Technical Report 71

# Advanced Event Correlation in Security Information and Event Management Systems


Carsten Elfers
Stefan Edelkamp
Hartmut Messerschmidt
Karsten Sohr

TZI, Universität Bremen



TZI-Bericht Nr. 71


**TZI-Berichte**

Herausgeber:
Technologie-Zentrum Informatik und Informationstechnik
Universität Bremen
Am Fallturm 1
28359 Bremen
Telefon: +49 421 218 94090
Fax: +49 421 218 94095
E-Mail: hq@tzi.de
http://www.tzi.de

ISSN 1613-3773

## Abstract

In this report, several open problems in current enterprise Security Information and Event Management systems, specifically with respect to their event correlation are discussed. An advanced event correlation using a special kind of soft pattern matching in conjunction with ontological background knowledge, Description Logic inference and a probabilistic post processing by Conditional Random Fields is proposed to address these problems. It is shown that this approach improves the detection accuracy by detecting even unknown incidents in contrast to currently applied rule-based correlations.

## 1 Introduction

The cyber criminal threat against the IT infrastructure is a well-known and steadily growing problem for many organizations. As a consequence, organizations need to protect their business-critical resources appropriately. One essential component of a comprehensive security management is to monitor the IT infrastructure at different levels, such as the operating system, network, and applications.

A single product, such as a firewall or an Intrusion Detection System (IDS) cannot be assumed to recognize all kinds of IT incidents [3, p.665]. Therefore, large organizations, e.g., car manufacturers or financial institutes, have deployed Security Information and Event Management (SIEM) systems [46]. These systems manage security-related events generated by different sources like IDSs, firewalls, system health monitors, antivirus programs, and database logs. The process of combining the events generated by these sources with background knowledge, such as IT infrastructure knowledge or known vulnerabilities for the purpose of detecting incidents, is called event correlation in the SIEM domain.

From the intrusion detection perspective, two incident detection methods are conceivable, i.e. anomaly detection [19, 35, 38, 52] where each deviating system behavior of a previously trained normal behavior is suggested to be an incident or a misuse or rule-based method [36, 30, 49, 55, 22] which detects incidents by correlating the input events with previously specified patterns.

SIEM systems like the market-leading product ArcSight [4, 46] or the Symantec Security Information Manager [56] typically use predefined rules to correlate the events. While some products like NitroSecurity SIEM [47], AlientVault Unified SIEM [2], RSA enVision [16] or the Q1 Labs correlation (used by Enterasys, Juniper and Nortel) are making use of integrated anomaly detection methods (such as detecting baseline deviations), the final decision making is almost rule based.

The application of a rule-based final decision making in these products is understandable due to the requirement of processing huge amounts of events and the problems inherent to anomaly detection such as an increased false positive rate and the absence of an interpretation of the detected incident [20]. However, the focus on rule-based methods leads to possibly unrecognized incidents due to a lack in the rule set.

In detail, we have identified the following problems in current SIEM correlation processes which we address by the approach presented in this report:

**Actuality Problem** One problem of detecting incidents by predefined patterns—as normally done in SIEM systems—is to detect incident variations. These variations cannot be discovered by convenient rule-based detection methods if the rules to detect these variations are unknown. Therefore, the need of actuality of the rule databases rises to detect actual incidents.

A similar problem arises from the actuality of background knowledge. For example, the IT assets in enterprises steadily change. Therefore, these changes must be recognized and the corresponding asset database must be kept up-to-date to guarantee correct background knowledge to enable a correct correlation.

**Balance Problem** The most important problem for the SIEM correlation and incident detection is called the balance problem. This problem describes that an incident detection approach must find an adequate balance between the false positive (also called type-I error) rate and the false negative (also called type-II error) rate. There is the difficult objective to maximize information security by analyzing all suspicious evidences on the one hand and to minimize the need of computational and human resources for this analysis on the other hand. Therefore, finding a good balance is essential for a successful application of a SIEM system. Even a small change in the false positive rate may have a drastic influence on the amount of generated incidents due to the high rate of processed events as discussed by Axelsson as the base-rate fallacy problem [5].

**Dependency Problem** Events from different sources and over different points in time may be highly dependent. For example, a failed log-in event is more suspicious if several failed log-ins have been detected previously or a preceding port scan from the same source address has been recognized. Additionally, log-in attempts may be temporally distributed, e.g., an attacker may perform only one log-in attempt each day which is hardly to be separated from regular users mistyping passwords. Further, there are dependencies to the background knowledge, such as known vulnerabilities or

servers running specific software. Several of such relations are conceivable [51]. Therefore, the identification of the relevant relations—and using them for incident detection—is very difficult due to their complexity.

**Knowledge Acquisition Bottleneck Problem** In enterprises, Information Security experts are continuously overloaded or not available at all. Specifically, the security domain requires lots of experts to gain expertise in all fields. Expert knowledge is a necessary requirement to assess incidents and their security impact [10]. Wagner [62] structured this problem into four subproblems: narrow bandwidth, acquisition latency, knowledge inaccuracy, and maintenance trap. The narrow bandwidth problem describes the problem of missing input from experts since experts are typically a sparse resource. The acquisition latency problem describes that it takes time until knowledge is formalized and shared. In the SIEM domain, this leads to missing information for the correlation. The knowledge inaccuracy problem means that even experts may produce mistakes which may lead to a misclassification of incidents. Finally, the maintenance trap is the problem of keeping knowledge up-to-date which is related to the actuality problem as stated above. These problems lead to the conclusion that background knowledge must be assumed to be incomplete or not up-to-date.

**Heterogeneity Problem** The sources of evidences that are used for detecting malicious behavior of software or users are typically heterogeneous in the SIEM domain. For example, an antivirus program generates different events than a firewall or system health monitor due to the different monitoring purposes. There exists no common accepted standard for the different types of event sources. The generated events are vendor or even product specific [43]. Considering the task of correlation, the events from all these different event sources must be understood in a common way to be correlated. Due to different detection methods, the reliability of the sensors is different and even depends on the incident to detect. These problems are summarized as heterogeneity problem in this report.

**Interpretation Problem** After an incident has been detected, an interpretation needs to be determined which describes this incident. This is a minor problem of rule-based correlation engines as used in most SIEM enterprise systems since the rules can be considered to be an adequate interpretation. However, this is a major problem of anomaly detection methods [20]. These methods can detect that there is something wrong (e.g., abnormal high network traffic), but they have difficulty in saying what the cause is [65]. This problem is stated here since it must be considered in the development of new correlation techniques. Only an adequate interpretation supports the security operator in determining appropriate countermeasures.

**Sparse Data Problem** The vast amount of attacks can be considered as a rich pool of possible reference data. However, the amount of professional successful attacks is very sparse [48] since these attacks are either prevented before they have been successful or they have not been recognized at all. This leads to the problem that there are not sufficiently annotated reference data to automatically derive dependencies (cf. the dependency problem) and to train a Machine Learning model of serious and successful intrusion attacks or incidents. Moreover, it is difficult to exchange or deploy concrete examples of incidents for training, e.g., in the form of tcp dumps, due to reasons of data privacy.

**Tracking and Persistency Problem** Another problem is the high amount of data that must be processed by the detection modules. Specifically, this is a problem of huge enterprise networks and comes hand in hand with the need to make these data persistent for later forensic analysis [34]. Further, the consideration of events from a long time span may be necessary for correlation, but, it is also computationally challenging and rises the need for efficient correlation algorithms.

## 2 Related Work

Besides the aforementioned rule-based detection as used in current SIEM approaches, the approach of Fan et al. [17] for intrusion detection is important due to the potential for application in a distributed way and its hybrid detection method. In their approach, an artificial anomaly generator is used to train a classifier to distinguish between (artificially generated) anomalies, known attacks and known normal data which partially addresses the problem of sparsity since unknown but suspicious events are classified as anomaly. However, it only uses examples of known attacks and known normal data without a representation of deep domain-specific expert knowledge as required in SIEM systems.

The approach of He et al. [28] is a rule-based detection approach which uses background knowledge represented in an ontology. This background knowledge is used to generalize the known rules to detect even variations of known attacks. Therefore, this approach addresses the requirement of handling sparsity and is shown to improve the detection accuracy by example. However, the approach lacks to validate the modeled expert knowledge

and is limited with respect to detecting unknown incidents since a threshold must be defined which controls the maximum generalization of the given rules for the detection.

The Iterative Boolean Combination (IBC) [33] approach of Khreich et al. learns boolean fusion functions applied to anomaly intrusion detection. The approach learns combination rules and underlying Hidden Markov Models which can represent uncertainty and incompleteness. The expert knowledge of this approach is limited to the combination rules. Expert knowledge with respect to IT asset information, relations to vulnerabilities, etc. cannot be modeled, which violates the requirement of SIEM systems.

With the approach of Yu et al. [64], it is possible to model background knowledge of attacks in Colored Petri-Nets which addresses the requirement of incorporating expert knowledge. Uncertainty has been addressed by extending the Petri-Nets by using Hidden States with trainable probabilities. However, the correlation of the events is performed by predefined pre- and postconditions which limits the approach to detect only known attack sequences.

MADAM ID [39] is noteworthy since the system consists of a misuse detection method with learning capabilities. MADAM ID can learn symbolic patterns which may be edited by security officers which addresses the requirement to model expert knowledge and allows learning from examples. However, the system cannot handle or represent incompleteness and uncertainty. Further, it does not address the requirement to handle sparsity since it is constrained to use a sufficient amount of reference data for learning. In 2007, Hwang et al. [29] continued this work by using the generated misuse patterns of a similar approach to MADAM ID in the Snort misuse detector. They could recognize a significant increase in the detection rate with only a small increase of the false positive rate. The combination of Snort misuse detection fed by an anomaly detection engine which mines patterns has also been proposed in [32]. This system should use several agents with a Bayesian network model to represent known attack types as well as normal behavior to combine anomaly and signature detection.

Gupta et al. [25] proposed to use Conditional Random Fields (CRFs) in the domain of intrusion detection. Their approach uses normal data as well as abnormal data for training. They investigated—based on the KDD cup 1999 data set [1]—that their Conditional Random Field approach outperforms the usage of Decision Trees and a naïve Bayes method. Later, they extended this approach by multiple layers of Conditional Random Fields with different features for the detection of different attack types [26]. However, this approach does not use modeled background knowledge and requires sufficient reference data to be trained.

In summary, all these methods contain concepts for partly addressing the problems stated in the introduction, however, none of them is covering the full spectrum.

## 3 Our Approach

The approach of this work is split into a preprocessing part called Tolerant Pattern Matching and a post-processing part by a Conditional Random Field. The Tolerant Pattern Matching (TPM) approach presented here is a special kind of Soft Pattern Matching. The TPM term is used to clarify the difference to other Soft Pattern Matching approaches that do not use ontological representations, logical expressions and generalizations of them in the matchmaking process. The post-processing part takes the matching values of the TPM as input for a statistical interpretation of incident hypotheses by the use of Conditional Random Fields.

We have chosen the TPM approach to fulfill the requirement of handling sparsity and to address the problem of incomplete background knowledge. TPM is used to transfer modeled background knowledge to unknown cases. Additionally, expert knowledge must be acquired by comprehensible concepts, like modeling logical expressions as in other SIEM systems. A CRF (as also used in this approach) cannot directly use or represent logical expressions. However, logical background knowledge can be represented and used by the combined approach of TPM and CRFs as we will see.

Further, the deployment of a correlation process which requires concrete examples for training—like Machine Learning models such as a CRF—is a difficult problem in the area of enterprise applications since the examples may contain confidential data. However, patterns as used by TPM abstract from individual examples and, therefore, are deployable. This provides an out-of-the-box solution for the correlation by still having the ability to refine the decision making by successively learning from examples by using a CRF post processing.

Empirical probabilities which are helpful for the decision making process must be derived from examples. Pattern matching approaches—including Tolerant Pattern Matching—are not capable of representing this. In the combination with probabilistic models like Conditional Random Fields, these probabilities can be represented and even be transferred to previously unknown cases by considering modeled background knowledge from the TPM. This allows us to adapt the modeled background knowledge to the application domain by learning from examples. Using empirical probabilities from the application domain in combination with TPM refines the handling of noisy input data by making use of probability

theory by still being able to exploit the given background knowledge.

Further, the trained CRF can determine which modeled patterns are most significant or insignificant for the inference with respect to the application domain (for example, directly by analyzing the trained weights or by feature selection as shown in [60]). This supports the revision of the modeled background knowledge and avoids redundant and useless patterns. Vice versa, the improvement of the pattern set supports the CRF inference. A synergy effect is given that successively improves the detection model and the modeled background knowledge.

## 3.1 Tolerant Pattern Matching

In the following, the method of Tolerant Pattern Matching is described which handles variations in the input data by generalizing patterns according to ontological background knowledge.

Tolerant Pattern Matching is realized by successively generalizing the patterns and determining a residual degree of satisfaction with respect to the input data (the observations a.k.a. the events). A pattern consists of logical compositions of constraints. Each constraint is expressed by a relation between two entities in the form entity, relation, entity. For example, the constraint

```
<observationEntity:SourceIP>
<relation:hasNetwork>
<individual:internalNetwork>
```

indicates that the SourceIP value of the observation must have the network `internalNetwork` which is assumed to be an individual from the ontology. An entity is assumed to be a variable in the scope of a pattern, an individual from the ontology or a place holder for a value from the observation. Briefly, an individual is a specific element in the ontology and a concept is a more abstract element in the ontology building a superset of several individuals. For further information about ontologies, the reader might refer to [6].

Each constraint in a pattern can be expressed as a query triple in a Description Logics query language like SPARQL [61] to use description logic reasoning in the ontology by using reasoners like pellet [14].

This allows to easily reason in complex domains and allows to use variables in the constraints. For example, the ontology may contain background knowledge about software, their vulnerabilities and information about the IT infrastructure. The conjunction of the following two constraints may be used to check if the target IP of the event has a specific vulnerability. The first constraint is:

```
<observationEntity:TargetIP>
<relation:hasSoftware>
<variable:ListOfTargetSoftware>
```

The second constraint is:

```
<variable:ListOfTargetSoftware>
<relation:hasVulnerability>
<individual:CVE-2012-2341>}
```

If this pattern does not match the observation since the IP does not have a software with this vulnerability, the TPM abstracts this pattern to determine the residual degree of matching. Therefore, the second constraint can be abstracted to a more general case, for example, the individual `CVE-2012-2341` may be abstracted to the concept `drupalVulnerability` which comprises of all CVE vulnerabilities that target the drupal software.

Now, we are looking for a similarity function for the TPM approach which guarantees that the best matching pattern (or in this context of CRFs, the best matching feature) dominates all other less matching patterns during the inference. This avoids that a huge number of slightly matching patterns overwhelm a strong (or even perfectly) matching one. In the following, a measure $\theta(\gamma^j, \gamma^k)$ for constraints $\gamma^j$ and $\gamma^k$ is assumed to quantify the similarity of an abstracted constraint $\gamma$ from the original level $j$ to an abstract (or generalized) level $k$.

$\gamma^\perp$ denotes the constraint on the most specific level $\perp$ (where $\perp$ is a positive integer) and $\theta(\gamma^j)$ is the short form of $\theta(\gamma^j, \gamma^\perp)$.

**Definition 1 (Similarity function)** *The similarity function $\theta$ is defined for the number of abstractions $\perp - k$ of a constraint $\gamma^k$ with respect to the number of patterns $|\boldsymbol{p}|$ by:*

$$\theta(\gamma^k) = \left(\frac{1}{|\boldsymbol{p}|}\right)^{\perp - k} \tag{1}$$

It can be shown that this measurement builds a similarity function according to [18] and [7].

This measurement is assumed to be 1 if the constraint is not abstracted, and decreases, if the constraint is getting more abstract by always being greater than or equal to 0. It can be assumed to be a special case of the SimRank similarity measurement [31].

The similarity function values of the constraints are combined to a matching degree of the whole pattern by applying some fusion operator $F(\theta_1, \ldots, \theta_n)$ similar to fuzzy pattern matching [11]. This is necessary to consider the semantics of the logical operators while abstracting the pattern. Therefore, a fusion approach is suggested by using the tree of logical operators in each pattern as follows:

**Definition 2 (Fusion Function)** *The fusion function $F(p)$ of a pattern p is recursively defined with respect to some similarity function $\theta$ of constraints $\gamma$ composed by logical operators as:*

$$\begin{aligned}
F(\gamma_1^j \wedge \gamma_2^j) &= \min\left(F(\gamma_1^j), F(\gamma_2^j)\right) \\
F(\gamma_1^j \vee \gamma_2^j) &= \max\left(F(\gamma_1^j), F(\gamma_2^j)\right) \\
F(\neg\gamma^j) &= \begin{cases} 1 - F(\gamma^j), & \text{for } i = \perp \\ \beta \cdot F(\gamma^j), & otherwise \end{cases} \\
F(\gamma^j) &= \theta(\gamma^j),
\end{aligned}$$

*where $\beta \in [0,1]$ is a penalty factor to additionally penalize the abstraction of negations.*

$\beta$ is a design parameter which is dependent on the used similarity function and the depth of the ontology. Optionally, a small $\beta$ may be chosen without multiplying it with the similarity function, i.e., the similarity function is omitted in the case of negation since it directly abstracts to a tautology.

The reason to choose the min-max fusion of conjunction and disjunction is to avoid that patterns with a huge number of disjunctions have a stronger tendency to be interpreted as true and patterns with a huge number of conjunctions stronger tend to be false.

The fusion function $F$ is monotonic with respect to $\theta$. It builds a partial order of patterns regarding the generality of their containing constraints. From this basis, it is necessary to find the best matching pattern with respect to some input data, i.e., the matching pattern with the biggest $F$.

For example, if we have a pattern set with two patterns and one pattern is specified to be the conjunction of one matching constraint and one constraint that must be abstracted two times to match, the residual degree of matching of the pattern is calculated for the similarity value of the first constraint $\theta_1 = 1$ and the similarity value of the second abstracted constraint $\theta_2 = \frac{1}{2}^2 = 0.25$ and the fusion function for the conjunction $F = \min(\theta_1, \theta_2) = 0.25$. Next, the combination of the presented Tolerant Pattern Matching approach with a post-processing by Conditional Random Fields is shown.

## 3.2 Tolerant Pattern Matches as Feature Function Values for Conditional Random Fields

The matches of the Tolerant Pattern Matching process are used as input features (i.e., the sufficient statistics) for a Conditional Random Field.

Conditional Random Fields are discriminative probabilistic models which have been suggested by Lafferty et al. [37] to overcome the label bias problem known from Maximum Entropy Markov Models [41]. Briefly, the CRF inference follows the theorem of random fields [27, 50] as stated by Lafferty. Wallach [63] suggested a simple notation of the CRF inference by assuming that the feature functions $f_j \in \boldsymbol{f}$ which are used to describe the input data are uniquely parameterized by the sequence of labels $\boldsymbol{y}$ and the sequence of observations $\boldsymbol{x}$ [63]. Wallach proposes the following equation for inference (adapted from [63, p. 4]):

$$Pr(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\lambda}) = \frac{1}{Z(\boldsymbol{x})} \exp\left(\sum_j \lambda_j f_j(\boldsymbol{y}, \boldsymbol{x})\right) \tag{2}$$

This equation relaxes the assumption of Maximum Entropy Markov Models by assuming that feature functions may depend on the whole sequence of observations (i.e., using $\boldsymbol{x}$ instead of $\boldsymbol{x}_j \in \boldsymbol{x}$). The model parameters $\boldsymbol{\lambda} = \{\lambda_1 \ldots \lambda_n\}$ are determined during training to efficiently infer the posterior distribution $Pr(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\lambda})$ based on the feature function values $f_j$ (the values are determined by the fusion function $F$ for the arbitrary patterns) parameterized by the observations (the events) and the labels (incidents or threats) to infer. Details about Conditional Random Fields can also be obtained from [21, p. 108].

The application of a discriminative model like Conditional Random Fields is used in this approach to be more robust with respect to the unknown class prior of incidents since before the training of the model by examples from the application domain, the frequency of incidents in this domain is unknown. Further, Conditional Ran-

dom Fields have already been successfully applied in the domain of intrusion detection, e.g. [26].

One feature is built for each combination of labels and patterns, i.e., $\boldsymbol{f} = \{f_1, \ldots, f_n\}$ with $n = |\boldsymbol{y}||\boldsymbol{p}|$, and $|\boldsymbol{y}|$ being the number of labels and $|\boldsymbol{p}|$ the number of patterns. Each feature matches exactly on one label and returns—in the case that the associated label is queried—the result of the fusion function of the associated less abstracted but matching pattern, i.e. the biggest F for this pattern.

The proposed combination of Tolerant Pattern Matching and Conditional Random Fields requires that a higher degree of matching leads to an increased influence to the posterior probability of the CRF which can be proven by the monotonicity of Equation 2. The intuition is that better matching patterns should stronger account for the final decision making. Further, the used similarity function can be shown to ensure that the best matching feature dominates the sum of all less matching features with equal (or less) significance (i.e., the assigned weights) and with a comparable abstraction lattice in the CRF inference. This guarantees that several slightly matching patterns do not overwhelm a perfectly matching pattern and avoids a too strong smoothing of the posterior probability distribution.

## 3.3 Modeling Incidents - The Incident Matrix

In this report three threat levels are used: A **normal** threat ($N \in \boldsymbol{t}$) is assigned to an incident to indicate that the pattern matches triggering this incident does not indicate an increased threat. In other words, a normal threat indicates that the detected incident is a false positive and indeed is no serious incident.
A **suspicious** threat ($S \in \boldsymbol{t}$) is assigned if the triggering pattern matches are suspicious but do not indicate a high threat at this time. This threat level is used to mark potentially interesting situations which should be kept for further correlation.
A **dangerous** threat ($D \in \boldsymbol{t}$) produces the highest prioritization and is used to indicate dangerous pattern matches.

For modeling incidents, each pattern is assumed to either **match** ($M$), **mismatch** ($\neg M$) or to have an **unspecified** ($U$) value indicating that the matching value does not affect the decision making.

The modeling of incidents by pattern matches can be represented as a single two dimensional matrix as visualized for three patterns $p_1, p_2, p_3 \in \boldsymbol{p}$ and two incidents $i_1, i_2 \in \boldsymbol{i}$ in Table 1 which we call incident matrix.

For example, the ping incident ($i_2$) does not care about the source IP of the event, therefore, the matching value to $p_3$ is set to unspecified. $p_2$ must match since the ping incident requires the ping pattern to match. Further, if $p_2$ is known to match, $p_1$ cannot match since the event cannot be a ping and a port-scan event at the same time.

| | $p_1$ | $p_2$ | $p_3$ | threat level |
|---|---|---|---|---|
| description | port-scan | ping | source is admin | |
| $i_1$ (non admin scan) | $M$ | $\neg M$ | $\neg M$ | dangerous |
| $i_2$ (ping) | $\neg M$ | $M$ | $U$ | normal |

Table 1: Example of an incident matrix that relates pattern matches to incidents and threat levels.

## 3.4 Two Layers of Conditional Random Fields

After specifying the input for the Conditional Random Field (i.e., the similarity function, the fusion function and the pattern matches as feature values), we focus on the output, i.e., the labels of the CRF or the inference target. In this work, two disjunct Conditional Random Fields are used, one for detecting and assessing threats (called Detection Layer) and one for explaining them (called Explanation Layer):
**Detection Layer** This layer has three labels representing three threat levels a) dangerous, b) suspicious and c) normal according to the threats specified in the incident matrix. All patterns are used as input which results in maximally $3|\boldsymbol{p}|$ feature functions (and weights) for the CRF of this layer if all threat levels are used. The Detection Layer is used to detect incidents out of the stream of events by determining the threat level. Further, this layer is essential for prioritizing incidents as we will see.
**Explanation Layer** This layer has one label for each modeled incident and is used for already detected incidents to explain the arbitrary steps belonging to the incident. The Explanation Layer may only be used in succession of a high prioritized incident from the detection layer and, therefore, may not affect the computational efficiency of the incident detection. If $|\boldsymbol{i}|$ is the number of modeled incidents, there are $|\boldsymbol{i}||\boldsymbol{p}|$ feature functions (and weights) in this layer.

There are two major reasons for splitting the detection and the explanation layer. The first is obviously a smaller inference effort for detecting incidents since the threat layer normally has a lower number of labels (only three). The second is the inability to derive the threat level from the explanation layer since the probabilities of the labels are not independent with respect to the semantically overlapping features.

## 3.5 Prioritization of Incidents - The Hypotheses Pool

Let $t$ be a certain threat level from $\boldsymbol{t}$ and $j$ an index over the sequence of observations $\boldsymbol{x}$ with each observation $\boldsymbol{x}_j$.

The probability of all observations belonging to a certain threat level is given by the inference of the Detection Layer $Pr_{det}$ by:

$$Pr_{det}(t|\boldsymbol{x}) = \prod_{j=1}^{|\boldsymbol{x}|} Pr_{det}(t|\boldsymbol{x}_j) \tag{3}$$

In this work the following prioritization is used:

**Definition 3** *Given a sequence of observations* $\boldsymbol{x}$*, the prioritization* prio *is determined by:*

$$\text{prio}(\boldsymbol{x}) = \log_{10}\left(\frac{0.5Pr_{\text{det}}(D \cup S|\boldsymbol{x}) + 0.5Pr_{\text{det}}(D|\boldsymbol{x})}{Pr_{\text{det}}(N|\boldsymbol{x})}\right) \tag{4}$$

The prioritization compares the likelihood that all observations belong to an incident with dangerous or suspicious threat against the likelihood that all observations belong to an incident with a normal threat. This measurement is similar to the likelihood ratio as often used in sensor fusion approaches (e.g., [24]). Please note that a trade-off of false positives and true positives can be specified by a threshold for this prioritization.

The objective of the correlation process is to determine a group of incidents which most likely are all dangerous. Each such group builds a hypothesis in the following, i.e., each hypothesis comprises of a sequence of events and incidents. Potentially, each permutation of the incidents may build a hypothesis. For example, a ping event might lead to a hypothesis with a ping incident, next a port-scan event generates two new hypotheses, i.e. the hypothesis with the combination of the ping and port-scan incident and solely the port-scan incident.

In practice, this is not feasible due to the exponentially growing inference effort. Therefore, we use a concept we call Hypotheses Pool which keeps the hypotheses with the highest priority and drops hypotheses with the lowest priorities to limit the total number of hypotheses. Further, the Hypotheses Pool consists of buckets where each bucket holds hypotheses with a predefined number of incidents. Each such bucket is individually checked with respect to a predefined maximum size of the bucket and the least prioritized hypotheses are dropped from these buckets. These buckets are required since hypotheses with more incidents have typically a higher prioritization than hypotheses with less incidents due to the increased certainty that at least one of their incidents is dangerous. The absence of these buckets would avoid the generation of complete new hypotheses which is at least undesired in the long term.

Further, the introduction of Hypotheses allows one to define temporal relations in their scope. Besides the typical description logical constraints and logical compositions, patterns can describe temporal relations in their constraints in the scope of a hypothesis. This can be used to express dependencies over time, e.g., a failed login attempt may be considered more suspicious with a preceding port-scan. Therefore, the three temporal relations **currently**, **previously** $\ominus(\gamma(\boldsymbol{x}_j)) = \gamma(\boldsymbol{x}_{j-1})$ and **once** $\Diamond\!\!\!-(\gamma(\boldsymbol{x}_j)) = \gamma(\boldsymbol{x}_1) \vee \ldots \vee \gamma(\boldsymbol{x}_{j-1})$ can be used in the constraints of the patterns to express temporal relations.

## 3.6 Training

The training of the Conditional Random Fields is done by Improved Iterative Scaling (IIS) [9]. The empirical probabilities for training are determined by the incident matrix. The matrix can be filled by modeled knowledge as well as by concrete examples of attacks which offers to use modeled expert knowledge as well as experienced misclassifications during the application of the system. This offers the ability to train the proposed system during application, for example, to consider the individual network behavior of the application domain.

The modeling of incidents by experts often produces an artificial imbalance [12] between benign and malign incidents. This occurs since incidents are modeled without the information about how frequent the incidents occur. This problem is reduced by learning from examples, but remains to be a challenge for the first deployment of the detection engine. Imbalanced Data is a serious problem for several machine learning approaches [8, 24] and a known problem in the Intrusion Detection domain, too [12]. Batista et al. discovered that over-sampling methods are well-suited for imbalanced datasets [8], therefore, this method is also tested in this work.

One problem of CRFs trained with IIS is that they tend to overfit the data. In the original version of IIS, the model parameters (the weights $\boldsymbol{\lambda}$ in Equation 2) are not limited and may even converge to infinitely huge numbers—which has obviously a high impact on the posterior distribution. Therefore, regularization is typically used to overcome this problem, briefly, by tying the model parameters near to zero. In regularization, the model parameters themselves are considered as random variables with a specified prior distribution. Smith et al. discovered that regularization priors as Gaussian, Laplacian or Hyperbolic *perform roughly equally well* if they are appropriately parameterized [54]. Chen et al. agreed with that even while comparing further regularization techniques [13]. They also derived the gradient for Improved Iterative Scaling with a Gaussian prior used in this work.

With these methods—IIS, regularization and oversampling—we are well prepared to train the CRFs with examples as well as with modeled incidents.

## 4 Empirical Evaluation

Due to the confidentiality of real data including serious attacks, only a few security benchmarks are available [59]. One frequently used benchmarking dataset is the KDD CUP'99 dataset [1], which consists of connection records and is well-suited for testing low-level IDSs. However, this dataset is less appropriate for benchmarking SIEM systems on the higher event level. The KDD CUP underlying DARPA dataset [40] refers to raw data, and is, therefore, even less appropriate for testing SIEM correlation engines. Further, several problems of both datasets have been investigated [57, 42] which lead to the decision to use a more up-to-date data set which more accurately fits the level of SIEM event correlation. Hence, we used the sandnet data set consisting of 1407 recorded malware samples and their generated Snort events. These samples are a subset of the generated samples from the sandnet project [53]. This collection of sample files is further called the sandnet dataset containing just malign Snort events. Benign traffic has been recorded in the Artificial Intelligence workgroup of our institute (TZI) which is further called "TZI data set". One month (31 days) of traffic has been analyzed by a Snort IDS with the same rule set as used in the sandnet malware analysis. The resulting data set has been filtered by just using static IP addresses of the institute. These 24 static IP addresses are reserved for staff members to avoid that the benign dataset becomes contaminated by mobile computers with potential malware infections.

The rule-based pattern matching method as used by most SIEM systems has been reimplemented for automatic testing to generate statistically significant results. Further, a naïve Bayes and a CRF approach compete to compare models from the two major fields of probabilistic models, i.e. generative vs. discriminative models [45]. For the performance analysis of the detection, we used probabilistic sampling to generalize from a few samples to the whole population. We have chosen Simple Random Sampling with replacement due to its acceptance for producing a representative evaluation [58, 15] and due to its simplicity. The test parameters are set to a level of significance of 0.05 and a tolerable sampling error of 0.03. Hence, at least 1068 samples are required to guarantee these test conditions.

**Test 1** uses one benign sample (X) and one malign sample (sandnet) for generating patterns. The generated patterns avoid to use network specific characteristics like IP addresses and ports to ensure that the correlation does not use characteristics from the different networks from which the samples are recorded. The test evaluates the approaches against one sample from the combined dataset of sandnet and TZI which uses two hours of malign hidden in one day of benign events.

This combination is done by inserting the malign events into the stream of benign events. Since no specific network characteristics are used in the evaluation, this does not bias the test results. Specifically, each test result is collected by a) the modeling of benign incidents based on a sample of the TZI dataset b) the modeling of malign incidents based on a sample from the sandnet dataset c) testing against benign events from the TZI dataset to measure true negatives and false positives and d) testing against malign hidden in benign events from the combined dataset to measure true positives and false negatives.

In the Receiver Operating Characteristic (ROC) curve in Fig. 1, we compared 7 test models using the same test and training data for each model. A varying threshold over the prioritization function has been used to create the ROC curve based on 1254 test results. The test investigates the following seven models: 1) The naïve Bayes model $\mathrm{NB}_{pm}$ using a rule-based approach with pattern matching that does not use abstractions (PM)—indicated by the index $pm$ 2) $\mathrm{NB}_{tpm}$, the same model as in model number one, but with using tolerant pattern matching instead of rule-based hard pattern matching (indicated by $tpm$). 3) $\mathrm{NB}_{tpm,\alpha=0.1}$, the same naïve Bayes model 2, but with using Laplace Smoothing with $\alpha = 0.1$, i.e., adding 10% to the samples to smooth the distribution to avoid a so-called wipe-out in the naïve Bayes model which might occur by training with a low number of training samples. 4) The $\mathrm{CRF}_{pm,\sigma=1}$ model using PM and a Gaussian Prior with $\sigma = 1$. 5) $\mathrm{CRF}_{tpm,\sigma=1}$, as number four but using TPM instead of PM. 6) $\mathrm{CRF}_{tpm,\sigma=5}$, the same CRF model as number five, but with a different parameter for the Gaussian Prior, i.e., $\sigma = 5$. 7) STRICT, a method only using PM without any probabilistic post-processing. This method performs similar to most enterprise SIEM systems. This has been verified in several tests using an instance of ArcSight ESM 5 which performed equally to STRICT as expected.

Fig. 1 shows that both naïve Bayes and CRFs benefit from using TPM. Specifically, model five and six—using CRFs and TPM—significantly perform better than the STRICT method and the naïve Bayes models. In the area around the false positive rate 0.25, the gap between model five and six to the other models is quite large. This can be explained by the threshold used to generate the samples for the ROC curve. If the threshold deviates from zero, the influence of the TPM is fading since partially matching patterns from TPM are producing less sharp probability distributions (to express their uncertainty) which results in prioritization values near to zero. This conjecture is underpinned in Table 2 showing the false positive rate for a prioritization threshold—to treat a hypothesis as serious incident—of zero.

As we see, this threshold leads to a false positive rate

| no. | model | tp | fp | prec. | rec. | acc. | F1 |
|---|---|---|---|---|---|---|---|
| 1 | $NB_{pm}$ | .82 | .42 | .66 | .82 | .7 | .73 |
| 2 | $NB_{tpm}$ | .86 | .37 | .7 | .86 | .75 | .77 |
| 3 | $NB_{tpm,\alpha=.1}$ | .87 | .37 | .7 | .87 | .75 | .78 |
| 4 | $CRF_{pm,\sigma=1}$ | .79 | **.22** | .78 | .79 | .79 | .79 |
| 5 | $CRF_{tpm,\sigma=1}$ | **.95** | **.22** | **.81** | **.95** | **.87** | **.88** |
| 6 | $CRF_{tpm,\sigma=5}$ | .94 | **.22** | **.81** | .94 | .86 | .87 |
| 7 | STRICT | .79 | **.22** | .78 | .79 | .79 | .79 |

Table 2: Test results of test one with a prioritization threshold of zero.

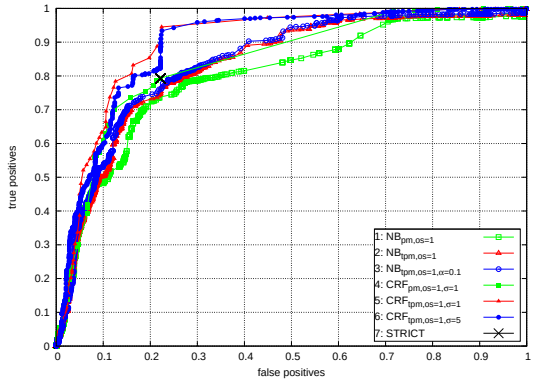


Figure 1: ROC curve of Test 1; parameters are Laplace smoothing $\alpha$, oversampling factor *os*, and Gaussian Prior $\sigma$.

of 22% and a true positive rate of over 90%. Model five has a 16% higher true positive detection rate than the STRICT method by keeping the false positive rate of 22%. Further, we see that the CRF performs slightly better with a Gaussian Prior of $\sigma = 1.0$ than with $\sigma = 5.0$. For lower false positive rates, Fig. 1 shows that smaller $\sigma$ values often lead to an improved detection. TPM improves the detection accuracy for CRFs and for naïve Bayes. CRFs with TPM perform better than all the other tested methods. However, we show that a post-processing by naïve Bayes performs poorly with the given test parameters, even worse than using conventional pattern matching (PM) alone. This surprising result is investigated in the next test. Please note that the results are given for a SIEM level of detection, i.e., they are relative to the underlying events produced by the sensors, e.g., an IDS. In this case, based on the underlying Snort sensor.

**Test 2** investigates the poor performance of the naïve Bayes models from Test 1 in detail by analyzing the detection performance with different training parameters. Therefore, the smoothing factor $\alpha$ of the Laplace smoothing has been varied. The result of this test with 1086 samples is visualized in Fig. 2. As we see, the varying parameters do mostly not lead to significant changes in the detection accuracy. Only the change of the oversampling factor *os* to a ratio of two times more benign

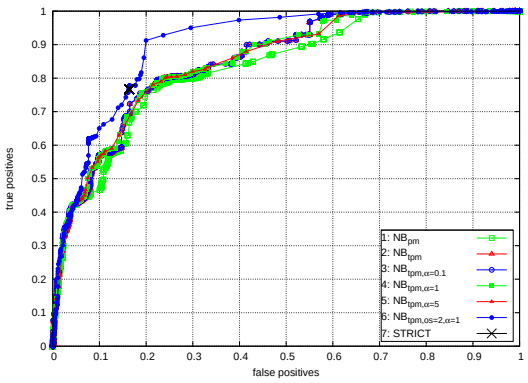


Figure 2: ROC curve of test 2 comparing different naïve Bayes parameterizations with activated oversampling in model 6.

samples than malign samples in model 6 significantly improves the detection accuracy. Even in comparison with an oversampling factor of 1 (model 2 and 3 in Fig. 1), an oversampling factor of 2 leads to a significant improvement (model 6 in Fig. 2). The explanation of this behavior is due to the nature of generative models to strongly depend on the prior of the distribution to infer. In this case the ratio between benign and malign incident threats which is specified by the oversampling factor or (in case of disabling oversampling) by the ratio in the modeled data highly affects the detection accuracy. Since the threat ratio in the modeled data does not reflect the real empirical frequency of benign and malign incident occurrences, naïve Bayes models are highly dependent on a correct oversampling factor. Since the oversampling for naïve Bayes models is critical to tune, a model which is more robust to this parameter is desirable.

**Test 3** evaluates 1151 samples with respect to different parameters to test the robustness of the CRFs with respect to the oversampling (*os*) and the regularization prior ($\sigma$). Fig. 3 shows that all CRF inferences (model 2–6) do not vary significantly according to varying parameters. This is a great advantage of the CRF model in comparison to the naïve Bayes approach. Only the CRF inference without oversampling (model 2) performs slightly worse than the other CRF models using oversampling. Another interesting derivation is that naïve Bayes with a strong oversampling ($os = 3$) outperforms the CRF at a false positive rate near 0.27. However, naïve Bayes slower increases the true positive rate near to STRICT at a false positive rate of 0.18. We see that naïve Bayes with TPM, an oversampling factor of 3, and a Laplace smoothing of 1 is also a reasonable probabilistic model in comparison with the CRF models for this dataset and test setup.

**Test 4** evaluates 1077 samples. Each sample consists of modeling two benign and two malign incidents. In

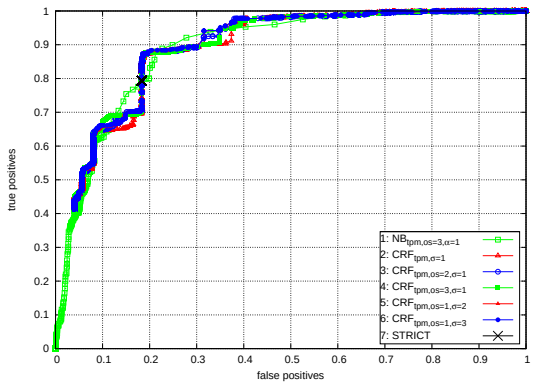


Figure 3: ROC curve of test 3 with varying parameters for CRF, naïve Bayes, and STRICT.

| test | tp | fp | prec. | rec. | F measure |
|---|---|---|---|---|---|
| 1 | 0.79 | 0.22 | 0.78 | 0.79 | 0.79 |
| 2 | 0.77 | 0.16 | 0.83 | 0.77 | 0.8 |
| 3 | 0.79 | 0.18 | 0.81 | 0.79 | 0.8 |
| 4 | 0.91 | 0.34 | 0.73 | 0.91 | 0.81 |

Table 3: Performance of the STRICT method for all test setups.

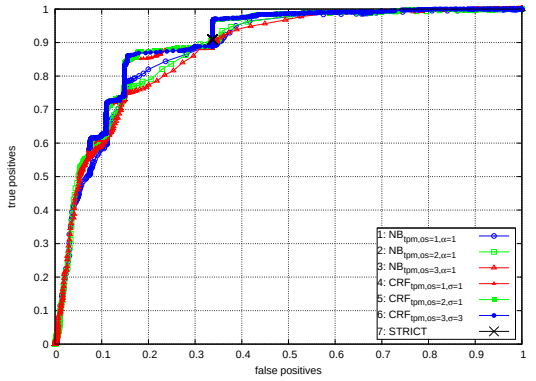


Figure 4: ROC curve of test 4 with two malign and two benign samples trained.

contrast to the previous tests, this one evaluates how the models behave when the trained modeled data are less sparse. Therefore, instead of one benign and one malign training sample two benign and two malign samples are used. As expected, the detection rate of STRICT increases — as can be obtained from Table 3 for all four test setups — since more events are known to be malign. However, the false positive rate increases significantly, too.

As can be seen in Figure 4, approaches using TPM are not significantly varying with respect to the last test; their learning has already converged and, therefore, this underlines that the models are more appropriately handling sparse data.

In comparison with STRICT, CRF/TPM can still better discriminate malign and benign incidents.

The aim of **test 5** which uses 1244 samples is to discover the influence of the reference data in combination with the modeled knowledge. This test has been performed with the same simulation data sets and a similar setup as the tests before. For each test, one sample from the TZI data set and one sample from the sandnet data set have been used to model incidents. Beyond this, two events from randomly chosen TZI samples have been used as reference data for benign incidents. All associations to further incidents are assumed to be unknown to make the scenario as realistic as possible since an expert cannot be assumed to necessarily assign all matching and mismatching incidents for an observation. Accordingly, two events from randomly chosen sandnet samples have been used as reference data for training malign incidents.

On average, 60.12 malign events have been sent to the correlation engine for each sandnet sample and 20,17 benign events have been sent for each TZI sample. Please remember that the events are generated by Snort which explains the low amount of benign events. The correlation engine has been trained with 3 modeled patterns belonging to malign incidents and 3.23 modeled patterns belonging to benign incidents on average.

One interesting behavior of the naïve Bayes models in this test is the increased false positive rate of the $NB_{tpm,os=2,\alpha=1}$ model with respect to the model with a lower oversampling factor $NB_{tpm,os=1,\alpha=1}$ as to obtain from Figure 5. One might expect that by increasing the normal to dangerous oversampling rate ($os$) the false positive rate should decrease. However, this is not guaranteed in general. The process of oversampling is a random process, which might lead to different inference results, specifically in cases with a small number of training data. Further, by the use of reference data the risk emerges that ambiguous patterns are used for the oversampling. For example, a previously modeled harmless incident for one pattern might be extended by a new malign incident — from the reference data — indicating a malign incident for the same pattern. During the oversampling process, this pattern might be selected for oversampling both, malign and benign incidents that might confuse the model for observations matching to this pattern. Additionally, in cases where the benign-to-malign rate is already fitting to the oversampling factor and, therefore, no oversampling is necessary, the absence of the oversampling can lead to an increased uncertainty due to a model with a smaller number of oversampled training data. This effect might become even worse while applying Laplace smoothing.

Therefore, there are several reasons why the uncertainty of the models might increase for changes in the oversampling factor. This increased uncertainty is re-

flected by a smaller distance between the prioritizations of the malign and benign hypotheses. For example, one sample from this test has been assessed by the $NB_{tpm,os=2,\alpha=1}$ model to have a prioritization of $prio = 1,72$ for the malign test data and a prioritization of $prio = 0,16$ for the benign test data. The same model for a smaller oversampling factor, i.e. $NB_{tpm,os=1,\alpha=1}$, has prioritized these hypotheses as $prio = 1,76$ and $prio = -0,01$. This small shift in the discriminative power of the model has already generated a false positive detection for a prioritization threshold of zero.

While further increasing the oversampling factor, e.g., in the model $NB_{tpm,os=3,\alpha=1}$, these side-effects are getting more and more redeemed by the strong class prior of the naïve Bayes model that suppresses the detection of false positives and coincidently the true positives. This shows again, that the independence of the oversampling factor should be preferred.

As to obtain from the ROC curve of Figure 5, the CRF performance is similar, but shifted with respect to the results from test one (see Figure 1). For example, the true positive rate for the false positive rate 0.2 in test one for the $CRF_{tpm,os=1,\sigma=1}$ model is near to 0.8, whereas for the same false positive rate the true positive rate for the same model in test five is near to 0.87. However, with a shifted false positive rate to 0.25 the true positive rate in test one straightly increases to around 0.95 and in test five the true positive rate is still near to 0.87. This lower value can also be explained by the additional reference data that may confuse the model due to ambiguous pattern matches.

Further, the comparison of the ROC curves from test one and five shows that the CRF model does not significantly benefit from the additional reference data. One reason is that the reference data only confirm the modeled knowledge. In other words, if the modeled knowledge is already sufficiently representing the domain, the additional reference data are not needed. Additionally, if the modeled knowledge more appropriately describes the domain than the few reference data, the incorporation of reference data may even lead to a lower discriminative power of the model, specifically if several matching values are set to unknown like in this test.

In contrast, the naïve Bayes model $NB_{tpm,os=2,\alpha=1}$ of this test performs worse than the same model from test two. For example, the true positive rate in test five for a false positive rate of 0.3 is near to 0.85, whereas the true positive rate of the same model in test two is at 0.95. However, the other naïve Bayes models—$NB_{tpm,os=1,\alpha=1}$ and $NB_{tpm,os=3,\alpha=1}$—are performing very well with the use of the additional reference data in this test. They even outperform the CRF models for some false positive rates, such as 0.3. They are, however, strongly dependent on the oversampling factor and are less rapidly increasing

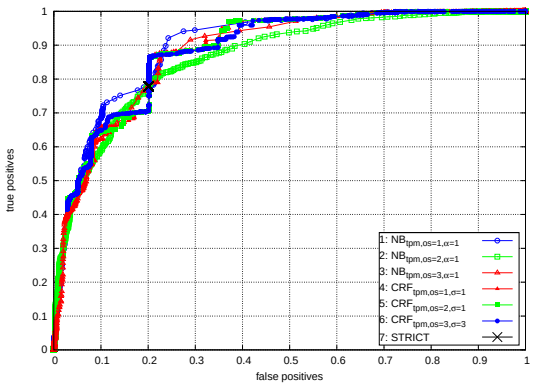


Figure 5: ROC curve of test 5 with different models that are trained by modeled dangerous and benign incidents each from one sample file with additionally two randomly chosen dangerous and benign concrete observations.

the true positive rate for a false positive rate near to 0.21. In comparison with test four, these naïve Bayes models perform better in this test without the additional patterns from test four. The CRF models have nearly the same false positive to true positive ratio in test four and test five, i.e., the use of the additional reference data in test five compensates the missing modeled patterns from test four.

The exact true positive and false positive rates as well as the precision, recall, accuracy and F1 score can be obtained from Table 4 for a prioritization threshold of zero. The approaches with the lowest false positive rates are the *STRICT* method and the $NB_{tpm,os=3,\alpha=1}$ naïve Bayes model. However, both have a low true positive rate and, therefore, cannot be recommended to detect incidents. The highest true positive rate is given by the $NB_{tpm,os=1,\alpha=1}$ naïve Bayes model, however, with a high false positive rate, too. The CRF models have nearly the same false positive rate as the best models, i.e., $NB_{tpm,os=3,\alpha=1}$ and *STRICT*, but a significantly higher true positive rate. Further, they are independent of the oversampling factor for this threshold. The $F1$ measure is equal for the best naïve Bayes model and all the CRF models. The naïve Bayes models are performing very well and even can potentially outperform the CRF models for some false positive rates if they are properly parameterized. However, the parametrization is not trivial and can be avoided by using Conditional Random Fields while keeping the F1 score. Therefore, this test shows again that the CRF models should be preferred with respect to naïve Bayes models in most cases, but also that naïve Bayes models are competitive.

| model | tp | fp | precision | recall | accuracy | F1 |
|---|---|---|---|---|---|---|
| $NB_{tpm,os=1,\alpha=1}$ | **0.95** | 0.3 | 0.76 | 0.95 | 0.83 | **0.84** |
| $NB_{tpm,os=2,\alpha=1}$ | 0.89 | 0.37 | 0.71 | 0.89 | 0.76 | 0.79 |
| $NB_{tpm,os=3,\alpha=1}$ | 0.78 | **0.2** | 0.8 | 0.78 | 0.79 | 0.79 |
| $CRF_{tpm,os=1,\sigma=1}$ | 0.87 | 0.21 | 0.81 | 0.87 | 0.83 | **0.84** |
| $CRF_{tpm,os=2,\sigma=1}$ | 0.87 | 0.21 | 0.81 | 0.87 | 0.83 | **0.84** |
| $CRF_{tpm,os=3,\sigma=1}$ | 0.87 | 0.21 | 0.81 | 0.87 | 0.83 | **0.84** |
| *STRICT* | 0.78 | **0.2** | 0.8 | 0.78 | 0.79 | 0.79 |

Table 4: The performance measures for a prioritization threshold of zero for the models in test five.

## 5 Computational Complexity

It can be shown that the TPM finds the best match of a pattern (the smallest generalization) in $O(n^{\lg(2^d-1)})$ Description Logic reasoner calls with $n$ being the depth of the abstraction lattice (in the maximum the depth of the ontology) and $d$ being the number of dimensions to find a solution, i.e. the number of constraints that share a common variable in a pattern. Further, a CRF is known to take $O(|\boldsymbol{f}|)$ with $|\boldsymbol{f}|$ being the number of feature functions, i.e. in the case of incident detection $|\boldsymbol{f}| \leq 3|\boldsymbol{p}|$. Therefore, the TPM of small dimensions and the CRF inference is very efficient. The Hypotheses Pool needs to extend and check each contained hypothesis. Using the Hypotheses Pool has the great advantage that it makes the system difficult to evade since an attacker may hardly predict which hypotheses are currently in the pool and which may become extended to build a dangerous or most suspicious hypotheses (especially for long term correlations). However, using the Hypotheses Pool can be computationally expensive if the CRF/TPM inference is complex.

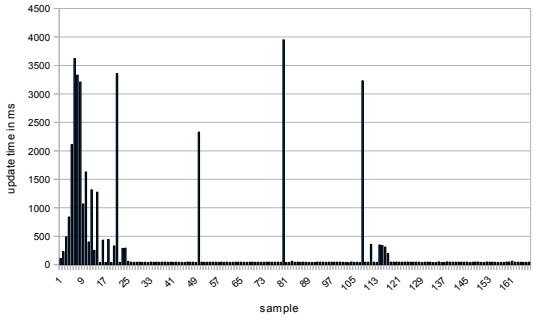


Figure 6: One example from the runtime performance test for a case with eight patterns, 64 feature functions for the Explanation Layer and 16 feature functions for the Detection Layer.

On average, the correlation takes 403,57 ms with a standard deviation of 1110.97 for each sample. This high value is explained by the use of the immediate generation of explanations, i.e., in the current implementation the explanations (from the explanation layer CRF) are always generated for new hypotheses instead of only on demand of the user. The consequence can be seen in Figure 6. At the beginning of each test, the Hypotheses Pool is empty, therefore, each new observation will create a set of new hypotheses. For all these new hypotheses, explanations are calculated and presented to the user. Therefore, the correlation engine requires more computational power in the beginning of the test since at this time the highest number of explanations must be calculated. Later, only a few observations are incoming that result in new hypotheses that are not dropped by the Hypotheses Pool due to their priority assessment. The generation of new hypotheses is reflected by peeks in Figure 6. For example, at sample 21 and sample 50 a comprehensive extension of the hypotheses can be recognized, whether, at sample 114 a smaller number of extended hypotheses can be seen.

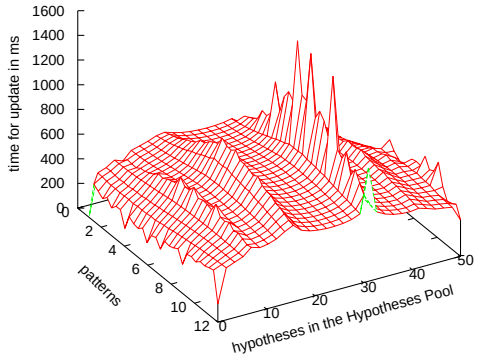


Figure 7: The average values of the runtime performance measurements.

Figure 7 shows the average update time in ms for all 13372 performance measurements with respect to the number of patterns and the number of hypotheses in the Hypotheses Pool. As to obtain from this figure, the peeks correlate with the initial development of the number of hypotheses in the Hypotheses Pool. The Hypotheses Pool expands by the series $2^n - 1$ without considering the reduction step (i.e., dropping hypotheses from the Hypotheses Pool). Therefore, one would expect peeks at 1, 3, 7, 15 and 31 hypotheses since for the given Hypotheses Pool size (five buckets with size ten) these steps are without any required reduction, i.e. a maximum expansion. In all other cases, the Hypotheses Pool dropped some hypotheses due to the reduction step which decreases the inference effort. While further hypotheses are generated, the Hypotheses Pool is guaranteed to drop some of the hypotheses since the capacity of at least some hypotheses buckets has been reached. This is the reason why the maximum inference effort is recognized for 31 hypotheses in the Hypotheses Pool which is the maximum extension of the number of hypotheses in the Hypotheses Pool that is possible. Further, it is interesting

that the number of patterns do not increase the inference effort significantly.

In our current implementation we were able to process between 15 and 20 events per second (see Figure 6) for an Hypotheses Pool with size 50 — if the Hypotheses Pool has been filled — on an Opteron Processor 275 with 2,2 GHz. We explain this by not using parallelizations, for example each inference might be performed in parallel, and the absence of code optimization. For example, we build a query string in SPARQL to query a Description Logic reasoner for each pattern check instead of using the direct API of the pellet reasoner. However, it is clear that the presented approach cannot compete with respect to the computational efficiency to current enterprise SIEM correlations. Therefore, we propose to use this approach as an add-on to existing correlations.

## 6 Conclusion

Since current SIEM correlation engines can process huge amounts of events, we propose to use such systems as a basis to process all known events in advance. Known benign events may be filtered out and known suspicious or dangerous events and unknown events may be forwarded to the proposed method of TPM/CRF for an advanced correlation and assessment. Further, the TPM/CRF approach may be used to revisit the forensic offline data to adapt the current pattern / rule set of the underlying SIEM system. SIEM systems like ArcSight ESM already use a taxonomy of event categorization which makes the integration of TPM reasonable.

The approach presented in this report addresses the initially mentioned problems by the following properties: The **actuality problem** is addressed by the combination of TPM and CRF. The TPM offers to abstract the modeled patterns to assess even unknown cases by the use of a similarity measurement which also addresses the **knowledge acquisition bottleneck problem**. The CRF has the potential to analyze the patterns with respect to their significance in the decision making to remove or revisit obsolete patterns. Further, the CRF can learn from examples and incorporate this into the decision making to keep the decision making up-to-date. The evaluation has shown that the false positive rate can be kept low while increasing the detection rate. Therefore, the balance of true to false positives could be improved to current SIEM correlations which addresses the **balance problem**. The **dependency problem** has been considered while using TPM which offers a comprehensible way for security experts to model complex dependencies among the events. The use of an ontology and a Description Logic reasoning provides the option to model even complex dependencies to background knowledge such as IT infrastructure information. The **heterogeneity problem** has been addresses by using the IDMEF [44] to provide a common syntax normalization and an ontology has been used for mapping the heterogeneous events to a common semantic for all sensors. This is a pragmatic solution and there are already advanced approaches to aggregate the events from several sensors, for example in the research field of sensor fusion [23]. However, this pragmatic approach does not require any training data and is also used in current enterprise SIEM systems. The drawback of this approach is that the outcomes of the sensors must be known in advance to provide a semantic normalization. In case of the Snort IDS, which is used in this work, we used a small program that parses the detection rule set of Snort to generate the set of all possible outcomes and assigns these outcomes to the most similar elements (concepts) in the ontology. The **interpretation problem** has been addressed by mapping the incoming events to named incidents. In case of detected unknown incidents, the most similar and most likely incidents are determined by the combined TPM/CRF approach to give the security officer the best explanations for the incident as possible. The **sparse data problem** has been addressed by the use of TPM which offers to deploy the system with predefined rules and abstracting these rules for incoming events that cannot be classified by the modeled rule set. A probabilistic discriminative classifier (CRF) has been chosen to be most robust with respect to the absence of concrete domain information, specifically the ratio of incidents to normal events. The drawback of CRFs to typically require more training data than their generative counterparts like Bayes models is compensated by the use of the pattern matches from TPM as input features. The **tracking and persistency problem** has been addressed by the Hypotheses Pool which offers long term correlations by keeping the most promising hypotheses for correlation until other more promising hypotheses require the reduction of the Hypotheses Pool. One might think that an attacker might easily evade this system by flooding the Hypotheses Pool with lots of suspicious events. However, several suspicious events/incidents would lead to an increased prioritization and the security officer will at least be warned. Further, it is difficult for an attacker to know how the events are prioritized since this depends on the modeled rules, the ontology and the probabilistic model which has learned from the application domain. Beyond this, the events are stored into a database for further offline analysis.

In summary, all stated problems have been successfully addressed by the proposed approach. However, it requires additional computational power which avoids to use this correlation technique as a self-contained solution. Therefore, we propose to use this approach as an add-on in current SIEM systems.

## References


[1] ACM SPECIAL INTEREST GROUP ON KNOWLEDGE DISCOVERY AND DATA MINING. Kdd cup 1999: Data. Download, provided online at http://www.sigkdd.org/kddcup/index.php?section=1999&method=data; visited on September 28, 2011.

[2] ALIENVAULT LC. Alientvault unified siem system description version 1.0. Website, available online at http://alienvault.com/docs/AlienVault_Unified_System_Description_1.0.pdf; visited on December 5, 2011.

[3] ANDERSON, R. J. *Security Engineering: A Guide to Building Dependable Distributed Systems*, second ed. Wiley Publishing, Inc., 2008.

[4] ARCSIGHT, INC. Smartrules and cross-correlation. PDF, provided online at http://www.snaiso.com/Documentation/Arcsight/arcsight_correlation.pdf; visited on November 1, 2012.

[5] AXELSSON, S. The base-rate fallacy and the difficulty of intrusion detection. *ACM Transactions on Information and System Security 3*, 3 (Aug. 2000), 186–205.

[6] BAADER, F., HORROCKS, I., AND SATTLER, U. *Handbook of Knowledge Representation*. Elsevier, 2008.

[7] BATAGELJ, V., AND BREN, M. Comparing Resemblance Measures. *Journal of Classification 12*, 1 (1995), 73–90.

[8] BATISTA, G. E. A. P. A., PRATI, R. C., AND MONARD, M. C. A study of the behavior of several methods for balancing machine learning training data. *Sigkdd Explorations 6* (2004), 20–29.

[9] BERGER, A. L., PIETRA, S. A. D., AND PIETRA, V. J. D. A maximum entropy approach to natural language processing. *Computational Linguistics 22* (1996), 39–71.

[10] BRIDGES, S. M., AND VAUGHN, R. B. Intrusion detection via fuzzy data mining. In *Proceedings of the 12th Annual Canadian Information Technology Security Symposium* (2000).

[11] CADENAS, J. M., GARRIDO, M. C., AND HERNÁNDEZ, J. J. Heuristics to model the dependencies between features in fuzzy pattern matching. In *Proceedings of the Joint 4th Conference of the European Society for Fuzzy Logic and Technology* (2005).

[12] CHAWLA, N. V., JAPKOWICZ, N., AND KOTCZ, A. Editorial: special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter 6* (June 2004), 1–6.

[13] CHEN, S. F., AND ROSENFELD, R. A gaussian prior for smoothing maximum entropy models. Tech. Rep. CMV-CS-99-108, Carnegie Mellon University, 1999.

[14] CLARK & PARSIA, LLC. Pellet: Owl 2 reasoner for java. Website, available online at http://clarkparsia.com/pellet/; visited on December 12, 2011.

[15] DAS, N. G. *Statistical Methods - Combined Edition (Volumes I & II)*. Tata McGraw-Hill Publishing Company, 2009.

[16] EMC CORPORATION. Rsa envision platform function, attribute and feature topics. Website, available online at http://germany.rsa.com/node.aspx?id=3171; visited on December 8, 2011.

[17] FAN, W., MILLER, M., STOLFO, S. J., AND LEE, W. Using artificial anomalies to detect unknown and known network intrusions. In *Proceedings of the first IEEE International conference on Data Mining* (2001), IEEE Computer Society, pp. 123–130.

[18] FANIZZI, N., AND D'AMATO, C. A similarity measure for the ALN description logic. In *Proceedings of the Italian Conference on Computational Logic (CILC)* (2006), pp. 26–27.

[19] FORREST, S., PERELSON, A. S., ALLEN, L., AND CHERUKURI, R. Self-nonself discrimination in a computer. In *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy* (1994), IEEE Computer Society Press, pp. 202–212.

[20] GATES, C., AND TAYLOR, C. Challenging the anomaly detection paradigm: A provocative discussion. In *Proceedings of the 15th Workshop on New Security Paradigms* (2006), ACM Press, pp. 21–29.

[21] GETOOR, L., AND TASKAR, B., Eds. *Introduction to Statistical Relational Learning*. MIT Press, 2007.

[22] GONZALEZ, J. M., PAXSON, V., AND WEAVER, N. Shunting: a hardware/software architecture for flexible, high-performance network intrusion prevention. In *Proceedings of the 14th ACM conference on Computer and communications security* (New York, NY, USA, 2007), ACM, pp. 139–149.

[23] GU, G., CÁRDENAS, A. A., AND LEE, W. Principled reasoning and practical applications of alert fusion in intrusion detection systems. In *Proceedings of the 2008 ACM symposium on Information, computer and communications security* (New York, NY, USA, 2008), ASIACCS '08, ACM, pp. 136–147.

[24] GU, Q., CAI, Z., ZHU, L., AND HUANG, B. Data mining on imbalanced data sets. In *Proceedings of the 8th International Conference on Advanced Computer Theory and Engineering* (December 2008), IEEE, pp. 1020–1024.

[25] GUPTA, K. K., NATH, B., AND RAMAMOHANARAO, K. Conditional Random Fields for Intrusion Detection. In *Proceedings of 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW)* (2007), IEEE Press, pp. 203–208.

[26] GUPTA, K. K., NATH, B., AND RAMAMOHANARAO, K. Layered approach using conditional random fields for intrusion detection. *IEEE Transactions on Dependable and Secure Computing 7*, 1 (2010), 35 –49.

[27] HAMMERSLEY, J., AND CLIFFORD, P. Markov field on finite graphs and lattices. 1971.

[28] HE, Y., CHEN, W., YANG, M., AND PENG, W. Ontology based cooperative intrusion detection system. In *Network and Parallel Computing*, vol. 3222 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2004, pp. 419–426.

[29] HWANG, K., CHEN, Y., AND QIN, M. Hybrid intrusion detection with weighted signature generation over anomalous internet episodes. *Transactions on Dependable and Secure Computing 4*, 1 (2007), 41–55.

[30] ILGUN, K., KEMMERER, R. A., AND PORRAS, P. A. State transition analysis: A rule-based intrusion detection approach. *IEEE Transactions on Software Engineering 21* (1995), 181–199.

[31] JEH, G., AND WIDOM, J. SimRank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2002), KDD '02, ACM, pp. 538–543.

[32] JONNALAGADDA, S. K., AND MALLELA, S. S. An intelligent hybrid structure for improving intrusion detection. *International Journal of Research and Reviews in Software Engineering 1*, 2 (2011).

[33] KHREICH, W., GRANGER, E., MIRI, A., AND SABOURIN, R. Iterative boolean combination of classifiers in the roc space: An application to anomaly detection with hmms. *Pattern Recognition 43*, 8 (2010), 2732–2752.

[34] KRUEGEL, C., VALEUR, F., VIGNA, G., AND KEMMERER, R. Stateful intrusion detection for high-speed networks. In *Proceedings of the Symposium on Security and Privacy* (2002), IEEE Press, pp. 285–294.

[35] KRÜGEL, C., TOTH, T., AND KIRDA, E. Service specific anomaly detection for network intrusion detection. In *Proceedings of the 2002 ACM symposium on Applied computing* (New York, NY, USA, 2002), ACM, pp. 201–208.

[36] KUMAR, S., AND SPAFFORD, E. H. A software architecture to support misuse intrusion detection. In *Proceedings of the 18th National Information Security Conference* (1995), pp. 194–204.

[37] LAFFERTY, J., MCCALLUM, A., AND PEREIRA, F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning 2001* (2001), pp. 282–289.

[38] LASKOV, P., SCHÄFER, C., AND KOTENKO, I. Intrusion detection in unlabeled data with quarter-sphere support vector machines. In *Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment* (2004), pp. 71–82.

[39] LEE, W., AND STOLFO, S. J. Data mining approaches for intrusion detection. In *Proceedings of the 7th USENIX Security Symposium* (January 1998).

[40] LIPPMANN, R., FRIED, D., GRAF, I., HAINES, J., KENDALL, K., MCCLUNG, D., WEBER, D., WEBSTER, S., WYSCHOGROD, D., CUNNINGHAM, R., AND ZISSMAN, M. Evaluating intrusion detection systems: the 1998 darpa off-line intrusion detection evaluation. In *Proceedings of the DARPA Information Survivability Conference and Exposition* (2000), vol. 2, pp. 12–26.

[41] MCCALLUM, A., AND FREITAG, D. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the ICML 2000* (2000), Morgan Kaufmann, pp. 591–598.

[42] MCHUGH, J. Testing intrusion detection systems: A critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. *ACM Transactions on Information and System Security 3*, 4 (2000), 262–294.

[43] MILLER, D. R., HARRIS, S., HARPER, A., VANDYKE, S., AND BLASK, C. *Security Information and Event Management (SIEM) Implementation*. McGraw-Hill, 2011.

[44] NETWORK WORKING GROUP. The intrusion detection message exchange format (idmef). RFC, available online at `http://www.ietf.org/rfc/rfc4765.txt`; visited on July 13, 2011.

[45] NG, A. Y., AND JORDAN, M. I. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems* (2001), pp. 841–848.

[46] NICOLETT, M., AND KAVANAGH, K. M. Magic quadrant for security information and event management. Gartner Research document G00176034, 2010.

[47] NITROSECURITY, INC. Putting the "security" back into siem. Website, available online at `http://www.nitrosecurity.com/solutions/enterprise-security/`; visited on July 12, 2011.

[48] OURSTON, D., MATZNER, S., STUMP, W., AND HOPKINS, B. Applications of hidden markov models to detecting multi-stage network attacks. In *Proceedings of the 36th Annual Hawaii International Conference on System Sciences* (2003).

[49] PAXSON, V. Bro: A system for detecting network intruders in real-time. *Computer Networks: The International Journal of Computer and Telecommunications Networking 31* (December 1999), 2435–2463.

[50] PEARL, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[51] RAMPRASATH, R., AND GNANJEYARAMAN, R. Detection and classification of network intrusion using ilacr. *International Journal of Computing Technology and Information Security 1*, 1 (March 2011), 62–75.

[52] RIECK, K., AND LASKOV, P. Language models for detection of unknown attacks in network traffic. *Journal in Computer Virology 2* (2007), 243–256.

[53] ROSSOW, C., DIETRICH, C. J., BOS, H., CAVALLARO, L., VAN STEEN, M., FREILING, F. C., AND POHLMANN, N. Sandnet: Network traffic analysis of malicious software. In *Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS)* (2011), pp. 78–88.

[54] SMITH, A., AND OSBORNE, M. Regularisation techniques for conditional random fields: Parameterised versus parameter-free. In *International Joint Conference on Natural Language Processing* (2005), pp. 896–907.

[55] SOMMER, R., AND PAXSON, V. Enhancing byte-level network intrusion detection signatures with context. In *Proceedings of the 10th ACM conference on Computer and communications security* (New York, NY, USA, 2003), ACM, pp. 262–271.

[56] SYMANTEC CORPORATION. Symantec security information manager. PDF, available online at `http://www.symantec.com/de/de/business/security-information-manager`; visited on December 8, 2011.

[57] TAVALLAEE, M., BAGHERI, E., LU, W., AND GHORBANI, A. A. A detailed analysis of the kdd cup 99 data set. In *Proceedings of the Second IEEE international conference on computational intelligence for security and defense applications* (Piscataway, NJ, USA, 2009), IEEE Press, pp. 53–58.

[58] TAYLOR-POWELL, E. Sampling. Tech. rep., University of Wisconsin-Extension, Cooperative Extension, 1998.

[59] TSAI, C.-F., HSU, Y.-F., LIN, C.-Y., AND LIN, W.-Y. Intrusion detection by machine learning: A review. *Expert Systems with Applications 36*, 10 (2009), 11994 – 12000.

[60] VAIL, D. L., LAFFERTY, J. D., AND VELOSO, M. M. Feature selection in conditional random fields for activity recognition. In *Proceeding of the International Conference on Intelligent Robots and Systems* (2007), pp. 3379–3384.

[61] W3C RDF DATA ACCESS WORKING GROUP. Sparql query language for rdf. Website, available online at `http://www.w3.org/TR/rdf-sparql-query/`; visited on December 13, 2011.

[62] WAGNER, C. Breaking the knowledge acquisition bottleneck through conversational knowledge management. *Information Resources Management Journal 19* (January-March 2006), 70–83.

[63] WALLACH, H. M. Conditional random fields: An introduction. Tech. Rep. MS-CIS-04-21, Department of Computer and Information Science, University of Pennsylvania, 2004.

[64] YU, D., AND FRINCKE, D. Improving the quality of alerts and predicting intruder's next goal with hidden colored petri-net. *Computer Networks 51* (February 2007), 632–654.

[65] ZANERO, S., AND SAVARESI, S. M. Unsupervised learning techniques for an intrusion detection system. In *Proceedings of the ACM symposium on applied computing* (New York, NY, USA, 2004), ACM, pp. 412–419.