Investigative Analysis of Bot Army Attacks and Defense against them

Igor Kotenko

Laboratory of Computer Security Problems
SPIIRAS

Metaphor

"When all you have is a hammer, everything looks like a nail." - The success of advancing critical technologies, to a large extent, depends on the available tools that can help effectively prototype, test, and analyze new designs and new ideas.

/Jason Liu, Florida University/

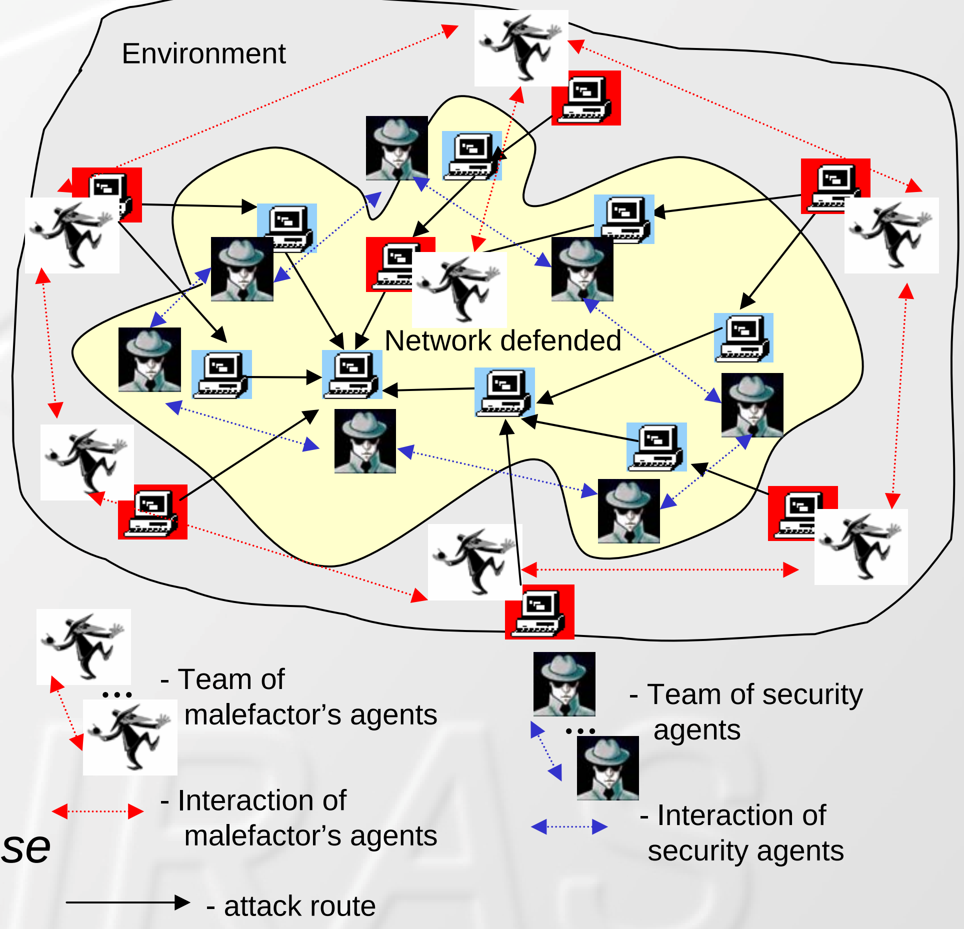
We suppose that in the context of cyberwarfare we need advanced instrumental tools which can be used for analyzing current and future network attacks and defense mechanisms, investigating the scenarios of cyber attacks and cyber defense, as well as be applied for "laboratorial" forensic investigation of cyber attacks fulfilled.

Modeling and simulation of cyberwarfare - Research objectives

Development of the *formal framework, models, architecture, and software for agent-based modeling and simulation of adversarial interaction of teams of malefactors and security teams* aimed to create theoretical bases for construction of ***integrated intrusion-aware trusted security systems operating in adversarial environments***.

The approach is based on using a hierarchy of macro and micro level models of network attacks and defense (analytical, packet-based, emulation-based), and real small-sized networks.

Interaction of team of malefactors and computer network assurance system components



Environment

Network defended

- Team of malefactor's agents

- Team of security agents

- Interaction of malefactor's agents

- Interaction of security agents

- attack route

Basic Assumptions

- Counteraction, competition or cooperation in a particular system (subject domain) is represented as the interaction of various teams of software agents.
- The aggregated behavior of this system is revealed due to local interactions of particular agents in the dynamic environment defined by the agent interaction model.
- Classes of agent teams are selected depending on the solved task and subject domain. We assume to select several classes of agent subsystems (teams): - Adversary attacking teams; - Security (defense) system; - Users (Background processes).
- Agents of different teams can be in indifference relationship, cooperate to reach the same goal or various consistent goals or compete to reach the opposite intentions.
- Agents of the same team cooperate to achieve joint intention.

Teamwork and Taskwork Maintenance Mechanisms

- The general intentions of agents are determined in several plans which describe the actions of teams as well as the actions of particular agents.
- Coordinated tasks and competing tasks are carried out due to installation of constraints on teams' and agents' roles and functionalities.

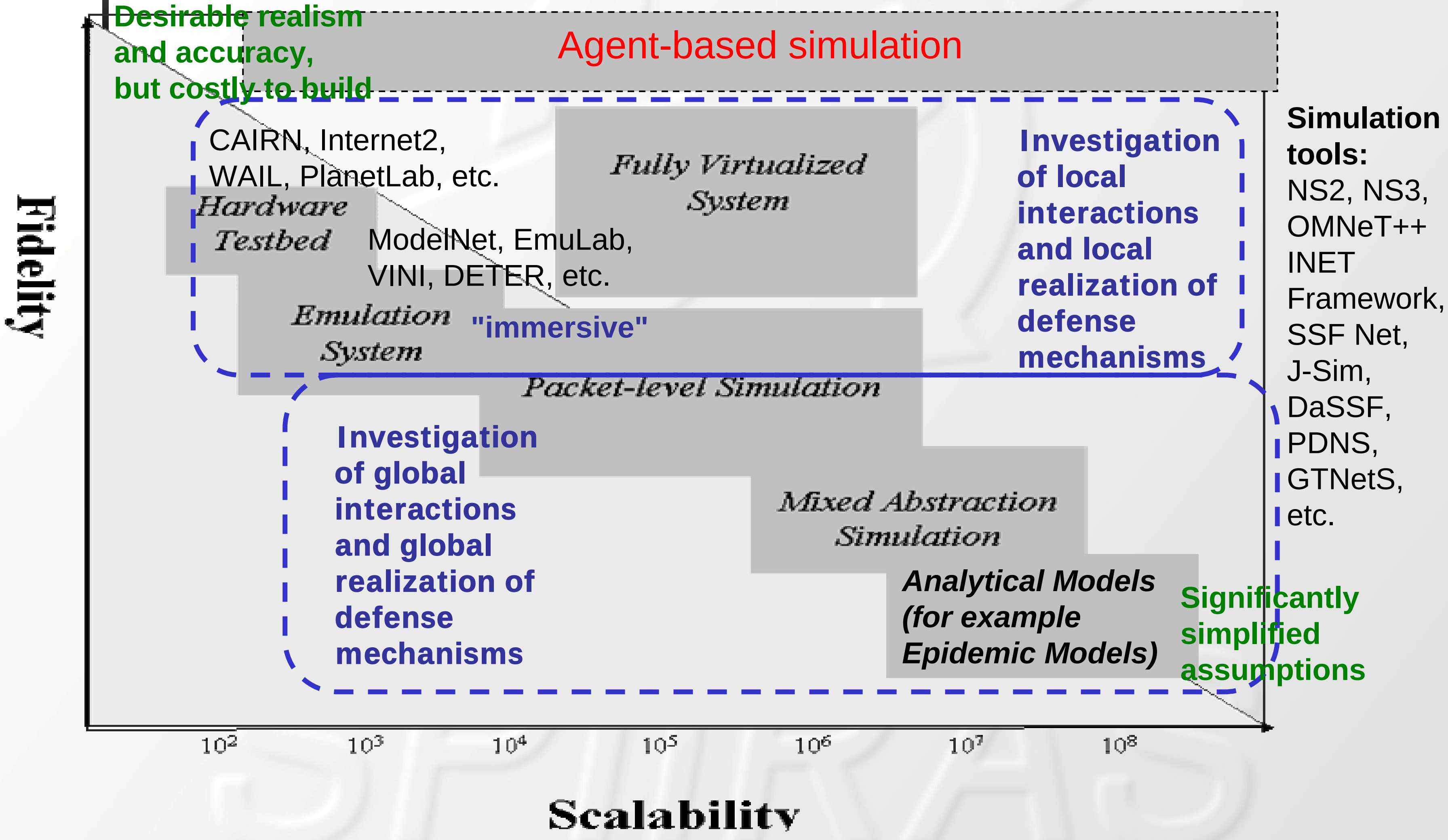
Taskwork maintenance mechanisms support agents' collaborative actions to maintain conditions for executing the task appointed to the team.

Teamwork maintenance mechanisms support agents' actions to maintain the team.

Basic mechanisms for taskwork and teamwork support:

- maintenance of actions coordination;
- monitoring and restoration of agents' functionality;
- maintenance of communication selectivity.

Range of Alternatives for Investigation



Desirable realism and accuracy, but costly to build

Agent-based simulation

CAIRN, Internet2, WAIL, PlanetLab, etc.

Hardware Testbed

ModelNet, EmuLab, VINI, DETER, etc.

Fully Virtualized System

Emulation System

"immersive"

Packet-level Simulation

Investigation of local interactions and local realization of defense mechanisms

Investigation of global interactions and global realization of defense mechanisms

Mixed Abstraction Simulation

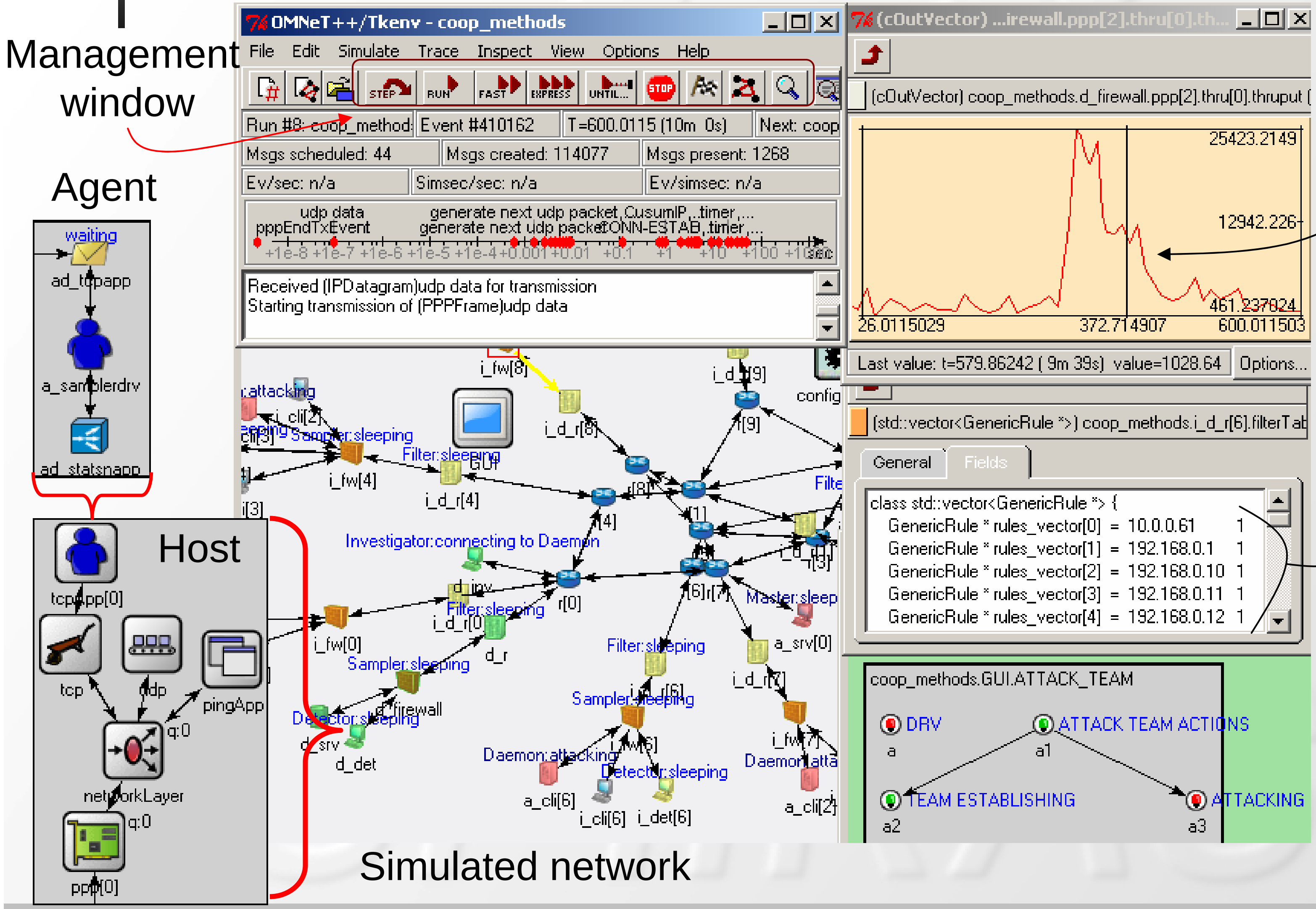
Analytical Models (for example Epidemic Models)

Significantly simplified assumptions

Simulation tools:
NS2, NS3, OMNeT++ INET Framework, SSF Net, J-Sim, DaSSF, PDNS, GTNetS, etc.

Source: [K.Perumalla, S.Sundaragopalan-04]

User Interface of Simulation Environment



Management window

Agent

Host

Simulated network

Network parameters

Agent work parameters

Teamwork parameters

OMNeT++/Tkenv - coop_methods

Run #8: coop_method | Event #410162 | T=600.0115 (10m 0s) | Next: coop

Msgs scheduled: 44 | Msgs created: 114077 | Msgs present: 1268

Ev/sec: n/a | Simsec/sec: n/a | Ev/simsec: n/a

```
Received (IPDatagram)udp data for transmission
Starting transmission of (PPPFrame)udp data
```

(cOutVector) ...irewall.ppp[2].thru[0].th...

(cOutVector) coop_methods.d_firewall.ppp[2].thru[0].thruput

Last value: t=579.86242 (9m 39s) value=1028.64

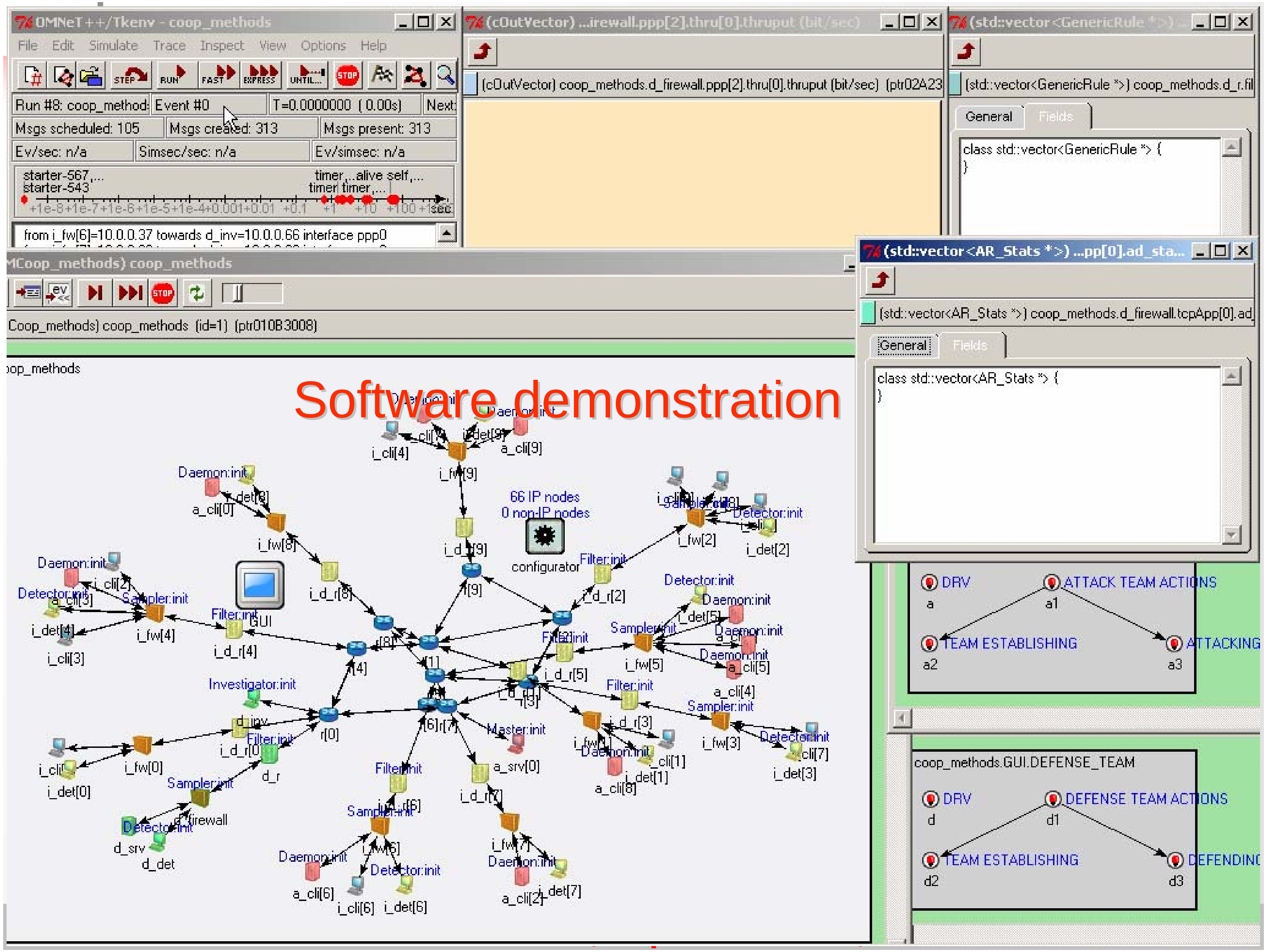
(std::vector<GenericRule *>) coop_methods.i_d_r[6].filterTab

General | Fields

```
class std::vector<GenericRule *> {
   GenericRule * rules_vector[0]  =  10.0.0.61     1
   GenericRule * rules_vector[1]  =  192.168.0.1   1
   GenericRule * rules_vector[2]  =  192.168.0.10  1
   GenericRule * rules_vector[3]  =  192.168.0.11  1
   GenericRule * rules_vector[4]  =  192.168.0.12  1
```

coop_methods.GUI.ATTACK_TEAM

Software demonstration



OMNeT++/Tkenv - coop_methods

Run #8: coop_method: Event #0 T=0.0000000 (0.00s) Next:

Msgs scheduled: 105 Msgs created: 313 Msgs present: 313

Ev/sec: n/a Simsec/sec: n/a Ev/simsec: n/a

starter-567,...
starter-543
timer,..alive self,...
timer timer,...
+1e-8 +1e-7 +1e-6 +1e-5 +1e-4 +0.001 +0.01 +0.1 +1 +10 +100 +1sec

```
from i_fw[6]=10.0.0.37 towards d_inv=10.0.0.66 interface ppp0
```

(cOutVector) ...irewall.ppp[2].thru[0].thruput (bit/sec)

(cOutVector) coop_methods.d_firewall.ppp[2].thru[0].thruput (bit/sec) (ptr02A23

(std::vector<GenericRule *>) ...

(std::vector<GenericRule *>) coop_methods.d_r.fil

```
class std::vector<GenericRule *> {
}
```

(std::vector<AR_Stats *>) ...pp[0].ad_sta...

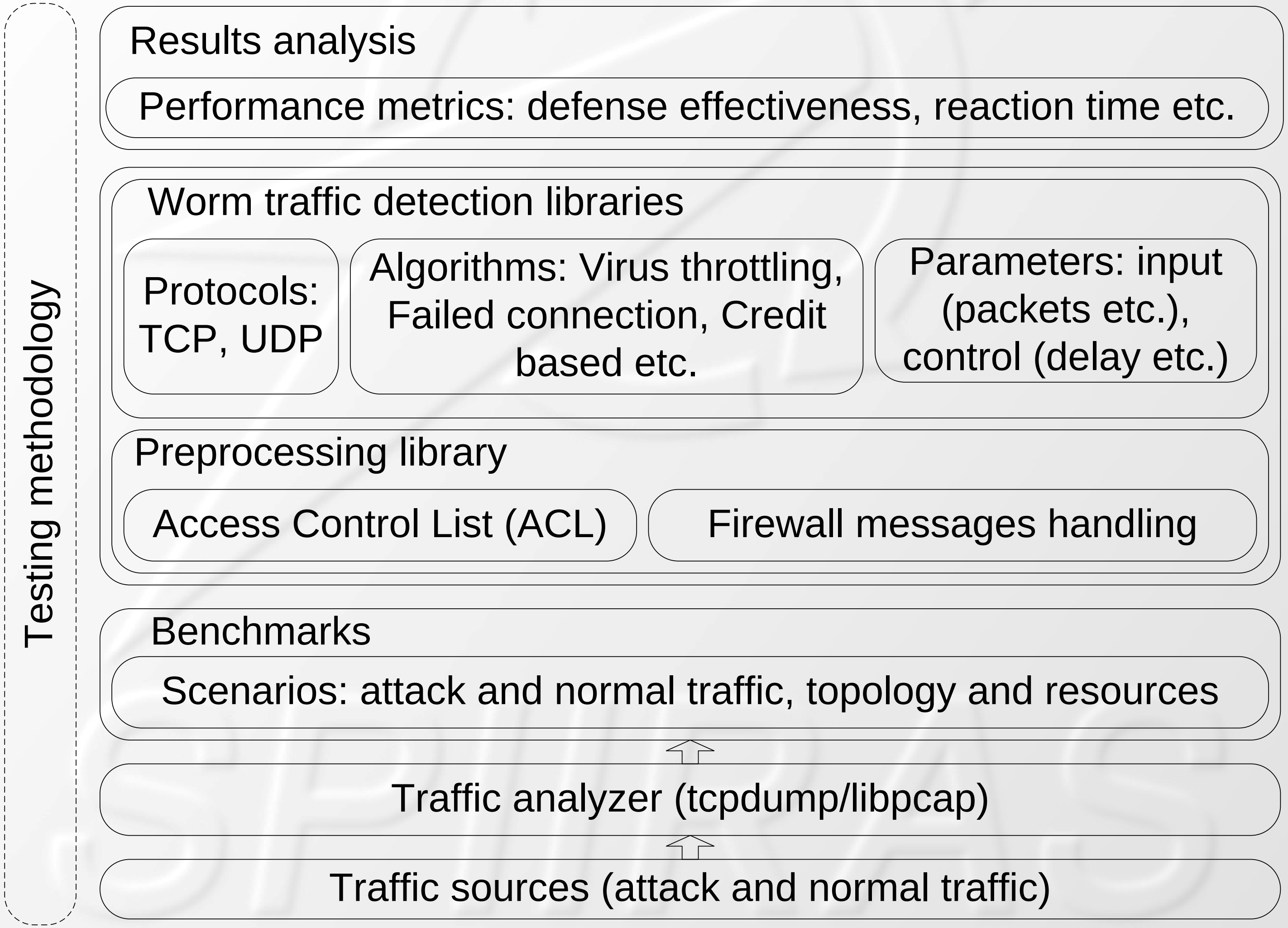
(std::vector<AR_Stats *>) coop_methods.d_firewall.tcpApp[0].ad

```
class std::vector<AR_Stats *> {
}
```

(Coop_methods) coop_methods

(Coop_methods) coop_methods (id=1) (ptr010B3008)

Example of simulation tool architecture for worm defense mechanisms investigation



Testing methodology

Results analysis

Performance metrics: defense effectiveness, reaction time etc.

Worm traffic detection libraries

Protocols: TCP, UDP

Algorithms: Virus throttling, Failed connection, Credit based etc.

Parameters: input (packets etc.), control (delay etc.)

Preprocessing library

Access Control List (ACL)

Firewall messages handling

Benchmarks

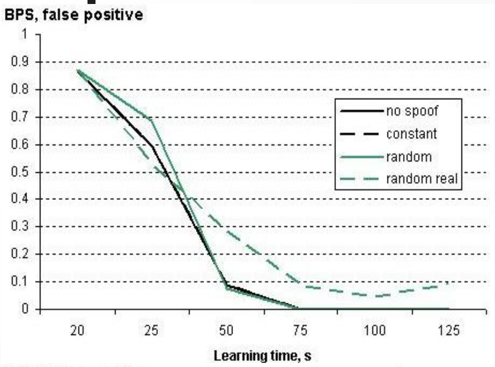
Scenarios: attack and normal traffic, topology and resources

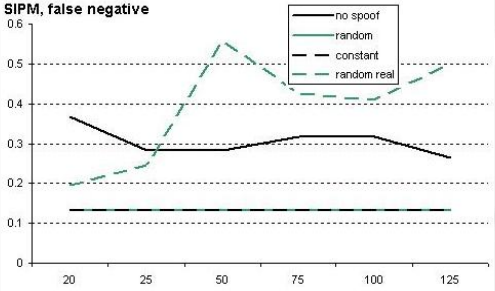
Traffic analyzer (tcpdump/libpcap)

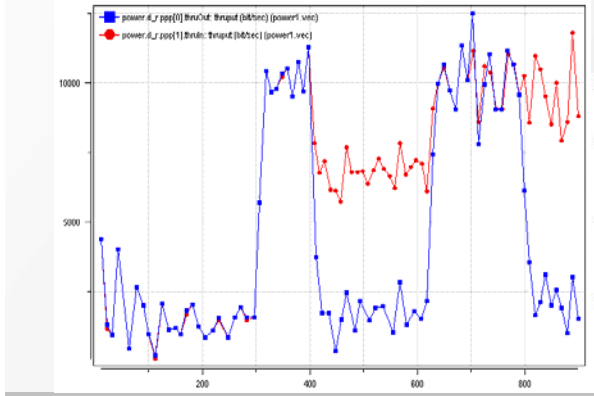
Traffic sources (attack and normal traffic)

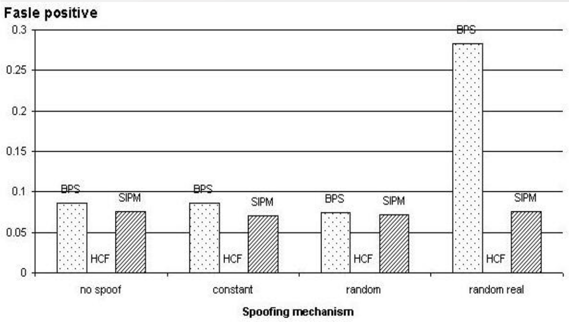
Experiments

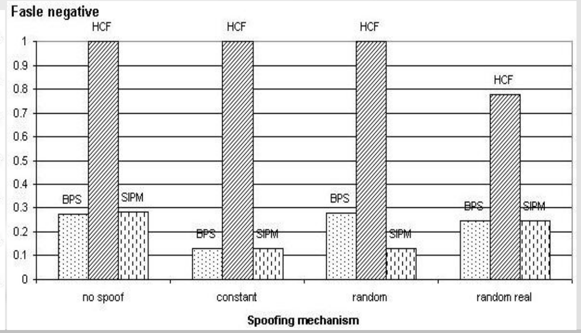
- The developed environment allows to carry out various experiments to investigate the strategies of attacks and prospective defense mechanisms.
- It allows to vary the following parameters during experiments: Network topology and configuration; Structure and configuration of defense and attack teams; Attack and defense mechanisms and their internal parameters; Team cooperation parameters etc.
- Examples: cooperation schemas are used in cooperative DDoS defense methods (COSSACK, Perimeter-based DDoS defense, DefCOM, Gateway-based, ACC pushback, MbSQD, SOS, tIP router architecture, etc.)


BPS, false positive
Learning time, s
no spoof
constant
random
random real


SIPM, false negative
no spoof
random
constant
random real


power.d_r.ppp[0].thruOut: thruput (bit/sec) (power1.vec)
power.d_r.ppp[1].thruIn: thruput (bit/sec) (power1.vec)


Fasle positive
BPS
HCF
SIPM
no spoof
constant
random
random real
Spoofing mechanism


Fasle negative
BPS
HCF
SIPM
no spoof
constant
random
random real
Spoofing mechanism