M. Meier, D. Reinhardt, S. Wendzel (Hrsg.): Sicherheit 2016,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2016 197

Towards Adaptive Event Prioritization for Network Security - Ideas and Challenges

Leonard Renners[1]

Abstract: In the network security domain Intrusion detection systems (IDS) are known for their problems in creating huge amounts of data and especially false positives. Several approaches, originating in the machine learning domain, have been proposed for a better classification. However, threat prioritization has also shown, that a distinction in true and false positives is not always sufficient for a profound security analysis. We therefore propose an approach to combine several aspects from those two areas. On the one hand, threat and event prioritization approaches are rather static with fixed calculation rules, whereas rule learning in alert verification focuses mostly on a binary classification and does not target rule parameters. In this paper we highlight specifics and challenges in event prioritization rules and describe first ideas and challenges towards solutions by the means of automatically learning these aspects.

Keywords: Network Security, Intrusion Detection, Machine Learning, Event Priority

1 Introduction

A lot of effort has been spent in reducing the number of false positives in network security alerts. Especially IDS are known for their problems in creating huge amounts of data, and studies have shown that up to 99% of them belong to the group of false positives, meaning they either were not successful or generally not relevant[JD02]. Multiple approaches have been conducted to tackle these challenges. Mostly, additional information is used to confirm or contradict an alert. Some approaches suggest to utilize techniques from the machine learning domain and learn from labeled attacks and the analysts behavior.

Nevertheless, the amount of reports can still be huge, due to the fact that many alerts can be generated for a single attack and also non-important, but successful, incidents are also classified as true positives. On the other hand, not all true negatives should be discarded. The information gained, even from unsuccessful attacks, can be substantial in protecting the network and especially preventing future incidents. For example an attack, which has not been successful, but was conducted from an IP-address in the corporate network may very well be a security incident, although the alert itself was a false positive.
Promising results towards this problem have been shown by trying to evaluate more information and calculate event priorities, instead of solely classifying the alerts in true/false positives. One major problem still is the fact, that these rather complex rules have to be written by a domain expert and mostly consist of static calculation formulas or cumbersome property files. These approaches therefore require high configuration efforts and are limited in their applicability in dynamic environments, which networks typically are.

[1] Hochschule Hannover, Fakultät IV, Abteilung Informatik, Ricklinger Stadtweg 120, 30459 Hannover, leonard.renners@hs-hannover.de

We are arguing, that one possible solution is to specifically design rules to target prioritization parameters in order to enable an easier understanding and maintenance of such rules. Furthermore we want to try to adapt some of the learning processes from advanced IDS to learn these rules from rated events and related background knowledge. In this paper we identify specifics of prioritization focused rule based systems and discuss resulting challenges for an automated learning. Note that event prioritization and thus the proposed approach is well suited, but not limited, to the domain of IDS alerts.
The remainder of this paper is structured as follows. Section 2 covers important work in related research areas, including automatic IDS alert classification and threat prioritization. The subsequent section 3 discusses our ideas towards the specifics of rules for event prioritization and gives first thoughts on the learning of each aspect. The final section 4 gives concluding remarks and an outlook on the next steps and future lines of research.

2 Related Work

Alert correlation and alert verification are techniques to improve the precision in classifying IDS alerts in order to identify true positives and thereby reduce the amount of alerts generated. While alert correlation uses the information from different IDS sensors to rate alerts, alert verification is the process to include additional information, mostly about the underlying topology and especially the target, to determine the probability of a successful alert [Yu04][MHT05]. Further research went towards the automated learning of IDS rules with background information in order to generate automatic classifiers. Pietraszek et al. [Pi04] built an adaptable learner for alert classification - especially suitable for alert verification - using machine learning techniques.

More recently, some research has been conducted towards a more general rating of events instead of classifying true and false positives. An implementation framework approach on cyber threat prioritization has been presented by Townsend and McAllister [TM13] where they analyze the aspects that influence the priority of an event. They conclude on three major aspects - impact, likelihood and risk - and further describe each one of them. Kim et al. [Ki14] propose a similar approach in their ACCEPT framework - a threat prioritization framework for cyber network defense. They develop why and how incoming cyber events can be prioritized in order to make quick and effective decisions, but also give several concrete metrics and formulas for different aspects of a cyber event, e.g. the importance of the target host or connected entities, beside the severity of the attack itself.

The approach presented in this paper utilizes some of the related concepts, but extends the functionality, especially by learning the concepts in the background knowledge in order to identify important rule elements, instead of solely focusing on the outcome. Furthermore the nature of rules for IDS alert verification and event prioritization differ, as the more recent approaches have shown, and concepts for learning can not be adopted directly. Event and threat prioritization thus far relies on pre- and user-defined rules without the possibility to automatically learn or adapt these aspects and is also lacking an appropriate syntax dedicated to this problem area.

3 Adaptive Prioritization Rules: Specifics and Challenges

State-of-the-art rule languages and rule based systems are generally capable of event pattern evaluation and can also be used to prioritize the events. Meanwhile, it is important to keep in mind the need for understandable and modifiable security policies in order to not only identify and deal with the events, but to also make sure to include the security administrator and follow the company compliance guidelines.
In this section we identify some special properties of prioritization focused rules. These can be used to construct a concept to explicitly target these challenges and the resulting rules become much more intuitive to understand and write.
Figure 1 gives a definition of the structural elements of a rule and an example (in a pseudo-code event-condition-action syntax) to further explain the different aspects of a rule and their specifics for prioritization.

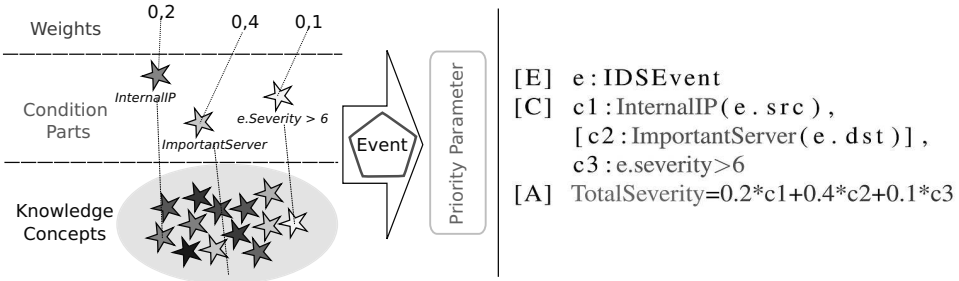

Figure 1: Structural Elements of a Rule

A rule for prioritization, in our understanding, can be divided in four constituents: *Knowledge concepts*, *condition parts*, *weights* and *priority parameters*.
Knowledge concepts comprise all available building blocks which can (potentially) be used as part of a rule condition. Condition parts then define the set of concepts which are used to describe one particular rule and form the influential factors on the focused priority parameter - the calculation result - of the event. They can be optional (square brackets) or necessary for a rule to fire.
Finally, weights are those factors which define the effect of each condition part on the priority parameter.

3.1 Knowledge Concepts

So most rule based system do not really target and highlight a defined set of possible concepts to use. Typically only attributes of the events are used in Boolean expressions, as in the example regarding the severity. For prioritization it is advantageous to define more abstract concepts to generalize from single attribute values. This makes the conditions reusable and thereby allows for an easier maintenance, since only the concept and not every single rule has to be adapted. Another aspect of the generalization is the encapsulation of more complex expressions. While a pattern concerning an integer is expressed quite easily inline, other conditions can be more complex. And often times not each individual attribute

value is important or crucial for decision making but its affiliation to a certain group of values. In the example the concrete ip-address is not really relevant, but its affiliation to the group *InternalIP* (the ip-addresses belonging to the internal corporate network). An inline expression would make the rule increasingly complex and harder to understand, especially for non-domain experts and analysts confronted with the rule.

Meanwhile, one can argue, that the introduction of knowledge concepts only shifts the complexity from the rules towards the background knowledge. While this is true, we believe that this shift also helps to cope with the complexity of the overall system since each part can now be tackled separately. Nevertheless, the definition of concepts and rules still is a complex problem and we believe that the user needs assistance in this area. So the task is to identify potentially relevant attributes and more importantly group and abstract from single values. Such learning can mostly be achieved unsupervised, e.g. by clustering techniques, but the following challenges have been identified and have to be overcome in future research:

Attribute Types: The value space of attributes can vary and need to be treated differently. E.g. numerical attributes could be separated into ranges whereas strings might be grouped with respect to their Levenshtein distances.

Indirect grouping: Some information might also not be relevant directly, but specific constellations of attribute manifestations. An example for a relevant relation is if the exploited vulnerability of a specific attack is actually present in a software running on the target system.

Evaluation: The quality assessment of different abstractions and grouping is difficult, but necessary to choose suitable concepts. The definition of a group or cluster does not always have a direct result, but can only be implicitly observed by its impact on the other aspects, especially the condition parts and finally the classification results.

3.2 Condition Parts

The concrete choice of elements out of the subset of knowledge concepts used to describe one specific rule are called condition parts. *Required* arguments specify the necessary parameter for the rating process and are the typical elements of rule based system. They can also describe a superimposed detection of a relevant situation (incident detection or alert correlation), and naturally influence the priority parameter.
Optional condition parts on the other hand are evaluated if all necessary requirements are fulfilled and used for a more accurate description and calculation of the resulting priority parameter. The example describes a rule to consider all rules that originate from the internal network and have a high base severity. But the fact, that the target is also important is an optional factor, that makes the alert even more relevant. To put it the other way around, we want to express patterns comparable to the following: an attack with a low severity is not relevant, even if it targets an important server, but the importance of the server is an influential factor for generally relevant attacks.

In order to automatically infer these parts typical rule induction techniques can serve as a basis for supervised learning. Although extensions are necessary in order to use the numerical priority parameters and not a static true/false classification. Furthermore we have to keep in mind the dependencies to the defined knowledge concepts. If the concepts are too specific, i.e. too many attribute characteristics exist, the condition parts also become increasingly specific and target only special cases, whereas the goal is to generalize rules and provide a generic rating, applicable for upcoming alerts.

3.3 Weights

Most use-cases for rule based systems define rules in order to detect relevant situations or abstract from a huge amount of low-level data. In these environments the rule parameters typically do not really influence the rule result, beside the fact they lead to a new (complex) event and some attributes may be adopted. In prioritization rules however, specifically with optional parameter, the condition parts need to explicitly influence the calculation as the rule result. In addition, rules and their results have to be comparable in order use the calculated rating to sort the events. The considered parameters need to have specific ranges in which all rules map their individual calculations, although the weights are specific for each rule. The same holds true for the relations between condition parts themselves. While most parts influence the value of a prioritization aspect directly, there might be some situations where all other factors become completely irrelevant. One example is the knowledge, that a target is not vulnerable against a certain attack (e.g. the software is known to be patched against this certain attack) and therefore the alert probability should always be zero.

The calculation of weights might be included in the rule induction process, e.g. by applying a linear regression for rule induction. Otherwise this challenge has to be tackled separately. The weights could be inferred from the already defined condition parts, when examining different calculation results in similar rules. Either way, it is an interesting question how the identified peculiarities regarding relativity and absolute values can be targeted.

4 Outlook and Future Work

In this paper, we presented our ideas towards adaptive event prioritization to tackle the challenge of the manual and complex rule design of threat prioritization systems. Therefore, we identified special requirements and challenges for a better rule design in order to provide a distribution of complexity and thereby more understandable rules. Further, we gave some remarks towards challenges in automatically learning these aspects in order to reduce the effort and assist the user. However, the proposed ideas are only first thoughts and further research is required in every single direction and especially in the combination of the subproblems.

For knowledge concepts, most importantly a way is needed to validate learned concepts, probably using the whole framework, but this involves the other learning techniques.
Condition parts should be analyzed with regard to the extracted patterns. For example

whether only complete patterns are valid rules or if sub-expressions can be considered. And further how they can be used to refine the concepts, e.g. if too complex rules emerge from the rule induction.
Weights can be learned from similar rules, but a concept has to be developed to include dependencies between parameters and rules.

Lastly, and parallel to the conceptual parts of the learning process, we also want to take a closer look at the syntactical representation. As highlighted in the paper, prioritization rules have some specifics which could best be represented by a syntax that explicitly tackles these requirements. The learning process could preferably generate such *complete* rules in a resulting framework.

One side issue occurring in the analysis of the proposed techniques is to acquire suitable data for testing purposes. This also concerns the task of an evaluation of the system. Labeled test data could be used to compare the overall result, e.g. the prediction accuracy. But further tests regarding the induction of the single rule aspects and especially towards the usability are desirable and need to be designed and conducted. Anyhow, not much data is publicly available in this area, especially rated alerts with background knowledge. So one auxiliary task will be to find or generate and prepare training data, i.e. complement existing data sets by suitable priority parameters.

References

[JD02] Julisch, Klaus; Dacier, Marc: Mining intrusion detection alarms for actionable knowledge. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 366–375, 2002.

[Ki14] Kim, Anya; Kang, Myong H.; Luo, Jim Z.; Velasquez, Alex: A Framework for Event Prioritization in Cyber Network Defense. Technical report, July 2014.

[MHT05] Mu, Chengpo; Huang, Houkuan; Tian, Shengfeng: Intrusion Detection Alert Verification Based on Multi-level Fuzzy Comprehensive Evaluation. In (Hao, Yue; Liu, Jiming; Wang, Yuping; Cheung, Yiu-ming; Yin, Hujun; Jiao, Licheng; Ma, Jianfeng; Jiao, Yong-Chang, eds): Computational Intelligence and Security, Lecture Notes in Computer Science 3801, pp. 9–16. Springer Berlin Heidelberg, December 2005.

[Pi04] Pietraszek, Tadeusz: Using Adaptive Alert Classification to Reduce False Positives in Intrusion Detection. In (Jonsson, Erland; Valdes, Alfonso; Almgren, Magnus, eds): Recent Advances in Intrusion Detection, Lecture Notes in Computer Science 3224, pp. 102–124. Springer Berlin Heidelberg, September 2004.

[TM13] Townsend, Troy; McAllister, Jay: , Implementation Framework â Cyber Threat Prioritization, September 2013. ref 28 from kim et al.

[Yu04] Yu, J.; Ramana Reddy, Y.V.; Selliah, S.; Kankanahalli, S.; Reddy, Sumitra; Bharadwaj, Vijayanand: TRINETR: an intrusion detection alert management systems. In: 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2004. WET ICE 2004. pp. 235–240, June 2004.