Agent-based Network Security Simulation (Demonstration)

Dennis Grunewald Marco Lützenberger Joël Chinnow
Rainer Bye Karsten Bsufka Sahin Albayrak
DAI-Labor | TU Berlin | Ernst-Reuter-Platz 7 | 10587 Berlin, GERMANY
NeSSi@dai-labor.de

ABSTRACT

We present $NeSSi^2$, the Network Security Simulator, a simulation environment that is based on the service-centric agent platform JIAC. It focuses on network security-related scenarios such as attack analysis and evaluation of countermeasures. We introduce the main $NeSSi^2$ concepts and discuss the motivation for realizing them with agent technology. Then, we present the individual components and examples where $NeSSi^2$ has been successfully applied.

Categories and Subject Descriptors

I.6.3 [**Simulation and modeling**]: Applications; I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Security, Design, Experimentation

Keywords

AAMAS proceedings, Network simulation, Demo, Network security, Application-level simulation

1. INTRODUCTION

The design and development of security solutions such as Intrusion Detection Systems (IDS) is a challenging and complex task. In this process, the evolving system needs to be evaluated continuously. There are several ways to study a system or technology. The most accurate is the analysis of the deployed production system. However, in the case of IDS evaluation, real experiments incorporating attack scenarios cannot be done in an operational environment because the induced risk of failures such as service loss is too high.

For this very reason, evaluation is often carried out in small testbeds. Virtual machines are a solution for modeling mid-scale networks, but the representation of very large networks with thousands or millions of devices and links is out of scope. There exist scientific initiatives such as PlanetLab[1] providing computational resources to a larger extent. This is an important opportunity for researchers to evaluate network or security functionality, but although they provide detailed results, experiments are time consuming and remain complex to setup and maintain.

Another approach is to represent the system with the aid of mathematical models and find analytical answers, i.e. logical and quantitative relationships between the entities. Typically, such models also become very complex, in particular for a concurrent system such as IDS. Therefore, simulations are useful for the evaluation of distributed systems and protocols. Depending on the evaluation metrics, the simulations allow the abstraction from irrelevant properties. In addition, hazard scenarios, called "what-if scenarios", can be constructed which may not be possible in real-world test environments.

2. SOLUTION APPROACH

We introduce $NeSSi^2$, an agent-based simulation environment [3], providing telecommunication network simulation capabilities with an extensive support to evaluate security solutions such as IDS. In contrast to other network simulators, like e.g. NS-3 [2], $NeSSi^2$ also provides a comprehensive *detection API* for the integration and evaluation of IDS. In particular, special common attack scenarios can be simulated. Worm-spread scenarios and botnet-based DDoS attacks are only two of the supported example attacks. In addition, customized profiles defining the node behavior can be applied within the simulation.

$NeSSi^2$ is built upon the JIAC [1] framework, a service-centric agent-framework. The most recent version, JIAC V[2], is used in $NeSSi^2$. The network entities, i.e. routers, clients, servers, or IDS (*nodes* in the following) are simulated with the aid of JIAC agents. Dependent on configuration parameters and hardware characteristics, each agent simulates one or more nodes. $NeSSi^2$ is benefiting from agent technology in general and JIAC in special through the service-centric, modular and flexible approach to realizing distributed execution environments. In addition, a common semantic data model enables interoperability of agents executing even different simulation models at the same time.

This semantic *model* also incorporates the main modeling concepts for the creation and administration of simulations. The first concept and step to setup a simulation is the creation of the *network* topology. This topology can then be re-used for different *scenarios*. The scenario is comprised of elementary building blocks for each device in the network, the node *profiles*. They allow the customization of node

[1] `http://www.planetlab.org`

[2] `http://www.jiac.de/`

Cite as: Agent-based Network Security Simulation (Demonstration), Grunewald et al., *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1325-1326.

behavior to automatically generate traffic, simulate failures or apply network-based defense measures. Every profile consists of *applications*, representing mechanisms to be executed on an individual node, e.g. an attack, a detection mechanism or an application protocol such as HTTP. The sum of all profiles for a given *network* is called the *scenario*. In order to execute it, the length of simulation execution, the number of simulation *runs* and a recording configuration are configured within a *session*. As simulations often contain stochastical components such as distribution functions, e.g. the number/timing of HTTP-requests, multiple runs allow for the statistical analysis of mean values and standard deviations.

3. ARCHITECTURE

$NeSSi^2$ has been structured into three distinct components, the *graphical frontend*, the *agent-based simulation backend* and the *result database*. Each of these modules may be run on separate machines. The modular design facilitates the exchange of network topologies, scenario definitions and simulation results.

The *graphical frontend* of $NeSSi^2$ (c.f. Figure1) allows to create and edit the necessary components of a network simulation as described in Section 2. On the other hand, finished (or even currently executing, long-running) simulations can be retrieved from the database server and the corresponding simulation results are visualized in the GUI. Accordingly, there exist two different perspectives in the GUI, the *Network Editor* perspective for the creation of simulations as well as the *Network Simulation* perspective to investigate simulation results.

In the *backend*, different agent roles carry out the task of the parallel simulation execution. On each backend, i.e. separate machine, there exists the *Simulation Control Agent* (SCA) administrating access to the resources of the system as well as the interaction with the GUI. In this way, the SCA interacts with the individual *Network Simulation Coordination Agents* (NCAs). For every executed simulation *run*, an NCA is invoked which starts a number of *Device Management Agents* (DMAs). The number of DMAs depends either on particular user configurations, e.g. "one agent for every node", "x agents in total", or follows the computational power of the backend system, i.e. "one agent per CPU core".

Finally, the *result database* stores simulation results according to the configuration specified during the creation process of the simulation in the GUI. For every simulation *run*, the agents record selected events and traffic data to a specified log4j[3] appender which handles the output according to the recorder configuration. By default, the results – such as attack-related events – as well as the model are recorded to a database which allows for replaying the simulation. In addition, the recorded data can be used for evaluation purposes.

4. SUCCESSFUL UTILIZATION

$NeSSi^2$ has demonstrated its value in recent research and was employed as a simulation environment for various security-related approaches. In this regard, $NeSSi^2$ was used to investigate optimal placement strategies for IDS, analyze worm propagation strategies and evaluate the benefit of collaborative IDS. $NeSSi^2$ has also been used in lectures

[3] `http://logging.apache.org/log4j/`

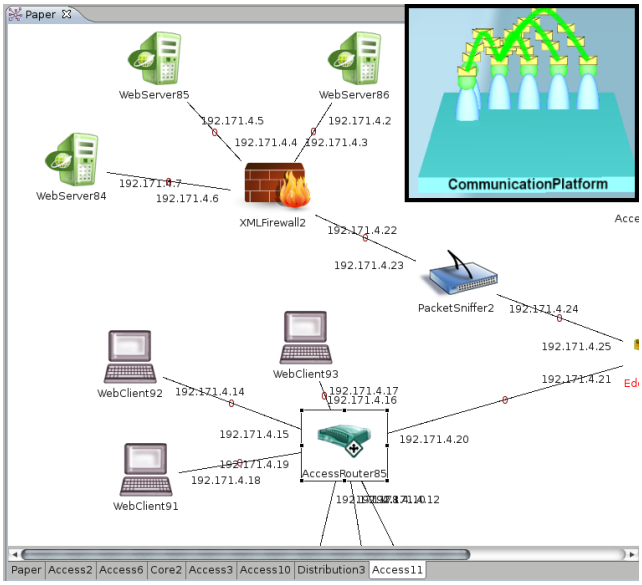

Figure 1: *GUI and Backend illustrated*: The GUI enables the creation and administration of arbitrary networks and node configurations. After the setup process is finished, an agent-based simulation backend ("CommunicationPlatform") executes the simulation and the results are stored in a database.

to generate attack data and evaluate detection algorithms implemented by students. In a recent industry research project, $NeSSi^2$ has been incorporated in an agent-based Decision Support System to forecast upcoming link congestions in the access network of a big German DSL-provider. $NeSSi^2$ is Open Source since January of 2009 and has been downloaded more than 6000 times.

5. CONCLUSION

We have presented $NeSSi^2$, a network simulation environment with a focus on security-related scenarios. The simulation backend is based on agent technology benefiting from the service-centric, modular and flexible design of the JIAC framework to load balance the complexity of the simulation runs. $NeSSi^2$ incorporates a semantic data model to reflect simulations of arbitrary networks and individual node configurations and has been used in various (industry) research projects as well as lectures. Related publications, documentation and source code can be looked up on the web site, c.f. `http://www.nessi2.de`.

6. REFERENCES

[1] B. Hirsch, T. Konnerth, and A. Heßler. Merging agents and services — the JIAC agent platform. In *Multi-Agent Programming: Languages, Tools and Applications*, pages 159–185. Springer, 2009.

[2] ns 3 project. NS-3 network simulator. `http://www.nsnam.org/docs/architecture.pdf`, last accessed on 02/24/2011.

[3] S. Schmidt, R. Bye, J. Chinnow, K. Bsufka, A. Camtepe, and S. Albayrak. Application-level simulation for network security. *SIMULATION*, 86(5-6):311–330, May/June 2010.