Threat-Model-Driven Runtime Adaptation and Evaluation of Intrusion Detection System

Martin Rehak†, Eugen Staab£, Volker Fusenig£, Jan Stiborek†,
Martin Grill†‡, Karel Bartos†‡, Michal Pechoucek† and Thomas Engel£
† Department of Cybernetics, Czech Technical University in Prague, Czech Republic
£ Faculty of Science, Technology and Communication, University of Luxembourg, Luxembourg
‡ CESNET, z.s.p.o., Czech Republic
martin.rehak@agents.felk.cvut.cz, eugen.staab@uni.lu

ABSTRACT

We present a mechanism for autonomous self-adaptation of a network-based intrusion detection system (IDS). The system is composed of a set of cooperating agents, each of which is based on an existing network behavior analysis method. The self adaptation mechanism is based on the insertion of a small number of challenges, i.e. known instances of past legitimate or malicious behavior. The response of individual system components to these challenges is used to measure and eventually optimize the system performance in terms of accuracy. In this work we show how to choose the challenges in a way such that the IDS attaches more importance to the detection of attacks that cause much damage.

Categories and Subject Descriptors: C.2.0 [Computer-Communication Networks]: Security and Protection.

General Terms: Security, Management, Measurement.

Keywords: intrusion detection, attack trees, self-adaptation, network behavior analysis.

1. INTRODUCTION

To overcome limitations of current intrusion detection techniques the CAMNEP IDS [4] was proposed. The system is based on a multi-stage cooperation process of agents that use different network behavior analysis methods to classify network flows into *malicious* and *legitimate* flows. The results of the agents are aggregated by different aggregation functions (see Fig. 1). We employ a self-adaptation procedure that selects the aggregation function which optimally integrates the results of the individual agents. To this end, we insert *challenges* [6], i.e. known malicious and legitimate flows, into the real traffic. This allows us to evaluate the accuracy of the different aggregation functions and eventually select the output of the function that performed best.

In an earlier work [5] we showed how to determine an optimal number of challenges. In this paper we show which challenges to choose such that the IDS attaches more importance to the detection of very harmful attacks. This is possible because the accuracy of the IDS is optimized with respect to the used challenges. More precisely, if the system is evaluated on challenges that are representative for a certain class of attacks, the detection of real attacks in that class becomes more probable. Thus, the composition of challenges should reflect the expected damage of known attack classes. In the following we show how the expected damage of realized *threats*, each of which is represented by an attack tree, can be used to find a suitable challenge composition.

ICAC'09, June 15–19, 2009, Barcelona, Spain.
ACM 978-1-60558-564-2/09/06.

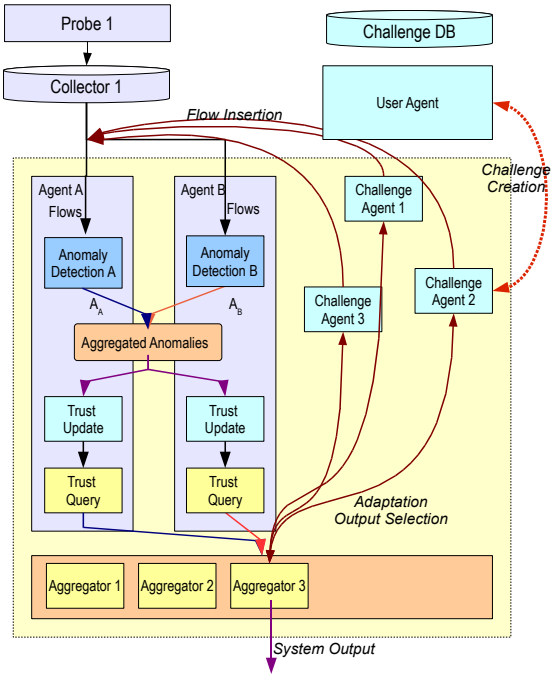

Figure 1: CAMNEP's Adaptation Process.

2. ATTACK TREE VALUATION

First, we define a set of *attack classes* $\{AC_1, \ldots, AC_k\}$. Each attack class AC_i contains several *atomic attacks* A_j (e.g., "horizontal scan"), which are known to the IDS in a sufficient number, and are used later to form the challenge flows. To structure these atomic attacks, we use the attack tree formalism (see also [1]). Figure 2 exemplifies the structure of such an attack tree. The root constitutes a *threat* that an attacker can try to implement, for instance a "server takeover". To reach the root, an attacker needs to carry out a number of atomic attacks represented by the leafs of the tree. To accomplish a (sub-)goal, the attacker needs to accomplish *at least one* of the children of the (sub-)goal, or,

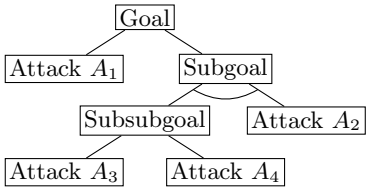

Figure 2: Example Attack Tree.

if the branch has the AND arc, *all* of the children. To each threat represented by an attack tree T we attach damage $D(T)$ that is expected if the threat was realized.

It is easy to see that any attack tree T can be transformed into a propositional logic formula, which we will denote by $F(T)$. We bring each $F(T)$ into *disjunctive normal form*, which is always possible (see [2]), and minimize it using the Quine-McCluskey algorithm [3]. A resulting formula consists of a disjunction of *clauses* C_i, which are conjunctions of *literals* l_{ij} (variables that may be negated), i.e. it is of the form: $\bigvee_i (\bigwedge_j l_{ij})$. For convenience we will write each formula as a set of clauses, and each clause as a set of literals.

For a given set of attack trees $\mathcal{T} = \{T_1, \ldots, T_n\}$ and expected damages $D(T_1), \ldots, D(T_n)$, an adequate composition of challenges should meet the following criteria:

1. The higher $D(T_i)$, the higher the attack class(es) should be prioritized to which the attacks in T_i belong.
2. An attacker realizes a threat corresponding to T_i if he satisfies at least one clause in formula $F(T_i)$. So, any satisfied clause in $F(T_i)$ causes damage $D(T_i)$, and so clauses within a formula should be equally prioritized.
3. For making a chosen clause true, an attacker needs to make true *all* literals in this clause. Thus, all literals belonging to the same clause should be equally prioritized.

To fulfill the last two criteria, we compute the priority of an attack A_i within a tree T_j as follows:

$$P(A_i, T_j) := \frac{1}{|F(T_j)|} \sum_{\substack{C_k \in F(T_j), \\ \text{with } A_i \in C_k}} \frac{1}{|C_k|} , \qquad (1)$$

where $|C_k|$ is the number of literals in clause C_k, and $|F(T_j)|$ is the number of clauses in the formula T_j. The reader can easily verify that if attack A_i is not in T_j, then its priority within the tree is zero. Also, the sum of the priorities of all attacks in the tree is 1. To fulfill the first criterion, we additionally weight each tree T_j according to the damage $D(T_j)$ and get the final priority for an attack A_i by summing over all attack trees:

$$P(A_i) := \frac{1}{\sum_{T_j \in \mathcal{T}} D(T_j)} \cdot \sum_{T_k \in \mathcal{T}} D(T_k) \cdot P(A_i, T_k) . \qquad (2)$$

Because of the normalization, again the priorities of all attacks sum up to 1. Finally, we compute the ratio of challenges $P(AC)$ taken from attack class AC as follows:

$$P(AC) = \sum_{A_i \in AC} P(A_i) . \qquad (3)$$

Given an optimal number of challenges n (see [5]), we choose $\lceil 0.5 \cdot n \cdot P(AC) \rceil$ many challenges from attack class AC.

Table 1: Difference between the anomaly of the given attack and the average anomaly of the traffic in the given set (expressed as a multiple of the standard deviation of the anomaly distribution of the flows from the current data set). Higher values are better, negative values refer to the traffic which is less anomalous than the average. First value refers to request traffic, the second to the response traffic.

Attack	All challenges	Selected chall.
Horizontal scan	1.1/-0.2	1.4/0.0
Vertical scan	1.2/-0.2	1.4/0.3
Fingerprinting	1.5/1.2	1.9/1.6
SSH passw. brute force	-0.2/0.6	0.2/1.2
Buffer overflow	-0.2/0.1	0.2/0.0

3. EXPERIMENTAL EVALUATION

Results of the experiments that evaluate the effects of threat modeling in the adaptation process are presented in Table 1. We can see that the use of threat modeling achieves better separation between the normal traffic and the exploit traffic, improving the performance against all components of the test scenario. This result was obtained in a series of experiments when we have trained the system to respond to horizontal scanning, fingerprinting and brute-force attacks/buffer overflow attacks.

4. CONCLUSION

In this work we extended the CAMNEP IDS by a mechanism for finding the right composition of challenges. For a set of threats and associated damages, the system is now able to choose challenges such that the detection of the most relevant threats is more probable. Also, the likelihood of attack detection can be reported to the network administrators.

Acknowledgment

This material is based upon work supported by the ITC-A of the US Army under Contract No. W911NF-08-1-0250. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the ITC-A of the US Army. Also supported by Czech Ministry of Education grants 6840770038 (CTU) and 6383917201 (CESNET).

5. REFERENCES

[1] A. P. Moore, R. J. Ellison, and R. C. Linger. Attack modeling for information security and survivability. Technical Report CMU/SEI-2001-TN-001, CMU Software Engineering Institute, March 2001.

[2] C. H. Papadimitriou. *Computational Complexity*. Addison Wesley, November 1993.

[3] W. Quine. A way to simplify truth functions. *American Mathematical Monthly*, 62(9):627–631, 1955.

[4] M. Rehak, M. Pechoucek, M. Grill, and K. Bartos. Trust-based classifier combination for network anomaly detection. In *Cooperative Information Agents XII*, volume 5180 of *LNAI/LNCS*, pages 41–54. Springer Verlag, September 2008.

[5] M. Rehak, E. Staab, M. Pechoucek, J. Stiborek, M. Grill, and K. Bartos. Dynamic information source selection for intrusion detection systems. In *Proc. of the 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS'09)*. IFAAMAS, 2009.

[6] E. Staab, V. Fusenig, and T. Engel. Towards trust-based acquisition of unverifiable information. In *Cooperative Information Agents XII*, volume 5180 of *LNAI/LNCS*, pages 41–54. Springer Verlag, September 2008.