Typed Linear Chain Conditional Random Fields And Their Application To Intrusion Detection

Carsten Elfers, Mirko Horstmann, Karsten Sohr, and Otthein Herzog

Center for Computing and Communication Technologies
Am Fallturm 1, 28359 Bremen, Germany
{celfers,mir,sohr,herzog}@tzi.de

http://www.tzi.de

Abstract. Intrusion detection in computer networks faces the problem of a large number of both false alarms and unrecognized attacks. To improve the precision of detection, various machine learning techniques have been proposed. However, one critical issue is that the amount of reference data that contains serious intrusions is very sparse. In this paper we present an inference process with linear chain conditional random fields that aims to solve this problem by using domain knowledge about the alerts of different intrusion sensors represented in an ontology.[1]

1 INTRODUCTION

Computer networks are subject to constant attacks, both targeted and unsighted, that exploit the vast amount of existing vulnerabilities in computer systems. Among the measures a network administrator can take against this growing problem are intrusion detection systems (IDS). These systems recognize adversary actions in a network through either a set of rules with signatures that match against the malicious data stream or detection of anomalous behavior in the network traffic. Whereas the former will not recognize yet unknown vulnerability exploits (*zero-day*) due to the lack of respective signatures, the latter has an inherent problem with false positives. Anomalies may also be caused by a shift in the network users' behavior even when their actions are entirely legitimate (see [12]). One strategy is to combine the signature and the anomaly detectors to a hybrid IDS by learning which detection method is reliable for a given situation (e.g. [4]). In this setup detecting false positives is the challenging task to avoid overwhelming the users of an IDS with irrelevant alerts but without missing any relevant ones.

Several well-known machine learning methods have already been applied therefore to the domain of intrusion detection, e.g., Bayesian networks for recognizing attacks based on attack-trees [11] and (hidden colored) Petri nets to infer

[1] This work was supported by the German Federal Ministry of Education and Research (BMBF) under the grant 01IS08022A.

the actions of the attacker by alerts [14]. For the detection of multi-stage intrusions in alert sequences especially hidden Markov models have been successfully investigated (e.g., [8, 10]). However, these models suffer from an implicit modeling of past alerts with the Markov property because in this domain the threat of an alert may highly depend on the context, e.g., the previously recognized alerts. This problem can be addressed by using Conditional Random Fields (CRF) [7] that can consider several (past) alerts to reason about the current state. It has been shown that CRFs are very promising for detecting intrusions from simulated connection information in the KDD cup '99 intrusion domain[2] compared to decision trees and naive Bayes [5, 6].

However, the high amount of reference data as in the KDD data set is only available in simulated environments and is not available in real network domains. The sparse reference data problem is due to the infrequent occurrence of successfully accomplished critical intrusions (cf. [2]) and the lack of annotation. This leads to the problem that most of the possible alerts are even unknown at the training phase of the alert correlator. One possibility to overcome this problem is described in this paper: Typed Linear Chain Conditional Random Fields.

This method uses type information of feature functions for the inference in linear chain conditional random fields and is motivated by filling the gap of missing reference data by considering semantic similarities. Earlier work has already considered the semantic similarity between states for the inference, e.g., in Markov models [1], in hidden Markov models [3], and in input-output hidden Markov models [9]. The latter is similar to linear chain conditional random fields. The inference can also be regarded as mapping a sequence of input values to a sequence of labels.

This paper is organized as follows: In the next section the intrusion detection domain representation in an ontology and its use for preprocessing the alerts from the different IDSs are described. In Section 3 we overcome the problem of sparse reference data by using the domain knowledge described in Section 2. In Section 4 the type extension to linear conditional random fields is evaluated by some real examples in the intrusion detection domain. At last we come to a conclusion and give an outlook of future research.

2 Preprocessing and Domain Knowledge

Hybrid IDSs that use both signature-based and anomaly-based detectors are a promising strategy to improve the precision of intrusion detection. Our approach therefore involves the correlation of alarms from several detectors that can be added if they are present in a particular network. As a first step, we use a syntactic normalization in the IDMEF[3] format, which is done by Prelude Manager[4], a well-known open source interface. This is followed by a semantic normalization that enables the system to handle each sensor's alarms according to their

[2] KDD '99 data set: http://kdd.ics.uci.edu/databases/kddcup99
[3] s. RFC 4765
[4] http://www.prelude-technologies.com/

meaning and a burst filtering that eliminates duplicates of alarms produced by several sensors or as a result of similar observations.

The semantic normalization is based on an ontology in OWL-DL[5] representation. This ontology contains several facets of the security domain, including e.g., the topology of the network in question, its computers (assets) and general configuration knowledge. Of particular interest for the recognition of multi-step attacks are definitions of possible observations that can be made by the sensors that are organized in a hierarchy of concepts (see Fig. 1). Among the concepts are some that have been derived from classes introduced by Snort[6]. Individuals that belong to these concepts are possible observations and can be imported from Snort's rules set by an automatic parser. When analysing multi-step attacks, these observations can be considered as describing adversary actions of an attacker, but from a security expert's perspective. Furthermore, the hierarchy denotes semantic similarity between nearby concepts and thereby supports the further correlation process.

If knowledge about further sensors is added to the ontology, several observations from one or more sensors can be unified when they are instances of the same concept from the observation ontology. E.g., if an observation according to the `ET EXPLOIT MS04-007 Kill-Bill ASN1 exploit attempt` rule has been made by the Snort IDS and the Prelude logfile parser LML recognizes a match of the `Admin login` rule in a log file it observes, they may be normalized to one concept `AttemptedAdminObservation` to which they both belong.

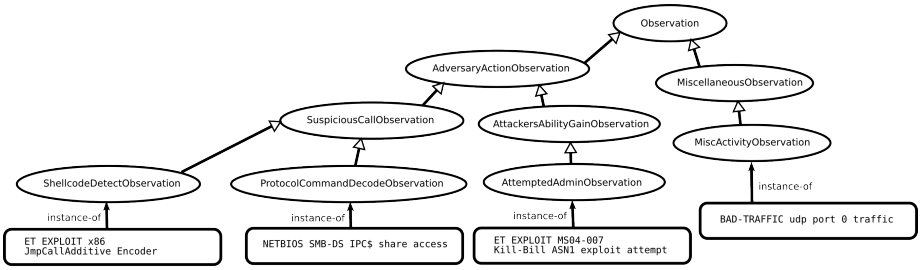

Fig. 1. Excerpt from the observation ontology. Specific observations (as defined by the sensors' rules) are instances of concepts in a hierarchy.

3 Typed Linear Chain Conditional Random Fields

In this section we briefly introduce conditional random fields and extend them by using a type hierarchy to fill the gap of missing feature functions due to

[5] http://www.w3.org/TR/owl-features/
[6] http://www.snort.org/

insufficient reference data. For ease of demonstration, this paper assumes that each observation corresponds to one feature function.

3.1 Prerequisites: Linear Chain Conditional Random Fields

The purpose of linear chain conditional random fields compared to hidden Markov models is to take multiple features (respectively observations) for computing the probability of the labels into account. Thereby they also address the label bias problem from maximum entropy Markov models (cf. [7]). In the following the simplified notation from [13] for linear chain conditional random fields is used with a sequence of labels X and a sequence of observations to be labeled Y with a normalization function Z:

$$p(\mathbf{y}|\mathbf{x}, \lambda) = \frac{1}{Z(\mathbf{x})} exp(\sum_j \lambda_j F_j(\mathbf{y}, \mathbf{x})) \tag{1}$$

The inference problem is to determine the probability distribution over a vector of labels $\mathbf{y}$ from a vector of observations $\mathbf{x}$. Conditional random fields are generally not restricted in the dependencies among the nodes, however in linear chain conditional random fields the nodes are only dependent on their predecessor and on the vector of observations. Each feature function F_j has a corresponding weight λ_j that is computed during training.

3.2 Typed Linear Chain Conditional Random Fields

One issue with linear chain conditional random fields is that there is a lack of information for computing the probability of the labels if the features are not known at training time. Our suggestion is to use a type hierarchy of feature functions to find the most similar feature functions that handle the observation. E.g., if no feature function matches a tcp port scan observation, it is dangerous to assume that the tcp port scan observation belongs to a normal system behavior. If there is a feature function matching a udp port scan observation and the type hierarchy expresses a high similarity between udp and tcp port scans, the feature function for udp port scan observation could be assumed to match instead. In our case we can derive the type hierarchy from the semantic normalization (cf. Section 2). The computation of the conditional probability is therefore extended by a parameter for the type hierarchy over feature functions T:

$$p(\mathbf{y}|\mathbf{x}, \lambda, T) = \frac{1}{Z(\mathbf{x})} exp(\sum_j \lambda_j F'_j(\mathbf{y}, \mathbf{x})) \tag{2}$$

In the case of not having a matching feature function for a given x we propose to instantiate a new feature function F' to match the currently unknown observation, i.e., the feature function is fulfilled (returns 1) iff the given observation arrives.
However, there is the need to determine the corresponding weights for the new

feature function. The original weights of the most similar feature functions should be regarded but with a loss to reduce the likelihood that the sequence of observations really belongs to that label. The weights $\lambda_{F'}$ of the new feature function F' are determined by the weights of the most similar feature functions. The most similar feature functions are given by a similarity measurement. The set of most similar feature functions SF is given by:

$$SF = \{F_s(x,y) | s \in \underset{k}{\operatorname{argmin}}\ sim(F_j(x,y), F_k(x,y), T).\ F_k(x,y) \in B(x,y)\} \quad (3)$$

$B(x,y) \subseteq T$ is the set of bound feature functions, i.e., the feature functions that have a value for the given parameters. The corresponding weights of the new feature function F' based on the most similar feature functions SF is given by:

$$\lambda_{F'} = \frac{1}{|SF|} \sum_s \lambda_s sim(F_j, F_s, T) \quad (4)$$

As mentioned there is the need for a similarity score between feature functions regarding the type hierarchy, denoted as $sim(a,b,T), a \in T, b \in T$. There are different possibilities to determine the similarity, e.g., the method of Zhong et al. [15]. This method uses the distance from a to the closest common parent in the type hierarchy denoted as $d(a, ccp, T)$ and the distance from b to the closest common parent $d(b, ccp, T)$ where the distance is defined as:

$$d(a,b,T) = |\frac{1}{2k^{l(a,T)}} - \frac{1}{2k^{l(b,T)}}| \quad (5)$$

$l(n,T)$ is the depth of $n \in T$ from the root node in the corresponding type hierarchy where the depth of the root node is zero ($l(root,T) = 0$). k is a design parameter to indicate how fast the distance increases depending on the depth in the hierarchy. In this paper $k = 2$ is used as proposed by Zhong. The similarity of two feature functions is given by the distances to the closest common parent by:

$$sim(a,b,T) = 1 - d(a,ccp,T) - d(b,ccp,T) \in [0;1] \quad (6)$$

4 Results

The evaluation of typed linear chain conditional random fields is done by two experiments to compare this model to traditional linear chain conditional random fields. In the experiments both models are trained with missing reference data. The first experiment shows how the type knowledge is used to overcome the lack of data. The second experiment is about the dependency of the model to the quality of the type hierarchy. The evaluation data consists of two real intrusions performed with the Metasploit Framework[7]: (1) the *Kill-Bill*[8] and (2) the *Net-Api*[9] exploit. The gathered sequences of alerts from the Snort detector and the normalized alerts by the preprocessor are presented in table 1 and 2.

[7] http://www.metasploit.com/
[8] Metasploit: windows/smb/ms04_007_killbill
[9] Metasploit: windows/smb/ms08_067_netapi

Time	Normalized alerts (after preprocessing)	Snort message
1	MiscActivityObservation	BAD-TRAFFIC udp port 0 traffic
2	MiscActivityObservation	BAD-TRAFFIC udp port 0 traffic
3	AttemptedAdminObservation	ET EXPLOIT MS04-007 Kill-Bill ASN1 exploit attempt
4	MiscActivityObservation	BAD-TRAFFIC udp port 0 traffic
5	MiscActivityObservation	BAD-TRAFFIC udp port 0 traffic

Table 1. Alert sequence of the Kill-Bill exploit

Time	Normalized alerts (after preprocessing)	Snort rule
1	MiscActivityObservation	BAD-TRAFFIC udp port 0 traffic
2	ShellcodeDetectObservation	ET EXPLOIT x86 JmpCallAdditive Encoder
3	MiscActivityObservation	BAD-TRAFFIC udp port 0 traffic
4	ProtocolCommandDecodeObservation	NETBIOS SMB-DS IPC$ share access
5	MiscActivityObservation	BAD-TRAFFIC udp port 0 traffic

Table 2. Alert sequence of the Net-Api exploit

4.1 Experiment 1

The two kinds of linear chain conditional random fields (typed and untyped) have been trained to detect the `MiscActivityObservation` as normal system behavior and the other observations from the Net-Api alert sequence as an attack. Both methods share exactly the same reference data and take the two preceding, the current and the two succeeding alerts for the labeling into account. The linear chain conditional random field has been tested against the typed linear chain conditional random field by performing the untrained Kill-Bill exploit. As expected the typed model detects the Kill-Bill exploit by using the type knowledge for this unknown observation. The typed model does not know the observation

Alert	Similarity
ProtocolCommandDecodeObservation	**0.882812**
ShellcodeDetectObservation	**0.882812**
MiscActivityObservation	0.519531

Table 3. Similarity of the different feature functions to the feature for AttemptedAdminObservation

`AttemptedAdminObservation` from training but it searches the available feature functions with the highest degree of similarity. These are the alerts `ProtocolCommandDecodeObservation` and `ShellcodeDetectObservation` (cf. Fig. 1 and 3) and it computes the corresponding weights as described in Eqn. 4. Both features refer to an attack and therefore the classification comes to the conclusion that the unknown observation `AttemptedAdminObservation` also refers to an attack. In contrast, the traditional untyped linear chain conditional random field has not detected the Kill-Bill exploit and generated a critical classification in the intrusion detection domain: A false negative. This shows how typed linear chain CRFs enrich traditional linear chain CRFs. In conclusion the typed classi-

fication outperforms the traditional classification if a type hierarchy expressing the correct semantic similarities is available.

4.2 Experiment 2

The second experiment shows how a type hierarchy expressing an ambiguous semantic similarity between contradictory feature functions influences the inference process. In this experiment, the `MiscActivityObservation` has been attached to the type hierarchy to have exactly the same similarity than the other similar observations `ProtocolCommandDecodeObservation` and `ShellcodeDetectObservation` (each one having a similarity of 0.882812). The high belief of the model that the `MiscActivityObservation` feature corresponds to normal system behavior and the circumstance that it is as similar as the contradictory feature functions led to the misclassification of the `AttemptedAdminObservation` to a normal system behavior like in linear chain conditional random fields. This behavior results by the contradictory weights associated with the similar observations leading to a nearly uniform probability distribution over the labels. In conclusion the increased inference accuracy of typed linear chain CRFs is highly dependent on a type hierarchy expressing the right semantic similarities. Ambiguous similar feature functions with contradictory semantics lean towards a uniform probability distribution pointing to the appropriate decreased certainty of the results. However, if the type hierarchy expresses the right semantic similarity, the typed model leads to an increased inference accuracy.

5 Conclusion and Future Work

Typed linear chain conditional random fields offer an improved way to handle missing feature functions. The missing feature functions' weights are approximated during runtime by searching semantically similar feature functions out of a type hierarchy. The type hierarchy is extracted out of an ontology and the semantic similarity between the concepts in the ontology (respectively the type hierarchy) are determined by a measurement from Zhong et al. [15]. Fortunately, the training process remains the same as for conditional random fields, only the inference process is adapted. Further, the computational effort of the inference process only increases if missing reference data influences the inference result, all other cases are not affected. First experiments in the domain of intrusion detection have shown that this is a useful extension to linear chain conditional random fields and that with this method variations of already known kinds of intrusions can be detected more reliably. In the future, the evaluation should be extended to a more expressive data set. Currently the benchmark sets of real intrusions are either very limited to the amount/kinds of intrusions or are only available for a low-level analysis. The search for similar features may be improved by suitable search algorithms. Also, the way of similarity measurement might be extended by not only considering a type hierarchy, but also considering different object properties / relations in the ontology, e. g. by considering IP-to-subnet relations

or host-to-asset relations. Overall, typed linear chain conditional random fields are a promising step in the direction of using complex domain knowledge to improve reasoning over time with only a few reference data.

References

1. Anderson, C., Domingos, P., Weld, D.: Relational Markov Models and their Application to Adaptive Web Navigation. Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2002)
2. Anderson, R.: Security Engineering 2nd Ed. Wiley Publishing, p. 664 (2008)
3. Wagner, T., Elfers, C.: Learning and Prediction based on a Relational Hidden Markov Model. International Conference on Agents and Artificial Intelligence (2010)
4. Gu, G., Crdenas, A. A., Lee, W.: Principled Reasoning and Practical Applications of Alert Fusion in Intrusion Detection Systems. ASIACCS '08 (2008)
5. Gupta, K. K., Nath, B., Ramamohanarao, K.: Conditional Random Fields for Intrusion Detection. 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07) (2007)
6. Gupta, K. K., Nath, B., Ramamohanarao, K.: Layered Approach Using Conditional Random Fields for Intrusion Detection. IEEE Transactions on Dependable and Secure Computing (2010)
7. Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. 18th International Conf. on Machine Learning (2001)
8. Lee, D., Kim, D., Jung, J.: Multi-Stage Intrusion Detection System Using Hidden Markov Model Algorithm. Proceedings of the 2008 International Conference on Information Science and Security (2008)
9. Oblinger, D., Castelli, V., Lau, T., Bergman, L. D.: Similarity-Based Alignment and Generalization. Machine Learning: ECML (2005)
10. Ourston, D.,Matzner, S., Stump, W., Hopkins, B.: Applications of Hidden Markov Models to Detecting Multi-stage Network Attacks. Proceedings of the 36th Hawaii International Conference on System Sciences (2003)
11. Qin, X., Lee, W.: Attack Plan Recognition and Prediction Using Causal Networks. Annual Computer Security Applications Conference (2004)
12. Garcia-Teodoro, P., Daz-Verdejo, J., Marci-Fernndez, G., Vzquez, E.: Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers and Security (2009)
13. Wallach, H.M.: Conditional random fields: An introduction. Technical Report MS-CIS-04-21, University of Pennsylvania (2004)
14. Yu, D., Frincke, D.: Improving the quality of alerts and predicting intruder's next goal with Hidden Colored Petri-Net. Computer Networks: The International Journal of Computer and Telecommunications Networking (2007)
15. Zhong, J., Zhu, H., Li, J., Yu, Y.: Conceptual Graph Matching for Semantic Search. Proceedings of the 2002 International Conference on Computational Science (2002)