.5355: UDP, length 46
16:49:21.529293 IP6 Wolfgang-PC.51010 > ff02::1:3.5355: UDP, length 90
16:49:21.529336 IP Wolfgang-PC.53416 > 224.0.0.252.5355: UDP, length 90
16:49:21.852959 IP6 Wolfgang-PC.58910 > ff02::1:3.5355: UDP, length 42
16:49:21.853101 IP Wolfgang-PC.53454 > 224.0.0.252.5355: UDP, length 42
16:49:22.054408 IP Wolfgang-PC.137 > 224.0.0.252.137: UDP, length 50
16:49:23.554370 IP Wolfgang-PC.137 > 224.0.0.252.137: UDP, length 50
16:49:25.054533 IP Wolfgang-PC.137 > 224.0.0.252.137: UDP, length 50
```

## Wireshark

**Wireshark** is a program that shows the package-traffic in the connected network. It can also open packages and read the contents. Wireshark great advantage in addition to its clear graphical user interface is that it is a completely free software.



## SNMP Framework

In this chapter we are going to compare the advantages as well the disadvantages of various SNMP-Frameworks. The quality of each framework depends on the following properties:

- Easy and intuitive usage
- Accomplishment of all tasks someone wishes
- Mostly bug-free
- Performance
- A lively community(Updates and solving compatibility problems)

## SNMP4J

[1] SNMP4J is an enterprise class free open source and state-of-the-art SNMP implementation for Java™ SE 1.4 or later\*. SNMP4J supports command generation (managers) as well as command responding (agents). Its clean object oriented design is inspired by SNMP++, which is a well-known SNMPv1/v2c/v3 API for C++ (see <http://www.agentpp.com>).

To call SNMP4J a marvelous framework would be an understatement. It provides an very indepth Java Documentation, which is indeed userfriendly and a must-have requirement. Each and every written package has its own summary, from which you gain more than enough information to know what exactly is going on.

SNMP4J has got its very own Wiki, where **Frequently Asked Question** and other useful features can be found. It also is equipped with simple-as-can-be example files for the own usage.

Package	Description
org.snmp4j	Provides classes and interfaces for creating, sending, and receiving SNMP messages.
org.snmp4j.asn1	Provides classes and interfaces for the mapping between Abstract Syntax Notation One (ASN.1) formatted values and their transfer syntax according to the Basic Encoding Rules (BER).
org.snmp4j.event	Provides classes and interfaces for SNMP4J event processing.
org.snmp4j.log	
org.snmp4j.mp	Provides classes and interfaces for the SNMP message processing.
org.snmp4j.security	Provides classes and interfaces for authentication and privacy of SNMP(v3) messages.
org.snmp4j.security.nonstandard	
org.snmp4j.smi	Provides classes for the representation of SMIPv2 data types (which also includes some basic ASN.1 primitive data types).
org.snmp4j.tools.console	
org.snmp4j.transport	Provides transport protocol mappings for SNMP.
org.snmp4j.transport.ssh	
org.snmp4j.transport.tls	
org.snmp4j.uri	
org.snmp4j.util	Contains table retrieval utilities and multi-threading support classes as well as miscellaneous utility classes.
org.snmp4j.version	

[3] To setup a default SNMP session for UDP transport and with SNMPv3 support the following code snippet can be used:

```
Address targetAddress = GenericAddress.parse("udp:127.0.0.1/161");
TransportMapping transport = new DefaultUdpTransportMapping();
snmp = new Snmp(transport);
USM usm = new USM(SecurityProtocols.getInstance(),
    new OctetString(MPv3.createLocalEngineID()), 0);
SecurityModels.getInstance().addSecurityModel(usm);
transport.listen();
```

#### Beispiel SNMP client/manager

```
import java.io.IOException;
import org.snmp4j.CommunityTarget;
import org.snmp4j.PDU;
import org.snmp4j.Snmp;
import org.snmp4j.Target;
import org.snmp4j.TransportMapping;
import org.snmp4j.event.ResponseEvent;
import org.snmp4j.mp.SnmpConstants;
import org.snmp4j.smi.Address;
import org.snmp4j.smi.GenericAddress;
import org.snmp4j.smi.OID;
import org.snmp4j.smi.OctetString;
import org.snmp4j.smi.VariableBinding;
import org.snmp4j.transport.DefaultUdpTransportMapping;

public class SNMPManager {
    Snmp snmp = null;
    String address = null;

    /**
     * Constructor
     * @param add
     */
    public SNMPManager(String add)
    {
        address = add;
    }
}
```

```

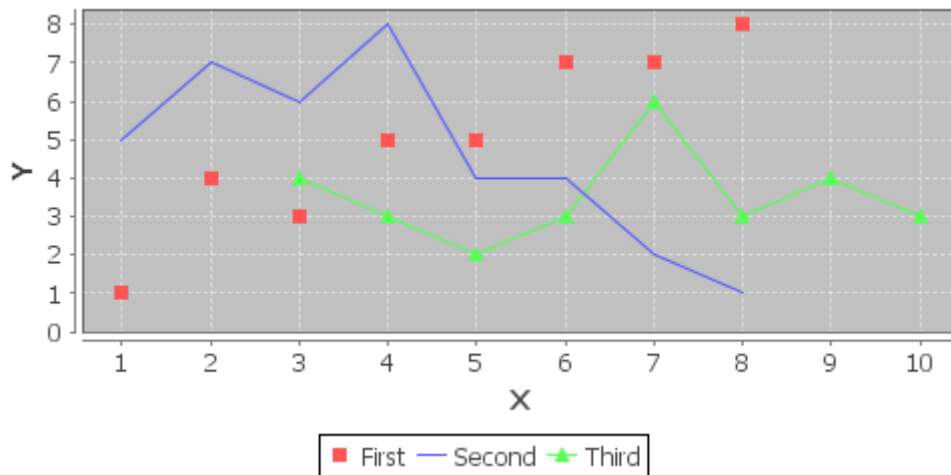
    }
    public static void main(String[] args) throws IOException {
        /**
         * Port 161 is used for Read and Other operations
         * Port 162 is used for the trap generation
         */
        SNMPManager client = new SNMPManager("udp:127.0.0.1/161");
        client.start();

        /**
         * OID - .1.3.6.1.2.1.1.1.0 => SysDec
         * OID - .1.3.6.1.2.1.1.5.0 => SysName
         * => MIB explorer will be usefull here, as discussed in previous article
         */
        String sysDescr = client.getAsString(new OID(".1.3.6.1.2.1.1.1.0"));
        System.out.println(sysDescr);
    }
    /**
     * Start the Snmp session. If you forget the listen() method you will not
     * get any answers because the communication is asynchronous
     * and the listen() method listens for answers.
     * @throws IOException
     */
    public void start() throws IOException {
        TransportMapping transport = new DefaultUdpTransportMapping();
        snmp = new Snmp(transport);
        // Do not forget this line!
        transport.listen();
    }
    /**
     * Method which takes a single OID and returns the response from the agent as a String.
     * @param oid
     * @return
     * @throws IOException
     */
    public String getAsString(OID oid) throws IOException {
        ResponseEvent event = get(new OID[] { oid });
        return event.getResponse().get(0).getVariable().toString();
    }
    /**
     * This method is capable of handling multiple OIDs
     * @param oids
     * @return
     * @throws IOException
     */
    public ResponseEvent get(OID oids[]) throws IOException {
        PDU pdu = new PDU();
        for (OID oid : oids) {
            pdu.add(new VariableBinding(oid));
        }
        pdu.setType(PDU.GET);
        ResponseEvent event = snmp.send(pdu, getTarget(), null);
        if(event != null) {
            return event;
        }
        throw new RuntimeException("GET timed out");
    }
    /**
     * This method returns a Target, which contains information about
     * where the data should be fetched and how.
     * @return
     */
    private Target getTarget() {
        Address targetAddress = GenericAddress.parse(address);
        CommunityTarget target = new CommunityTarget();
        target.setCommunity(new OctetString("public"));
        target.setAddress(targetAddress);
        target.setRetries(2);
        target.setTimeout(1500);
        target.setVersion(SnmpConstants.version2c);
        return target;
    }
}

```

## JFreeChart

We are going to use JFreeChart for the LineChart view. Below is a demo of a LineChart that we are going to use:



## JavaFx

- Compared to Swing, it provides a clear and clean architecture and features many enhancements:  
styling, event management, transitions, scene graph — to name a few.
- It provides the possibility of developing up-to-date user interfaces with animations, multitouch, and the like
- It is based on a clear and clean language: Java.
- It provides all the professional Java tooling required to debug, analyze, profile, and log a client application.
- It enables a simple app-like installation on the client side, without any prerequisites.

## **Task execution**



## GUI Design

### Mockup Design

RockTheNet - Login

File Edit View Tools Help

Name:

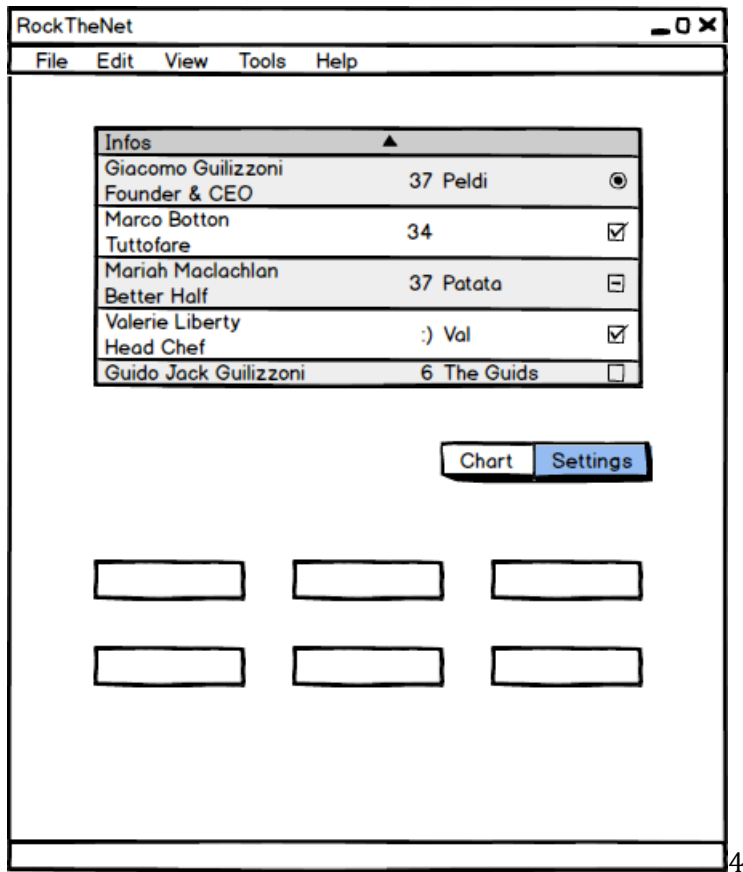
Password:

RockTheNet

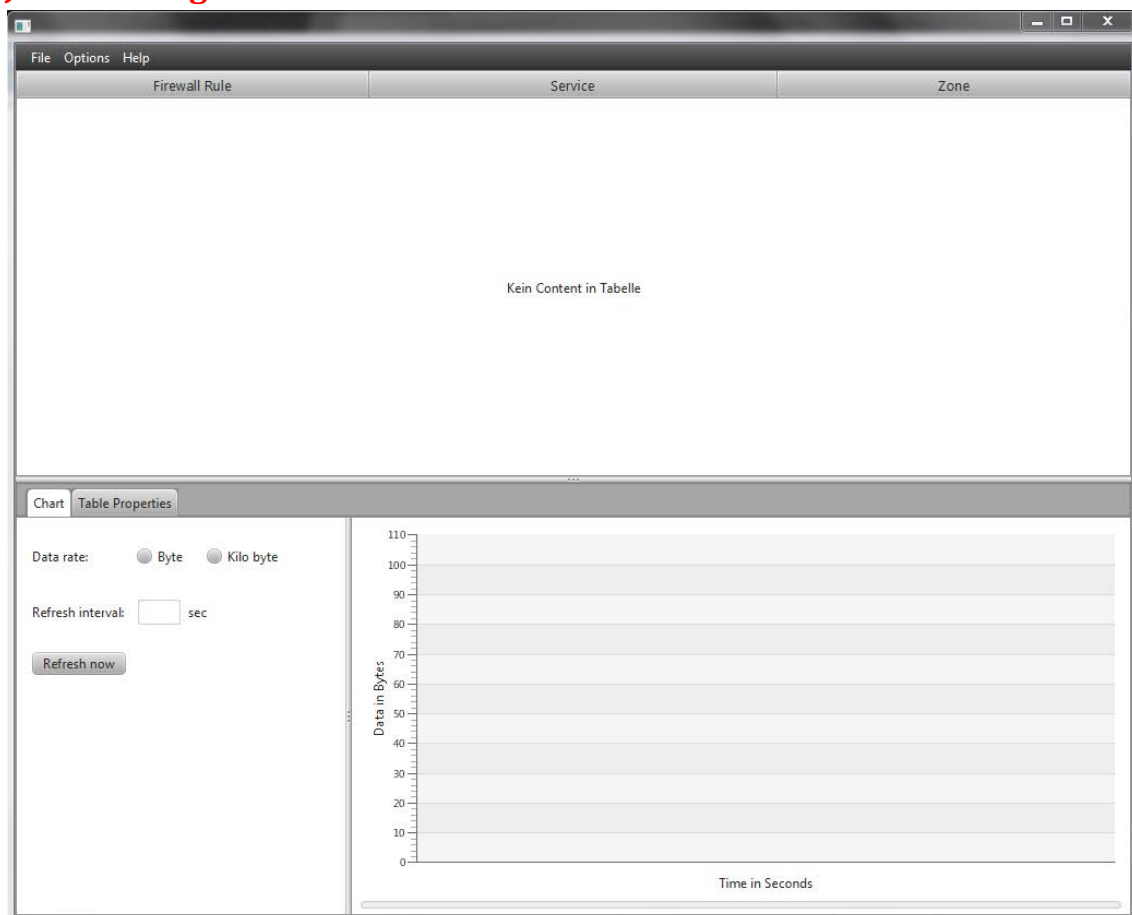
File Edit View Tools Help

Firewall Rule	Service	Zone	
Giacomo Guilizzoni Founder & CEO	37	Peldi	<input checked="" type="radio"/>
Marco Botton Tuttofare	34		<input checked="" type="checkbox"/>
Mariah MacLachlan Better Half	37	Patata	<input type="checkbox"/>
Valerie Liberty Head Chef	:)	Val	<input checked="" type="checkbox"/>
Guido Jack Guilizzoni	6	The Guids	<input type="checkbox"/>
{65L	0R	35	0Cs

Chart Settings

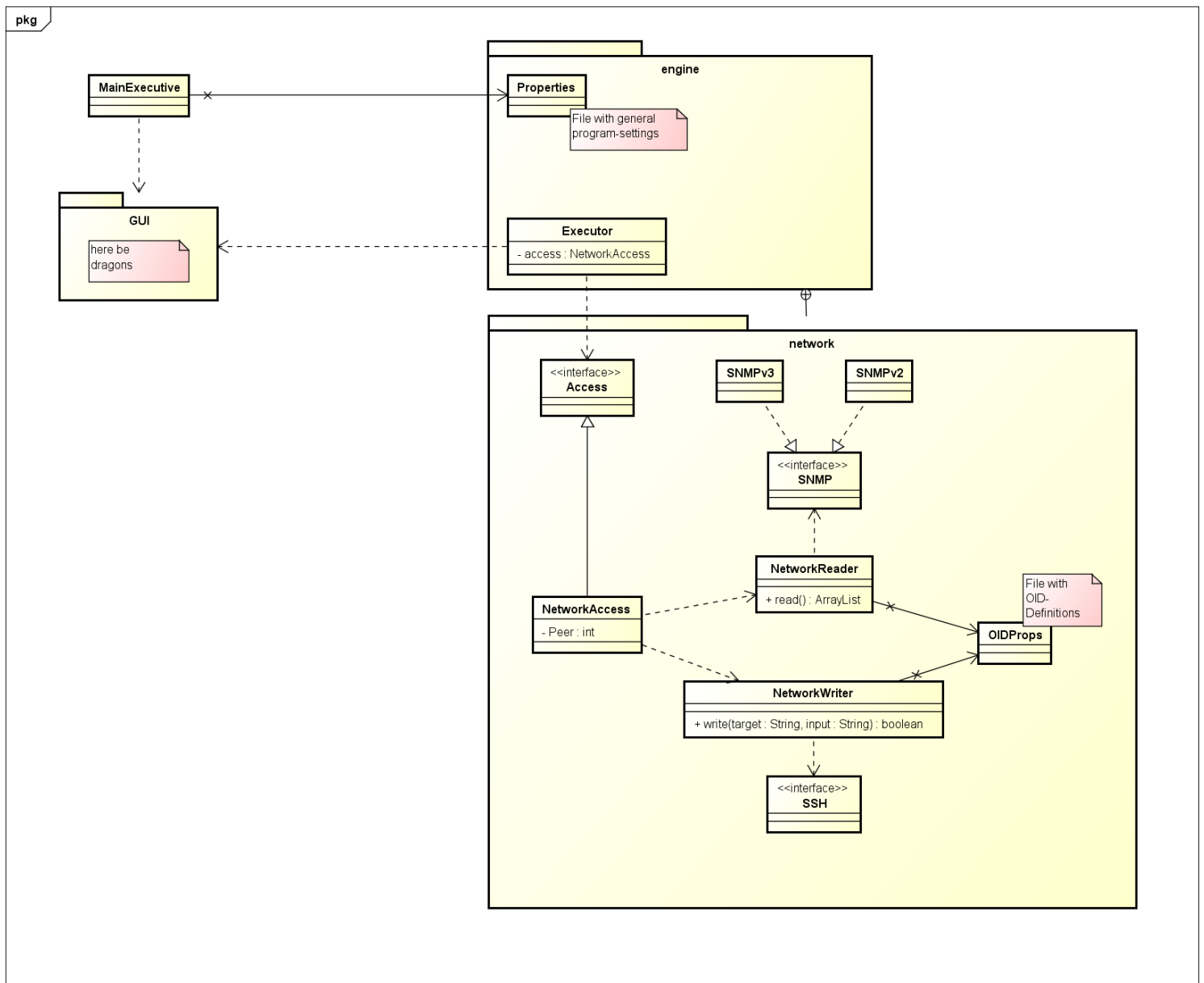


## JavaFX Design



## SW-Design consideration

## UML-Class diagram



## **Test report**

## Bibliography

---

- [1] Title: The SNMP4J Framework  
Author: ?  
Online-/Resource: <http://www.snmp4j.org/index.html>  
last modified: /  
abstracted: 9/23/2014
- [2] Title: inPcap Manual  
Author: man2html  
Online-/Resource: <http://www.winpcap.org/windump/docs/manual.htm>  
last modified: 1/12/2006  
abstracted: 9/23/2014
- [3] Title: SNMP4J Example + Description  
Author: ?  
Online-/Resource: <http://www.snmp4j.org/doc/org/snmp4j/Snmp.html>  
last modified: /  
abstracted: 9/23/2014
-