

Event Correlation Using Conditional Exponential Models with Tolerant Pattern Matching Applied to Incident Detection

A Thesis presented to the
Fachbereich Mathematik/Informatik
Universität Bremen

In Partial Fulfillment of the Requirements for the Degree
Dr.-Ing.

Dissertation

by
Carsten Elfers
born in Gronau

Submission: December, 2013

Colloquium: September, 2014

Advisor: Prof. Dr. Stefan Edelkamp

Co-Advisor: Prof. Dr. Norbert Pohlmann

This revised version includes minor corrections of the layout, a shortened appendix, minor clarifications and updates. This version comes without digital attachment.

Berichte aus der Informatik

Carsten Elfers

**Event Correlation Using Conditional
Exponential Models with Tolerant Pattern Matching
Applied to Incident Detection**

D 46 (Diss. Universität Bremen)

Shaker Verlag
Aachen 2014

Bibliographic information published by the Deutsche Nationalbibliothek

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at <http://dnb.d-nb.de>.

Zugl.: Bremen, Univ., Diss., 2014

Copyright Shaker Verlag 2014

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publishers.

Printed in Germany.

ISBN 978-3-8440-3168-3

ISSN 0945-0807

Shaker Verlag GmbH • P.O. BOX 101818 • D-52018 Aachen

Phone: 0049/2407/9596-0 • Telefax: 0049/2407/9596-9

Internet: www.shaker.de • e-mail: info@shaker.de

Thanks to...

- **Bettina Elfers**

my wife, for making square things round.

- **Hartmut Messerschmidt**

a colleague and friend, for his help in the analysis of the Conditional Exponential Model and his great — and funny — lectures about machine learning.

- **Heinz and Maria Elfers**

my parents, for always being there for me.

- **Jennifer Santa Lopes**

my sister-in-law, for spell-checking this thesis.

- **Karsten Sohr**

a colleague and friend, for his advises and his exhilarating and motivating work in the fides project.

- **Norbert Pohlmann**

my co-advisor, for his support.

- **Otthein Herzog**

for supporting me and my work.

- **Stefan Edelkamp**

my thesis advisor, for encouraging discussions, supporting me and my thesis and his help with the complexity analysis of the Pareto algorithm.

Contents

1	Introduction	1
1.1	Research Goals	2
1.2	Classification and Overview of this Approach	3
1.3	Research Results	5
2	Challenges in the Security Information and Event Management (SIEM) Domain	6
2.1	Intrusion Detection and SIEM Systems	6
2.2	SIEM Event Sources	12
2.2.1	Firewalls	12
2.2.2	System Monitoring	13
2.2.3	Antivirus Software and Appliances	14
2.2.4	Intrusion Detection Systems	15
2.3	SIEM Related Domain Knowledge	17
2.3.1	Events	17
2.3.2	Asset Information	18
2.3.3	Vulnerabilities	19
2.4	Open Problems of the SIEM domain	20
2.5	Requirements for Solving the Open Problems	25
2.6	Summary and Conclusion	26
3	Incident and Intrusion Detection Systems and Methods	29
3.1	Intrusion Detection Systems (IDS)	30
3.1.1	Spatial Coverage	30

3.1.2	Detection Methods	34
3.1.3	Summary and Discussion	44
3.2	Security Information and Event Management (SIEM)	49
3.2.1	Solutions	49
3.2.2	Conclusion	54
3.3	Summary and Conclusion	55
4	Knowledge Representations and their Usage for Incident Detection	57
4.1	Knowledge Representation	58
4.1.1	Ontologies and Description Logic	58
4.1.2	Fuzzy Logic	62
4.1.3	Summary and Discussion	66
4.2	Pattern Matching	68
4.2.1	Hard Pattern Matching	69
4.2.2	Soft Pattern Matching	71
4.2.3	Summary and Discussion	79
4.3	Summary and Conclusion	80
5	Inductive Learning with Probabilistic Models	82
5.1	Preliminaries and Notation	83
5.2	Bayesian Networks	85
5.2.1	Naïve Bayes	89
5.2.2	Dynamic Bayesian Network	90
5.2.3	Hidden Markov Model	90
5.3	Conditional Exponential Models	91
5.3.1	Maximum Entropy Markov Model	95
5.3.2	Conditional Random Field	97
5.3.3	Markov Logic Network	100
5.4	Summary and Conclusion	102

6	Conditional Random Fields with Tolerant Features for Intrusion Detection	105
6.1	Preliminaries	106
6.1.1	Observations	106
6.1.2	Ontological Representation and Preprocessing	106
6.2	Tolerant Pattern Matching	108
6.2.1	Patterns and Generalizations of Patterns	109
6.2.2	Measuring Abstraction	113
6.2.3	The Pareto Algorithm	117
6.2.4	Complexity Considerations	123
6.2.5	Summary and Discussion	128
6.3	Conditional Random Fields with Tolerant Features	129
6.3.1	Tolerant Pattern Matches as Feature Function Values	130
6.3.2	Monotonicity	131
6.3.3	Similarity Function	135
6.3.4	Two Layers of Conditional Random Fields	138
6.3.5	Modeling Incidents - The Incident Matrix	141
6.3.6	Prioritization of Incidents	144
6.3.7	Hypotheses Pool	145
6.3.8	Learning from Examples	149
6.3.9	Summary and Discussion	160
6.4	Implementation of the SIEM correlation process	161
6.4.1	Architecture Overview	161
6.4.2	Tolerant Pattern Matching	164
6.4.3	Conditional Random Fields	166
6.4.4	Web Front End Prototype	168
6.4.5	Summary and Discussion	172
6.5	Summary and Conclusion	172
7	Evaluation of the Proposed Incident Detection	174
7.1	Detection Performance	174
7.1.1	Data Sets	175

7.1.2	Performances Measures	177
7.1.3	Experiments with ArcSight	180
7.1.4	Simulation Experiment with Modeled Incidents	185
7.1.5	Simulation Experiments with Modeled and Trained Incidents	197
7.1.6	Summary and Conclusion	200
7.2	Use Cases	202
7.2.1	Use Case 1 - Reconnaissance Attempts	202
7.2.2	Use Case 2 - Different Sensors and Temporal Relations . . .	211
7.2.3	Use Case 3 - Vulnerabilities and Assets	216
7.2.4	Summary and Conclusion	218
7.3	Runtime Performance	219
7.3.1	Test Setup	219
7.3.2	Test Results	220
7.3.3	Summary and Discussion	225
7.4	Summary and Conclusion	225
8	Conclusion and Future Work	227
8.1	Retrospective	227
8.2	Conclusion	229
8.3	Future Work	231
A	Details and Additional Examples	260
A.1	Intrusion Detection Message Exchange Format (IDMEF)	261
A.2	CVE Examples	262
A.2.1	CVE-2011-2516	262
A.2.2	CVE-2012-2341	262
A.3	Description Logic Languages	263
B	Test Configurations	264
B.1	Configuration File of the ArcSight Connector	265
B.2	Categorization Mapping of the ArcSight SmartConnector	267
B.3	ArcSight Rules for the Comparison of this Work	269

Notation

In this work $\lg n$ refers to the dual logarithm $\log_2 n$, while $\ln n$ refers to the natural logarithm $\log_e n$. Vectors or sets are denoted by bold characters, e.g. $\mathbf{x}$. Elements are indexed by a subscript enumerator, e.g., x_j and are written in bold characters if they are vectors as well, for example, $\mathbf{x}_j$. A non-bold character without index, e.g., x , denotes an arbitrary element of $\mathbf{x}$. The power set is denoted by $\mathcal{P}$. The following table gives an overview of the most frequently used symbols in this work (sorted by Latin letters before Greek letters):

Symbol	Description
F	Fusion function
j, k, l, m, n	These symbols are reserved for indexing elements.
$\mathbf{e}$	Set of entities used in the constraints
$\mathbf{f}$	Set of feature functions
$\mathbf{h}$	Set of hypotheses
$\mathbf{i}$	Set of incidents
$\mathbf{p}$	Set of patterns
$\mathbf{q}$	Abstraction levels in the search space of the Pareto algorithm
$\mathbf{s}$	Set of samples
$\mathbf{t}$	Set of threat levels
$\mathbf{v}$	Set of pattern matching values
$\mathbf{x}$	Sequence of observations
$\mathbf{y}$	Sequence of labels
$\mathbf{R}$	Set of relations in the ontology
$\mathbf{S}$	Set of solutions of the Pareto algorithm
$\mathbf{Q}$	Search space of the Pareto algorithm
β	Penalty factor of the Fusion Function
θ	Similarity function
γ	Set of constraints
λ	Set of model parameters (weights)

Table 1: Table of symbols.