Development of a framework for performance testing of intrusion detection systems

Outline:

➢ **Organisation**

✗ Purpose
✗ Steps + Timing
✗ IDS - concepts
✗ Snort
✗ Requirements
✗ Design decisions
✗ Implementation decisions
✗ Issues
✗ Possible Improvements
✗ Scripts
✗ Test 1
✗ Test 2
✗ Test 3
✗ Conclusions

Organisation

Royal Military Academy (RMA)

Department of Communication, Information Systems and Sensors,

Computer Sciences Chair

- Belgian Army's Officers School
- They have research projects in various fields

Worked with Wim Mees and Olivier Thonnard.

Yannick LOTH, DIA3, SPP, University of Luxembourg, 2005
Organisation: CISS INFO - Royal Military Academy, Brussels

Development of a framework for performance testing of intrusion detection systems

Outline:

✔ Organisation
➢ **Purpose**
✗ Steps + Timing
✗ IDS - concepts
✗ Snort
✗ Requirements
✗ Design decisions
✗ Implementation decisions
✗ Issues
✗ Possible Improvements
✗ Scripts
✗ Test 1
✗ Test 2
✗ Test 3
✗ Conclusions

Purpose of the internship

look at the title...

Main goals:

- Develop a reusable framework to stress-test network devices, particularly IDSs
- Perform tests to show usage of the framework
- Show behaviour of IDSs under heavy network load
- Show interpretation of results

- Show that Snort is unreliable

Development of a framework for performance testing of intrusion detection systems

Outline:

- ✔ Organisation
- ✔ Purpose
- ➢ **Steps + Timing**
- ✗ IDS - concepts
- ✗ Snort
- ✗ Requirements
- ✗ Design decisions
- ✗ Implementation decisions
- ✗ Issues
- ✗ Possible Improvements
- ✗ Scripts
- ✗ Test 1
- ✗ Test 2
- ✗ Test 3
- ✗ Conclusions

Steps and timing

Weeks 1-7:

Searching and reading documentation, do some tests

Week 8:

Develop the program in C++ (completely inefficient)

Weeks 9-10:

Compile a custom kernel (only worked SuSE 9.2 Pro)

Weeks 11-12:

Develop the program in C

Test the program

Week 13-14:

Write the bash scripts

Do the tests

Week 15:

Interpret the results

Development of a framework for performance testing of intrusion detection systems

Outline:

- ✔ Organisation
- ✔ Purpose
- ✔ Steps + Timing
- ➢ **IDS - concepts**
- ✗ Snort
- ✗ Requirements
- ✗ Design decisions
- ✗ Implementation decisions
- ✗ Issues
- ✗ Possible Improvements
- ✗ Scripts
- ✗ Test 1
- ✗ Test 2
- ✗ Test 3
- ✗ Conclusions

Intrusion Detection Systems

- Passive network devices inspecting network data flow
- Alerts on detected attacks

Fundamental Components:

- Information sources
- Analysis
- Response

Uses:

- Detect attacks
- Document existing threats
- Used as quality control for security design and administration

Development of a framework for performance testing of intrusion detection systems

Outline:

- ✔ Organisation
- ✔ Purpose
- ✔ Steps + Timing
- ✔ IDS - concepts
- ➢ **Snort**
- ✗ Requirements
- ✗ Design decisions
- ✗ Implementation decisions
- ✗ Issues
- ✗ Possible Improvements
- ✗ Scripts
- ✗ Test 1
- ✗ Test 2
- ✗ Test 3
- ✗ Conclusions

Snort

- Open source
- Software
- Popular
- Ported to different OSes
- Present on almost any modern Linux distribution
- http://www.snort.org

- Studied by O. Thonnard for his Master Thesis

Development of a framework for performance testing of intrusion detection systems

Outline:

- ✔ Organisation
- ✔ Purpose
- ✔ Steps + Timing
- ✔ IDS - concepts
- ✔ Snort
- ➢ **Requirements**
- ✗ Design decisions
- ✗ Implementation decisions
- ✗ Issues
- ✗ Possible Improvements
- ✗ Scripts
- ✗ Test 1
- ✗ Test 2
- ✗ Test 3
- ✗ Conclusions

Requirements

- Reliably send packets at given bitrates
- Use 1 or 2 files A and B and mix their packets with a given ratio m:1
- Send a given total quantity of packets
- The application should run on computers available at the RMA:
  - 3 Intel Pentium 4 based Celeron, 2.60GHz, 768Mb RAM
  - Linux 2.6
  - Gigabit ethernet device cards

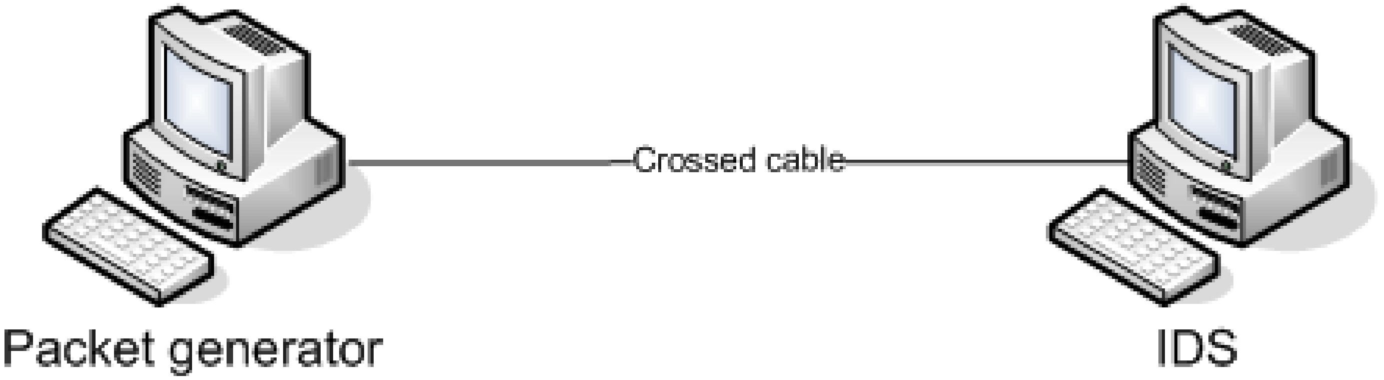
Development of a framework for performance testing of intrusion detection systems

Outline:

- ✔ Organisation
- ✔ Purpose
- ✔ Steps + Timing
- ✔ IDS - concepts
- ✔ Snort
- ✔ Requirements
- ➢ **Design decisions**
- ✗ Implementation decisions
- ✗ Issues
- ✗ Possible Improvements
- ✗ Scripts
- ✗ Test 1
- ✗ Test 2
- ✗ Test 3
- ✗ Conclusions

Design decisions

- Application to generate packets, scripts to synchronize
- Packets read from tcpdump formatted files(libpcap)
- Packets read into memory before sending
- In the critical loop, a minimum quantity of operations are done
- Busy-waiting



Yannick LOTH, DIA3, SPP, University of Luxembourg, 2005
Organisation: CISS INFO - Royal Military Academy, Brussels

Outline:

- ✔ Organisation
- ✔ Purpose
- ✔ Steps + Timing
- ✔ IDS - concepts
- ✔ Snort
- ✔ Requirements
- ✔ Design decisions
- ➢ **Implementation decisions**
- ✗ Issues
- ✗ Possible Improvements
- ✗ Scripts
- ✗ Test 1
- ✗ Test 2
- ✗ Test 3
- ✗ Conclusions

Implementation decisions

- Sockets API
- Packets sent to the data link layer (PF_PACKET, SOCK_RAW)
- C (for stressnet)
- Bash scripts (for the synchronization)
- 'Minimalist' Kernel 2.6
- Matlab (R14) scripts to plot the results

Issues

- Small packets:
  - Generate too much overhead
  - As overhead is not quantifiable => synchronization impossible, test scripts fail

Development of a framework for performance testing of intrusion detection systems

Outline:

- ✔ Organisation
- ✔ Purpose
- ✔ Steps + Timing
- ✔ IDS - concepts
- ✔ Snort
- ✔ Requirements
- ✔ Design decisions
- ✔ Implementation decisions
- ✔ Issues
- ➢ **Possible Improvements**
- ✗ Scripts
- ✗ Test 1
- ✗ Test 2
- ✗ Test 3
- ✗ Conclusions

Possible Improvements

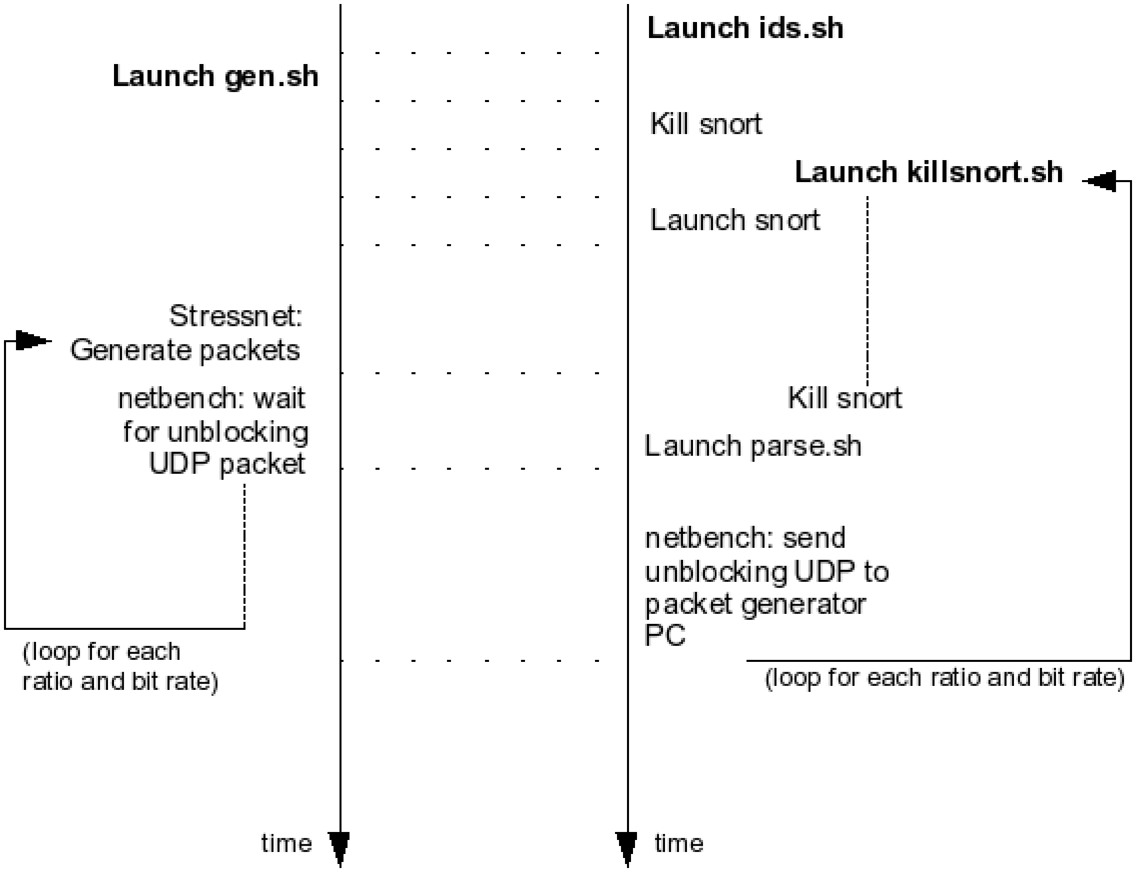
- Optimization of the algorithm used to put the packets into memory
- Use `sendmsg()` instead of `sendto()`
- Permit the use of more than 2 files
- Extend stressnet to a 2-tier application, which would enable to pre-calculate and exchange a complete tcp session between 2 computers

Development of a framework for performance testing of intrusion detection systems

Outline:

- ✔ Organisation
- ✔ Purpose
- ✔ Steps + Timing
- ✔ IDS - concepts
- ✔ Snort
- ✔ Requirements
- ✔ Design decisions
- ✔ Implementation decisions
- ✔ Issues
- ✔ Possible Improvements
- ➢ **Scripts**
- ✗ Test 1
- ✗ Test 2
- ✗ Test 3
- ✗ Conclusions

Scripts



Launch gen.sh

Stressnet: Generate packets

netbench: wait for unblocking UDP packet

(loop for each ratio and bit rate)

time

Launch ids.sh

Kill snort

Launch killsnort.sh

Launch snort

Kill snort

Launch parse.sh

netbench: send unblocking UDP to packet generator PC

(loop for each ratio and bit rate)

time

Outline:

- ✔ Organisation
- ✔ Purpose
- ✔ Steps + Timing
- ✔ IDS - concepts
- ✔ Snort
- ✔ Requirements
- ✔ Design decisions
- ✔ Implementation decisions
- ✔ Issues
- ✔ Possible Improvements
- ✔ Scripts
- ➢ **Test 1**
- ✗ Test 2
- ✗ Test 3
- ✗ Conclusions

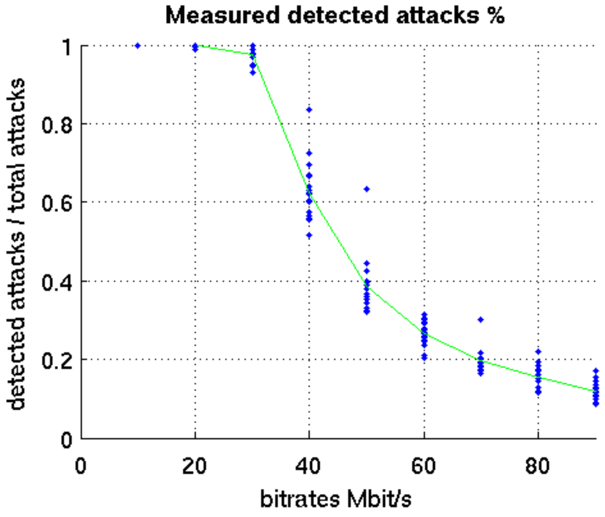
Test 1: SMTP with PCRE

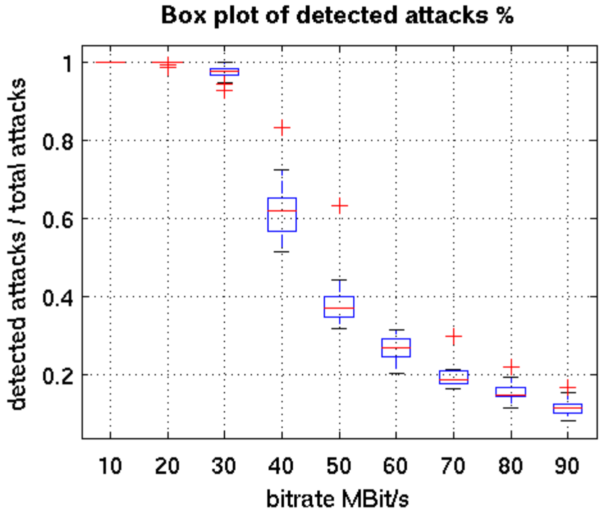
- Attack: UDP zero packets
- Standard flow: SMTP packets with repeating PCRE-detection triggering patterns
- 200 000 packets
- 20 tests for each of the following bitrates: 10, 20, 30, 40, 50, 60, 70, 80, 90 Mbit/s

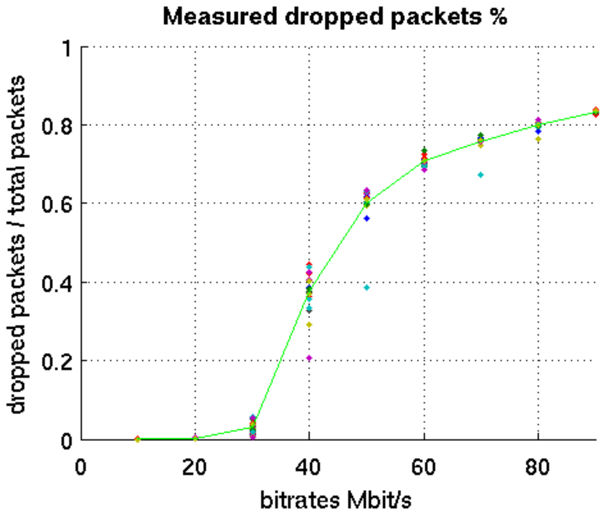
Development of a framework for performance testing of intrusion detection systems

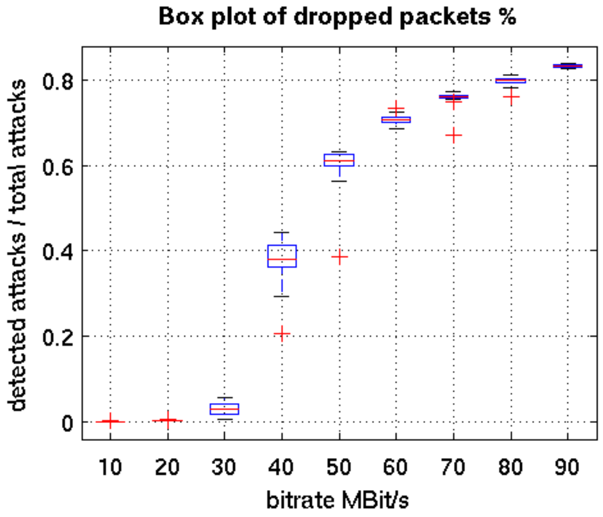
Outline:

✔ Organisation
✔ Purpose
✔ Steps + Timing
✔ IDS - concepts
✔ Snort
✔ Requirements
✔ Design decisions
✔ Implementation decisions
✔ Issues
✔ Possible Improvements
✔ Scripts
➤ **Test 1**
✗ Test 2
✗ Test 3
✗ Conclusions


Measured detected attacks %


Box plot of detected attacks %

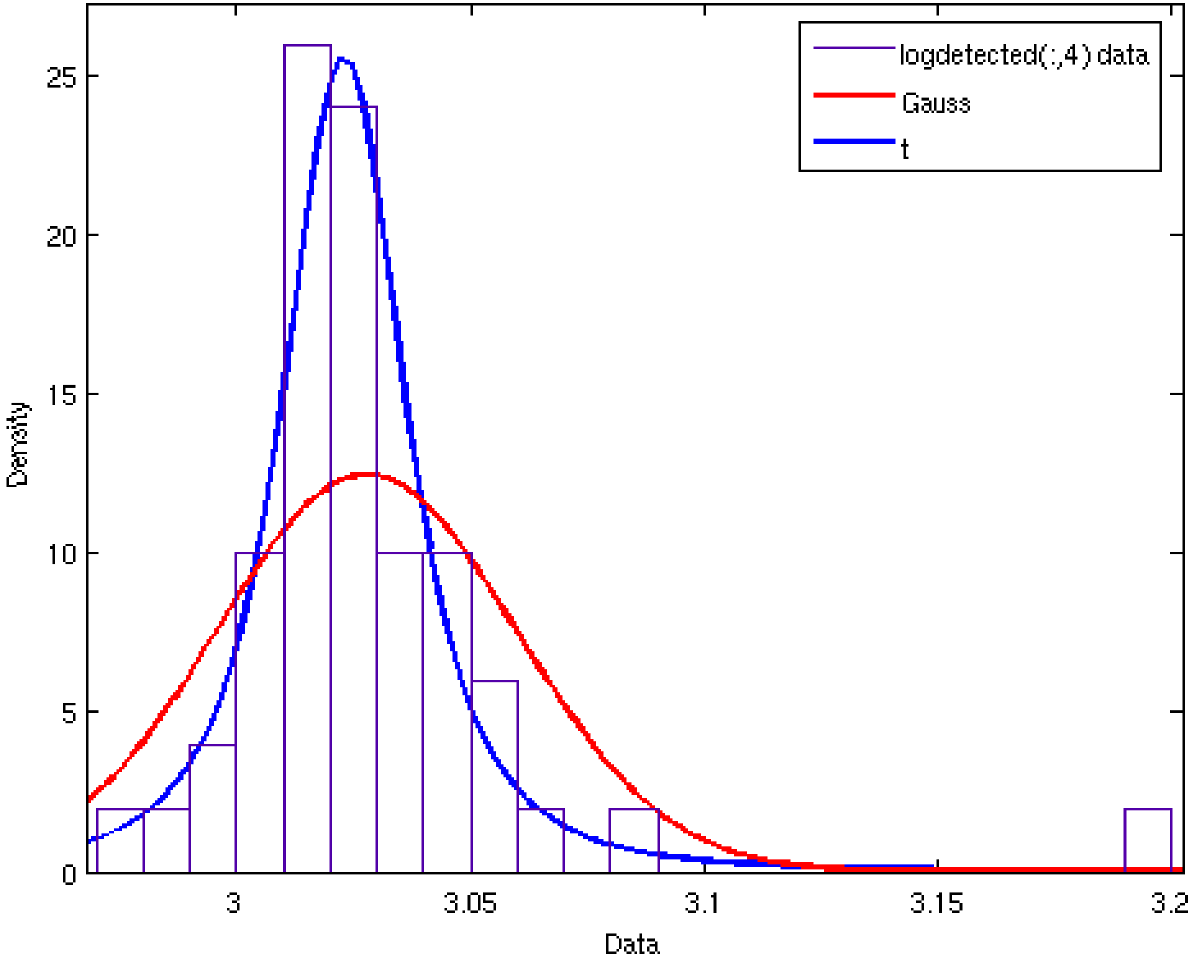

Measured dropped packets %


Box plot of dropped packets %

Development of a framework for performance testing of intrusion detection systems

Outline:

- ✔ Organisation
- ✔ Purpose
- ✔ Steps + Timing
- ✔ IDS - concepts
- ✔ Snort
- ✔ Requirements
- ✔ Design decisions
- ✔ Implementation decisions
- ✔ Issues
- ✔ Possible Improvements
- ✔ Scripts
- ➢ **Test 1**
- ✗ Test 2
- ✗ Test 3
- ✗ Conclusions



Yannick LOTH, DIA3, SPP, University of Luxembourg, 2005
Organisation: CISS INFO - Royal Military Academy, Brussels

Development of a framework for performance testing of intrusion detection systems

Outline:

- ✔ Organisation
- ✔ Purpose
- ✔ Steps + Timing
- ✔ IDS - concepts
- ✔ Snort
- ✔ Requirements
- ✔ Design decisions
- ✔ Implementation decisions
- ✔ Issues
- ✔ Possible Improvements
- ✔ Scripts
- ✔ Test 1
- ➢ **Test 2**
- ✗ Test 3
- ✗ Conclusions

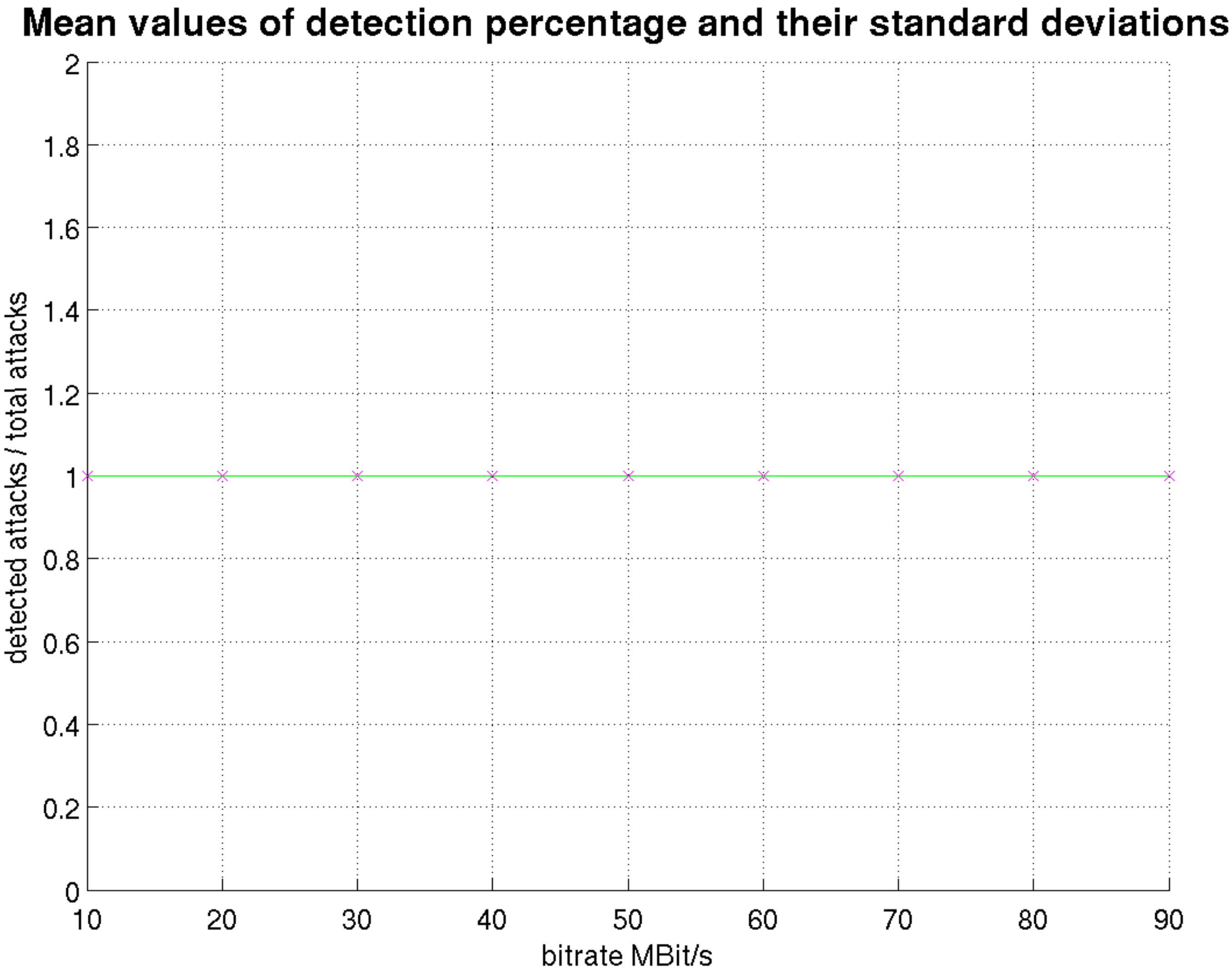
Test 2: Standard complete HTTP sessions

- Attack: UDP zero packets
- Standard flow: 3 complete HTTP sessions (MS IE)
- 200 000 packets
- 20 tests for each of the following bitrates:
  10, 20, 30, 40, 50, 60, 70, 80, 90 Mbit/s

Development of a framework for performance testing of intrusion detection systems

Outline:

- ✔ Organisation
- ✔ Purpose
- ✔ Steps + Timing
- ✔ IDS - concepts
- ✔ Snort
- ✔ Requirements
- ✔ Design decisions
- ✔ Implementation decisions
- ✔ Issues
- ✔ Possible Improvements
- ✔ Scripts
- ✔ Test 1
- ➢ **Test 2**
- ✗ Test 3
- ✗ Conclusions



Yannick LOTH, DIA3, SPP, University of Luxembourg, 2005
Organisation: CISS INFO - Royal Military Academy, Brussels

Development of a framework for performance testing of intrusion detection systems

Outline:

- ✔ Organisation
- ✔ Purpose
- ✔ Steps + Timing
- ✔ IDS - concepts
- ✔ Snort
- ✔ Requirements
- ✔ Design decisions
- ✔ Implementation decisions
- ✔ Issues
- ✔ Possible Improvements
- ✔ Scripts
- ✔ Test 1
- ✔ Test 2
- ➢ **Test 3**
- ✗ Conclusions

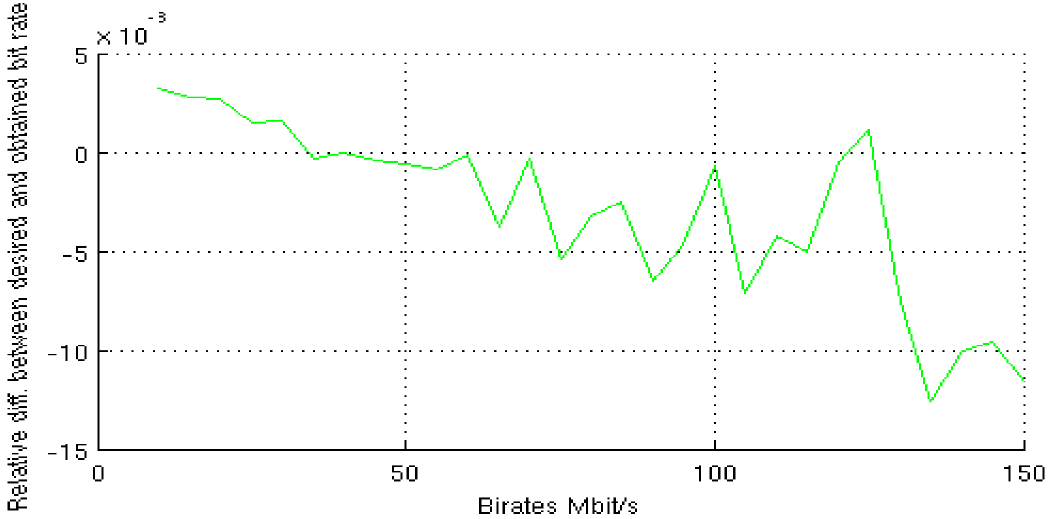
Test 3: Stressnet's reliability

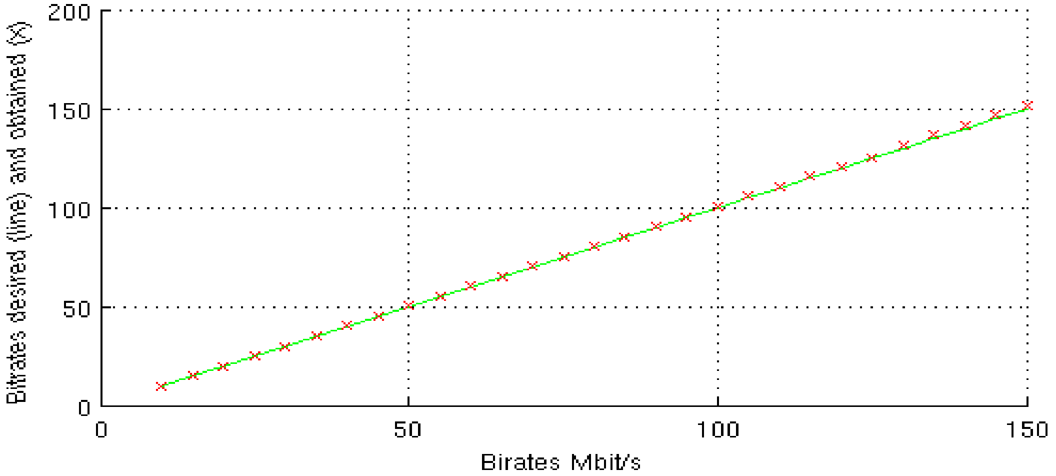
- Flow: 3 complete HTTP sessions (MS IE)
- 200 000 packets
- 20 tests for each bitrate multiple of 5 Mbit/s between 5Mbit/s and 150 Mbit/s
- Results plotted with Matlab (R14)
- Effective bitrates read with the tool capturecounter
  - Uses libpcap
  - Not documented here
  - Simple counter printing bitrate every 1 second (this is all the application does)

Development of a framework for performance testing of intrusion detection systems

Outline:

- ✔ Organisation
- ✔ Purpose
- ✔ Steps + Timing
- ✔ IDS - concepts
- ✔ Snort
- ✔ Requirements
- ✔ Design decisions
- ✔ Implementation decisions
- ✔ Issues
- ✔ Possible Improvements
- ✔ Scripts
- ✔ Test 1
- ✔ Test 2
- ➢ **Test 3**
- ✗ Conclusions





Development of a framework for performance testing of intrusion detection systems

Outline:

✔ Organisation
✔ Purpose
✔ Steps + Timing
✔ IDS - concepts
✔ Snort
✔ Requirements
✔ Design decisions
✔ Implementation decisions
✔ Issues
✔ Possible Improvements
✔ Scripts
✔ Test 1
✔ Test 2
➢ **Test 3**
✗ Conclusions

Test 3: Stressnet's reliability

- This test should be extended with packets of different sizes between 2 tests, but constant packet sizes in the same test

Development of a framework for performance testing of intrusion detection systems

Outline:

- ✔ Organisation
- ✔ Purpose
- ✔ Steps + Timing
- ✔ IDS - concepts
- ✔ Snort
- ✔ Requirements
- ✔ Design decisions
- ✔ Implementation decisions
- ✔ Issues
- ✔ Possible Improvements
- ✔ Scripts
- ✔ Test 1
- ✔ Test 2
- ✔ Test 3
- ➢ **Conclusions**

Conclusions

- Stressnet should be optimized (speed and memory)
- We've found how to make Snort inefficient at 30 Mbit/s with regular datagrams
- We've probably found the worst case
- We have the experience to extend stressnet, to solve issues and to interpret results
- I've got a much better knowledge of Linux
- I've got a professional experience about engineering tasks (understand concepts, apply techniques, solve issues, interpret results, work in a team)
- Exchanging ideas and experiences permits to raise efficiency, speed and correctness of work

- Now we have effective knowledge and experience to develop a high quality network device test tool