SPIIRAS Team in RE-TRUST: team background and preliminary analysis of tasks to be solved

Igor Kotenko

Computer Security Research Group,

St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences

Contents

1. St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS)
2. International collaboration, projects and accomplishments of Computer Security Research Group
3. Tasks in RE-TRUST
4. Computer Security Research Group research - General view
   - Modeling and simulation of computer attacks
   - Security analysis of computer networks
   - Intrusion detection
   - Deception systems, honeynets
   - Modeling and simulation of cyberwarfare
   - Security policy specification and checking
   - Security protocols analysis

1. St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS)

STRUCTURE OF THE RUSSIAN ACADEMY OF SCIENCES

PREZIDIUM
Yu. S. Osipov, President

Departments representing scientific fields

- Mathematical Sciences
- Physical Sciences
- Information Technologies & Computer Systems
- Power & Mechanical Engineering, Mechanics & Control Processes
- Biological Sciences
- Chemistry & Materials Science
- Social Sciences
- Historical & Philological Sciences
- Earth Sciences

Regional Branches

Far East	Siberian	Ural

Scientific Centres

Far East	Siberian	Ural
Amur	Buryat	Arkhangelsk
Kamchatka	Irkutsk	Komi
North-East	Kemerovo	Orenburg
Khabarovsk	Krasnoyarsk	Perm
Primorski	Novosibirsk	Udmurt
Sakhalim	Omsk	Chelyabinsk
	Tomsk	
	Tyumen	
	Yakutsk	

Regional Scientific Centres

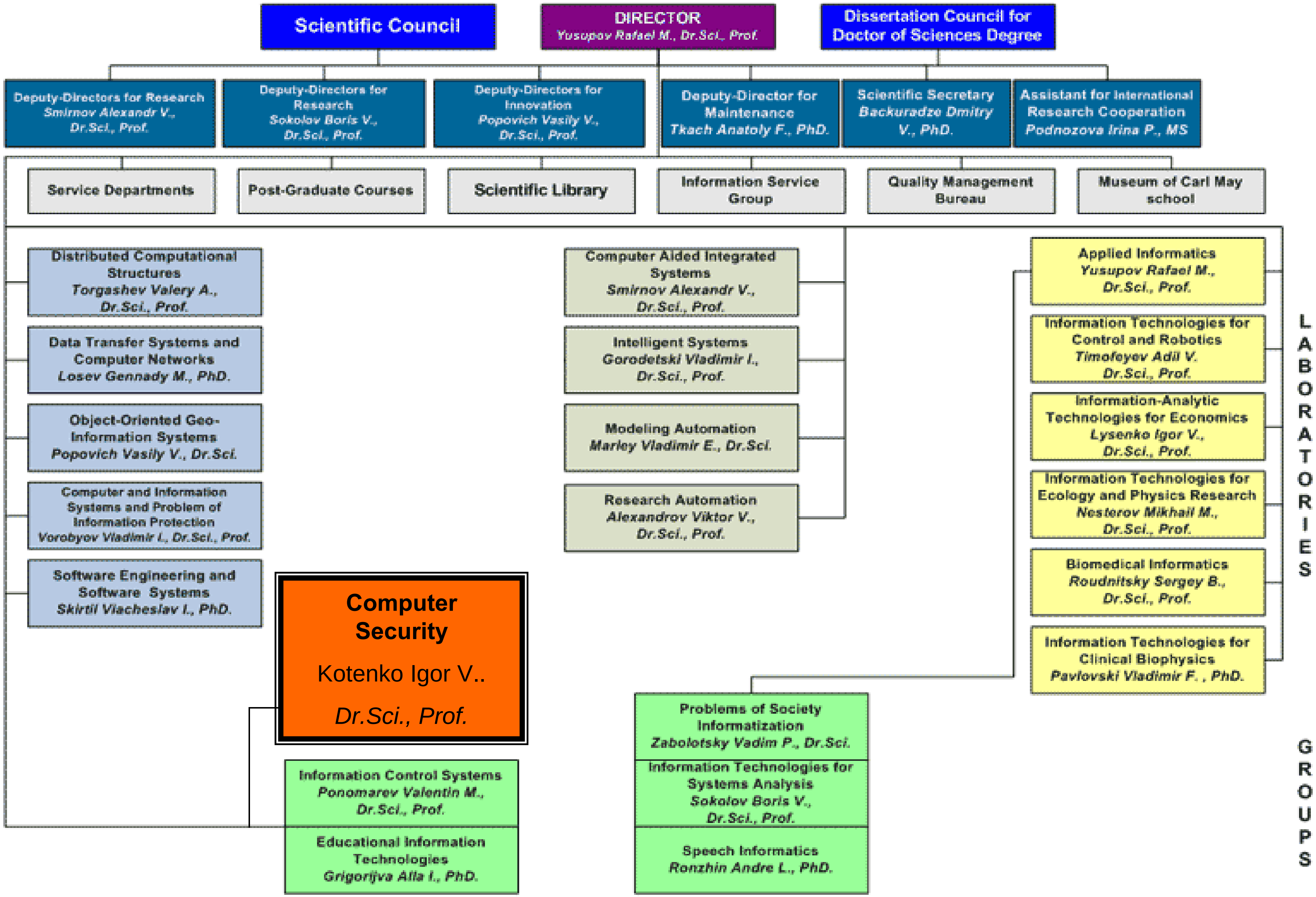
St. Petersburg
G.I. Alferov
Vice-President of RAS

- Chernogolovka
- Daghestan
- Kabardino-Balkariya
- Kazan
- Karelian
- Kola
- Puschino
- Samara
- Saratov
- Troitsk
- Ufa
- Vladikavkaz
- South

Short Profile of St. Petersburg Institute for Informatics and Automation (SPIIRAS)

- Founded in 1978
- The Russian Academy of Sciences Institute operating in the North-West of Russia in Information Technologies
- Personnel – 203
- SPIIRAS is a competence center in the area of advanced information technologies
- Experienced in collaboration with EC Countries

SPIIRAS Structure



SPIIRAS General Research Directions

- Development of Information and Control Systems for Real Time Signal Processing
- Fundamentals of Information Processes in Complex (Socio-, Eco-, Bio-,Geo-, etc) Systems
- Theoretic Basics in Developing Information Technologies for Research Automation, Control, Manufacturing, and Intelligent Systems

Research Directions of Computer Security Research Group

- **Computer security**, including security policy management, access control, authentication, network security analysis, intrusion detection, firewalls, deception systems, virus protection, verification of security systems, modeling, simulation and visualization technologies for counteraction to cyber terrorism;
- **Artificial intelligence**, including multi-agent frameworks and systems, soft and evolutionary computing, machine learning, data mining, data and information fusion;
- **Telecommunication**, including decision making and planning for telecommunication systems.

2. International collaboration, projects and accomplishments of Computer Security Research Group

International Projects

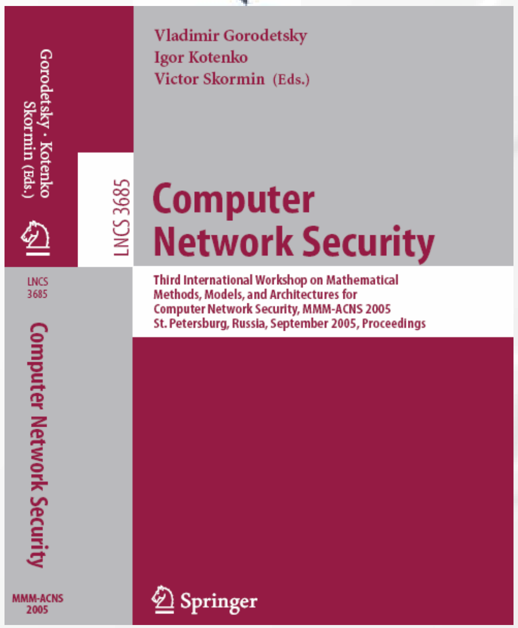
- Air Force Research Laboratory/ Information Directorate (European Office of Aerospace Research and Development) (1999-2003 - 3 computer security projects)
  - "Agent-Based Model of Information Security System: Architecture and Formal Framework for Coordinated Intelligent Agents Behavior Specification" (1999-2001)
  - "Formal Grammar-Based Approach and Tool for Simulation Attacks against Computer Network" (2001-2003)
  - "Mathematical Foundations, Architecture and Principles of Implementation of Multi-Agent Learning Components for Attack Detection Computer Networks" (2001-2003)
- INTEL – "Network traffic preprocessing algorithms" (2004-2005)
- Fraunhofer First (Germany) – "Intrusion detection learning systems" (MIND) (2004-2006)
- FP6 (EU Project) – "Security policy specification, checking and deployment" (POSITIF) (2004-2007)
- …
- FP6 (EU Project) – "Remote EnTrusting by RUn-time Software auThentication" (RE-TRUST) (2006-2009)

Russian Grants and Projects

- Government Budget Projects:
  - Models and methods of developing secure computer systems (2006-2008)
- Grants of Russian Foundation for Basic Research:
  - Mathematical models of information security assurance in computer networks based on MAS technology and its experimental evaluation (2001-2003)
  - Modeling and simulation of cyber warfare (2004-2006)
- Projects from Department of Information Technology and Computer Systems of the Russian Academy of Sciences:
  - Agent-based stochastic modeling and simulation of adversarial competition of teams in the Internet environment (2003-2004)
  - Mathematical models of active audit of computer network vulnerabilities, intrusion detection and response: Multi-agent approach (2003-2006)
- Projects from Government Institutions:
  - Models and prototypes of honeynets (2003-2004)
  - Monitoring of user activity in computer networks (2004)
  - …

International Computer Security Conferences Organized

- **Mathematical methods, models and architectures for computer network security (MMM-ACNS)**: 2001, 2003, 2005, 2007



The objectives of MMM-ACNS Workshops are to bring together leading researchers from academia and governmental organizations as well as practitioners in the area of computer networks and information security, facilitating personal interactions and discussions on various aspects of information technologies in conjunction with computer network and information security problems arising in large-scale computer networks engaged in information storing, transmitting, and processing.

Recent accomplishments (results) in 2005-2006 (1)

- **The theoretical basis, algorithms and software implementation of agent-oriented modeling and simulation of antagonistic counteraction of malefactors and computer network security components.**
  - The **principles of construction, the structure and a fragment of distributed ontology-based knowledge base** for modeling and simulation of protection mechanisms in an antagonistic environment.
  - The **formal models** of agents–malefactors, security agents and computer network under defense.
  - The **software environment** based on imitation of computer attacks and protection mechanisms at a network packets level using OMNeT ++ INET Framework.
  - The **attack and security agents** as compound modules containing simple modules, responsible for functioning of various network protocols, and agent kernel that controls these modules.
  - Different **experiments** with this environment were fulfilled on an example of simulation of DDoS attacks and particular defense mechanisms. These experiments included the investigation of attack scenarios and protection mechanisms for the networks with different structures and security policies.
  - The **received results are directed** on investigation of various aspects of antagonistic interactions of agent teams in the Internet and development of recommendations on design and implementation of advanced security systems.

Recent accomplishments (results) in 2005-2006 (2)

- **The theoretical basis and operation algorithms of deception systems (DS).**
  - These systems represent hardware-software tools for information protection that are based on the technology of “traps” and false targets. In particular, we developed **the requirements to DS, the generalized architecture of multi-agent DS, the generalized models and algorithms** of disguised counteraction to remote non-authorized access to information resources, including the models of malefactor detection and readdressing of non-authorized request to false components, determining the malefactor plan (strategy), generating a plan of false components operation, etc.
  - The offered approach is based on simulation of information systems components and on using **three levels of malefactor deception**: (1) a network segment level – the whole network segment is emulated; (2) a host level – among working servers the bait-host is used; (3) a services and applications level –the programs emulating services and applications are applied on servers.
  - The **deception software system** was implemented.
  - We fulfilled a set of **experiments** on investigating basic deception functions at realization of different attacks. These experiments are executed on several different scenarios determined according to various attack types.

Recent accomplishments (results) in 2005-2006 (3)

- **Models, techniques and prototypes for active security analysis of computer networks**.
  - The approach is based on **automatic generation and fulfillment of distributed attack** scripts taking into account a variety of goals and knowledge levels of malefactors, and intended for implementation at various stages of computer network life cycle including design and exploitation stages.
  - The offered approach is based on **application of a set of models** (including models of malefactor, attack scripts generation, security level evaluation, computer network, etc.) using expert knowledge.
  - Functioning of security analysis system based on the approach suggested is resulted in **determined vulnerabilities, the traces (graphs) of possible multistage attacks, "bottlenecks" (main "holes")** in a computer network on which these attacks are based, and also **various security metrics** which can be used for evaluating a security level of a computer network and its components, and also for comparisons of various network configurations and security policies.
  - These results provide the **development of justified recommendations** on elimination of revealed "bottlenecks" and on amplification of system security.

Recent accomplishments (results) in 2005-2006 (4)

- The **generalized architecture, particular models and prototypes of components for verifying security policies** of computer networks were analyzed and developed.
  - The **mechanisms for operation with policies** of three levels were offered:
    - (1) the top level, approximated to the user requirement language,
    - (2) the intermediate level, classifying rules according to several policy categories, and
    - (3) the bottom level, describing policies in the format of Common Information Model (CIM).
  - We developed and implemented the **research prototypes of the verification manager that handles the process of verification, and different verification modules**:
    - (1) based on Event Calculus,
    - (2) based on Model Checking,
    - (3) Specialized modules.

Some Recent research papers published or accepted for publishing in 2006 (1)

- *Kotenko I., Stepashkin M., Ulanov A.* Agent-based modeling and simulation of malefactors' attacks against computer networks. Security and Embedded Systems. D.N.Serpanos, R.Giladi (Eds.). IOS Press. 2006.
- *Kotenko I.V., Tishkov A.V., Chervatuk O.V.* Architecture and Models for Security Policy Verification. Mathematics and Security of Information Technologies. Amsterdam, 2006.
- *Kotenko I., Ulanov A.* Agent-based Simulation of Distributed Defense against Computer Network Attacks. Proceedings of 20th European Conference on Modelling and Simulation (ECMS 2006). Bonn. Germany. May 28th - 31st, 2006. P.560-565.
- *Kotenko I., Ulanov A.* Antagonistic Agents in the Internet: Computer Network Warfare Simulation. The 9th International Conference on Information Fusion. Florence (Italy), 10-13 July, 2006.
- *Kotenko I., Stepashkin M.* Network Security Evaluation based on Simulation of Malefactor's Behavior. SECRYPT - International Conference on Security and Cryptography. International Joint Conference on e-Business and Telecommunications. ICETE 2006. Setubal, Portugal. 7-10 August 2006.

Some Recent research papers published or accepted for publishing in 2006 (2)

- *Kotenko I., Ulanov A.* Agent-based modeling and simulation of network softbots' competition. The Joint Conference on Knowledge-Based Software Engineering (JCKBSE'06). Tallinn, Estonia. August 28-31. 2006.
- *Kotenko I., Ulanov A.* Simulation of Internet DDoS Attacks and Defense. 9th Information Security Conference. ISC 2006. Samos, Greece. August 30 - September 2, 2006. Lecture Notes in Computer Science, Vol. 4176, 2006.
- *Kotenko I., Stepashkin M.* Analyzing network security using malefactor action graphs. International Journal of Computer Science & Network Security, 2006.
- *Kotenko I., Stepashkin M.* Attack Graph based Evaluation of Network Security. The 10th IFIP Conference on Communications and Multimedia Security. CMS'2006. Heraklion, Greece. 19 - 21 October 2006. Lecture Notes in Computer Science, Vol. 4237, 2006.
- *Kotenko I., Ulanov A.* Agent Teams in Cyberspace: Security Guards in the Global Internet // International Conference on CYBERWORLDS. CW2006. Lausanne, Switzerland, November 28-30, 2006. Proceedings. IEEE Computer Society, 2006.
- …

3. Tasks in RE-TRUST

Main Tasks in RE-TRUST

Task T4.1 – Trust analysis of SW-based method

Task T4.5 – Remote entrusting and Internet secure protocols

- **T4.5.1 – Analysis of integration of remote entrusting with existing Internet security protocols**
- **T4.5.2 – Integration and analysis of secure protocols to support remote entrusting methods.**

4. Computer Security Research Group Research: General view

Examples of related developed solutions (mainly in network security analysis):

- **Modeling and simulation of computer attacks**
- **Modeling and simulation of cyberwarfare**
- **Security analysis of computer networks**
- **Intrusion detection**
- **Deception systems, honeynets**
- **Security policy specification and checking**
- **Security protocols analysis**

Security Evaluation Areas

- **Impact assessment** for determining how security measures affect system and application properties (performance, reliability, etc.) [D.Nicol, S.Smith, M.Zhao-04 ; S.Kent, C.Lynn, K.Seo-00 (Secure BGP); M.Zhao, S.Smith, D.Nicol-05; etc.]
- **Emulation**, in which real and virtual worlds are combined to study the interaction between malware and systems, and probe for new system weaknesses [G.Bakos, V.Berk-02 (Worm activity by metering ICMP); M. Liljenstam et al-03 (Simulating worm traffic); etc.]
- **Cyber attack exercises** and training scenarios [M. Liljenstam et al-05 (RINSE); B. Brown et al-03; etc.]
- **Risk assessment** based on known vulnerabilities, exploits, attack capabilities, and system configuration [R. Ortalo, Y.Deswarte, M.Kaaniche-99; Sheyner et al-02; V.Gorodetski, I.Kotenko-02 (Attack Simulator); B.Madam, K.Goseva-Popstojanova-02; etc.]
- …

Works on Security Evaluation

- Methodology and software tools for testing IDSs ([Puketza *et al*-96], [Puketza *et al*-97], [Debar *et al*-98], [Alessandri *et al*-01], [McHugh-00]);
- Evaluations of IDSs of MIT ([Lippmann *et al*-98, 00, 02]);
- Real-time test bed of AFRL [Durst *et al*-00];
- Dependability models for evaluation security [Nicol *et al*-04];
- Penetration testing of formal models of networks for estimating security metrics [Sheyner *et al*-02];
- Model checking for analysis of network vulnerabilities [Ritchey, Ammann-00];
- Global metrics for analyzing the effects of complex network faults and attacks [Hariri *et al*-03];
- Natural-deduction for automatic generation and analysis of attacks against IDS [Rubin *et al*-04];
- Knowledge-based network risk assessment [Shepard *et al*-05], etc.

Works directly coupled with Attack Modeling and Simulation

- Using Colored Petri Nets [Kumar *et al*-94];
- State transition analysis technique [Iglun *et al*-95], [Kemmerer *et al*-98];
- Conceptual models of computer penetration ([Cohen-99],[Stewart-99]);
- Descriptive models of attackers [Yuill *et al*-00];
- "Tree"-based models of attacks ([Moore *et al*-01], [Dawkins *et al*-02]);
- Modeling survivability of networked systems [Moitra *et al*-01];
- Object-oriented Discrete Event Simulation [Chi *et al*-01];
- Situation calculus and goal-directed procedure invocation [Goldman-02];
- Using and building attack graphs for vulnerability analysis ([Swiler *et al*-01], [Ortalo *et al*-01], [Sheyner *et al*-02], [Jha *et al*-02]);
- Game-theoretic models [Lye and Wing-03];
- Multi-stage attack analysis [Dawkins, Hale-04];
- Modeling and inference of attacker intent, objectives, and strategies [Liu, Zang-05]; etc.

Security Analysis

❶ Model system

❷ Model adversary

❸ Identify security properties

❹ See if properties preserved under attack

- Result
  - Under given assumptions about system, no attack of a certain form will destroy specified properties
  - There is no “absolute” security

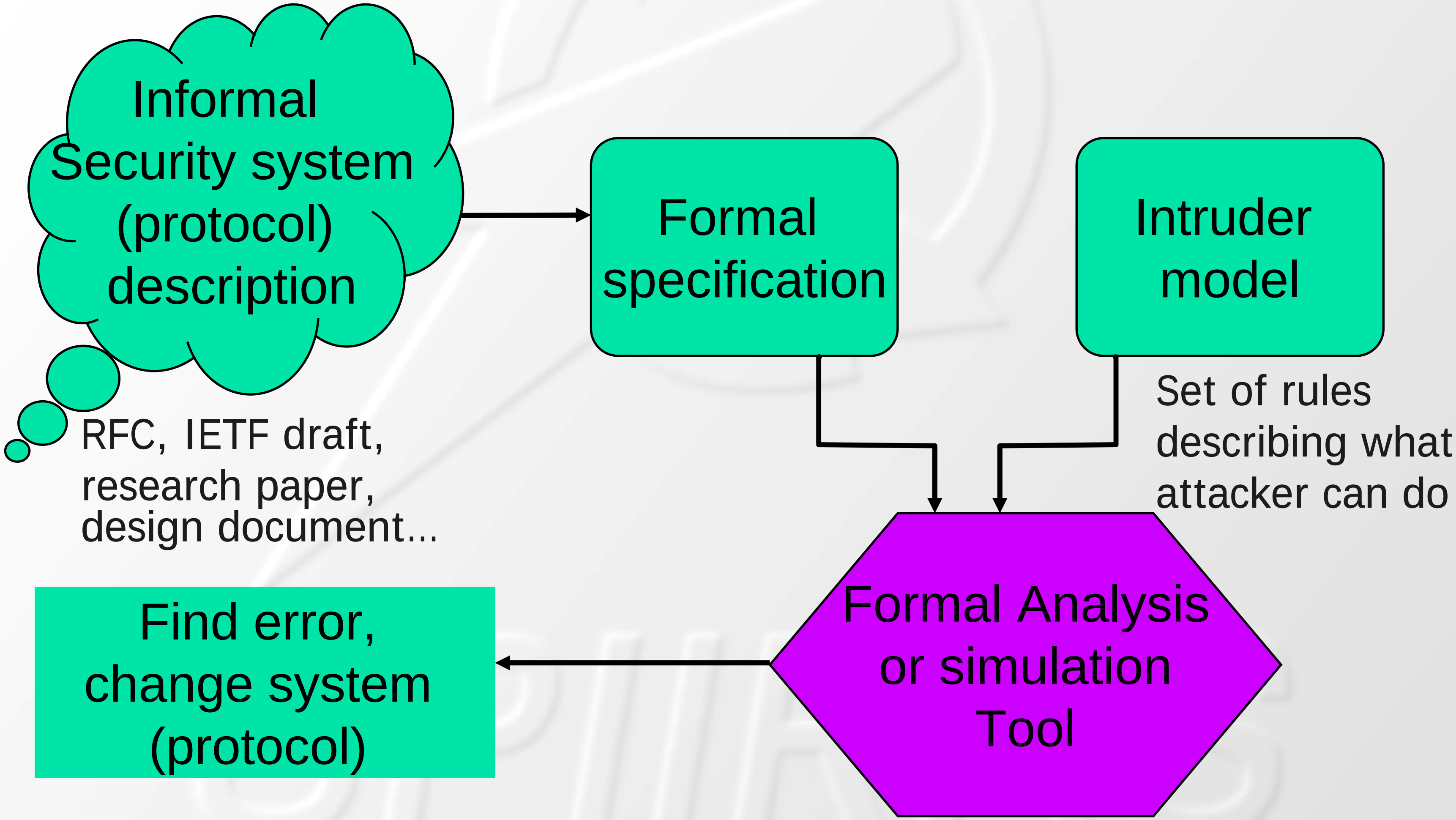
/Vitaly Shmatikov/

Fundamental Tradeoff

- Formal models are abstract and greatly simplified
  - Components modeled as finite-state machines
  - Security functions modeled as abstract data types
  - Security property stated as unreachability of "bad" state
- Formal models are tractable…
  - Lots of verification methods, many automated
- …but not necessarily sound
  - Proofs in the abstract model are subject to simplifying assumptions which ignore some of attacker's capabilities
- Attack in the formal model implies actual attack

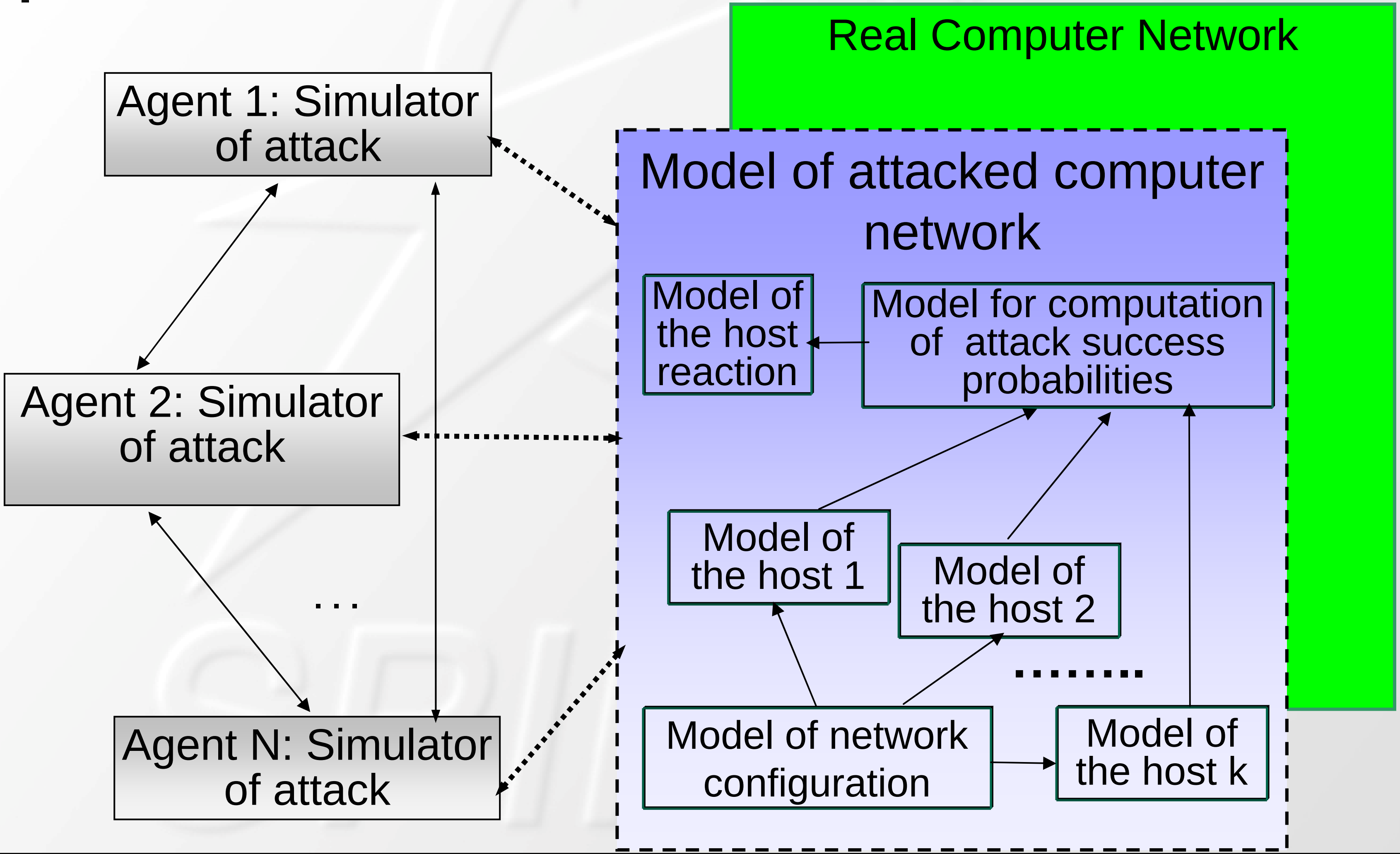
/Vitaly Shmatikov/

Explicit Intruder Method



- Modeling and simulation of computer attacks

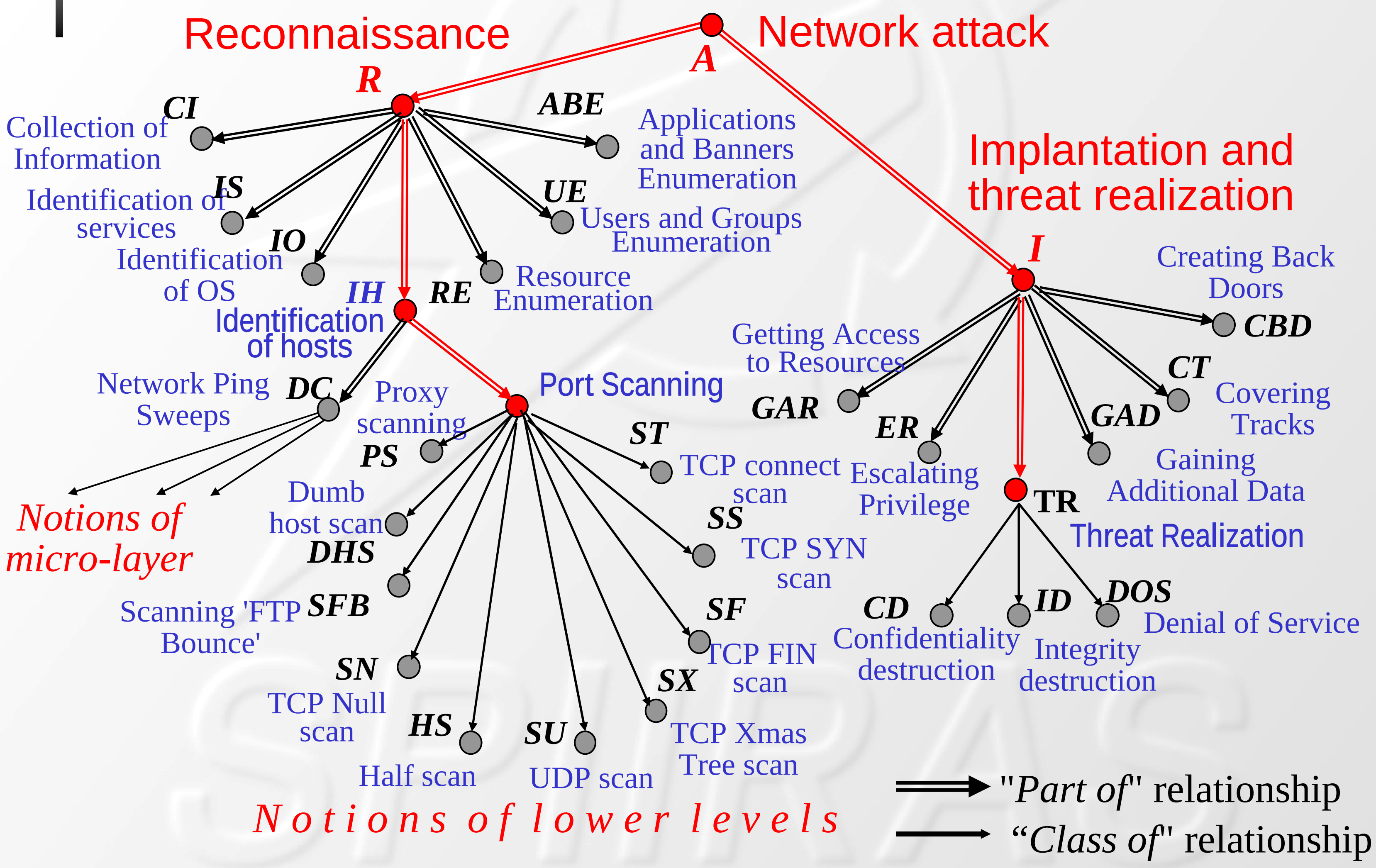
Conceptual simulation scheme



Basic Components of Attack Model

1. **Ontology of the Problem** "*Attacks against Computer Network*" : structure of the basic malefactors' intentions and actions.
2. Basic **malefactors' intentions** and **attack task specification**.
3. **Formal grammar-based framework** for specification of attack development.
4. Formal scenarios of a **representative multitude of attacks** and their development in time.
5. Formal model of the **attacked computer network**.
6. Model of **interaction** of malefactor's activity and victim computer network.

Partial Ontology of the Domain "*Attacks against Computer Network*" (Macro-levels)



Basic malefactors' intentions

Intention-centric approach to the specification of malefactor's activity: basic notions of the domain correspond to the malefactor intentions and all other notions are structured according to the structure of intentions.

List of Basic Classes of High-level Malefactor's Intentions

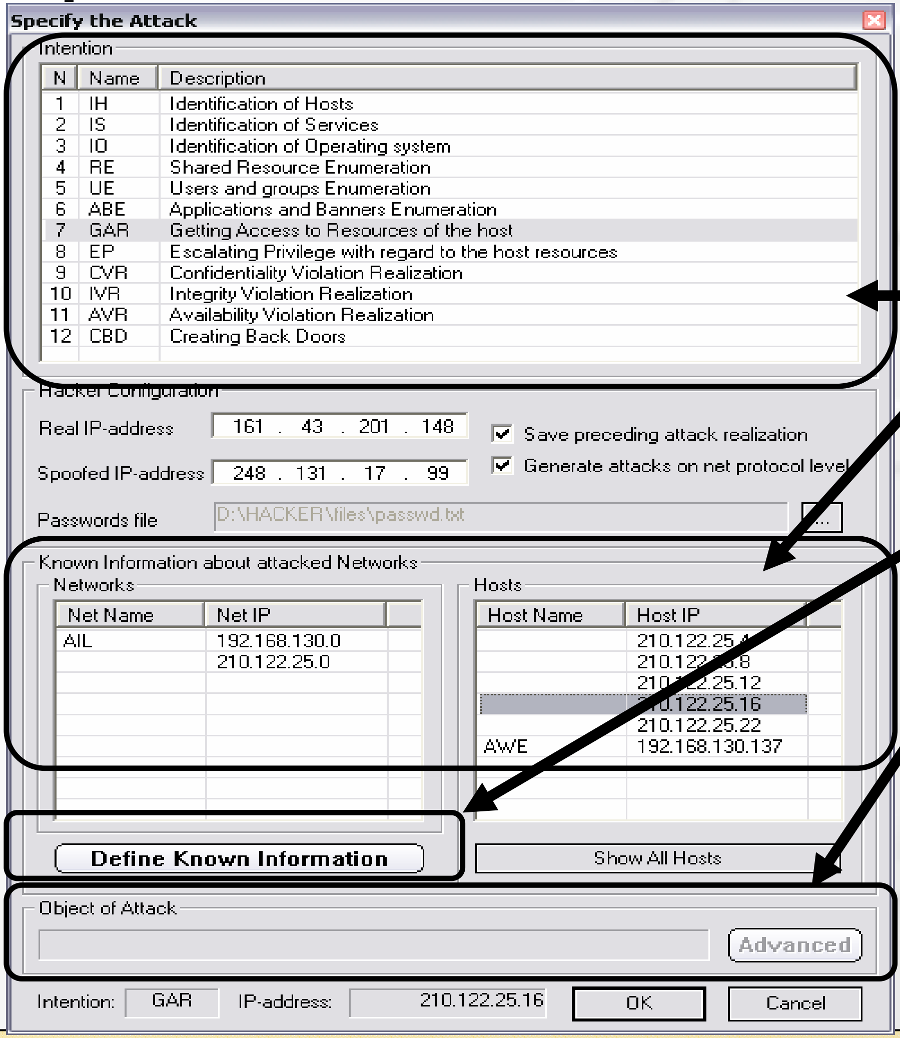
R – Reconnaissance:

- *IH* – *Identification* of the running *Hosts*
- *IS* – *Identification* of the host *Services*
- *IO* – *Identification* of the host *Operating* system
- *CI* – *Collection* of additional *Information*
- *RE* – shared *Resource Enumeration*
- *UE* – *Users* and groups *Enumeration*
- *ABE* – *Applications* and *Banners Enumeration*

I – Implantation and threat realization:

- *GAR* – *Getting Access* to *Resources* of the host
- *EP* – *Escalating Privilege* with regard to the host resources
- *GAD* – *Gaining Additional Data* needed for further threat realization
- *TR* – *Threat Realization*
  - *CD* – *Confidentiality Destruction*
  - *ID* – *Integrity Destruction*
  - *DOS* – Violation of resource availability (*Denial of Service*)
- *CT* – *Covering Tracks*
- *CBD* – *Creating Back Doors*

Attack task specification



Main elements of attack specification:

1) Malefactor's intention (1-12);
2) Address of the attacked host or network;
3) Available information about attacked host;
4) Attack object (file name, user account, resource, etc.);

Formal framework for specification of attacks

Formal grammar: $G_i = <V_N, V_T, S, P, A>$,

where G_i – formal grammar name (it coincides with the name of attack and the name of its axiom);

V_N – the set of non-terminal symbols; V_T – the set of terminal symbols; $S \in V_N$ – formal grammar axiom;

P – the set of productions which look like follows:

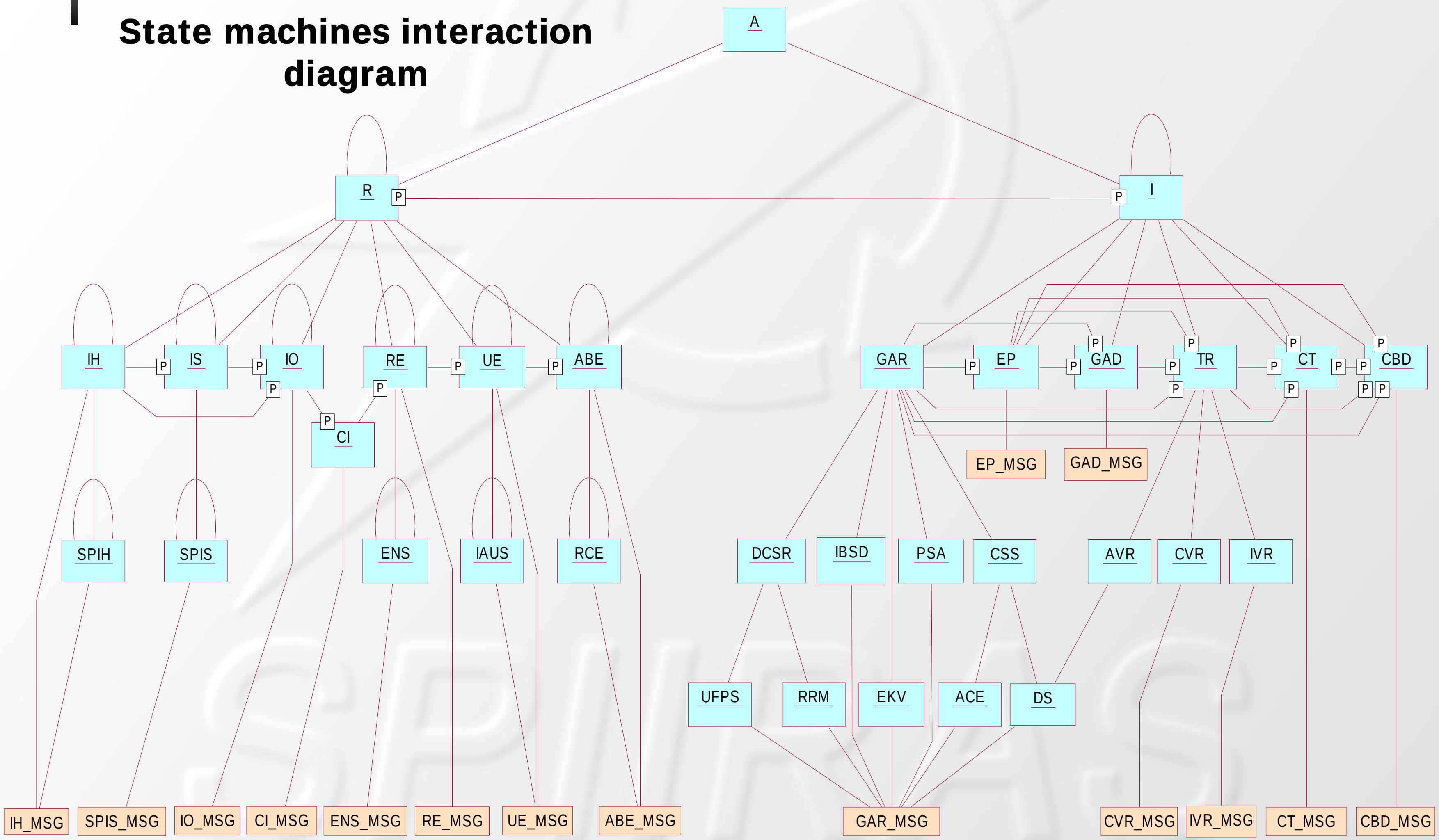
$$(U)\ X \rightarrow \alpha\ (Prob),$$

where $X \in V_N$, $\alpha \in (V_T \cup V_N)^*$, U – precondition of the production application; $Prob$ – probability of the production application;

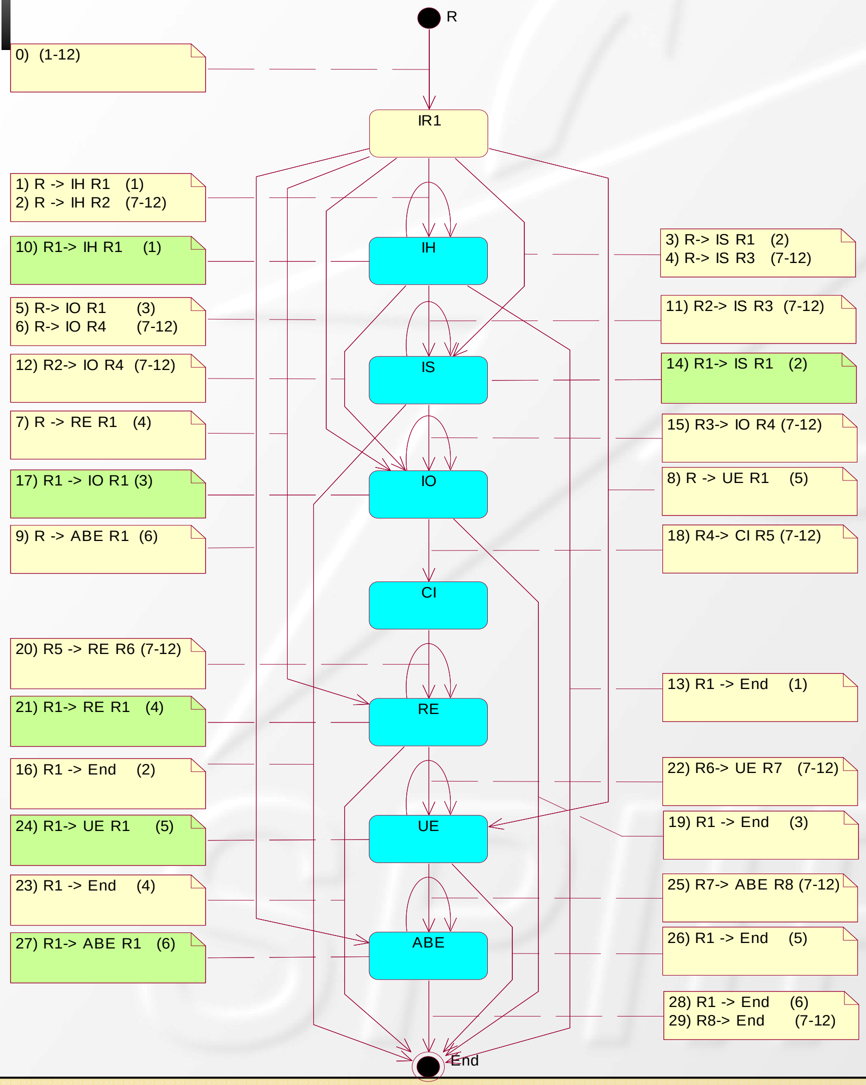
A – the set of attributes and their dependencies (functions having attributes as variables).

Implementation Issue: State Machine-based Representation of Attack Generation

State machines interaction diagram



Implementation Issue: State Machine-based Representation of Attack Generation



"Reconnaissance"
Attack Generation

IR1 – Intermediate state

IH – Identification of running Hosts

IS – Identification of Services

IO – Identification of OS

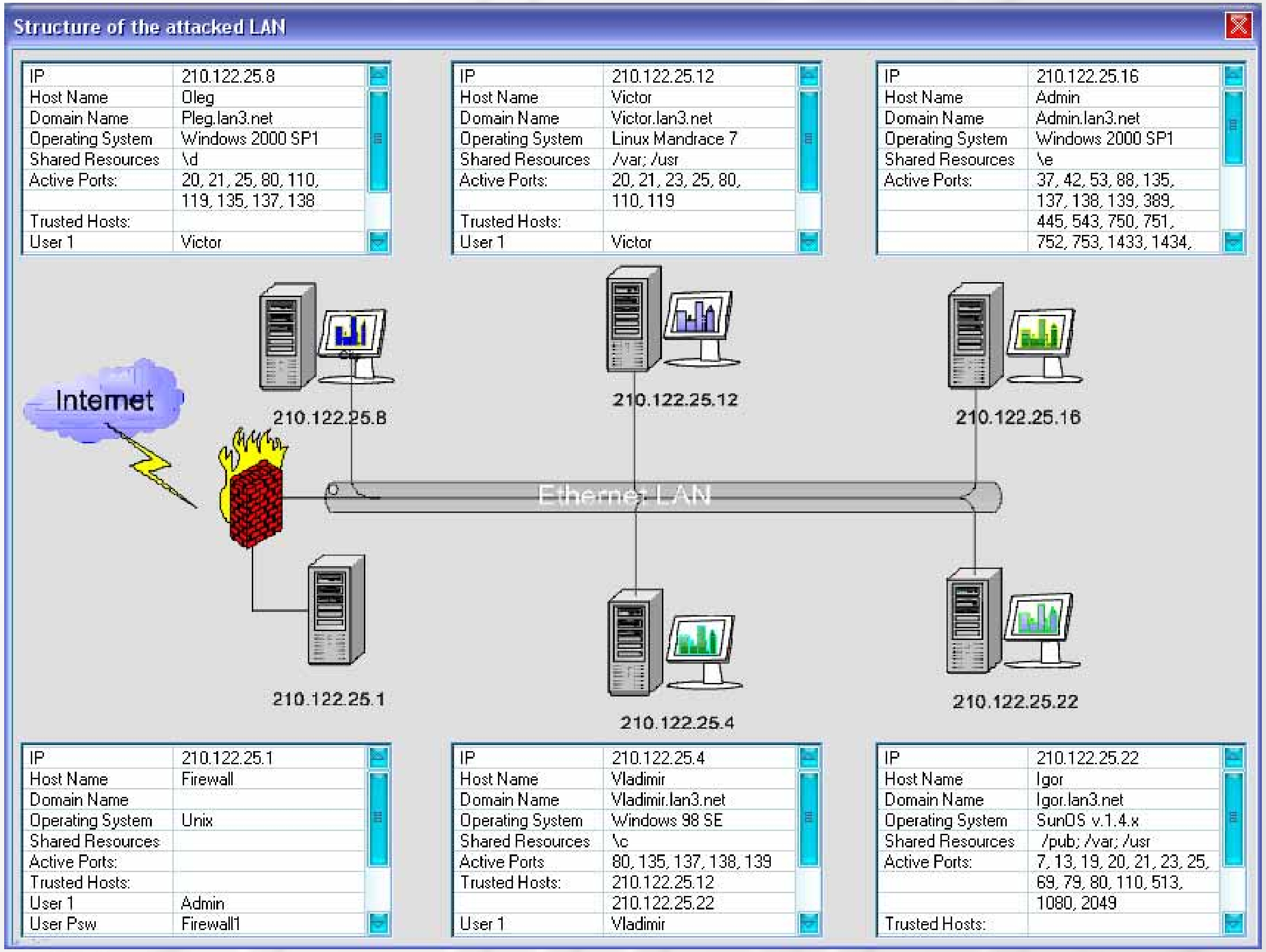
CI – Collection of Information

RE – Resource Enumeration

UE – Users and Groups Enumeration

ABE – Applications and Banners Enumeration

User Interface with Network Model



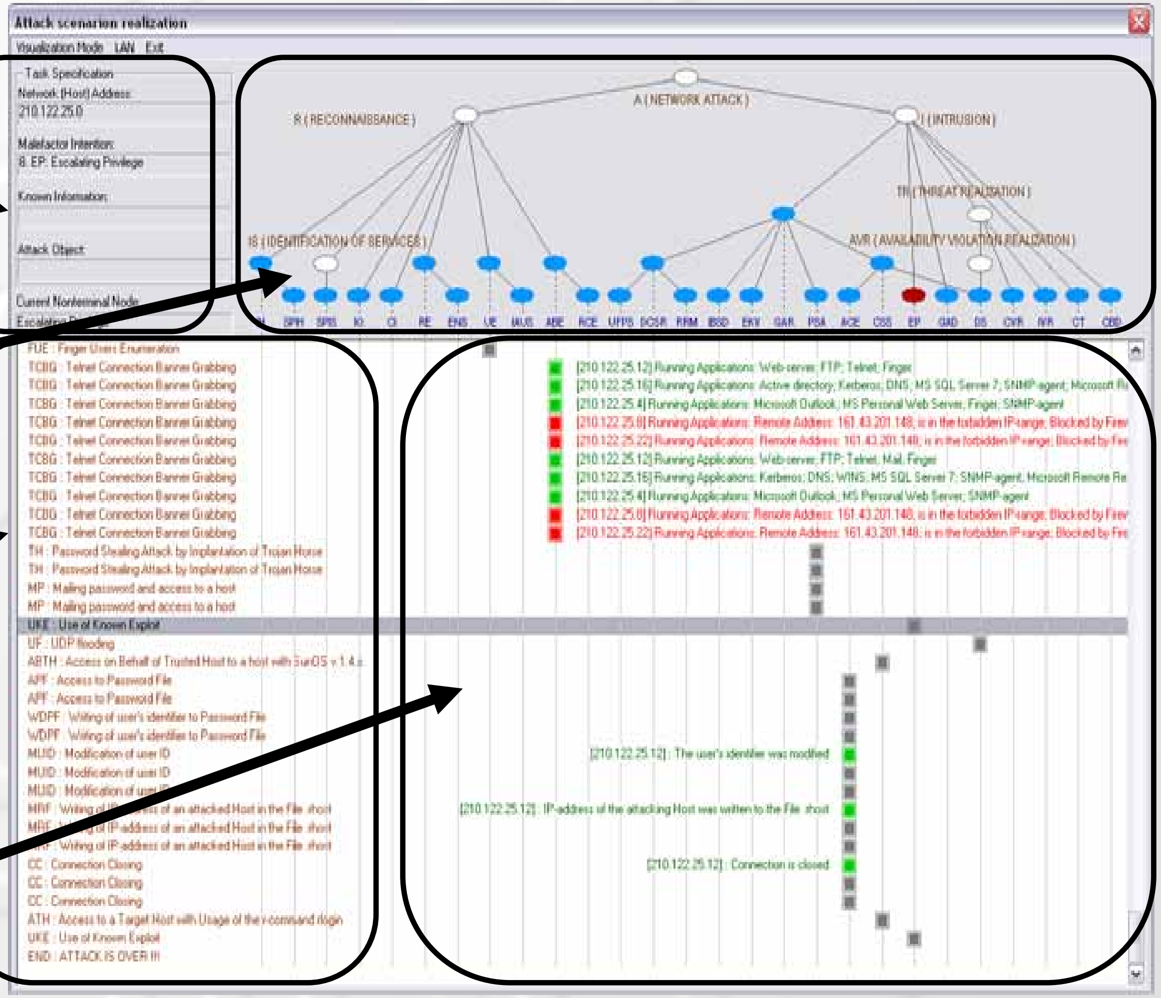
Visualization of an attack development on macro-level

Attack task specification

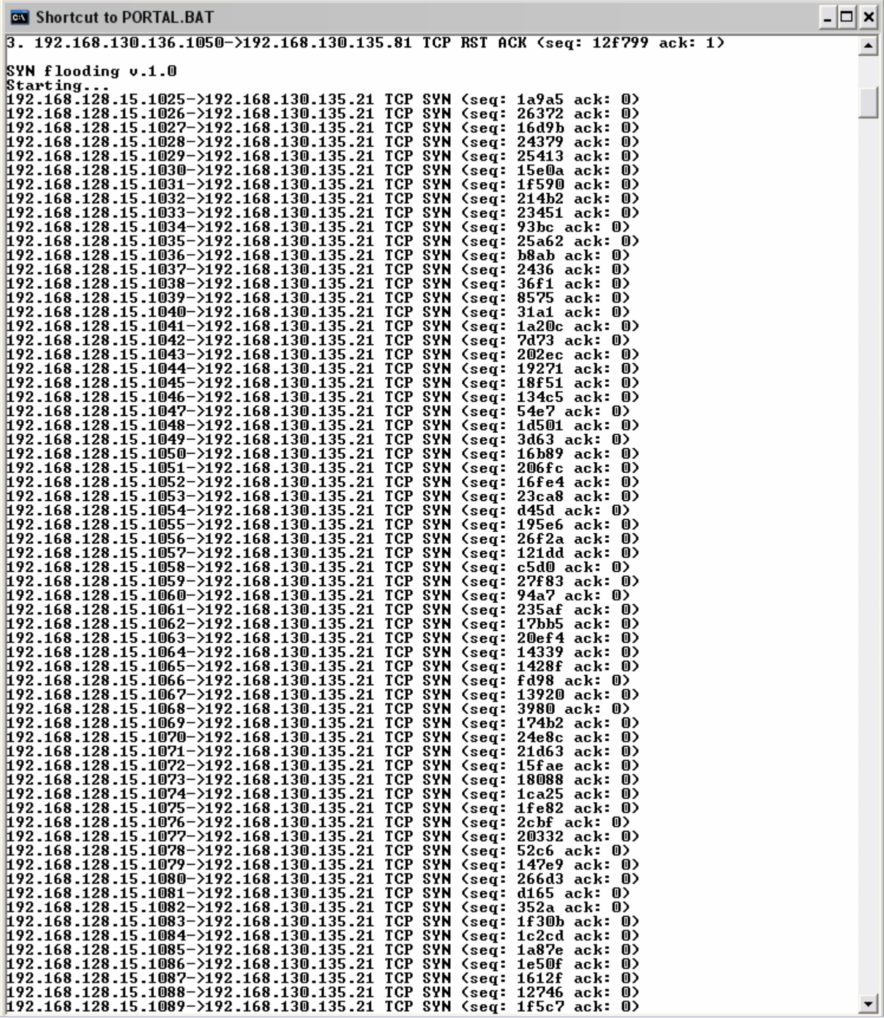
Attack generation tree

Malefactor' s actions

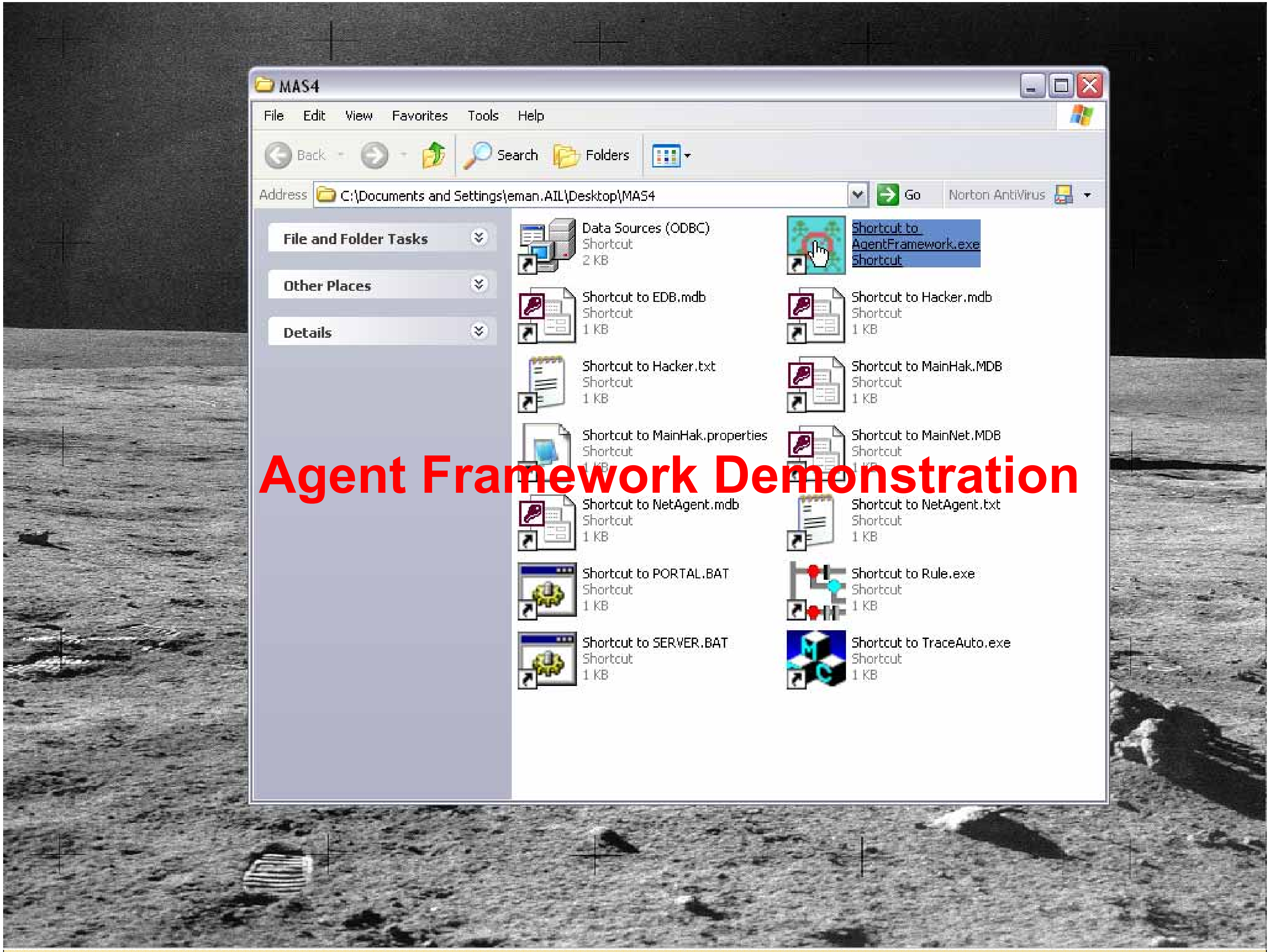
A tag of success (failure) and and data obtained from an attacked host (a host response)



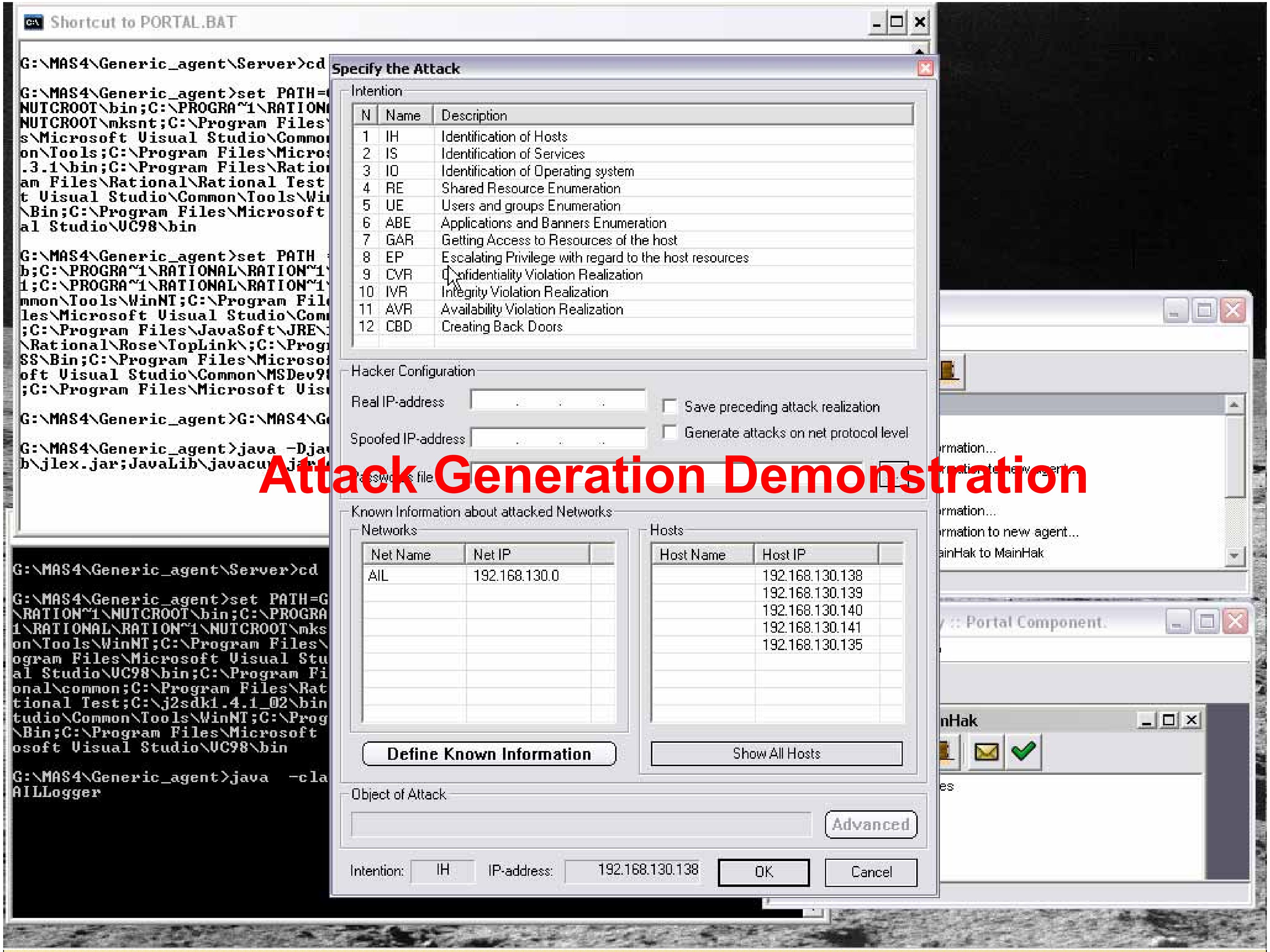
On-line Visualization of an Attack Development on Micro-Level



Agent Framework Demonstration



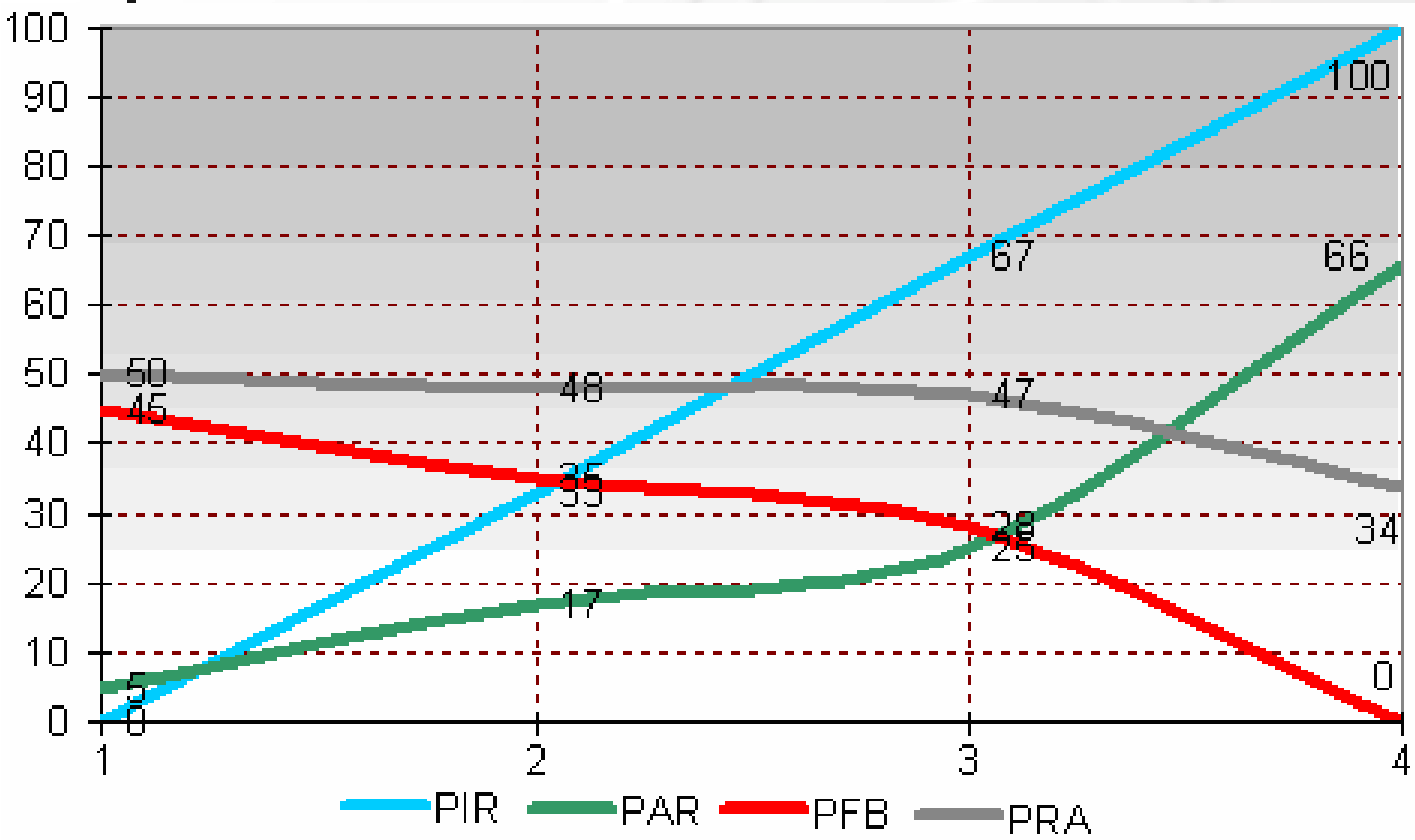
Attack Generation Demonstration



Parameters of attack realization outcome

- **NS (Number of attack Steps) –** number of terminal level attack actions;
- **PIR (Percentage of Intention Realization) –** percentage of the hacker's intentions realized successfully (for "Reconnaissance" it is a percentage of objects about which the information has been received; for "Implantation and threat realization" it is a percentage of successful realizations of the common attack goal on all runs);
- **PAR Percentage of Attack actions Realization –** percentage of "positive" messages (responses) of the Network Agent on attack actions (the "positive" messages are designated in attack visualization window by green lines);
- **PFB (Percentage of Firewall Blockage) –** percentage of attack actions blockage by firewall (red lines in attack visualization window);
- **PRA (Percentage of Reply Absence)** - percentage of "negative" messages (responses) of the Network Agent on attack actions (gray lines in attack visualization window) .

Example of experiment results for intention "Gaining Access to host Resources"



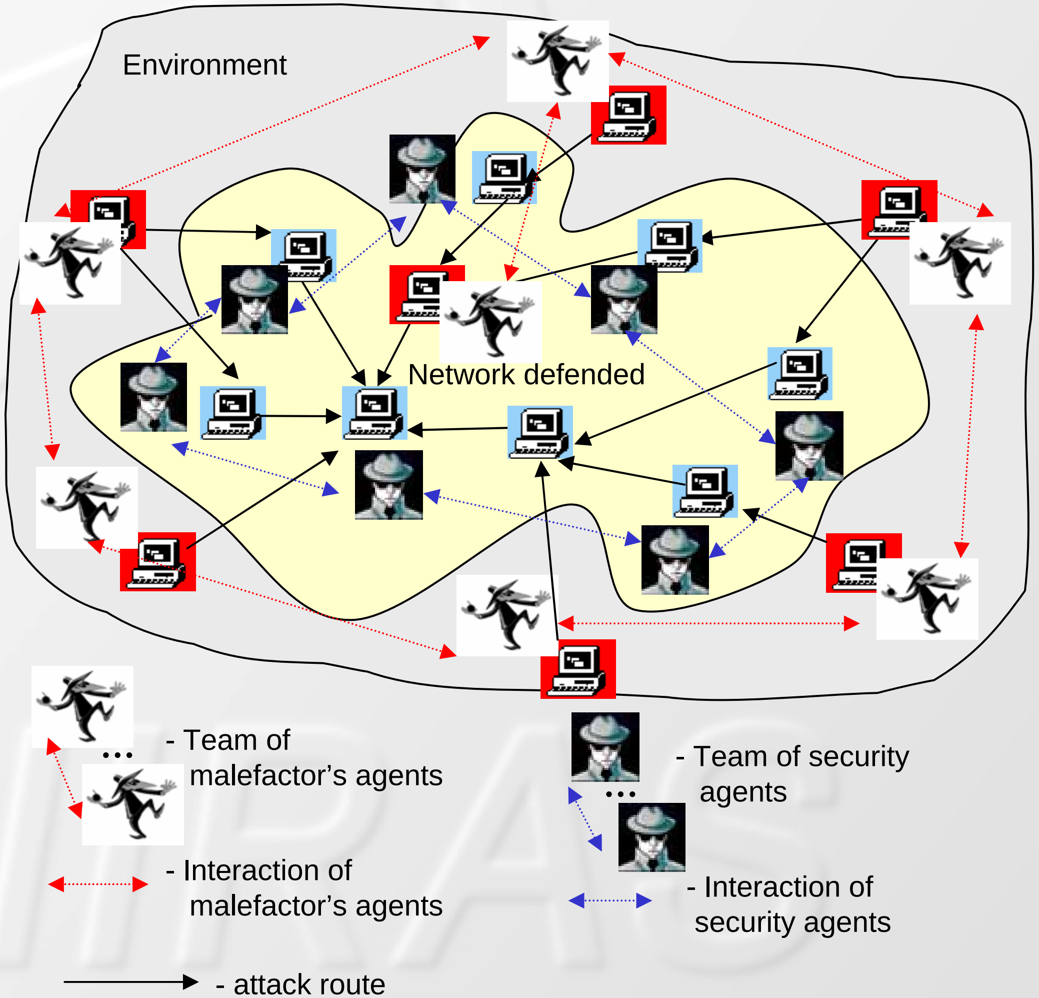
Configurations of firewalls: 1 - Both Net & Personal firewalls are active; 2 - Only Net firewall is active; 3 - Only Personal firewall is active; 4 - None of firewalls is active

- **Modeling and simulation of cyberwarfare**
  (between malefactors' teams and securty teams)

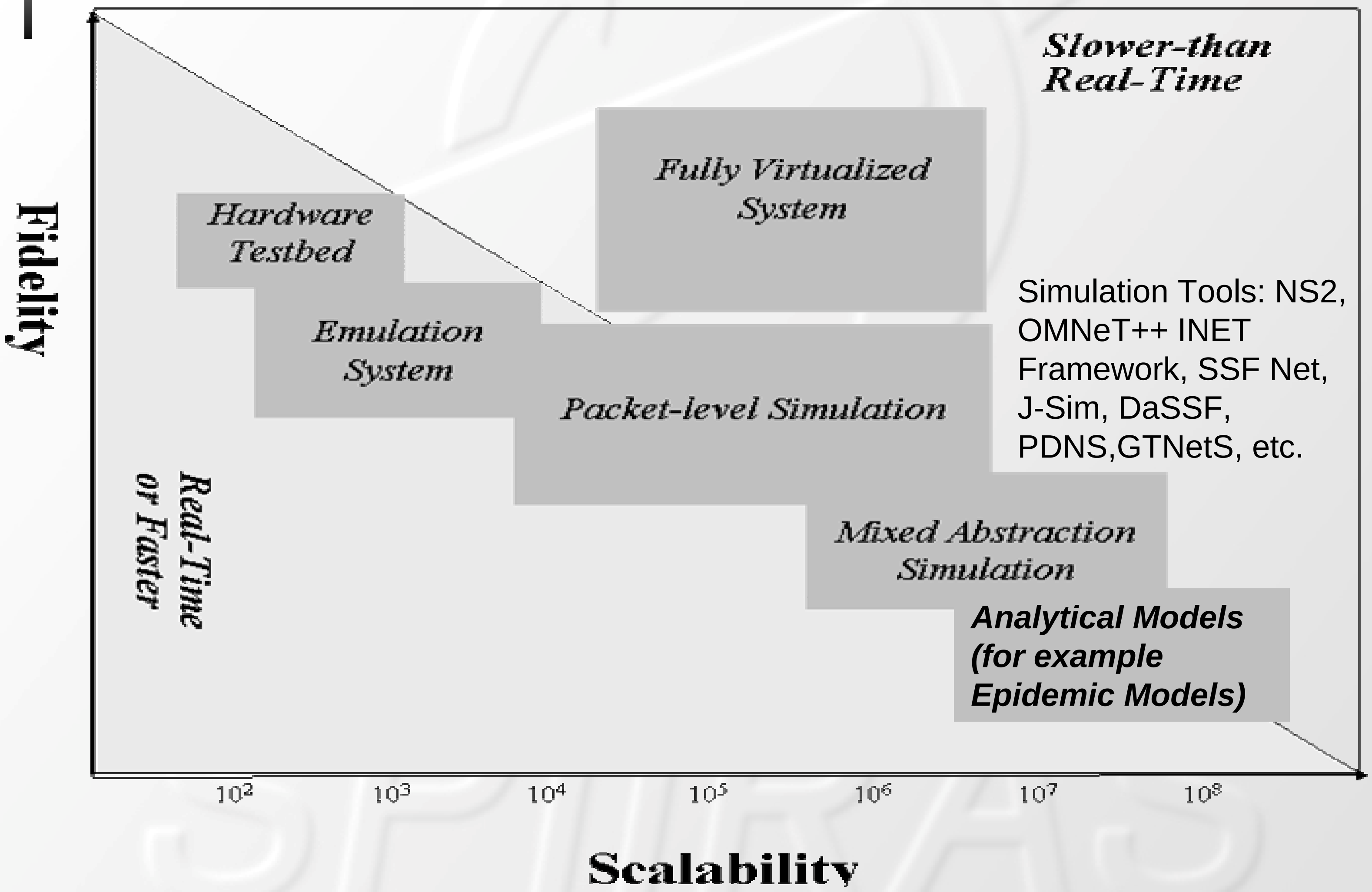
Research objectives

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***.

Interaction of team of malefactors and computer network assurance system components



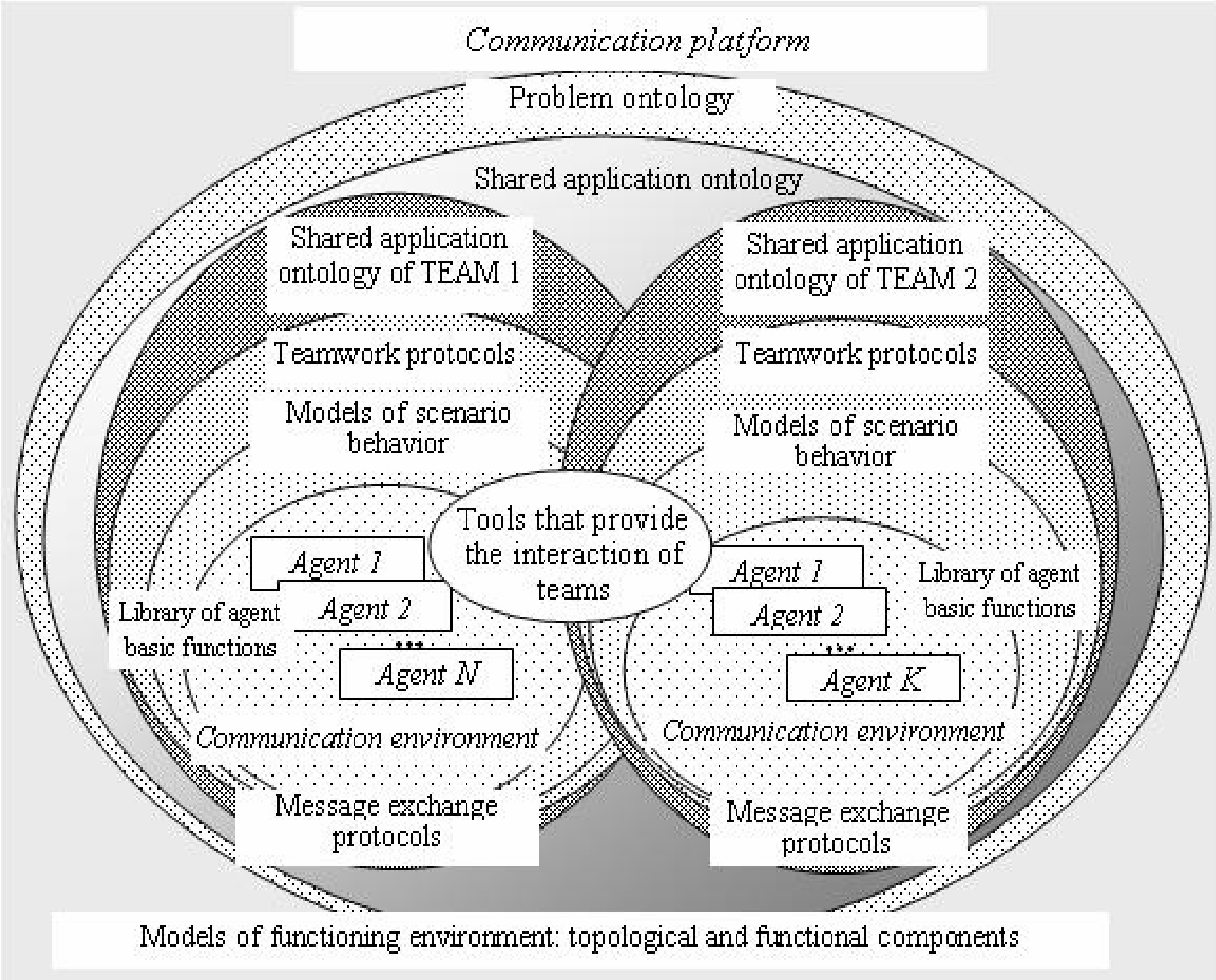
Range of Modeling Alternatives



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

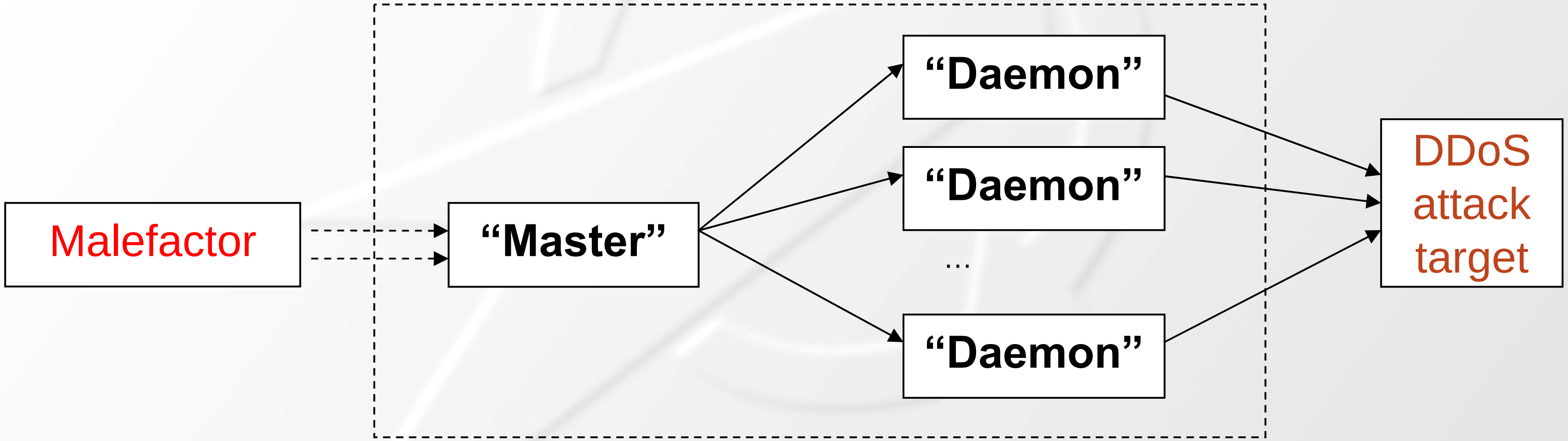
Source: [K.Perumalla, S.Sundaragopalan-04]

Abstract model of team interaction

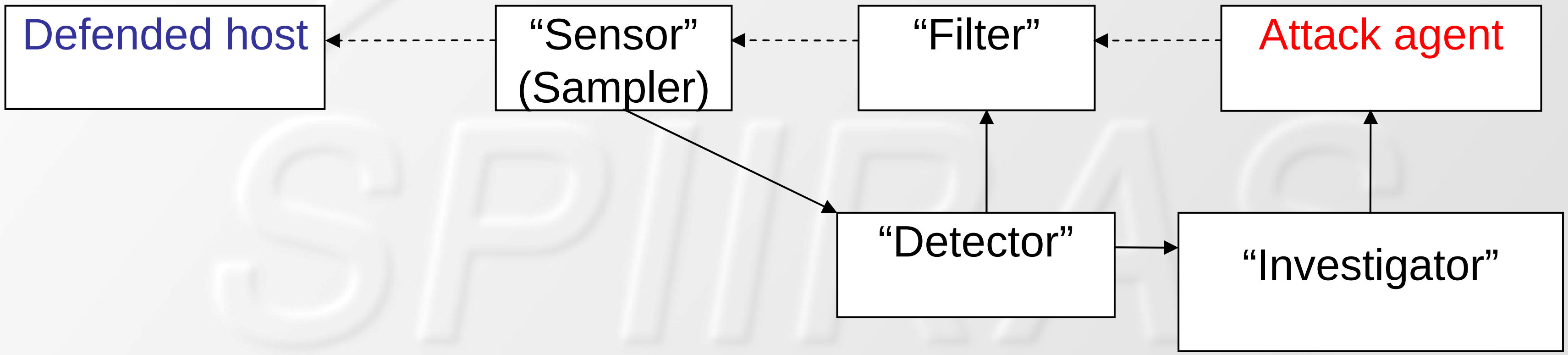


Structure of Teams

Structure of attack team



Structure of defense team



Main Classes of Attack and Defense Parameters. Parameters of Defense Efficiency

Attack module

- *Victim type*
- *Attack type*
- *Impact on the victim*
- *Attack rate dynamics*
- *Persistent of agent set*
- *Possibility of exposure*
- *Source address validity*
- *Degree of automation*

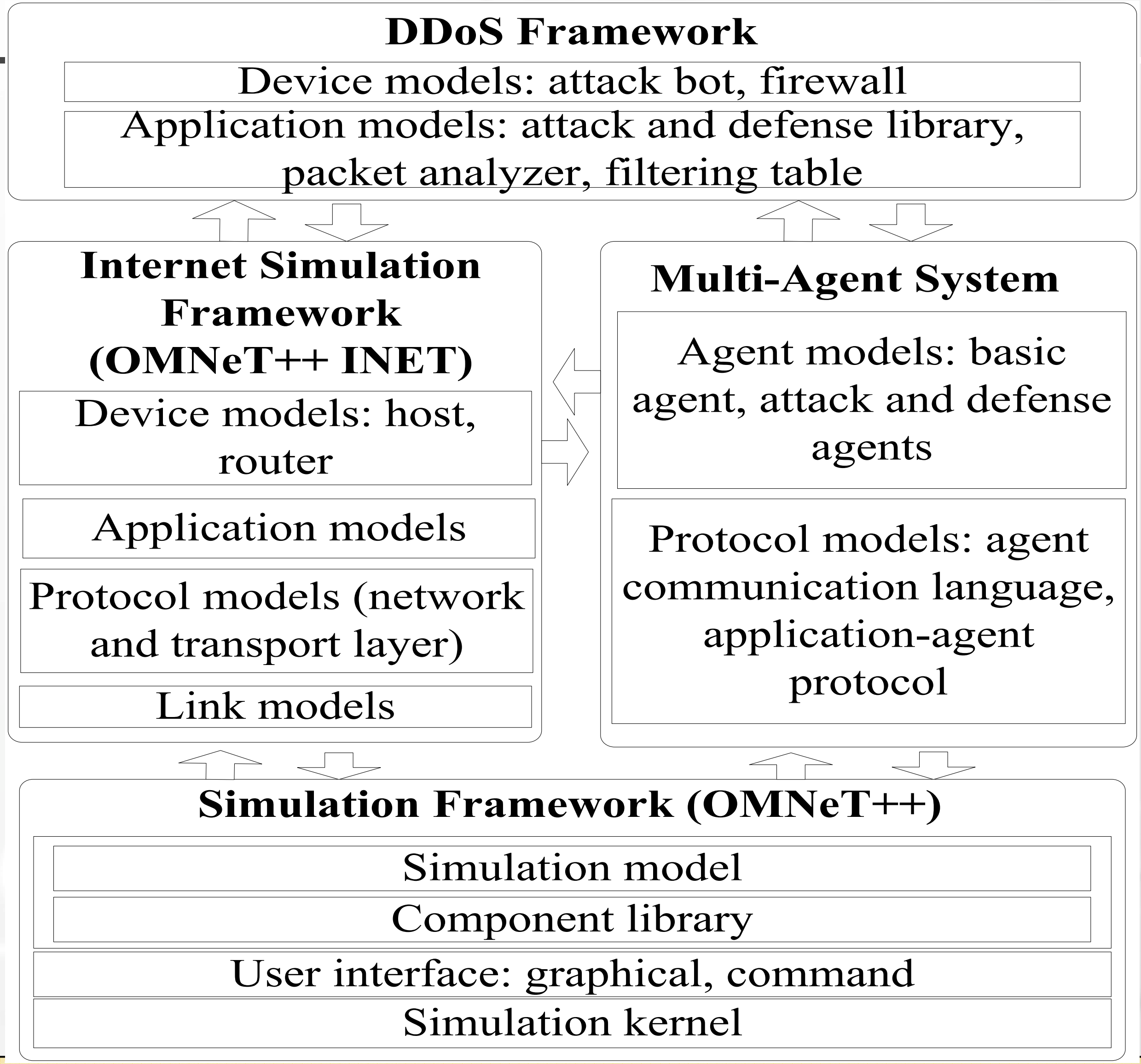
Defense module

- *Deployment location*
- *Mechanism of cooperation*
- *Covered defense stages*
- *Attack detection technique*
- *Attack source detection technique*
- *Attack prevention/counteraction technique*
- *Model data gathering technique*
- *Determination of deviation from model data*

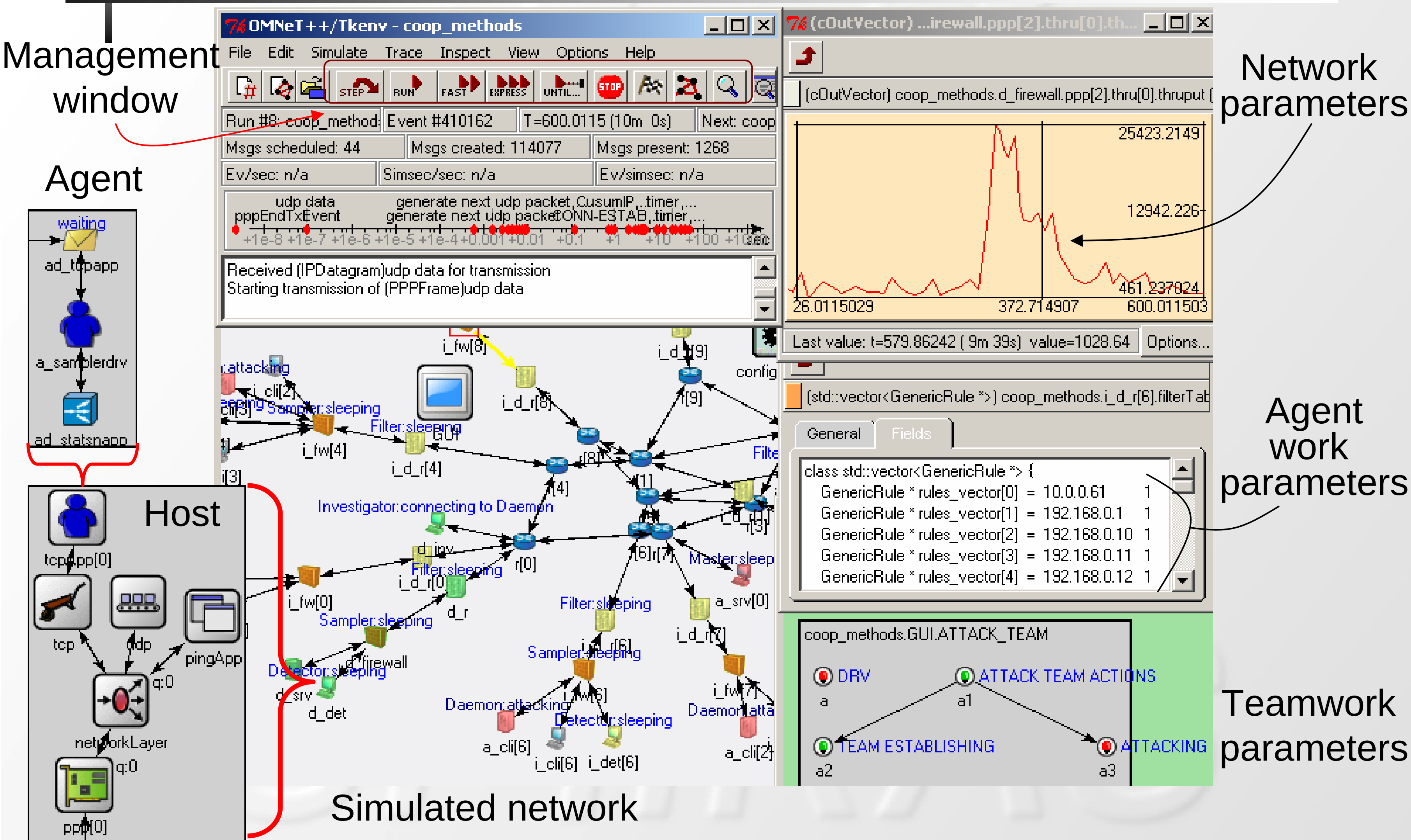
Efficiency Parameters:

- List of detectable attacks
- Volume of the input traffic before and after filters
- Percent of the normal traffic and the attack traffic on entrance to attacked network
- Rate of dropped legitimate traffic (false positive rate)
- Rate of admitted attack traffic (false positive rate)
- Attack detection and attack reaction times
- Computational complexity
- etc.

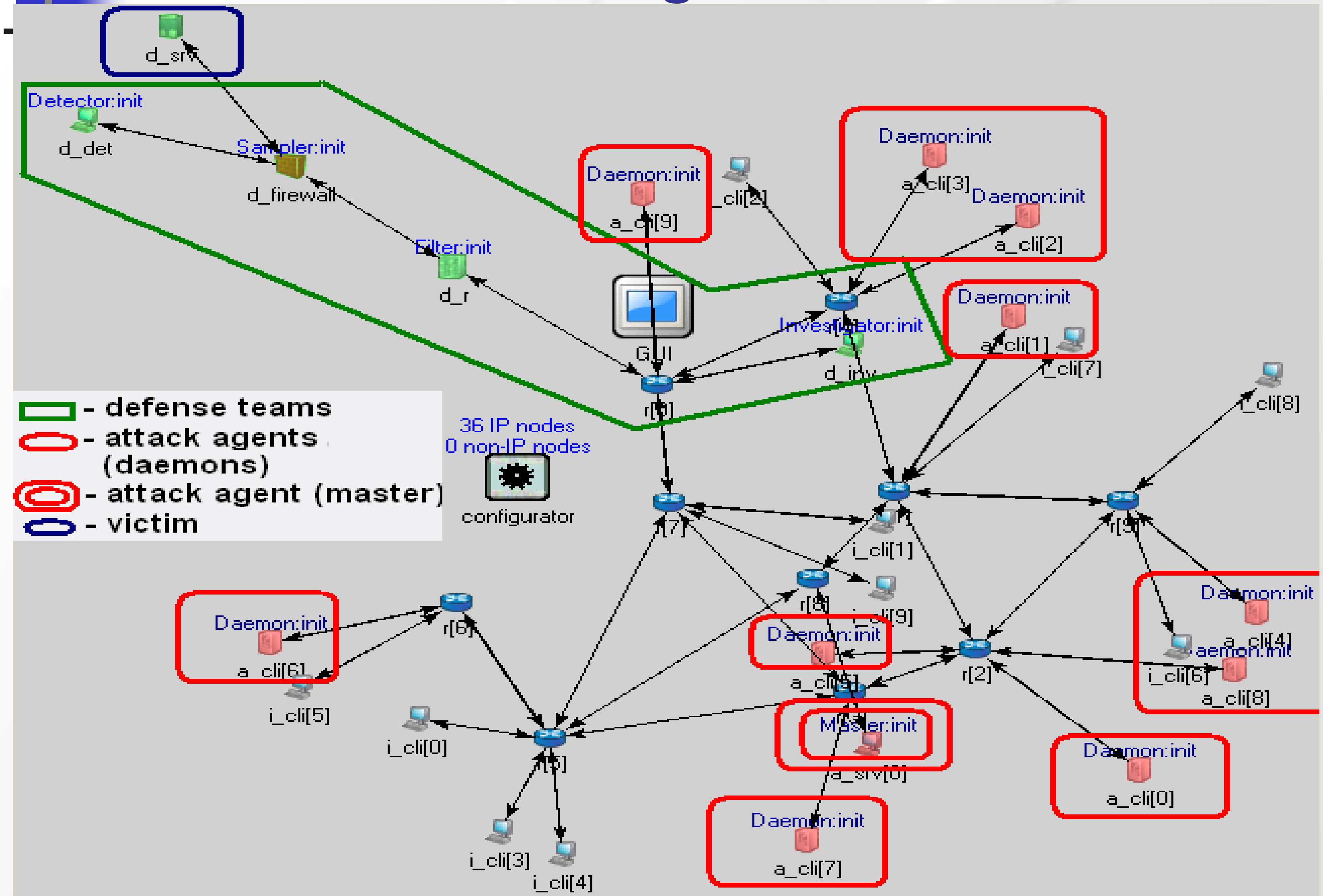
Architecture of Simulation Environment



User Interface of Simulation Environment



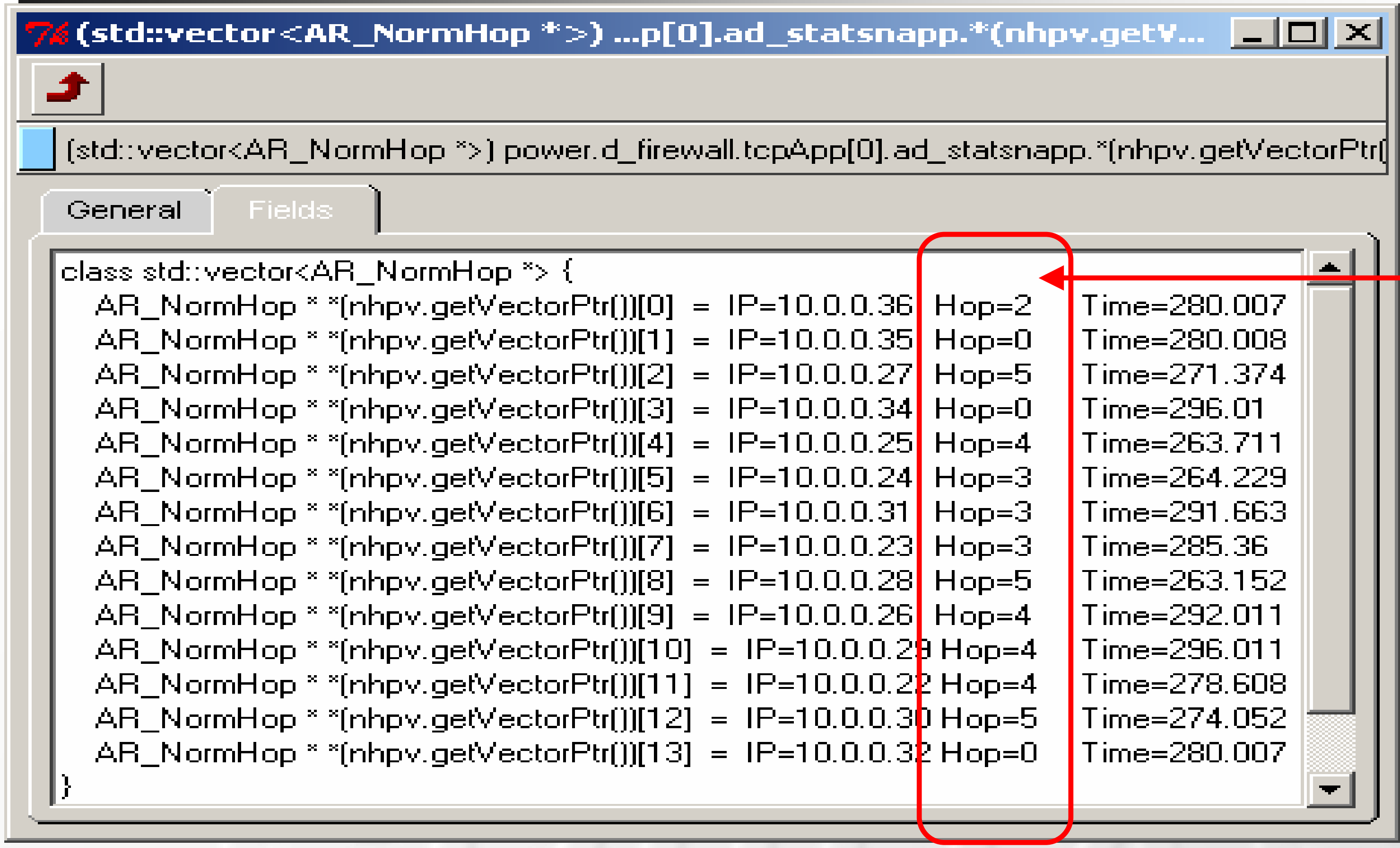
Configuration of the Internet Fragment and Agent Teams



Learning Mode (1)

- The main task of learning mode is to *create the model of generic traffic for the given network*.
- *The clients* send the requests to the server and it replies.
- At this time *sampler* analyses requests and uses them to form the models and parameters for defense different methods.
- During the learning it is possible to watch the *change of traffic models*.

Learning Mode (2)



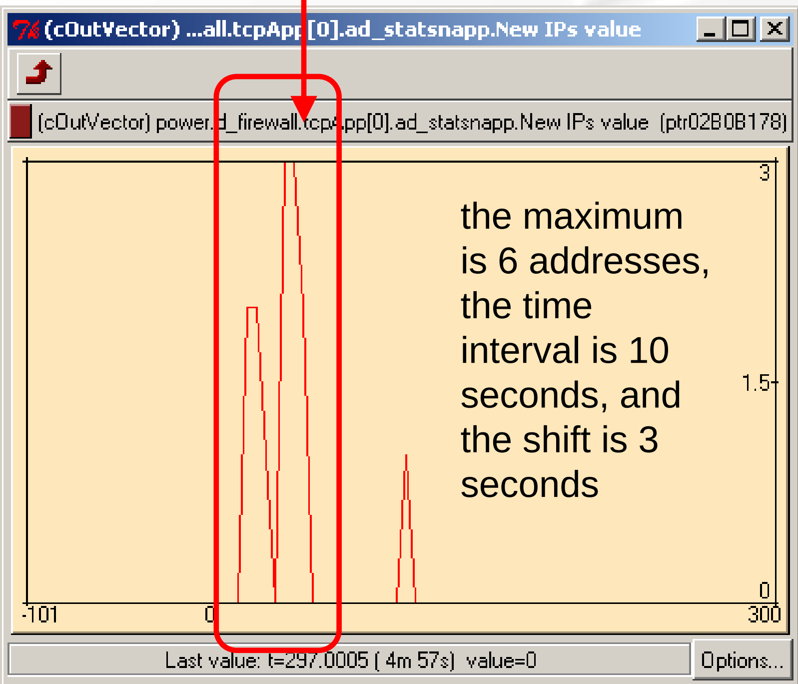
Number of hops

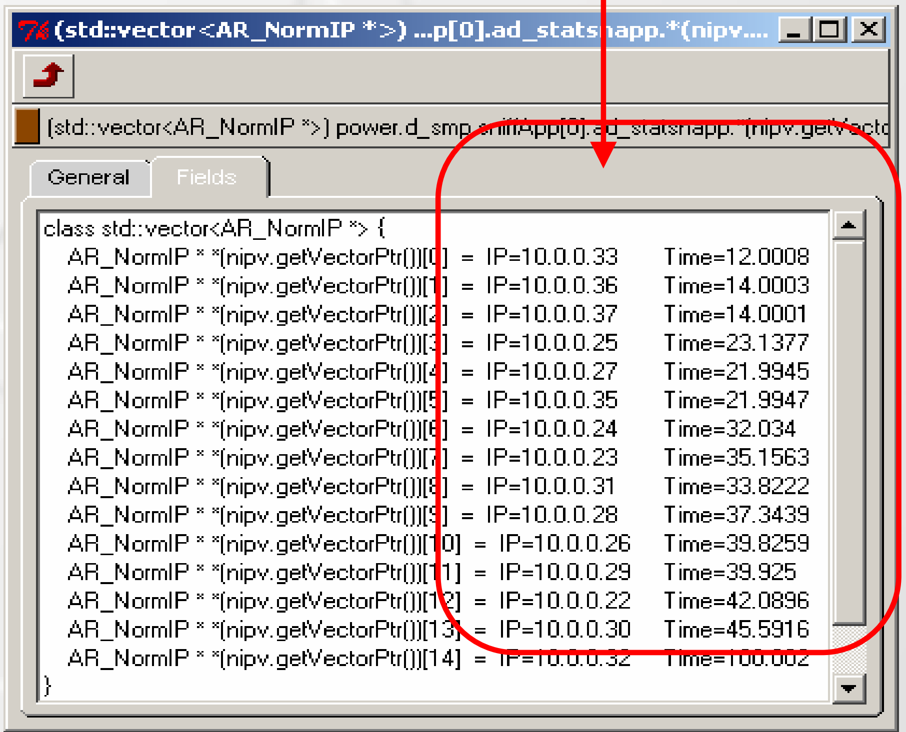
List of hosts that sent requests to server and <u>hops</u> to them after 300 sec of learning

Learning Mode (3)

many new addresses in the beginning

many new addresses in the interval between 0 and 50 seconds





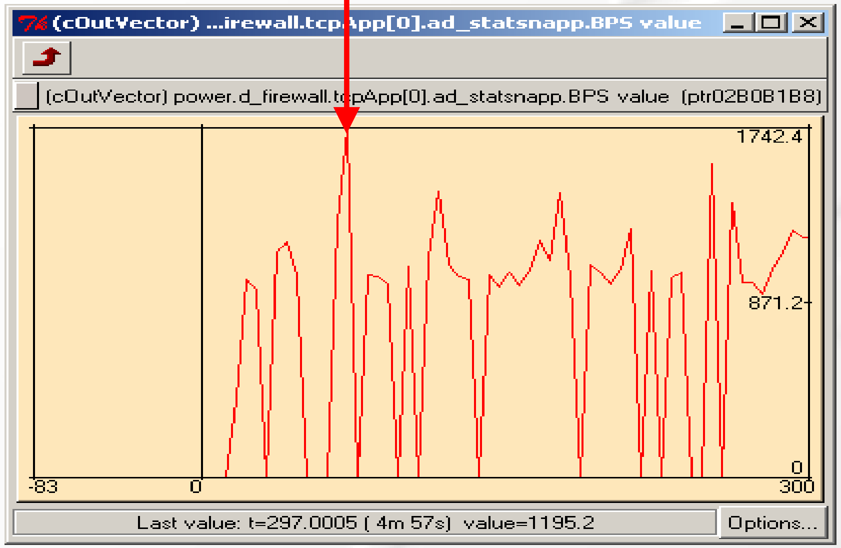
Change of new IP addresses amount

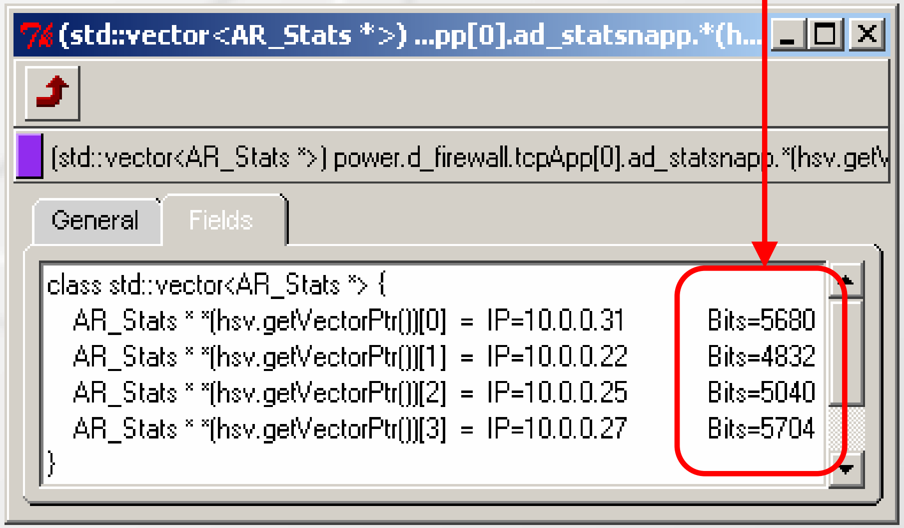
List of clients requested server and considered as legitimate after 300 sec of learning

Learning Mode (4)

The maximum value was 1742.4 bit/s

Values of bits in interval 10 seconds





Change of BPS (bit per second) parameter

Values of transmitted bits for different hosts

Decision Making and Acting (1)

- Normal work (interval 0 – 300 seconds)

- Defense team: Formation, start using BPS method

- Attack team: Formation

- Attack team: After 300 seconds - begins the attack actions (intensity of attack for every daemon - 0.5, **no IP spoofing**)

- Defense team: data processing, attack detecting (**using BPS**) and reacting (interval 300 – 350 seconds)

- Defense team: blocking the attack, destroying some attack agents (interval 300 – 600 seconds)

Decision Making and Acting (2)

- Attack team: After 600 seconds - **automatic adaptation** (redistributing the intensity of attack (0.83), changing the method of **IP spoofing (Random)**)

- Defense team: data processing, failing to detect the attack (**using BPS method**) – Detector sees that the input channel throughput has noticeably lowered, but does does not receive any anomaly report from sampler because BPS does not work.

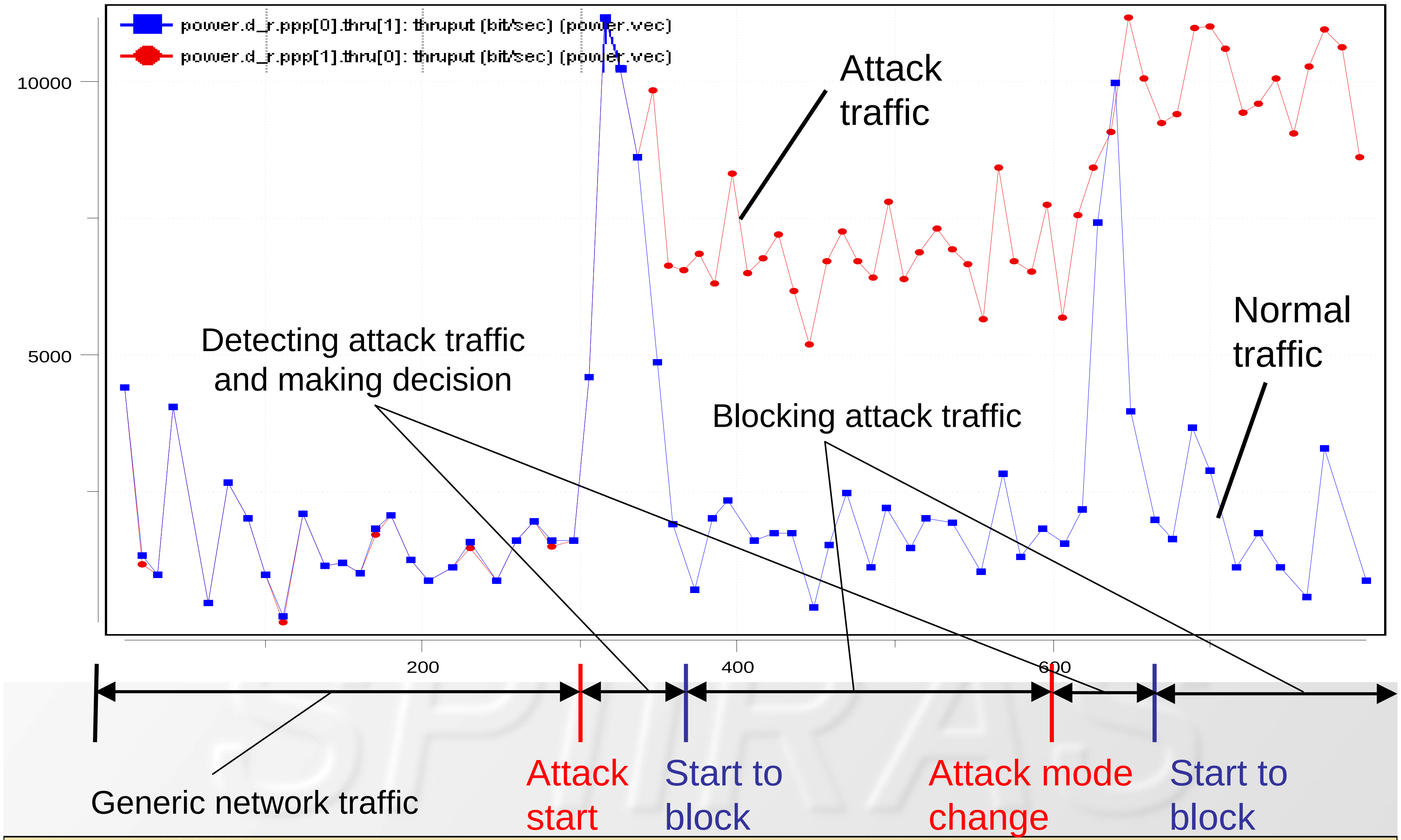
- Defense team: Changing defense method on **SIPM (automatic adaptation)**.

- Defense team: data processing, attack detecting (**using SIPM method**) and reacting – (interval 600 – 700 seconds)

- …………………………..

Scheme of Acting

Graphs of channel throughput



Cooperation between defense teams

***Models of cooperation between distributed defense teams*:**

(1) *filter-level cooperation*

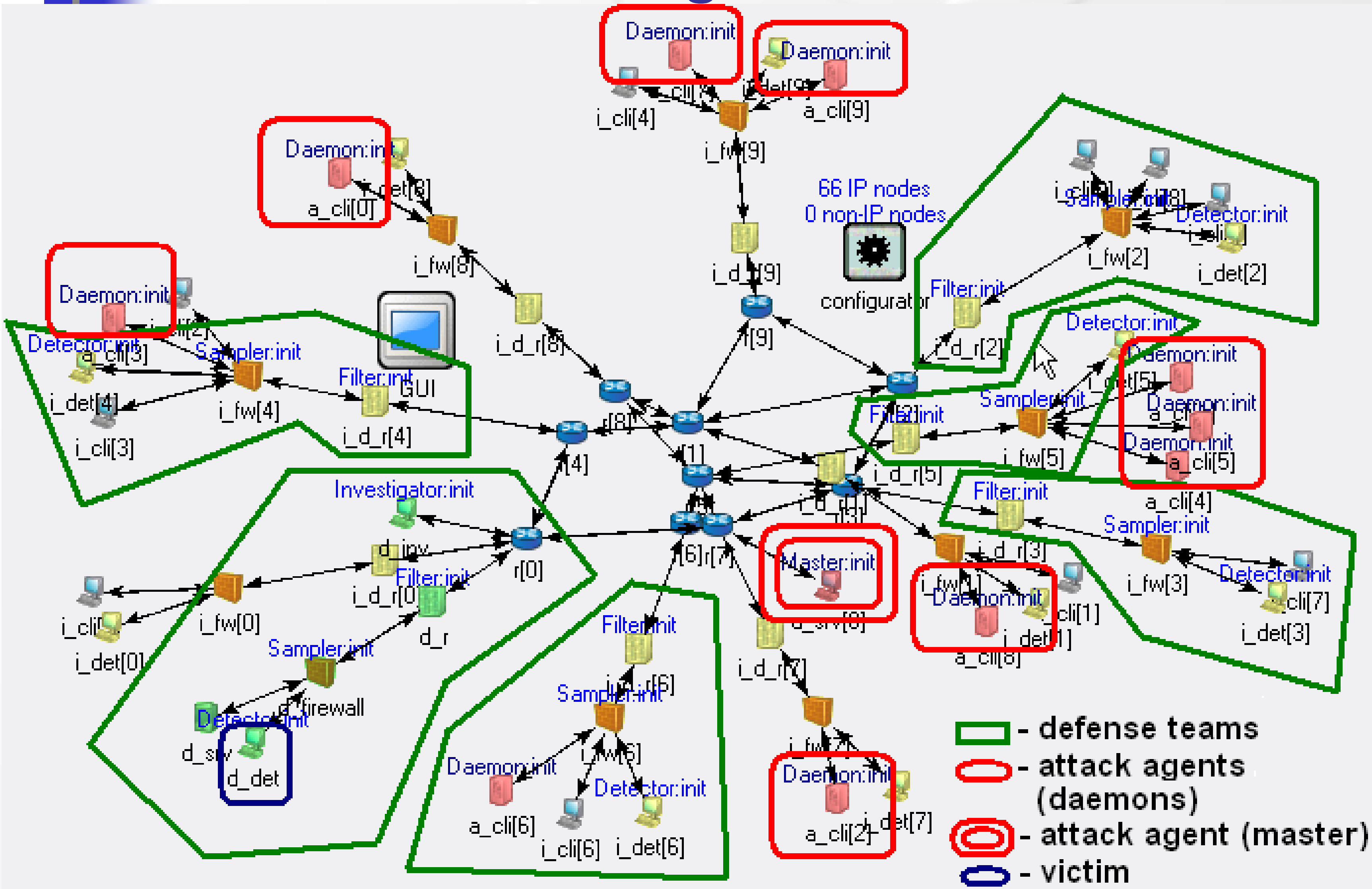
(2) *sampler-level cooperation*

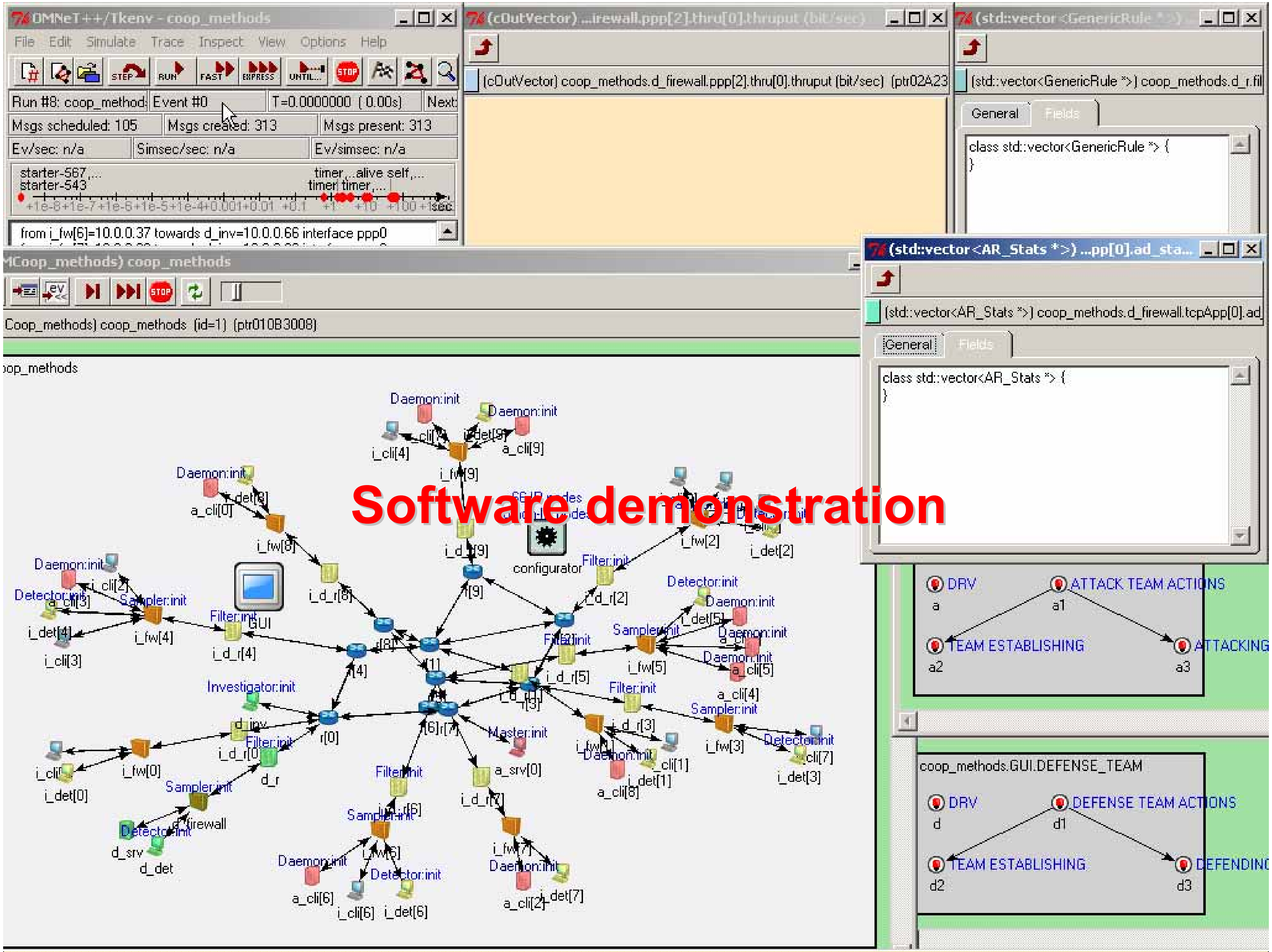
(3) *“poor” cooperation*:

(4) *“full” cooperation*

Such cooperation schemas are used in the cooperative DDoS defense methods: COSSACK, Perimeter-based DDoS defense, DefCOM, Gateway-based, ACC pushback, MbSQD, SOS, tIP router architecture, etc.)

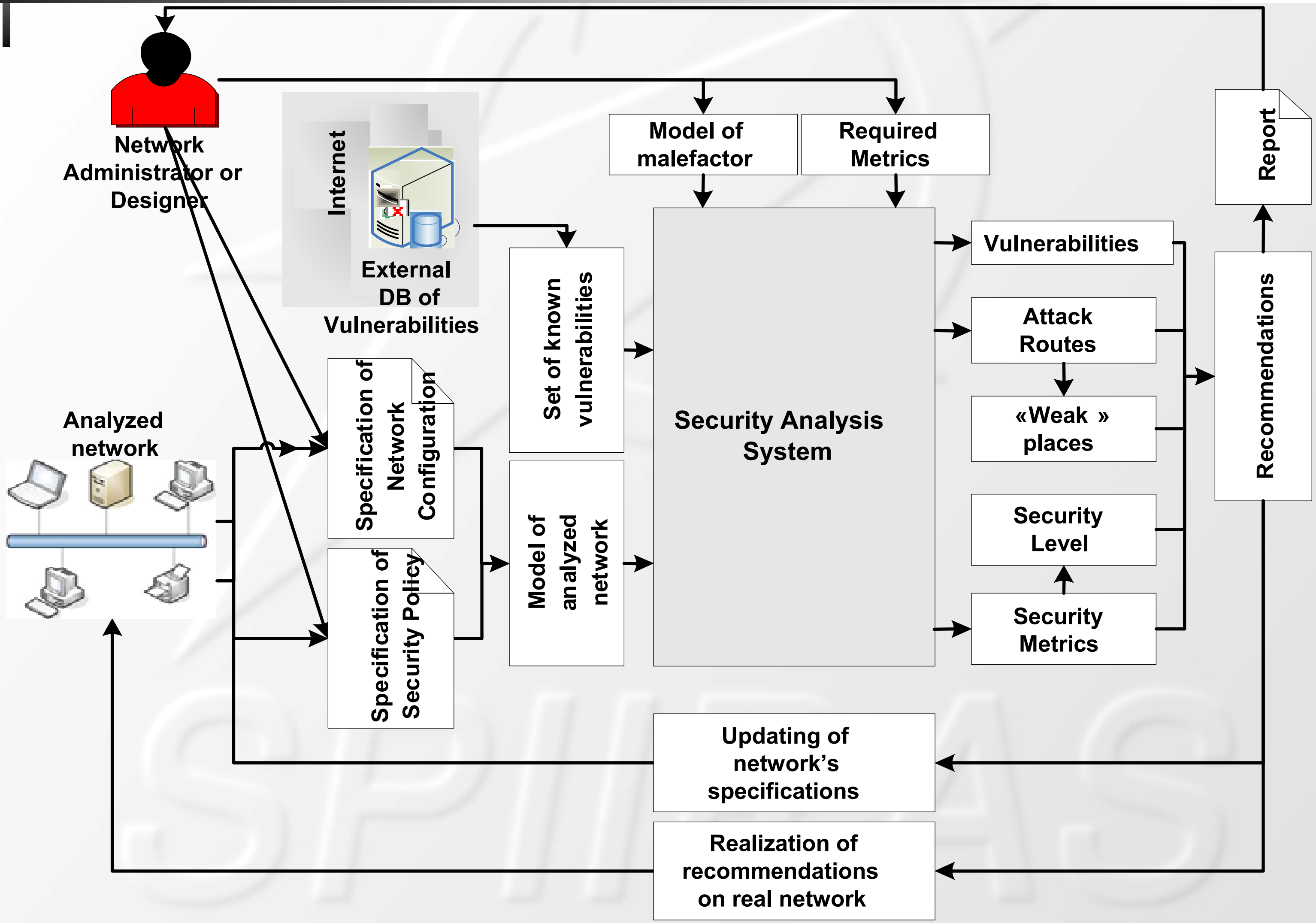
Configuration of the Internet fragment and agent teams




Software demonstration
OMNeT++/Tkenv - coop_methods
File Edit Simulate Trace Inspect View Options Help
Run #8: coop_methods
Event #0
T=0.0000000 (0.00s)
Msgs scheduled: 105
Msgs created: 313
Msgs present: 313
Ev/sec: n/a
Simsec/sec: n/a
Ev/simsec: n/a
from i_fw[6]=10.0.0.37 towards d_inv=10.0.0.66 interface ppp0
(cOutVector) coop_methods.d_firewall.ppp[2].thru[0].thruput (bit/sec)
(std::vector<GenericRule *>) coop_methods.d_r.fil
General
Fields
class std::vector<GenericRule *> {
}
(std::vector<AR_Stats *>) coop_methods.d_firewall.tcpApp[0].ad
class std::vector<AR_Stats *> {
}
Coop_methods) coop_methods (id=1)
configurator
GUI
DRV
ATTACK TEAM ACTIONS
TEAM ESTABLISHING
ATTACKING
coop_methods.GUI.DEFENSE_TEAM
DEFENSE TEAM ACTIONS

- Security analysis of computer networks

High Level Task Representation



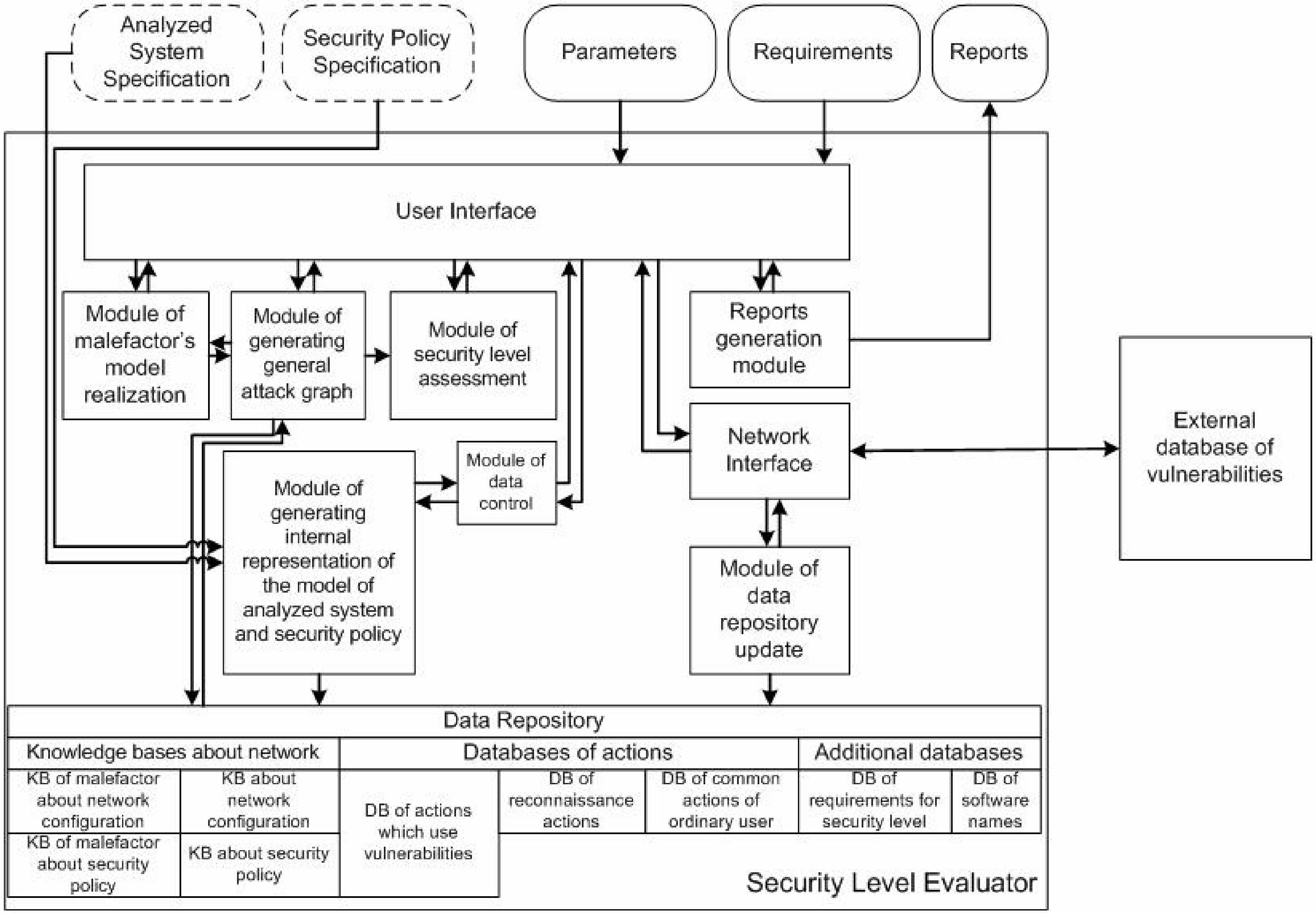
Main features of the Approach

- **Based on malefactor's action simulation and integrated family of various expert knowledge models**
- **Two main phases:**
  - (1) construction of attack graph and
  - (2) computation of different security metrics using combination of qualitative techniques of risk analysis
- **Taking into account diversity of malefactor's positions, intentions and experience;**
- **Estimating the influence of different configuration and policy data;**
- **Taking into account not only attack actions (which use vulnerabilities), but the common actions of legitimate users and reconnaissance actions;**

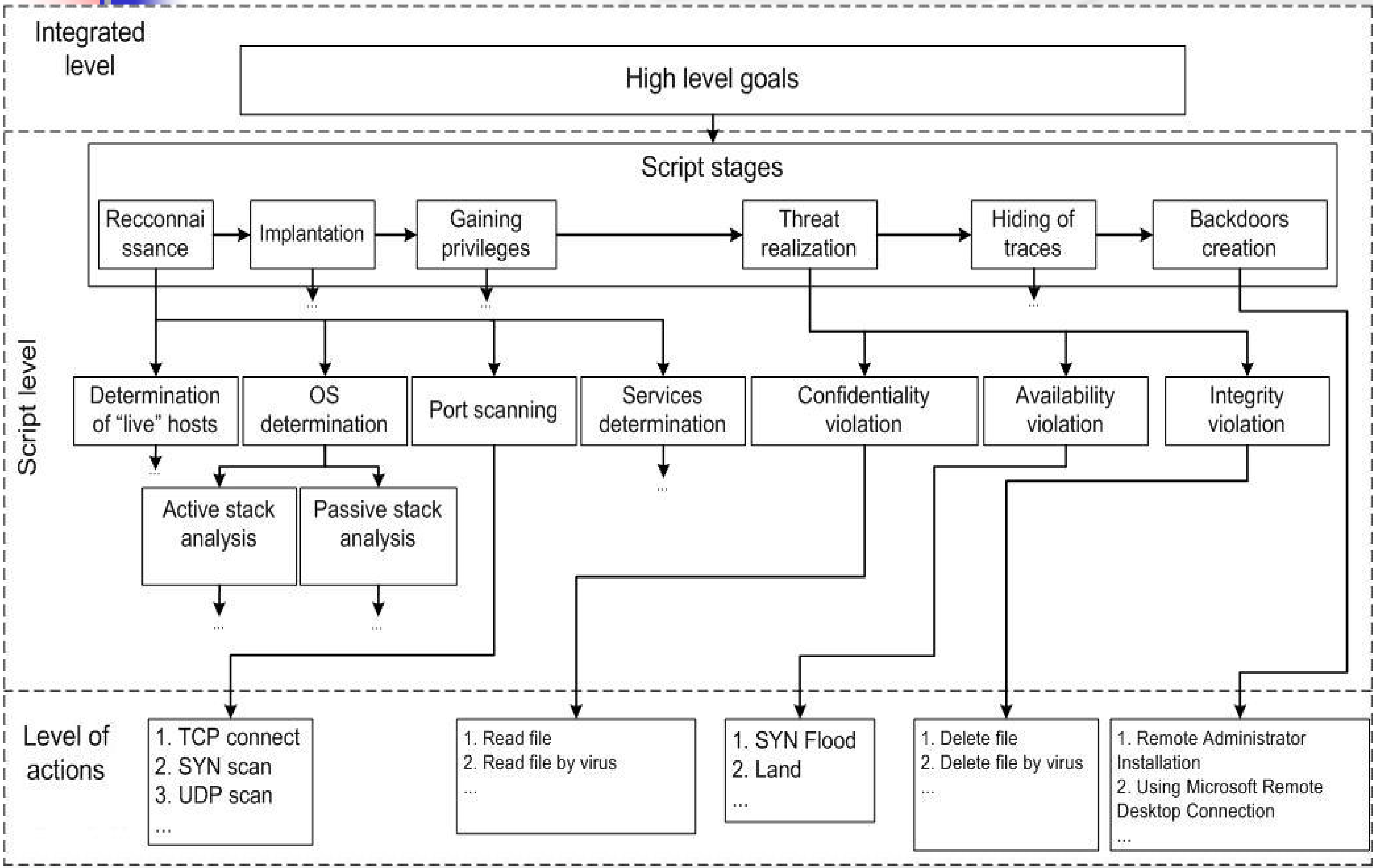
Main features of the Approach (cont'd)

- Investigation of various threats for different resources;
- Detection of “weak” places;
- Usage of up to date databases of vulnerabilities (NVD, OSVDB);
- The “CVSS. Common Vulnerability Scoring System” approach is used for computation of a part of primary security metrics;
- Comparing calculated metrics and user requirements
- The qualitative techniques of risk analysis are used for computation of security metrics (in particular SANS/GIAC and FRAP technique).

Generalized Architecture



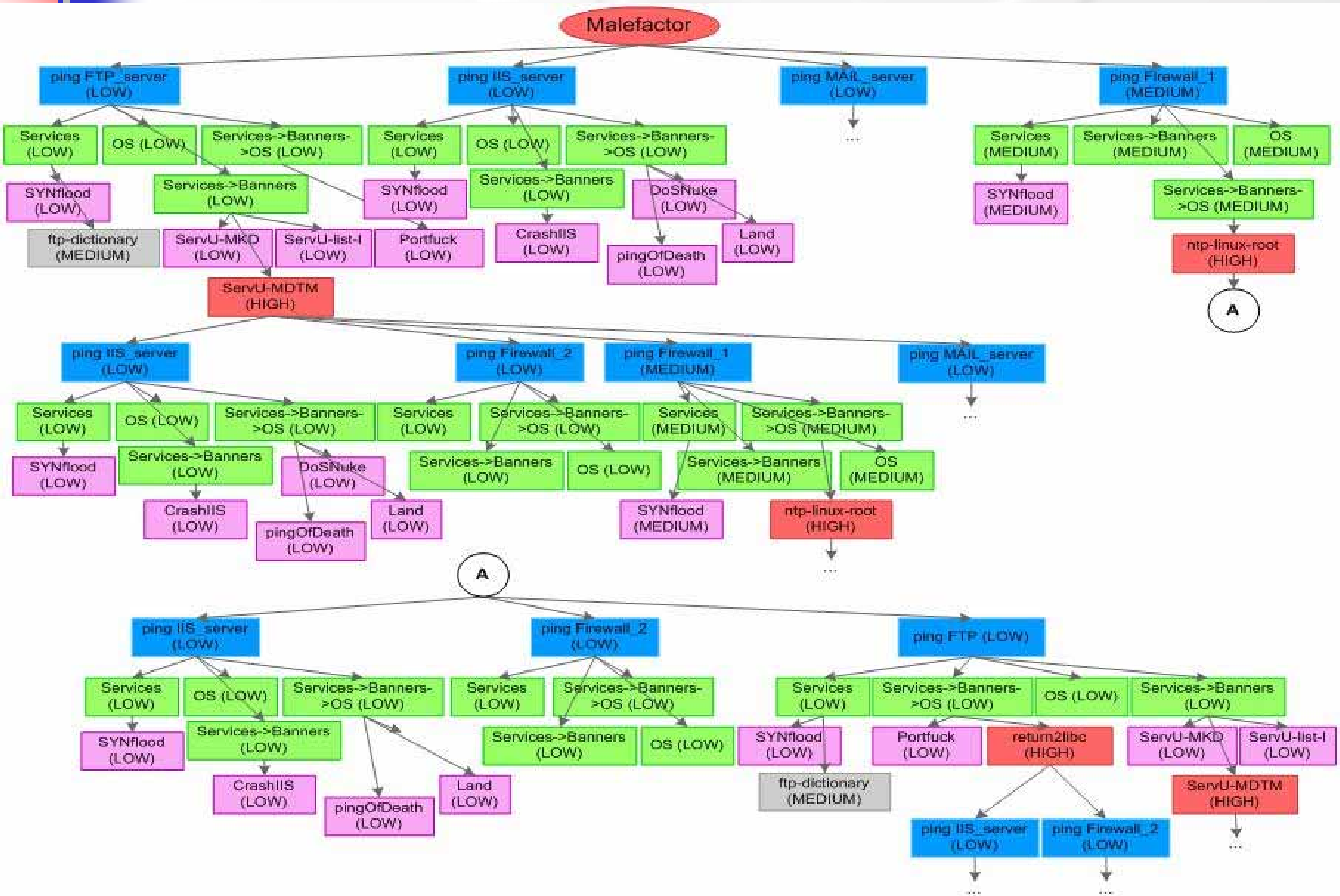
Generalized Attack Scenarios



Generating Common Attack Graph

- **Realization of actions which are intended for malefactor's movement from one host onto another in the following cases:**
  - There is a possibility to realize the attack actions which use vulnerabilities of software and hardware and which require that the malefactor already have the privileges of local user
  - Movement of the malefactor into the attacked host allows him to penetrate the another segment of network
  - Movement of the malefactor into attacked host allows him to use "trust relationships"
- **Realization of reconnaissance actions for detection of "live" hosts (ex., "ping" utility)**
- **Realization of reconnaissance scenarios (the sets of actions) for each of the detected hosts (ex, "nmap OS", "nmap services", "banners")**
- **Realization of attack actions which use vulnerabilities of software and hardware, and common actions of ordinary users**

Example: Attack Graph



Model of Security Level Evaluation

Model of Security Level Evaluation consists of:

- Security metrics
- Rules (formulas) for their calculation

Two approaches for security level evaluation:

- Qualitative express assessment (!)
- Quantitative computation

Taxonomy of Security Metrics (SM)

- According to division of objects of attack graph
  - SM of base objects (hosts, attack actions)
  - SM of complex objects (routes, threats, graph)
- According to the order of calculation
  - Primary (received directly from attack graph)
  - Secondary (calculated on the basis of primary)
- Whether metrics are used for evaluation of general security level
  - Basic (are used for evaluation of general security level)
  - Auxiliary (are not used for evaluation of general security level)

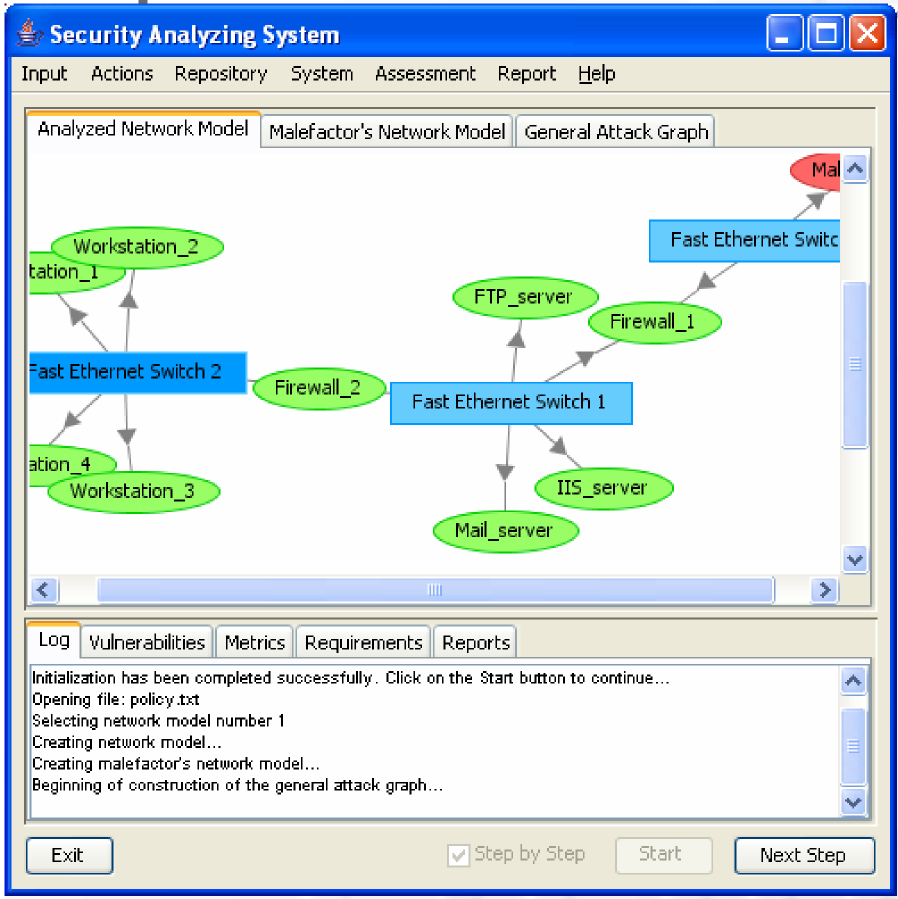
Basic Security Metrics

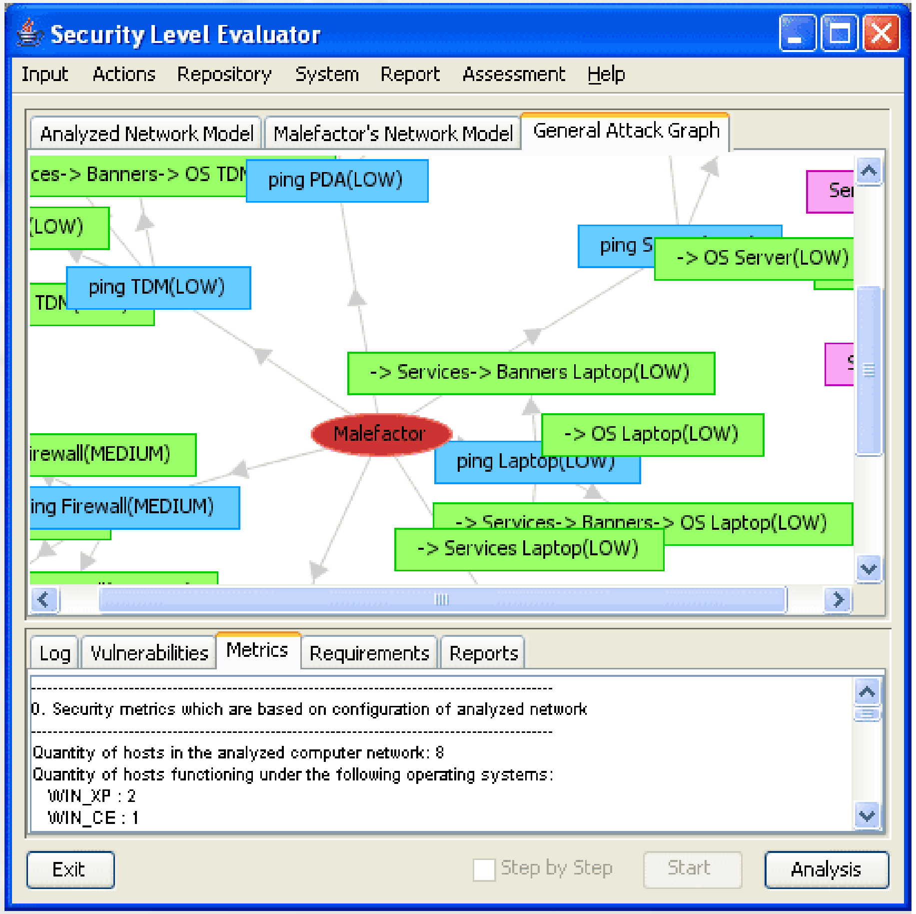
- Criticality Level of the host Criticality(h)
- Criticality Level of attack action Severity(a)
- Damage Level of attack action Mortality(a,h)
- Damage Level of route Mortality(S) or threat Mortality(T)
- Access Complexity Level AccessComplexity(a), AccessComplexity(S), AccessComplexity(T)
- Admissibility of threat realization Realization(T)
- Risk Level of threat RiskLevel(T)
- General Security Level of network SecurityLevel

Technique of General Security Level Evaluation

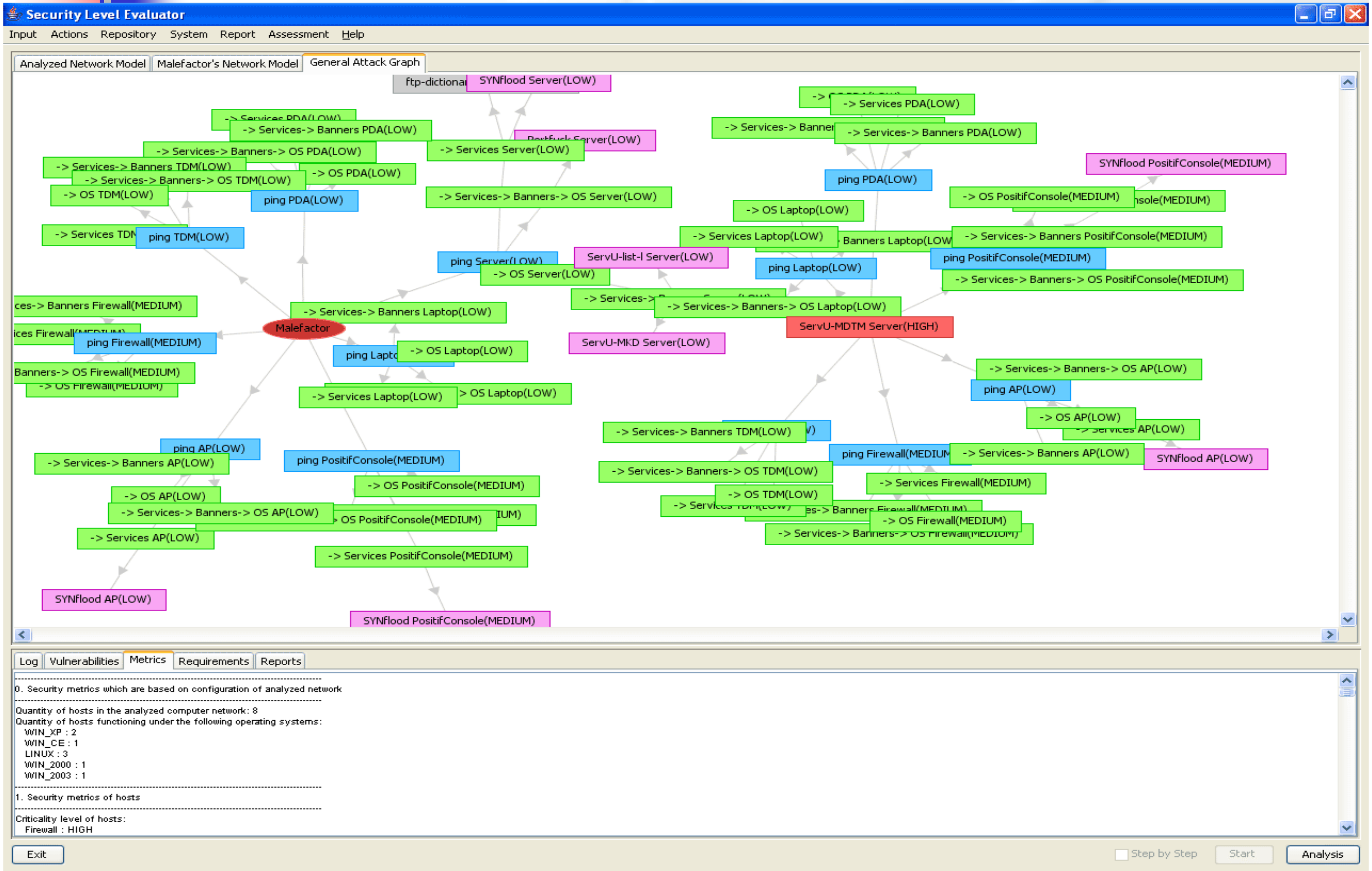
- Calculation of security metrics of basic and complex objects
- Estimation of qualitative risk level for all threats
- Evaluation of security level of analyzed computer network on basis of received values of risk levels for threats

Implementation: User Interface (1)



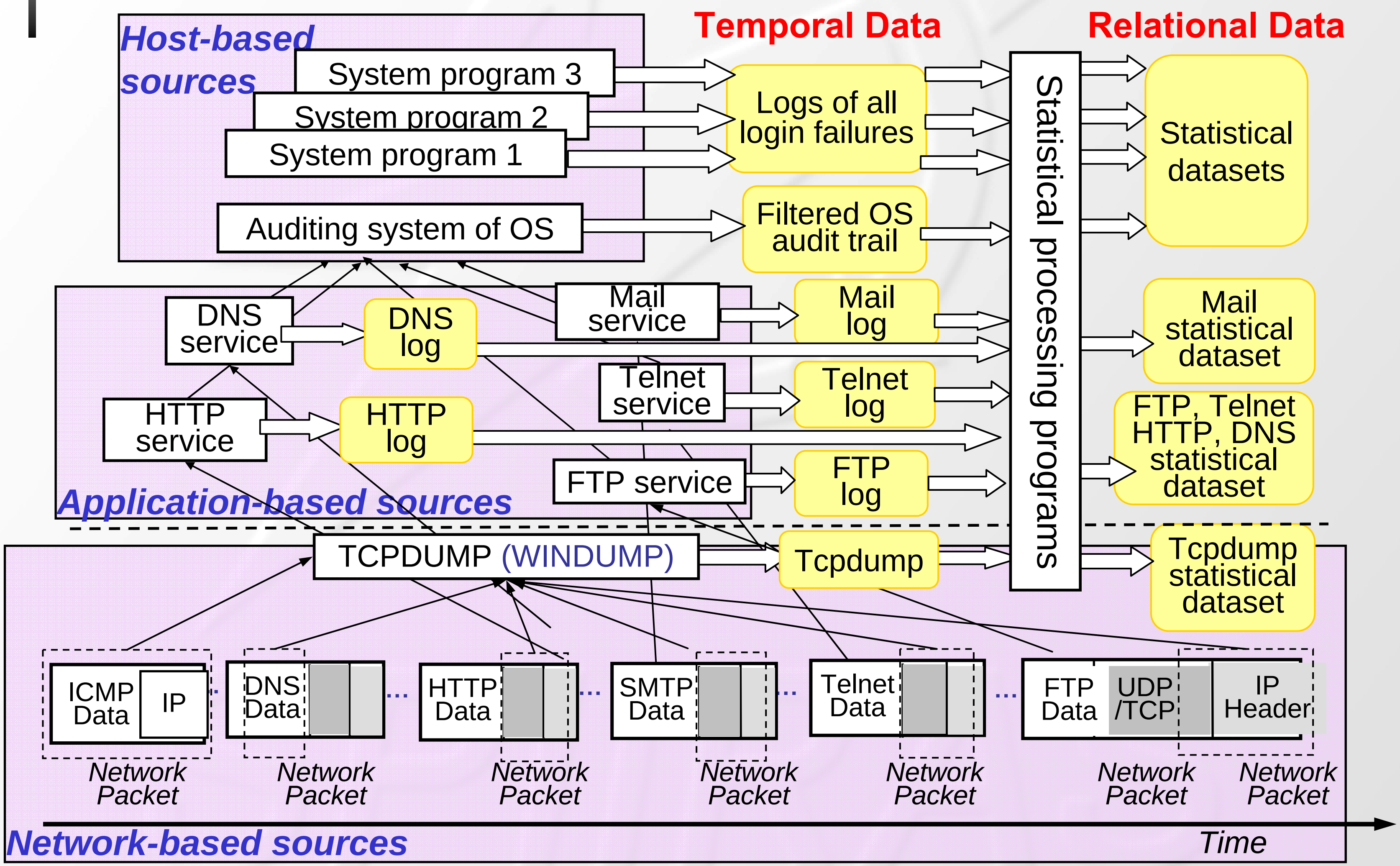


Implementation: User Interface (2)

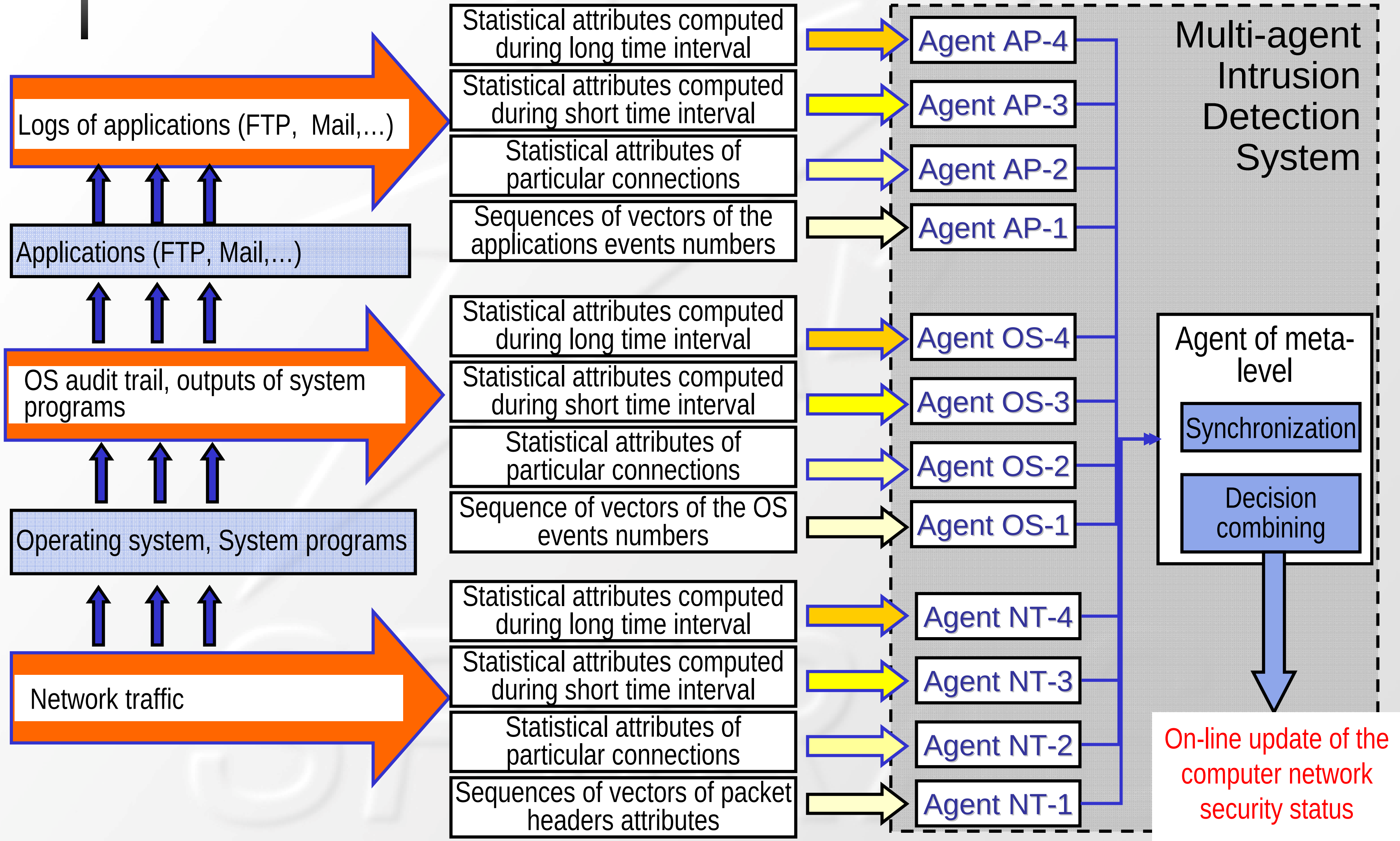


- **Intrusion detection**

Different Sources of information



Multi-agent Architecture of Raw Data Preprocessing and Intrusion Detection



Case Study: Anomaly Detection in Computer Network

Computer security status: {*Normal* , *Abnormal*}.

Types of attacks constituting class "*Abnormal*": {Probing, Remote to local (R2L); Denial of service (DOS) and User to root (U2R)".

Instances of attacks of respective classes: {SYN-scan, FTP-crack attack, SYN flood, and PipeUpAdmin}.

Information Source: Network traffic raw data.

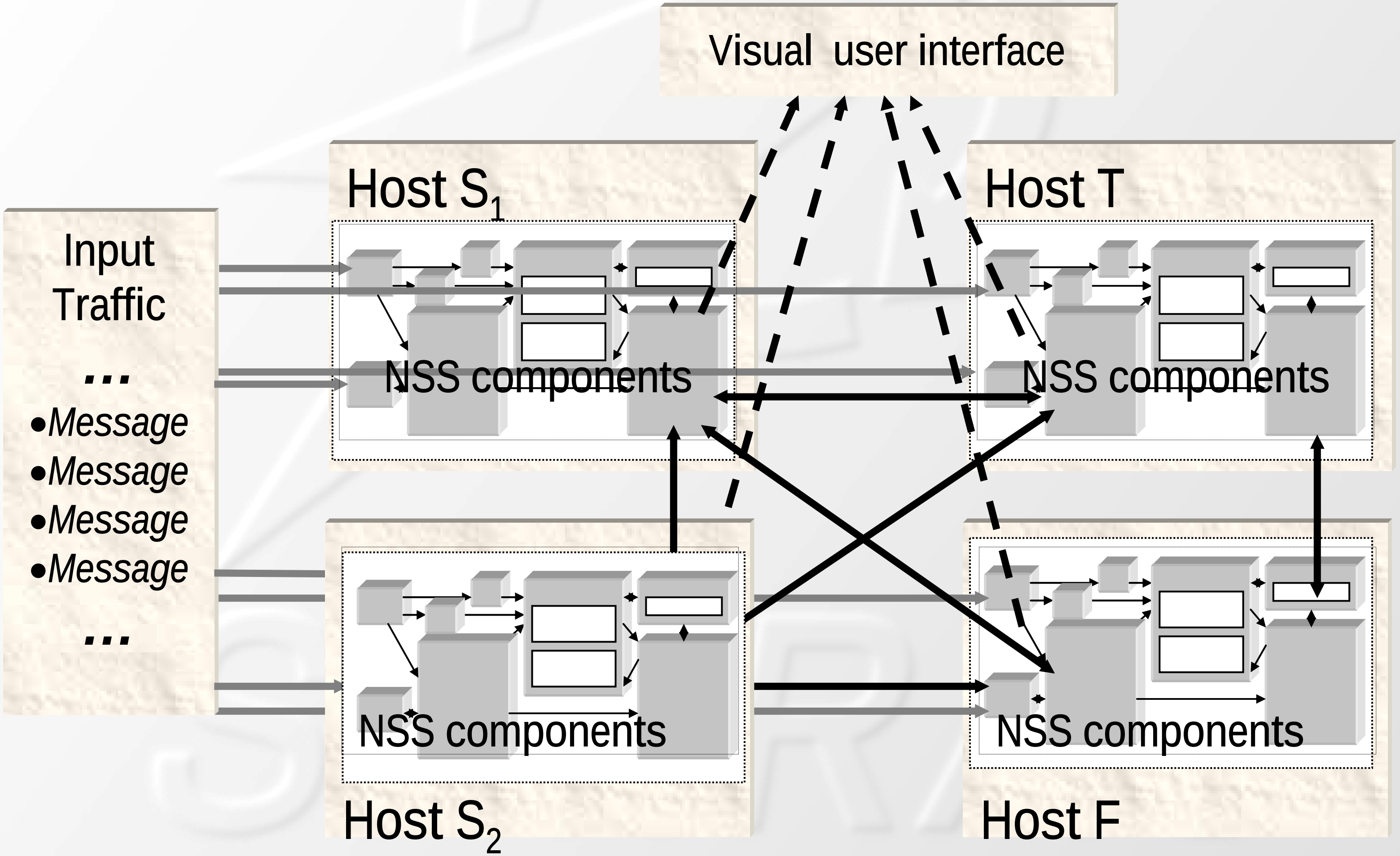
Data source 1: *Stream of binary vectors* specifying stream of headers of *IP* packets within a connection (sequence of binary vectors).

Data source 2: *Statistical attributes of particular connections manifesting in input traffic*. (duration, status, total number of connection packets and also other attributes specifying statistics of connections).

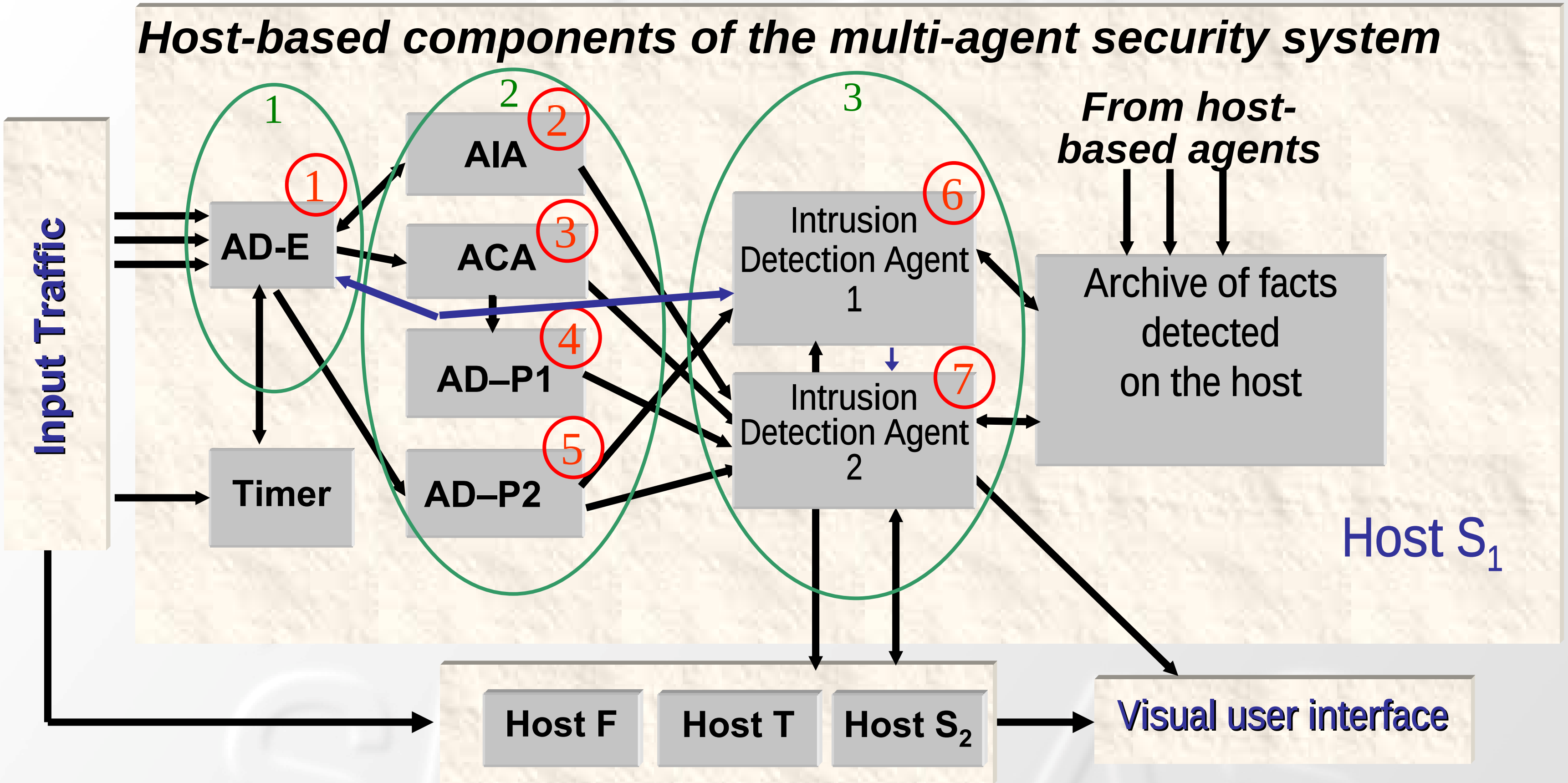
Data source 3: *Statistical attributes of traffic during the short time (5 sec) intervals* (four features specifying integral characteristics of input traffic-- numbers of connections and services of different types during last 5 sec).

Data source 4: *Statistical attributes of traffic for long time intervals* (composed of the same statistics as previous ones averaged over 100 connections).

High-level architecture of MIDS components

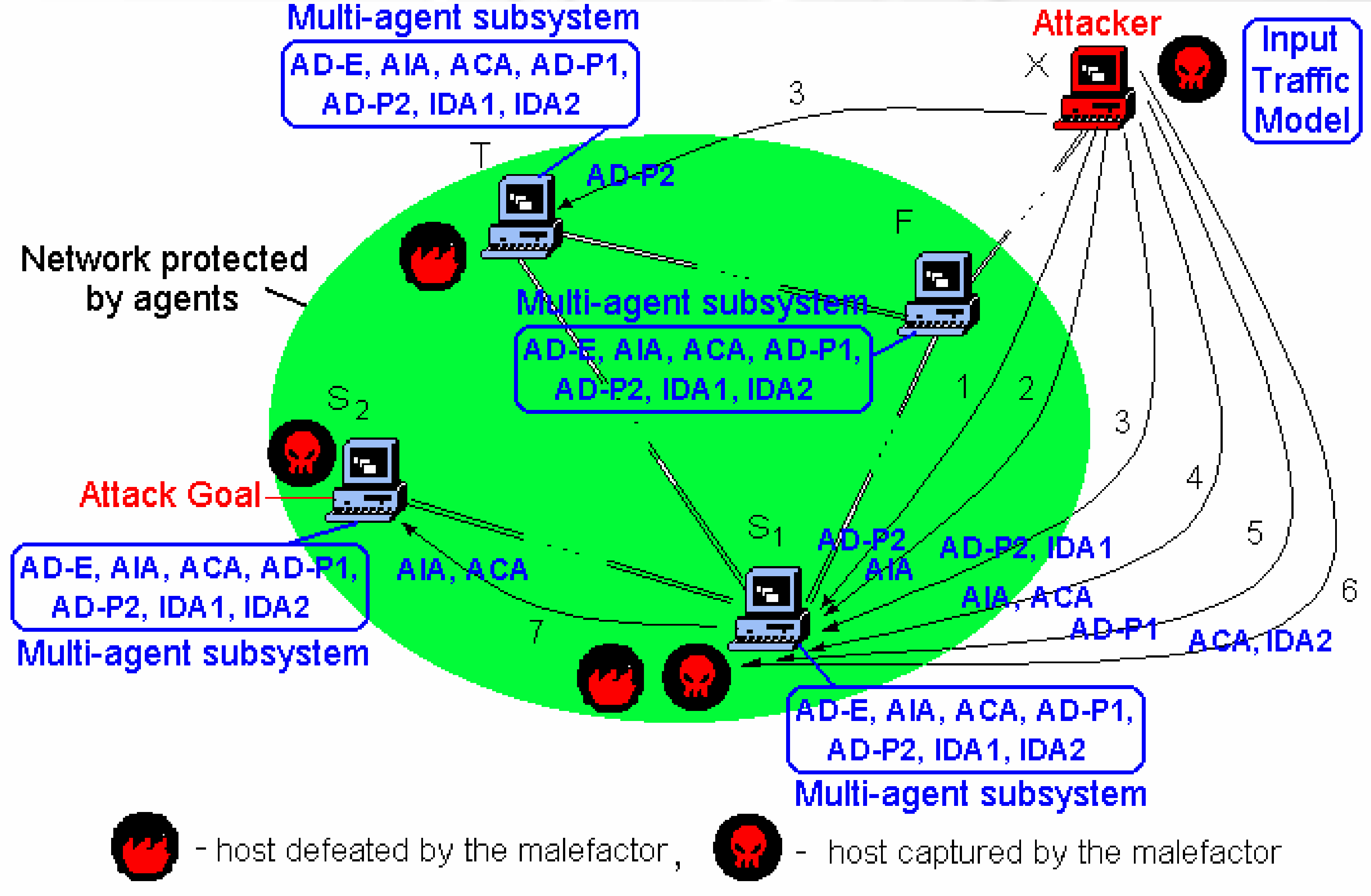


Architecture of host-based MIDS components

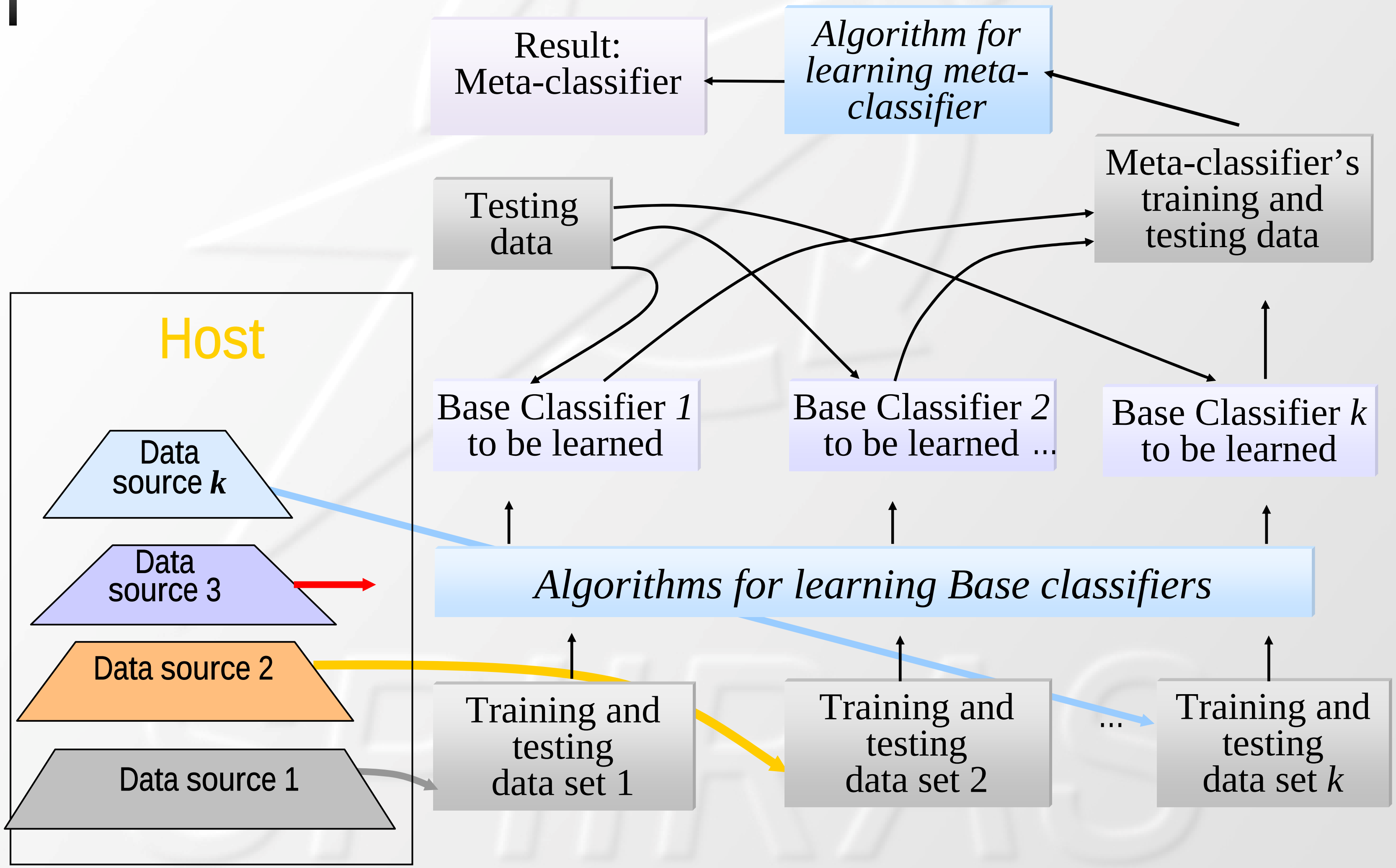


1,2,3 – levels of processing; 1,2,…,7 – types of agents.

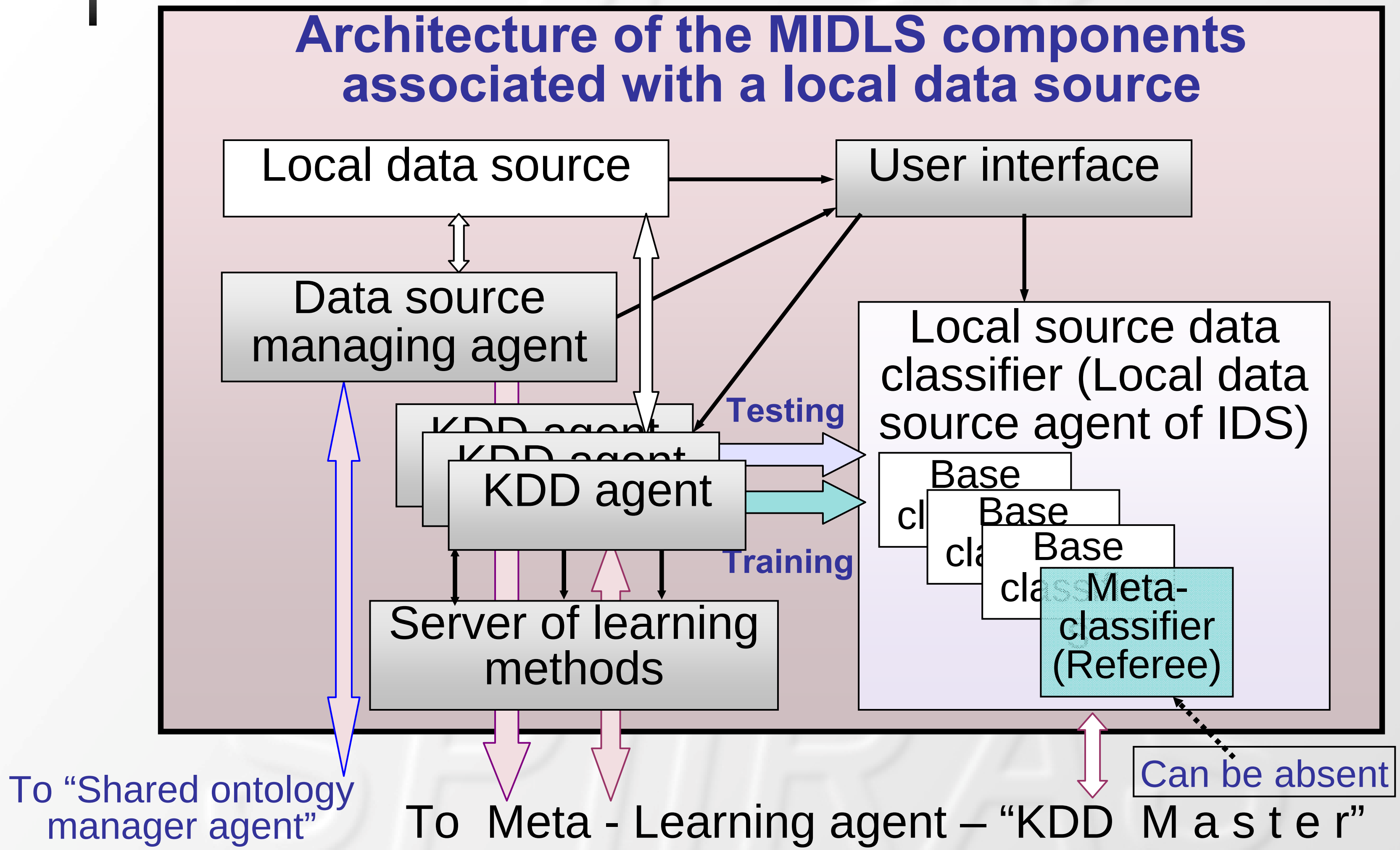
Case-study Simulation



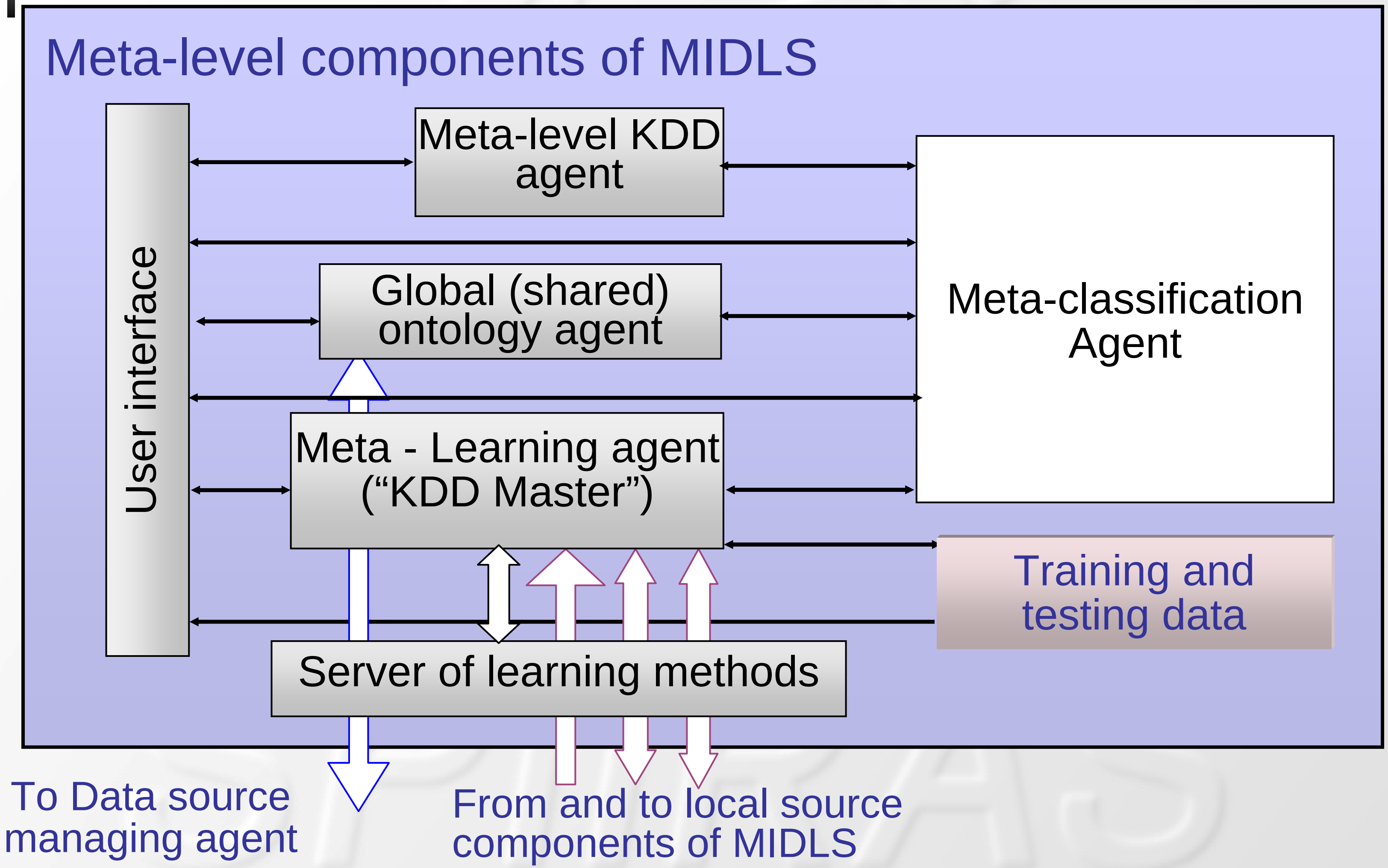
Multi-sensor DF learning: meta-classification scheme



MIDLS architecture

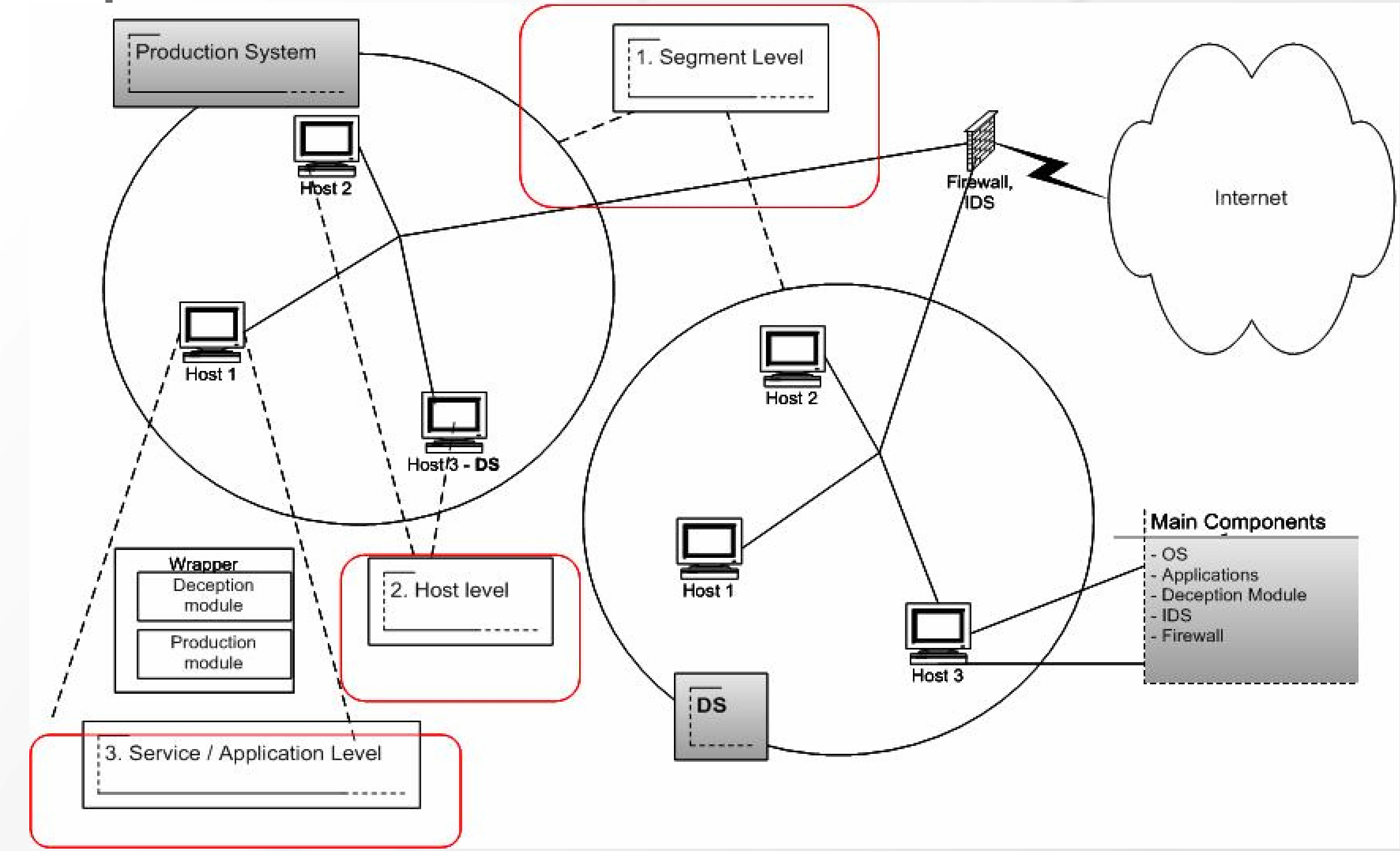


MIDLS architecture

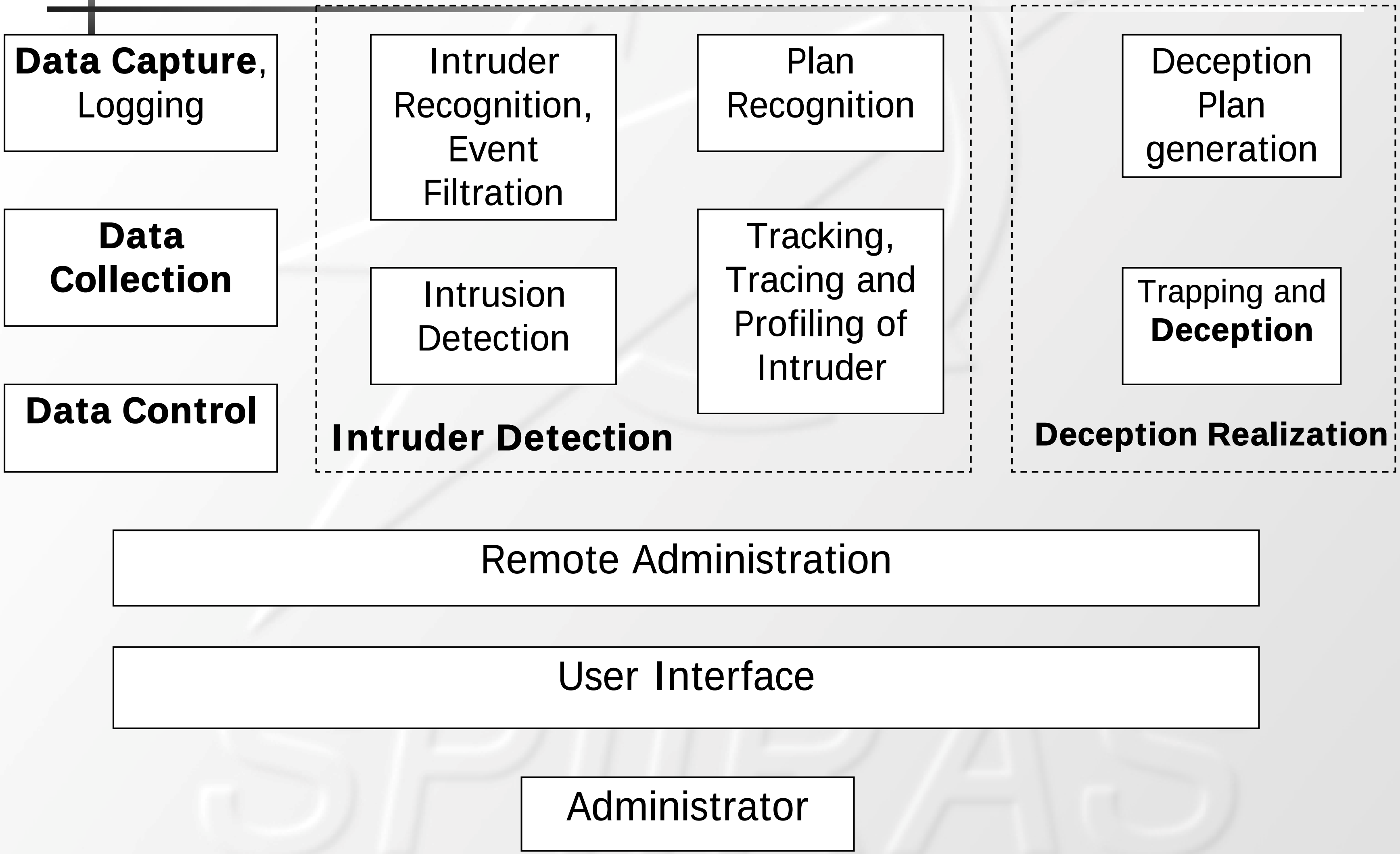


- **Deception systems, honeynets**

Deception System (DS) network architecture

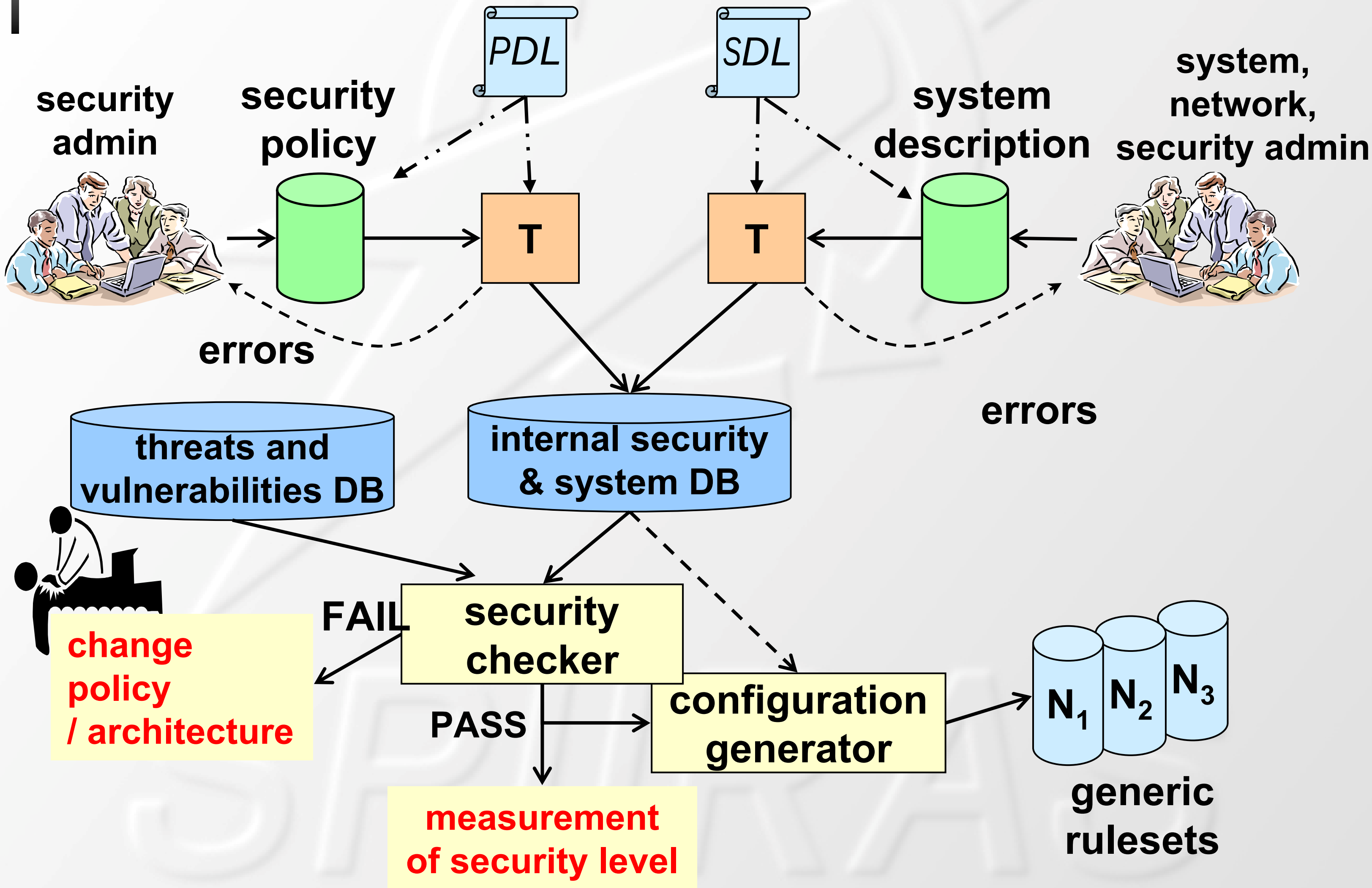


Functional DS architecture

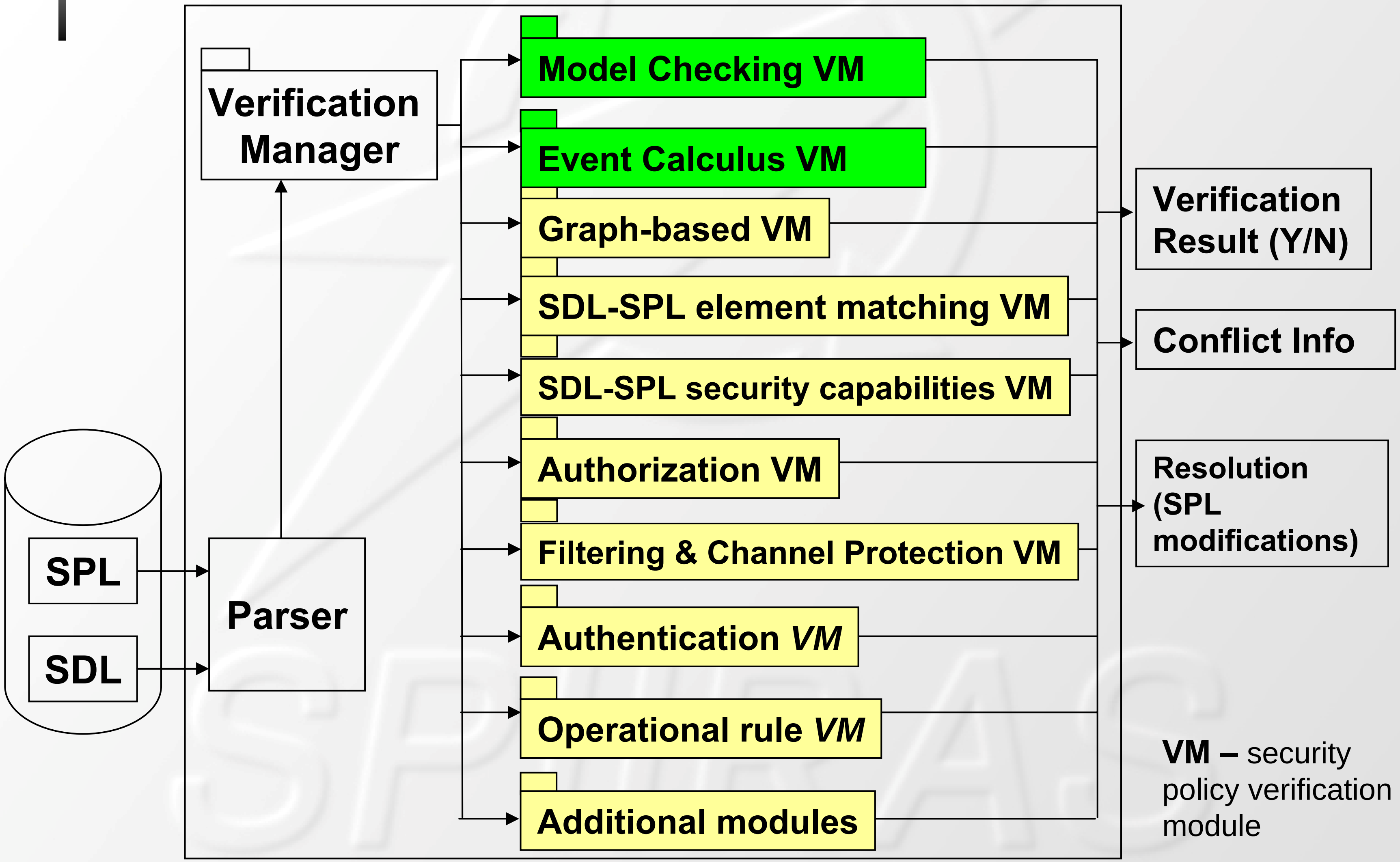


- **Security policy specification and checking**

Workflow of security policy specification and checking



Architecture of security policy verification system



Examples of Verification Modules

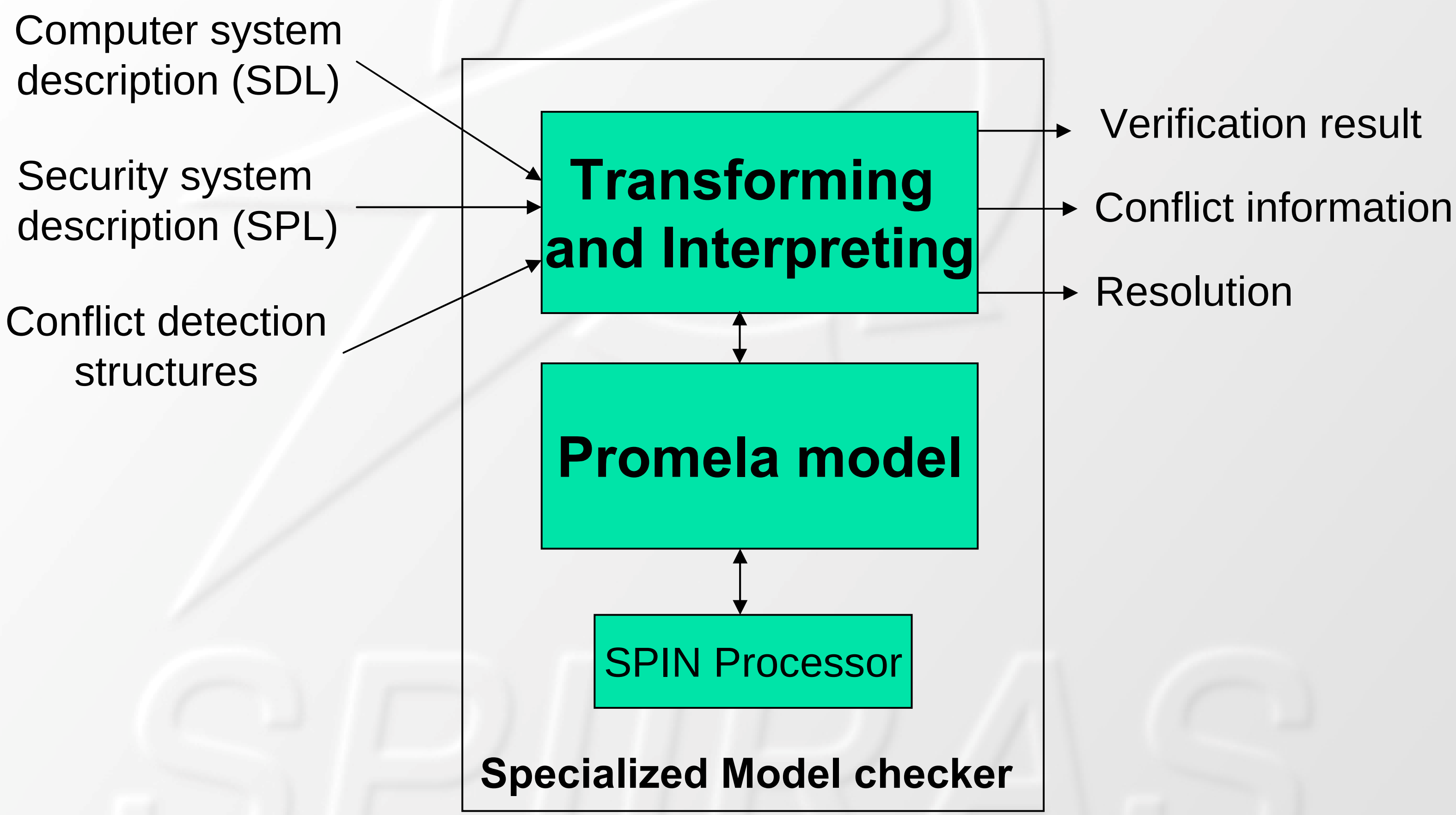
- **Event Calculus verification module**
  - security policies and system description are translated into domain-dependent Event Calculus axiomatic
  - conflict predicates are introduced
  - abductive inference is used for conflict detection
  - *Implemented in Jess*

- **SPIN verification module**
  - security policies and system description are translated into Promela data structures, processes, and assertions
  - policy conflicts introduced as additional assertions
  - *impemented in SPIN-Promela*

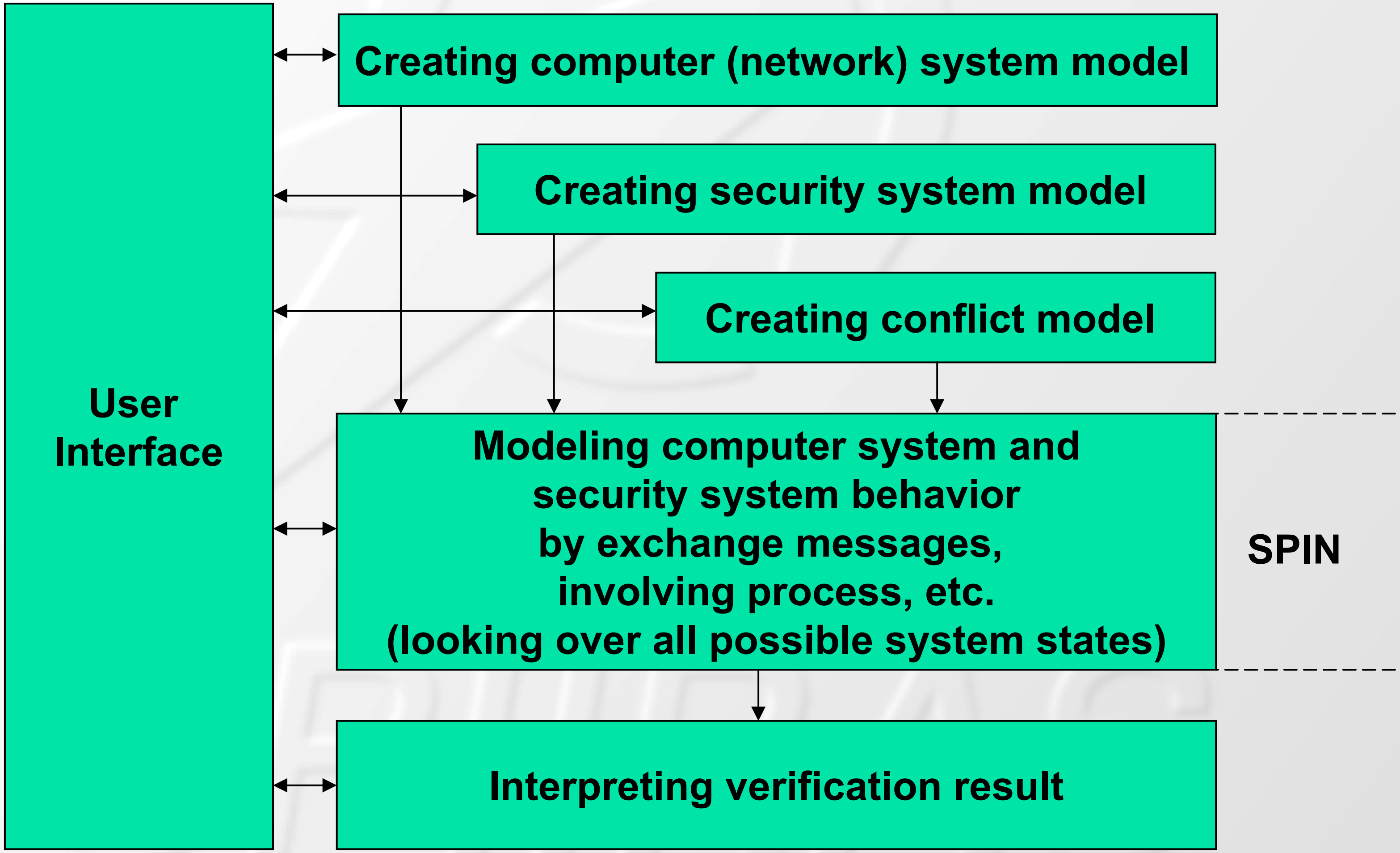
Implementation of security policy verification system

- VerificationManager
  - registers, loads and authenticates verification modules
  - invokes verify() method of verification modules for policy consistency and enforceability checks
  - debugs policy if conflicts have been found
  - open for new verification modules
    - semi-lattice-based, semantic approach
- VerificationModule
  - checks for policy consistency and predicts policy violations
  - resolves conflicts by means of a resolution strategy
- permits independent development of module
  - speed up implementation
- needs administrator to check or change results
  - choosing between proposed alternatives

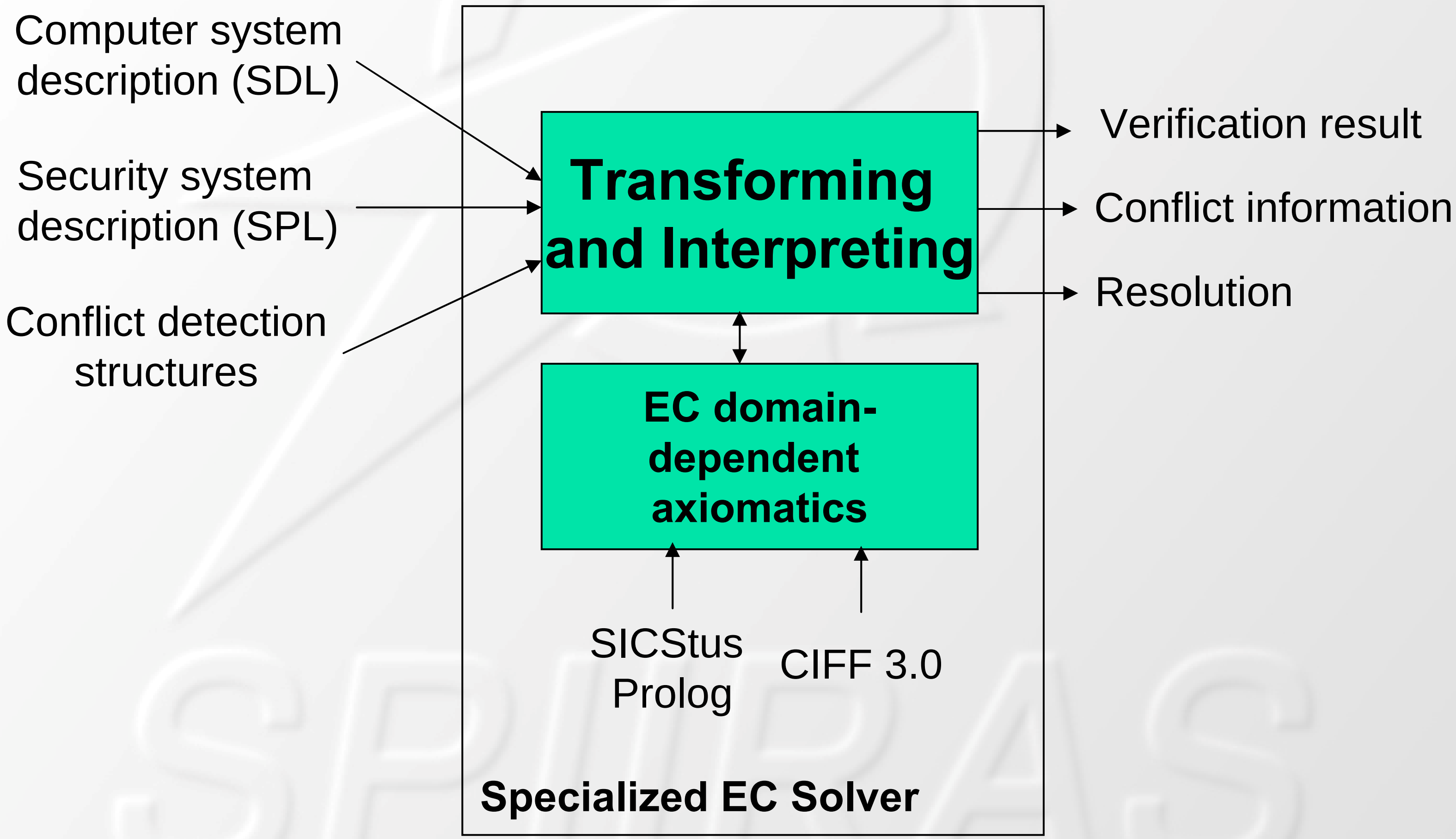
Model Checking VM: functional model (IDEF)



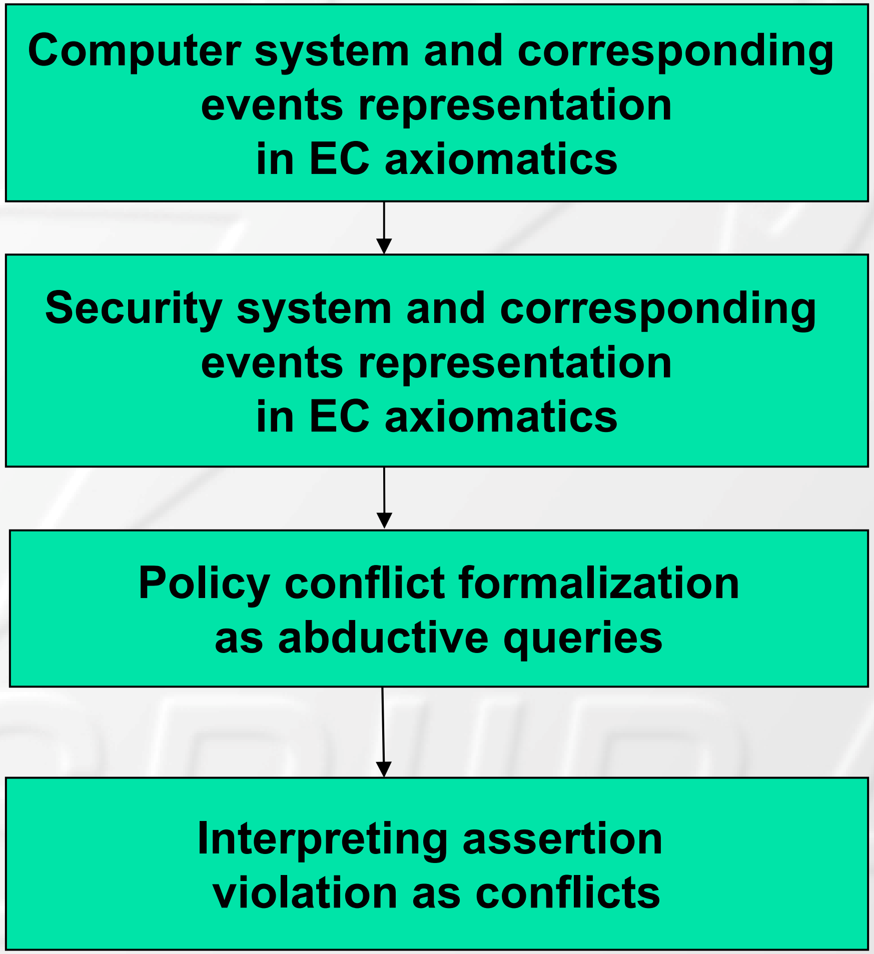
Model Checking VM: common architecture



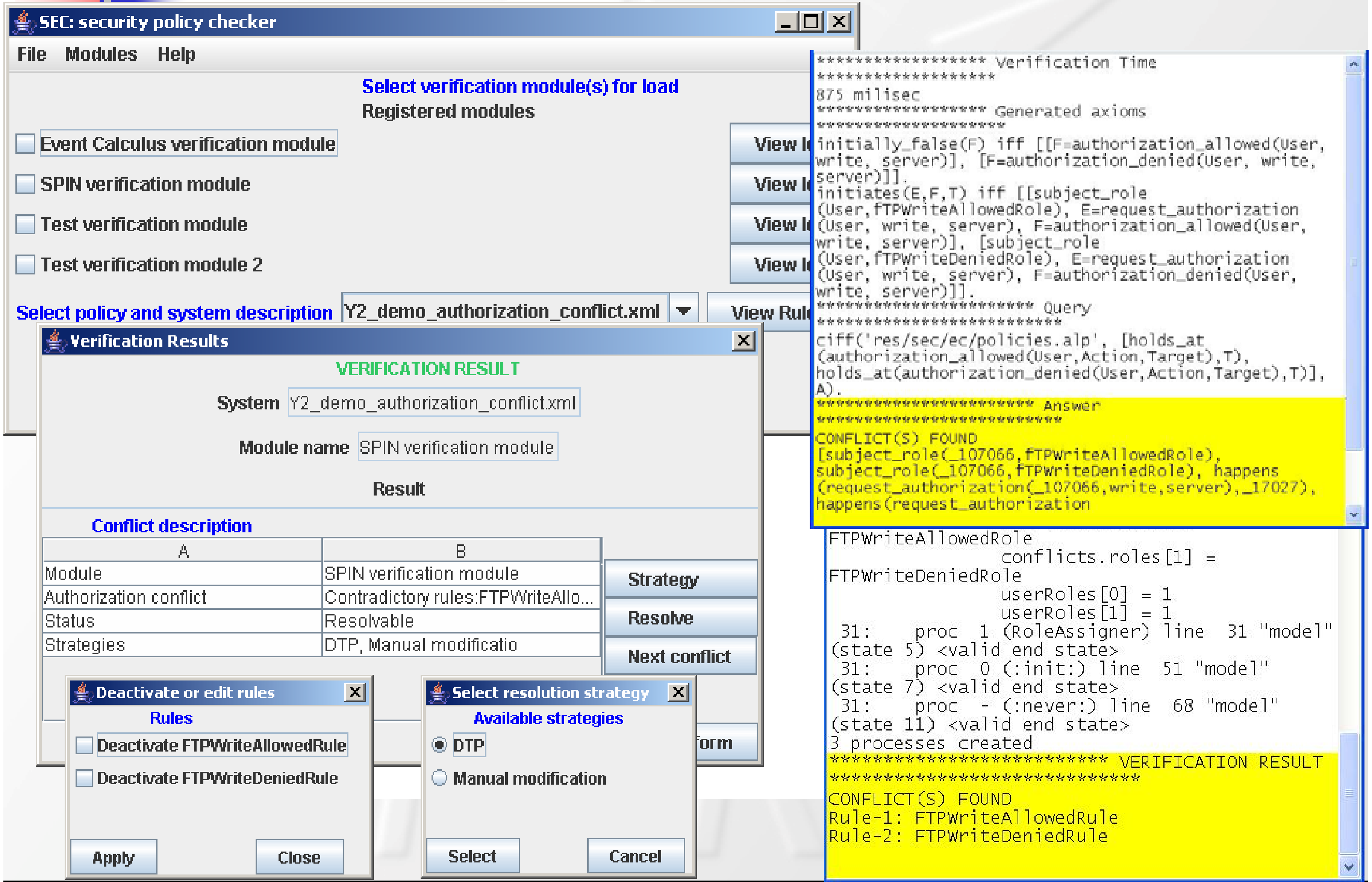
Event Calculus VM: functional model (IDEF)



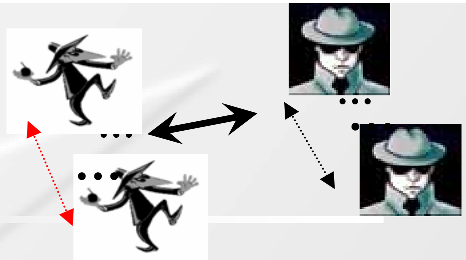
Event Calculus VM: technique



Security policy verification system GUI



Thanks!



For more information and
related publications please contact

Prof. Igor Kotenko

St. Petersburg Institute for Informatics and Automation
of the Russian Academy of Sciences
39, 14th Liniya, St. Petersburg, 199178, Russia
Telephone: +7-(812)-328-2642
Fax: +7-(812)-328-4450, +7-(812)-328-2642
E-mail: ivkote@iias.spb.su, ivkote@comsec.spb.ru
Web page: http://space.iias.spb.su/ai/kotenko/
http://www.comsec.spb.ru/kotenko/