Early Warning and Intrusion Detection based on Combined AI Methods

Stefan Edelkamp and **Carsten Elfers** and **Mirko Horstmann** and
Marcus-Sebastian Schröder and **Karsten Sohr** and **Thomas Wagner**
TZI, Universität Bremen

Abstract

In this paper we survey the architecture and AI aspects in our project on early warning- and intrusion detection based on combined AI methods. We address the problem of alarm assessment in intrusion detection and use plan reconstruction based on hierarchically organised procedural knowledge that contains descriptions of adversary actions. Reconstructed plans are supposed to correlate events and alarms from a SIEM and provide explanations for a security expert. We also aim at predicting the next steps of multi-stage intrusion attacks in computer networks. Therefore a probabilistic relational reasoning over time method based on hidden Markov models is proposed.

Introduction

The project *Early Warning and Intrusion Detection System Based on Combined AI Methods* (FIDeS) funded by the German Ministry of Research and Education (BMBF) aims at developing an advanced, intelligent assistance system for detecting attacks from the Internet both in local area networks and in wide area networks as early as possible. Within the framework, not only widely-used Internet protocols such as FTP, SMTP, and HTTP shall be considered, but also newer protocols such as telecommunication protocols, and SOAP. This also allows the early warning systems to detect attacks on Internet nodes which may originate from mobile devices. In addition, fraudulent access in security-critical, IT-based business processes of enterprises will be detected.

Conventional IDS and in particular IDS for anomaly detection usually produce a high false positive rate or do not detect all attacks (false negatives). Complementary to anomaly-based IDS, we develop an early warning system based upon heterogeneous methods of Artificial Intelligence (AI). This system supports a security officer in analyzing attacks and carrying out appropriate counter measures. Consequently, the project FIDeS focuses more on assistance (such as concrete instructions in case of an attack) rather than on mere intrusion detection. For this purpose, various AI-based methods are employed such as declarative knowledge representation, the generation of explanations, and cognitive assistance. However, the integration with an anomaly-based IDS is also envisioned.

The system addresses the following capabilities:

- availability of plausible (comprehensible) explanations for attacks,
- declarative representation of knowledge on attacks, systems to be attacked, and system components as well as counter measures,
- interactive assistance on the execution and selection of counter measures for attacks,
- forecast of the future course of an attack (including a plausibility check for the forecast),
- scalability of explanations and forecasts (depending on the expected risk),
- provision of a knowledge base (containing descriptions of attacks, counter measures, and instructions),
- maintainability and comprehensibility of the knowledge about attacks and counter measures,
- extensibility to timely react on new types of attacks.

In addition, a simulation tool will be developed that creates attack scenarios under realistic time- and system constraints. This simulation tool will be used to validate the functionality and the coverage of the early warning system.

Last but not least, privacy requirements are taken seriously during the entire development process of the project prototype. For example, a provider is not allowed to examine the data packets due to privacy laws.

In the intrusion detection research serveral methods have been proposed to either detect anomalies in host-sessions by learning the normal behavior of users (Garcia-Teodoro *et al.* 2009), or to detect intrusions by specified attack signatures in network-packets (J. M. Gonzalez 2007). It is a challenging task to improve the quality of intrusion detection by avoiding false positives regarding the detection rate. For this purpose, some development aims to the fusion of multiple and heterogeneous sensors, the so called *Security Incident and Event Managers* (SIEM), e.g., Prelude[1], OSSIM[2], and the ArcSight SIEM Platform[3].

In contrast to these systems, our aim is to develop a system that uses machine-learning methods to improve the

[1] http://www.prelude-ids.com
[2] http://www.ossim.net
[3] http://www.arcsight.com

quality of detecting network attacks and to support the users (IT specialists) with an enriched assistance in this scenario with the usage of event data offered by a SIEM. Therefore, we propose a system that is divided into three core areas: Detection and explanation of attacks, prediction of attacks and assistance for the user to react on a detection.

Intrusion detection alarms take a lot of effort to investigate. Together with the large number of false alarms that distract from the real threats, this is a fundamental problem of IDS in general. To address the problem of too many false alarms, a number of projects like REMIND (Rieck & Laskov 2007) have introduced machine learning techniques within IDS. However, the problem of tracking and eliminating the underlying causes of alarms remains.

To understand network attacks, it is a well established idea to look at the network from an attacker's perspective. Depending on the specific network topology and given environment, an attacker will usually exploit several vulnerabilities in succession. One way to describe multi-step adversary actions is the use of attack trees as first described by Schneier (1999). With a different focus, Templeton & Levitt (2001) introduced a model that describes atomic attack components in terms of their preconditions and postconditions.

Architecture of the System

Figure 1 depicts the architecture of the FIDeS system[4]. On the lowest layer of the system, the data traffic of a network is tapped and analyzed by a SIEM and other external tools, such as the Internet Analysis System (IAS)[5]. These tools usually provide their gathered data in real-time or save it into their own databases. Thus, a normalization layer unifies the access to the various sensors and tools used for data capture. For this, the IDMEF data format[6] is used to provide the other FIDeS components with uniform data regardless of its source. Another aim of the normalization layer is to enable its users to access sensor states from arbitrary points of time in the past, which makes it possible to replay attack scenarios and provides forensic integrity to the system.

The KB manager situates the expert knowledge. The emergency management is responsible for processing important incidents directly, while the context management maintains interfaces for different user groups. The normalization layer is accessed by the various modules of the application control layer. These modules form the AI-plugin layer, which contains replaceable parts for attack analysis and attack prediction. Additionally, an assistance component mediates between the AI modules and both user interactions as well as notifications from other parts of the system such as those generated by the emergency management module. Finally, the context management module provides information on how to represent the status of the system to users of a given competence level, while the knowledge base manager provides an uniform interface to various knowledge bases used by the AI modules.

[4]http://www.fides-security.org

[5]http://www.internet-sicherheit.de/IAS

[6]http://www.ietf.org/rfc/rfc4765.txt

FIDeS uses a web application for data visualization to users. The advantage of this approach is the instant availability of the application on any computer connected to the internet. Even from system on which a user has no rights to install software, such as computer pools in hotels or internet cafes, a user can quickly and easily check the system status and react to incidents from there.

A message queue is used to connect the various parts of the system with each other. The Advanced Message Queuing Protocol (AMQP)[7] is designed to be a platform- and programming-language-independent standard for creating a messaging middleware. O'Hara (2007) details the objectives for creating AMQP as well as the functionality of the protocol in. Using AMQP allows us to both write the individual components in the programming language most suited for the task as well as increase system performance by decentralizing the system onto multiple machines. We have chosen the Apache Qpid[8] implementation of AMQP since it is an actively maintained project which keeps up with updates of the standard and provides client libraries for the programming-languages that are used for FIDeS.

Tracing and Explaining Attacks using Planning Technology

Provided a model of a security domain, planning can be executed to trace an attack an network. Moreover, attack trees with finite branching can be realized in a planning domain description language.

In extension to the STRIPS formalism (Fikes & Nilsson 1971) for describing planning domains, the SAS$^+$ formalism (Helmert 2004) uses partial multi-valued state variables instead of propositional atoms. An SAS$^+$ stucture $M = (V, S, O)$ is defined by a set of state variables $V = (v_1, \ldots, v_m)$, defining a space $S = S_1 \times \ldots \times S_m$ of all possible states, where S_j is the domain *Domain*(v_j) of mutually exclusive values for the jth variable, $j = 1, \ldots, m$. Operators change assignments to states according to their pre- and postconditions. Preconditions are Boolean formulas over variable assignments and postconditions are updates of variables to new values. Partial states are states with some variable values being undefined.

Diagnosing an Attack

In diagnosis, we are not only concerned with detecting attacks, but additionally with explaining them. This is done by propagating the error in the model and probing on more and more specific issues. Since a diagnosis task is a search in a space of different hypotheses on the values of variables, it deals with uncertainties in background knowledge.

For multiple faults assumption-based truth maintenance systems (ATMSs) have been suggested (Forbus & de Kleer 1993). Their model is an undirected network with the edges labeled with discrete variables, whose values are of a certain range. Devices in the network to be diagnosed manipulate and propagate the information found at incident edges. They

[7]http://www.amqp.org

[8]http://qpid.apache.org

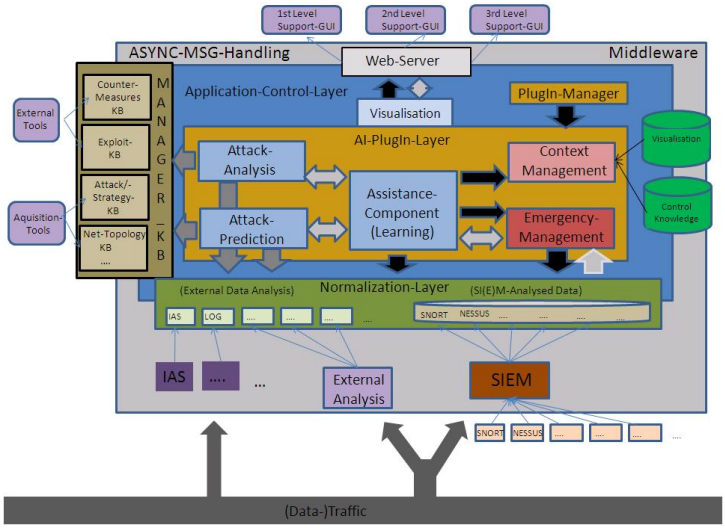

Figure 1: The Architecture of the FIDeS System.

represent the qualitative knowledge about and the influence the variables have on each other.

Abduction of an Attack

The problem of generating abductive explanations is divided into two subproblems that can be addressed separately:

- generating the set of all possible explanations, and
- selecting the most appropriate hypothesis among the set of possible explanations

For a logical theory T and some manifestation M of a set of individual hypotheses, we are interested in Δ such that $T \cup \Delta \models M$. We solve the abduction problem by using planning technology. The domain theory T is encoded in the planning operators O. Next, we construct a transition relation T_o, encoding all (predecessor, successor) state pairs valid under operator $o \in O$. This yields the domain theory $T = \bigvee_{o \in O} T_o$. Logical subsumption $\phi \models_T \psi$ has semantics that there is a sequence of operators applied to ϕ, which entails ψ. In ordinary SAS$^+$ planning the initial state is total, while the goal state is partial. For abduction, however, both states are partial. The specified part of initial state denotes the assumptions, one possible completion is a hypothesis. The specified part of the goal state denotes the observations.

Efficient approaches to abduction are limited. A tractable solution to the generation problem has been proposed by Eiter & Makino (1992) that is limited to Horn theories and positive observation literals.

Knowledge Acquisition for Security Attack Reconstruction

Single-step actions that belong to one and the same chained attack act as IDS-generated events and alarms with varying confidence. On the other hand, actions may be missing in these observations either because IDS sensors didn't recognize them as important or because the attacker took measures to hide his actions. If these observations can be correlated to possible adversary actions from a knowledge base, it will be possible to reconstruct a chained attack and to re-assess the confidence and importance of the alarms. Furthermore, it would allow to generate explanations for the administrator that assist him in estimating the importance.

In order to reconstruct past attacks with AI methods like classical planning, there is a need for procedural knowledge representation. Boddy *et al.* (2005) have applied classical planning to the problem of vulnerability analysis with a system that uses domain knowledge and descriptions of the environment and situation in a planning domain language (specifically, PDDL). A planning action is defined as a tuple of preconditions and effects that allows an automated planner to find a consistent sequence of actions if there is one. A related approach has been proposed by Bhattacharya & Ghosh (2008).

A well-known problem with knowledge-based reasoning in general and planning in particular is that knowledge acquisition is a tedious task: the quality of a knowledge base is hard to evaluate and acquisition tools are scarce and complicated to use. Furthermore, knowledge about attacks has to be extracted from several different sources: A large set of standard approaches to compromise network infrastructure may be available, but a security expert may also want to model a specific attack based on his experience. Atomic attack components will depend on certain vulnerabilities of software of which there is a very large number documented on sites such as the National Vulnerability Database [9].

For a convenient knowledge acquisition and reusability of the attack knowledge base, we propose an approach based on the decomposition of planning actions. We have started to develop a top level ontology of attack methods that provides the basis for a guided interactive integration of novel methods in the context in which they will later be used. The

[9] http://nvd.nist.gov/cvss.cfm

leaves of the specialization and partonomic trees correspond to single-step actions from the plan knowledge base.

Predicting Intrusion Trials with Relational Hidden Markov Models

Next we focus on the prediction of the next attack steps which provides useful information about the system frame the user should investigate and helps him to decide for the appropriate countermeasures. This is a challenging task because every attacker could use a different course of action.

To represent courses of action, a reasoning over time approach must be used. Considering that each attacker could use a different course of action to injure systems, multiple forecasts exist. A set of multiple forecasts might not be sufficient for the user because it will be still difficult to set a well-founded focus inside the set of forecasts. Allowing an assessment of the forecasts and to model the inherent uncertainty in the prediction (each attacker may act differently also by having the same goal) a probabilistic appoach is reasonable in contrast to rule-based methods that are currently widely used to detect (not to predict) attack steps. Furthermore the uncertainty in recognizing attack attempts is a problem that should be regarded when working out a reliable model for attack prediction (some attack-steps may probably not be detected by the underlying system). A model which addresses these problems is the hidden Markov model (HMM) (Rabiner 1989). The model consists of hidden states which represent the attack steps and observations which represent the events produced by the SIEM. In the context of HMMs observations (SIEM-events) are used to infere a probability distribution over the upcoming states (the possible attacks). The probability distribution can be interpreted as an assessment of the forecasts.

In the case of attack prediction the user may need information on different levels of granularity to be able to investigate the attack entirely (to identify the intention of the attack) and in detail. Therefore a taxonomy in the domain representation can be used. This knowledge also provides an improvement of the inference mechanism by the use of statistical smoothing techniques which addresses a second fundamental problem: Sparse training data for some course of action. One promising recent method is the relational hidden Markov model (Elfers *et al.* 2008) which combines the HMM features with the ability to use domain knowledge in the form of a taxonomy like in relational Markov models (Anderson, Domingos, & Weld 2002) which addresses these problems. This approach has already been successfully applied in the multi-agent system domain to predict actions of autonomous agents and is a promising approach in the security domain for predicting attack steps.

Conclusion

The results of this research will be used to combine conventional event correlation with plan/intention recognition techniques in order to assess the confidence of alarms issued by an IDS and generate explanations for a security expert. The vision is an integrated set of AI algorithms for collecting, analyzing, and managing enterprise event information.

A central problem is the modeling of the domain so that the complexity is manageable and the representation granularity is sufficient to generate a benefit for the user. Under these considerations a set of different models for each host or for a previously detected attack strategy could be taken into account. The training of each model could be done by learning from succeeded attacks specifically by learning from SIEM-event attack pairs. Our method offers the ability to train a prediction for generalized attack patterns.

References

Anderson, C.; Domingos, P.; and Weld, D. 2002. Relational markov models and their application to adaptive web navigation. In *8th ACM SIGKDD*, 143–152.

Bhattacharya, S., and Ghosh, S. K. 2008. Attack trees: Modeling security threats. *Journal of Information Assurance and Security* 3(2):119–127.

Boddy, M. S.; Gohde, J.; Haigh, T.; and Harp, S. A. 2005. Course of action generation for cyber security using classical planning. In *International Conference on Automated Planning and Scheduling (ICAPS)*, 12–21.

Eiter, T., and Makino, K. 1992. On computing all abductive explanations. In *AAAI*, 62–67.

Elfers, C.; Herzog, O.; Miene, A.; and Wagner, T. 2008. Qualitative abstraction and inherent uncertainty in scene recognition. In *Dagstuhl Seminar Proceedings of Logic and Probability for Scene Interpretation*.

Fikes, R. E., and Nilsson, N. J. 1971. Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* 2:189–208.

Forbus, K. D., and de Kleer, J. 1993. *Building Problem Solvers*. MIT Press.

Garcia-Teodoro, P.; Diaz-Verdejo, J.; Macia-Fernandez, G.; and Vazquez, E. 2009. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers and Security* 28(1-2):18–28.

Helmert, M. 2004. A planning heuristic based on causal graph analysis. In *ICAPS*, 161–170.

J. M. Gonzalez, V. Paxson, N. W. S. 2007. A hardware/software architecture for flexible, high-performance network intrusion prevention. In *CCS '07 - 14th ACM conference on Computer and communications security*, 139–149.

O'Hara, J. 2007. Toward a commodity enterprise middleware. *Queue* 5(4):48–55.

Rabiner, L. R. 1989. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, 257–286.

Rieck, K., and Laskov, P. 2007. Language models for detection of unknown attacks in network traffic. *Journal in Computer Virology* 2(4):243–256.

Schneier, B. 1999. Attack trees: Modeling security threats. *Dr. Dobb's Journal*.

Templeton, S. J., and Levitt, K. 2001. A requires/provides model for computer attacks. In *Proceedings of the 2000 Workshop on New Security Paradigms*, 31–38.