Self-Protecting Networks – How Cooperation Strategies Can Strengthen Network Security

Selbstschützende Netze – Netzwerksicherheit durch Kooperation

Tanja Zseby, Michael Kleis, Thomas Hirsch, Fraunhofer FOKUS, Berlin

Summary The Internet today is an insecure place. New attacks and vulnerabilities emerge every day, user education remains fragmentary and security solutions are only sparsely deployed. Protecting networks against new and unexpected attacks remains a challenging task, given the dynamic nature of network traffic. It is difficult to model or predict what to consider as normal behavior in current and future networks, which is a pre-requisite for detecting deviations from normality. In this paper we argue that to cope with these challenges network protection has to become an integral part of future networks. The evolving research field on autonomic communication provides a basis for introducing techniques for detection and defense directly into network nodes. Together with integration of learning methods, we see deployment of cooperation strategies as the main enabler for self-protection. ►►► **Zusammenfassung** Im heutigen Internet existieren eine Vielzahl von Bedrohungen. Täglich werden neue Angriffe und Schwachstellen registriert. Kenntnisse über Bedrohungen sind meist nur lückenhaft vorhanden und Sicherheitslösungen werden immer noch nicht ausreichend eingesetzt. Eine besondere Herausforderung stellt der Schutz vor neuen unbekannten Angriffen dar. Ein Grund dafür ist die hohe Dynamik und schlechte Vorhersagbarkeit von Netzwerkverkehr. Diese erschwert es, ein Normalverhalten in heutigen und zukünftigen Netzen zu modellieren, eine Grundvoraussetzung, um Abweichungen vom Normalfall zu erkennen. Der Schutz vor Angriffen muss ein integraler Bestandteil in zukünftigen Netzen werden, um sich diesen Herausforderungen zu stellen. Das noch junge Forschungsgebiet Autonomic Communication stellt eine gute Basis bereit, um Techniken zur Angriffserkennung und Verteidigung direkt in Netzwerk-Knoten zu integrieren. Neben der Integration von Lernverfahren sehen wir den Einsatz von Kooperationsstrategien als Schlüsseltechnologie für die Realisierung von selbstschützenden Netzen.

KEYWORDS C.2.3 [Computer Systems Organization: Computer-Communication Networks: Network Operations] network management; network protection, security, trust, cooperation

1 Introduction

Ever since the advent of communication networks, attackers have tried to disturb, interrupt and destroy network operations and their attacks are steadily increasing in quantity and quality. Classical methods of attack prevention such as authentication, secure protocols, firewalls and signature-based detection methods are no longer sufficient to cope with upcoming threats.

The deployment of new technologies and protocols, the increasing heterogeneity of current networks at all layers and the dynamic nature of network traffic generate an evolving environment under permanent evolution where it is hard to define which patterns can be considered as normal, and what can be construed as potentially harmful deviations from normality.

Autonomic Communication (AC) provides a framework for the introduction of self-management and self-protection capabilities in future networks. As an extension of classical methods of attack prevention, autonomic communication

includes concepts of learning and node cooperation. It is evident that future networks need to incorporate such techniques in order to cope with the growing number of challenges in the security area.

2 Autonomic Communication

Autonomic communication aims to move decision-making processes into the network in order to automate network management and to support application requirements. One much cited decision cycle for human decision making is the OODA-Loop. OODA stands for *Observe-Orient-Decide-Act* and describes the human decision cycle during battle situations [19].

To automate the management of complex systems, IBM started to apply decision cycles to computer systems under a paradigm called autonomic computing [1]. IBM defines autonomic managers that implement an intelligent control loop consisting of four function blocks similar to OODA: Monitor, Analyze, Plan, and Execute (MAPE). In [2] Brent Miller introduces self-configuration, self-healing, self-optimization, and self-protection as attributes for autonomic computing architectures. Self-protection is defined as the ability of a system to "anticipate, detect, identify, and protect against threats" [1].

Autonomic communication applies this idea to communication systems. In [20] the authors present an autonomic control loop along the lines of the MAPE concept in autonomic computing. Within the field of autonomic communication the cooperation among multiple nodes within networks or different domains provides a major opportunity for improving network security.

3 Problem Statement

Cooperation provides two advantages for intrusion detection: a) sharing of viewpoints and b) sharing of resources.

Components located at different places in the network have access to different views of the network situation. Some network nodes are specialized (e. g., DNS or AAA servers, BGP speakers, measurement nodes) and have access to specific data that is not seen by all nodes. Combining viewpoints helps to provide an overall picture of the network situation.

The second advantage is the ability to share resources. A significant limiting factor in anomaly detection is the amount of resources available for measurement and data analysis. Cooperation strategies can distribute resources in such a way that each node takes over tasks according to its capabilities to accomplish a joint mission.

But cooperation does not come for free. It requires a communication infrastructure, incentives to cooperate, and a certain level of trust in the behavior of the other nodes. A solution for cooperation in network protection should be scalable and is also subject to timing requirements with regard to the decision process.

Reaching joint decisions is a further challenge in a system of multiple interconnected control loops. Policy conflicts, and inconsistent information may complicate the problem. Furthermore, the protection of privacy may prevent information sharing across networks.

A final but nonetheless critical point is protection of the cooperation system against communication failures, malicious or faulty cooperation partners and dedicated attacks.

In Section 4 we discuss the different phases of the decision cycle at which cooperation can take place. In Section 5 we define enablers to achieve cooperation and show how different cooperation strategies use these enablers to provide solutions for network protection.

4 Cooperation Strategies

Intrusion detection can be separated into signature- and anomaly-based approaches. Signature-based approaches compare the current network traffic to a previously stored attack pattern. Such methods cannot cope with new, so called "zero-day" attacks unlike anomaly detection methods which model the normal state of a network and detect deviations from this behavior. Statistical methods, machine learning and data mining techniques are typically used to learn what can be considered as normal behavior [4].

But anomalies can also originate from legitimate changes in user behavior. Thus anomaly detection systems often report many false positives.

Most commercial Intrusion Detection Systems (IDSs) such as Cisco Guard rely on signature detection. They provide anomaly detection only as an advisory system to administrators. As described below, systems can cooperate at different phases of the decision cycle.

4.1 Observe

In contrast to signature-based methods, in anomaly detection it is less clear which metrics will reflect relevant deviation from normality. Originally designed for classical large operator networks, most IDSs have been realized as closed systems, with a central management entity.

For multi domain, Peer-to-Peer, Mobile Ad-Hoc or similar scenarios, an IDS must be prepared to operate without central coordination. For the first step of the decision cycle, distributed measurements should identify related events across network probes.

Multi-point measurements are essential to get a network wide view and are also required to calculate certain metrics such as one-way delays or routes. Existing tools face the challenges of clock synchronization, and accurate correlation of packet events at different observation points (e. g., [8–10]).

A sophisticated combination of data selection techniques with multi-point measurements ensures that the same packet is selected at different points. Hash-based selection techniques are proposed in [9]

and [11] that aim at an emulation of random sampling.

4.2 Orient

The second step is to create a situational view, which provides the information needed to achieve the systems objectives. Cooperation can provide additional data and enable joint analysis.

4.2.1 Sharing Information

Various information sharing strategies help to improve a nodes situation awareness and thus support the decision process. In [13] a system is proposed where neighboring nodes may be searched for specific intrusion detection events. Several context metrics, like validity and significance of the data, or distance of the reporting node and current work load are computed.

In [27] an infrastructure is described that enables the sharing of arbitrary network information among nodes. It also enables triggering of measurements to improve the situational view.

Context information helps to extend the network view. This includes data from different network layers, about the network environment such as geo location, user behavior or external events. Information from network services (e. g., DNS or AAA server) can further improve defense strategies [12].

Sharing information among network operators is a more difficult challenge. It can help to better identify an attack, to track an attacker faster and to isolate the source of the attack. But privacy and secrecy concerns make sharing of network data difficult as it can reveal information about network structure, users or vulnerabilities to competitors or potential attackers.

This problem is present at all levels. Incidents are often not reported, for fear of negative publicity. This is why the former IETF working group on Extended Incident Handling (INCH) developed the IODEF format to share information about security incidents.

Another attempt was made to standardize a Real-time Inter-Network Defense (RID) protocol to enable joint incident handling. Some implementations for sharing information were indeed engineered (e. g., Automated Incident Reporting AirCERT [14]), but due to a lack of supporters the INCH working group was closed in October 2006.

SPRINT also attempted to standardize an architecture for data sharing among operators, yet it failed to build an IETF working group around this topic. The majority of operators have not yet recognized the importance of sharing incident information.

4.2.2 Distributed Analysis

Delegating analysis tasks facilitates utilization of free resources, either centrally controlled or decentralized and means that data analysis tasks may be shared between entities. Furthermore, suspicious patterns can be forwarded to dedicated analysis components for further inspection. Commercial IDS, such as the Cisco Anomaly Detector, are a first step towards specialization of network components in a domain. In their systems, anomalous traffic detected by the Anomaly Detector in the network is forwarded to a more specialized DDoS Mitigation component, the Cisco Guard [29].

4.3 Decide

In conventional IDS, a central instance decides whether an incident has occurred. In the decentral case any node that locally detects an intrusion or anomaly with a strong evidence can trigger a response. If evidence is weak, communication with neighbor nodes may be required for broader investigation [5].

An important aspect of such joint decision-making is timely access to the required information. Decision-making algorithms need fallback solutions for cases where the required information is not available at the time when a decision is needed. Participation of multiple nodes in the decision-making process increases interdependencies and makes the process more complex.

4.3.1 Election of a Leader

The cooperative decision problem may be reduced to the centralized case. For ad hoc networks the authors of [6] propose to combine clustering with a periodic leader re-election in each cluster. The actual monitoring is performed by cluster members which propagate prefiltered monitoring results to the leader for analysis. If the cluster leader detects an intrusion it can coordinate the response.

4.3.2 Voting

A less centralized approach is offered by voting systems. For the case of sensor networks, [21] describes a voting system that can be realized without a priori knowledge of node behavior. Each sensor is able to observe its neighbors' activities and defines majority sensor behavior as "normal", based on its own local view. If one of its neighbors shows abnormal behavior, the observing sensor starts a voting process and presents evidence to its neighbors. Intruders are identified by a large number of negative responses.

4.3.3 Emergent Behavior

To address the cooperative monitoring problem, the authors of [7] study an emergent behavior based collaborative information processing strategy to address the cooperative monitoring problem. Using the model of ant colonies, the actual monitoring results are translated into a pheromone concentration. Thus a path of intrusion in the sensor network can be identified by its pheromone concentration.

4.4 Act

With an approach based on clustering and leader election the actual response to an attack can be initiated and coordinated by the corresponding cluster leader. Possible actions initiated by the cluster leader are: a) traffic blocking (e. g., adjustment of filter rules) b) traffic redirection

or c) elimination of infected systems and services.

A different kind of response strategies is proposed in [15]. The presented "currency approach" is based on the principle that service is provided only when a client pays for it with some form of currency. In the DDoS example such currency is bandwidth. The speak-up scheme is based on the assumption that attackers are already using a significant portion of their upload bandwidth for attacks while on the other hand standard clients have spare upload bandwidth. To exploit this fact a speak-up enabled server encourages all of its clients to speak up (i. e., invest more of their currency bandwidth into the session). Unlike attackers clients can follow this request, thus good clients crowd out the bad ones.

5 Enablers

In this section we define a set of enablers for cooperation between a group of actors. Starting from considerations about the requirements for the underlying communication infrastructure we describe trust, careful selection of incentives and emergent behavior as enablers for cooperation.

5.1 Communication

Communication is a pre-requisite for cooperation. But it also makes the cooperation system highly vulnerable. Unreliable communication complicates cooperation. Attacks on the communication infrastructure can fool the cooperating parties, hide attack activities or invoke unnecessary counteractions that bind system resources. Thus communication methods that facilitate network protection have themselves to be extremely well protected against attacks.

With large scale distributed systems spanning multiple administrative domains this is a challenging task since interactions between different domains are typically untrustworthy, and subject to business and privacy concerns. However, the required interactions may be regulated via Service Level Agreements (SLA) or protected by Virtual Private Networks. In addition Overlay Based Systems can operate without regard to physical structure. Their interaction is typically peer-based, or centered around certified trusted servers.

5.2 Trust

Any cooperation requires a certain level of trust in the assumption that the partners will act in a defined way. In many scenarios, this assumption must hold true both for past and future actions. In [18] trust is defined as "the subjective probability with which an agent assesses that another agent (…) will perform an action, both before he can monitor such an action and in a context in which it affects his own action." Trust can be justified by the existing administrative or contractual setting between actors. If those do not exist, trust has not to be established by the system itself. In the following we describe different means to measure and to establish trust and how they can be applied to distributed anomaly detection.

5.2.1 Trust Models

The most immediate approach to quantify the relationship of trust that can be put into other components comes from the definition of trust metrics. These metrics commonly take the form of a probability value, expressed in the unitless range $[0, 1]$. An actor would therefore attempt to compute and store a number of trust values, which represent his expectations about the success of certain cooperations.

Note that trust in this context is a subjective representation of the objective, yet unknown trustworthiness of the other actor [22]. It is the objective of every actor to increase, maintain, and correctly represent its own trustworthiness [23].

Furthermore, trust should not be confused with risk. As shown in [22] the two have only a weak correlation. The risk of an action failing may be high or low, independently of the trust placed in the actor. However, the trust and risk values, as well as the cost of a failure are the decisive factors that determine whether a node should engage in cooperation, and how it should act in response to information transferred from other parties.

5.2.2 Trusted Thirds

At the end of the day broad trust can be replaced by control in situations where it is not justified. An actor might have to validate all externally retrieved information using his own means. A more convenient means of ensuring control is to disclose necessary information to a third party who is trusted by all the involved actors. Trusted third parties however create a hierarchical structure and a security bottleneck, as we show below.

One way of measuring the inherent security of a system is to determine the size of the attack vector. The attack vector defines how many determined attackers it takes to compromise the system. In terms of a third party, this size equals one. If the trusted party decides to break the system, it is perfectly able to do so. Thus the system is not secure by design, but – hopefully – by mechanisms said to be established "out of band". Requiring trust in third parties has no computational cost. Where incidents are rare such trust is simple to establish. Providing actual protection is not however, and incidents have a tendency to be devastating for the system [24].

5.2.3 Distributed Security

Consequently, distributed security mechanisms aim to establish control by other means. Where algorithms cannot provide perfect security by design, it is argued that cooperation with untrustworthy actors can provide sufficient security [25]. The prime assumption is that each group of cooperating actors consists of a number of benevolent and malicious nodes. By algorithmically increasing the requirements for the

attack vector, secure computation is possible within given limits. Distributed Security is therefore measured by the following metrics: the size of the attack vector and the size of the access vector. The latter specifies how many actors are necessary and sufficient to perform a single transaction.

When related to the speed of communication between actors, these numbers eventually yield the number of transactions which can be processed per unit of time. The design of secure distributed systems aims at providing a reasonable number of transactions each time, while maintaining the given size of the two control vectors. Hence, the enabling technology for distributed security applications is in terms of the increased speed of communication and complex information processing available in the Internet. With dwindling information transfer delays, higher replication rates can be reached [26]. More actors can provide additional control for each single action in a system.

5.3 Incentives

In terms of a centrally administrated network, cooperation can be enforced by operators or administrators. In systems without central control other incentives have to be provided for cooperation. One assumption made by several "Incentive Systems" [16] is that nodes or people have an incentive to participate, e. g., in a cooperative anomaly detection if the utility they derive from cooperation is higher than the cost of joining the system. In asymmetric systems, wherein the set of contributors is different from the set of beneficiaries the situation can be more complex. From the viewpoint of cooperative anomaly detection such a disparity between contributors and beneficiaries can occur when a set of nodes is constantly monitoring network traffic but is never attacked. The Lottery Trees [16] incentive system covers such cases by using the principle of a lottery, where the lottery winner is selected among the participating nodes in a way that encourages contributions to, and growth of the system.

Game theory provides methods to model and analyze situations in which the decisions of multiple actors influence each other. One example is "Tit-for-Tat", first introduced by Anatol Rapoport.

Tit-for-Tat is an interaction strategy where an agent will initially cooperate, then respond in kind to an opponent's previous action. If the opponent previously was cooperative, the agent is cooperative. If not, neither is the agent. One of the most compelling demonstrations of the potential of the Tit-for-Tat approach has been the success of BitTorrent [17]. In BitTorrent no centralized resource allocation to regulate file downloads is made. Instead each peer is responsible for maximizing its own download rate. To do this, a peer tries to download from as many sources as possible. It also decides which peers it should cooperate with (by uploading content) and not to cooperate with (by temporarily refusing to upload). The "choking algorithm" used in BitTorrent is based on a variant of Tit-for-Tat and attempts to archive pareto efficiency.

5.4 Emergent Behavior

"Emergent behavior" can be seen as an orthogonal approach to the ones mentioned above. As emergent behavior we consider phenomena where a group of entities interact without central control and each entity in the system behaves according to "simple" rules (microscopic behavior) which engenders "sophisticated" behavior in the overall group (macroscopic behavior). Examples of such phenomena in biology are ant or bee colonies.

6 FOKUS Activities

As one strand of its research in the field of Autonomic Communication, Fraunhofer FOKUS is developing a Node Collaboration System (NCS) which serves to realize autonomic communication solutions using different cooperation strategies for network protection.

Its basis is an information sharing platform described in [27] and used to establish situation awareness for network nodes. The system allows nodes to request measurement results and network context information from neighbors and also allows for the invocation of new measurements if needed.

Information can be transferred using standard protocols like IPFIX [30]. The platform serves as the foundation for a set of distributed security applications, such as a Distributed Context-Aware Firewall (D-CAF). To enable trusted information sharing, in March 2008 the EU project PRISM [32] was initiated to investigate privacy-preserving measurement methods.

7 Conclusions

In this paper we have discussed cooperation strategies and the opportunities they present for strengthening network security.

We reviewed the benefits and challenges of cooperation in the security context and analyzed proposed cooperation strategies for Intrusion Detection Systems. We also enumerated a set of enablers for cooperation between network nodes.

We argue that cooperation is a key requirement for the protection of large scale and decentralized systems. As an integral part of autonomic communication solutions we fully expect that cooperative strategies will indeed prove to be a key component in the development of self-protecting networks.

References

[1] IBM. An architectural blueprint for autonomic computing. White paper, IBM, 2006.

[2] B. Miller. The autonomic computing edge: Can you chop up autonomic computing? Technical report, IBM, 2008.

[3] J. Coppens, S. de Smet, S. van den Berghe, F. de Turck, and P. Demeester. Performance evaluation of a prob-

abilistic packet filter optimization algorithm for high-speed network monitoring. In: *HSNMC*, pp. 120–131, 2004.

[4] A. Patcha and J.-M. Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. In: *Comput. Netw.*, 51(12):3448–3470, 2007.

[5] Y. Zhang and W. Lee. Intrusion detection in wireless ad-hoc networks. In: *MobiCom '00: Proc. of the 6th Annual Int'l Conf. on Mobile computing and networking*, pp. 275–283, New York, NY, USA, 2000.

[6] Y. A. Huang and W. Lee. A cooperative intrusion detection system for ad hoc networks. In: *SASN '03: Proc. of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 135–147, New York, NY, USA, 2003.

[7] S. Banerjee, C. Grosan, A. Abraham and P. K. Mahanti. Intrusion Detection on Sensor Networks Using Emotional Ants. In: *Int'l Journal of Applied Science and Computations*, USA, vol. 12, no. 3, pp. 152-173, 2005.

[8] I. D. Graham, St. F. Donnelly, S. Martin, J. Martens, and J. G. Cleary. Nonintrusive and accurate measurement of unidirectional delay and delay variation on the internet. In: *INET*, 1998.

[9] N. G. Duffield and M. Grossglauser. Trajectory sampling for direct traffic observation. In: *SIGCOMM*, pp. 271–282, 2000.

[10] T. Zseby, S. Zander, and G. Carle. Evaluation of building blocks for passive one-way-delay measurements. In: *Proc. of Passive and Active Measurement Workshop (PAM 2001)*, Apr. 2001.

[11] T. Zseby, M. Molina, N. Duffield, S. Niccolini, and F. Raspall. Sampling and Filtering Techniques for IP Packet Selection, June 2007. Internet Draft, work in progress.

[12] T. Zseby, E. Boschi, N. Brownlee, and B. Claise. IPFIX Applicability, June 2007. Internet Draft, work in progress.

[13] T. Gamer, M. Scharf, and M. Schöller. Collaborative Anomaly-based Attack Detection. In: *Proc. of 2nd Int'l Workshop on Self-Organizing Systems (IWSOS 2007)*, LNCS, pp. 280–287, English Lake District, Sep. 2007.

[14] AirCERT. http://aircert.sourceforge.net/.

[15] M. Walfish, M. Vutukuru, H. Balakrishnan, D. Karger, and S. Shenker. Ddos defense by offense. In: *SIGCOMM Comput. Commun. Rev.*, 36(4):303–314, 2006.

[16] J. R. Douceur and T. Moscibroda. Lottery trees: motivational deployment of networked systems. In: *SIGCOMM '07: Proc. of the 2007 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp. 121–132, New York, NY, USA, 2007.

[17] B. Cohen. Incentives build robustness in bittorrent. Technical report, bittorrent.org, 2003.

[18] D. Gambetta. *Can We Trust Trust?* Basil Blackwell, 1988. Reprinted in electronic edition from Department of Sociology, University of Oxford, chapter 13, pp. 213–237.

[19] J. R. Boyd. An organic design for command and control. In: A Discourse on Winning and Losing. Unpublished lecture notes, http://www.d-n-i.net/dni/about/john-r-boyd/, 1976.

[20] S. Dobson, S. Denazis, A. Fernández, D. Gaïti, E. Gelenbe, F. Massacci, P. Nixon, F. Saffre, N. Schmidt, and F. Zambonelli. A survey of autonomic communications. In: *ACM Trans. Auton. Adapt. Syst.*, 1(2):223–259, 2006.

[21] F. Liu, X. Cheng, and D. Chen. Insider attacker detection in wireless sensor networks. In: *INFOCOM 2007: 26th IEEE Int'l Conf. on Computer Communications. IEEE*, pp. 1937–1945, May 2007.

[22] Why trust is not proportional to risk. *ARES 2007: Proc. of the Second Int'l Conf. on Availability, Reliability and Security*, pp. 11–18, Apr. 2007.

[23] A. Josang, C. Keser, and Th. Dimitrakos. Can We Manage Trust? In: *Proc. of the Third Int'l Conf. on Trust Management*, *LNCS*, pp. 93–107. 2005.

[24] N. Szabo. Trusted third parties are security holes. Online essay, 2001–2005. http://szabo.best.vwh.net/ttps.html.

[25] C. Georgiou and A. A. Shvartsman. *Do-All Computing in Distributed Systems*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[26] M. Hirt, U. M. Maurer, and B. Przydatek. Efficient secure multiparty computation. In: *ASIACRYPT '00: Proc. of the 6th Int'l Conf. on the Theory and Application of Cryptology and Information Security*, pp. 143–161, London, UK, 2000.

[27] D. Witaszek and J. Tiemann. Context Dissemination System: Requirements, Architecture and Ability to Support Measurement Results. *Technical Report TR-2008-0130*, Fraunhofer FOKUS.

[28] L. Olsson. Anomaly detection using self/nonself discrimination for the linux kernel.

[29] CISCO Systems Cisco Anomaly Guard Module, Official Product Page. http://www.cisco.com/en/US/products/ps6235/index.html. May 15, 2008.

[30] B. Claise (Ed). Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information. Request for Comments: 5101 IETF, Jan. 2008.

[31] J. Kang, Y. Zhang, and J.-B. Ju. Classifying ddos attacks by hierarchical clustering based on similarity. In: *Proc. of the Int'l Conf. on Machine Learning and Cybernetics*, pp. 2712–2717, Aug. 2006.

[32] PRIvacy-aware Secure Monitoring (PRISM) Project. http://www.fp7-prism.eu/. Aug. 2008.


1


2


3

1 Dr.-Ing. Tanja Zseby received her degree in Electrical Engineering from the Technical University of Berlin in 1997 and her Ph.D. degree (Dr.-Ing.) in 2005. She joined Fraunhofer FOKUS as a scientist in 1997. In 2003 she was appointed head of the Competence Center for Measurement Technologies.

Since 2005 she has been head of the Competence Center for Network Research (NET). Her research interests are novel concepts for the self-protection and self-management of computer networks as well as network analysis with statistical sampling techniques.
Address: Fraunhofer FOKUS, Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany,
Tel.: +49-30-34637153,
Fax: +49-30-34638153,
E-Mail: tanja.zseby@fokus.fraunhofer.de

2 Dipl.-Math. Michael Kleis received his degree in Mathematics from the University of Saarbrücken (Germany) in 1999. He joined FOKUS in August 2001 and worked in the fields of signaling protocols for multimedia streaming, QoS support for multimedia traffic and error correction techniques. His current research activities are in the fields of peer-to-peer and overlay networks with an emphasis on autonomic networking principles and service composition.
Address: Fraunhofer FOKUS, Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany,
Tel.: +49-30-34637121,
Fax: +49-30-34638000,
E-Mail: michael.kleis@fokus.fraunhofer.de

3 Dipl.-Ing. Thomas Hirsch joined FOKUS in December 2001. He works in the Competence Center for Network Research (NET) where he manages projects on NetCentric Security and Critical Infrastructure Protection. Previously, he worked on European and industry research projects with a focus on measurement and security. Before his Master studies he worked as manager of a small Network Service Company. His research work focus is on distributed and autonomic application layer services for Measurement, Intrusion Detection, QoS and AAA. His key interest is in developing self-awareness and collaboration infrastructures in non-hierarchical networks.
Address: Fraunhofer FOKUS, Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany,
Tel.: +49-30-34637287,
Fax: +49-30-34638000,
E-Mail: thomas.hirsch@fokus.fraunhofer.de