

Adaptive Anomalous Behavior Identification in Large-Scale Distributed Systems



THE UNIVERSITY
of ADELAIDE

Javier Álvarez Cid-Fuentes

School of Computer Science

The University of Adelaide

This dissertation is submitted for the degree of

Doctor of Philosophy

Supervisors: Dr. Claudia Szabo and Prof. Katrina Falkner

August 2017

© Copyright by

Javier Álvarez Cid-Fuentes

August 2017

All rights reserved.

No part of the publication may be reproduced in any form by print, photoprint, microfilm or any other means without written permission from the author.

A mi familia.

Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint award of this degree.

I give consent to this copy of my thesis when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968. The author acknowledges that copyright of published works contained within this thesis resides with the copyright holder(s) of those works.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

Javier Álvarez Cid-Fuentes

August 2017

Acknowledgements

First of all I would like to thank my two supervisors, Dr. Claudia Szabo and Prof. Katrina Falkner, for their support and guidance during these three years (and a bit). Thank you for your patience, help, insight, and above all, thank you for giving me the opportunity to work with you and for teaching me everything about research.

I would also like to thank the people who, besides my supervisors, helped me to carry out my studies in Adelaide. Francesc, Cruz, Rosa, and Wamberto, this thesis would not have been possible without your support.

I am also very thankful to the university staff and students for creating a really enjoyable atmosphere that makes things much easier. I am especially grateful to the people who, in one way or another, helped me during my studies. Thanks to Dan and Marianne for their immense technical support; to Yongrui and Ali for our fruitful talks; to Dídac for his brief but really insightful assistance; to Yuval for his help in the area of security; and to Lachlan and Jamal for the good times spent together. Finally, I would like to give special thanks to Pádraig and Zhigang for their academic and personal support. These years would not have been the same without you.

Por otro lado, también me gustaría darle las gracias a todas las personas que, a nivel personal, han estado a mi lado durante estos años. En primer lugar, muchísimas gracias a mis padres, que han trabajado durísimo toda su vida para darnos a mí y a mi hermana unas oportunidades que ellos no tuvieron. Sin vosotros no sería quien soy, ni estaría donde estoy. Gracias también a mi hermana, por estar siempre ahí, y a mis

tíos, Maribel y José Luís, por haber hecho esta experiencia más llevadera aun estando a miles de kilómetros de aquí.

Gracias a todos los amigos que están lejos pero están, gracias a Esteve por el soporte y por haberme traído un trocito de casa contigo en los días en los que estuviste aquí, y gracias a Juan por aquel fin de semana en Montreal que se pareció a los míticos jueves. Agradecérselo también a todas esas *aves de paso* que han hecho de mi estancia en Australia una experiencia inolvidable: Carles, Vicent, Zulia, Sandra, Nacho, Mehdi, Luís, Tania, Andrea, Carmen, Gorka, Rafa, Cristian, Jonathan, Alex y Jorge. A Alfonso y Gerard por los buenísimos ratos y las risas. Y en especial a Pablo y Cassia por su gran ayuda y las muchas tardes y noches que hemos compartido.

Por último, *last but not least*, dar las gracias a Ana, una de las personas más importantes de mi vida, por haber compartido conmigo este viaje que no ha sido fácil, por la cantidad y la calidad de los momentos, y por haber estado a mi lado incondicionalmente.

Abstract

Distributed systems have become pervasive in current society. From laptops and mobile phones, to servers and data centers, most computers communicate and coordinate their activities through some kind of network. Moreover, many economic and commercial activities of today's society rely on distributed systems. Examples range from widely used large-scale web services such as Google or Facebook, to enterprise networks and banking systems. However, as distributed systems become larger, more complex, and more pervasive, the probability of failures or malicious activities also increases, to the point that some system designers consider failures to be the norm rather than the exception.

The negative effects of failures in distributed systems range from economic losses, to sensitive information leaks. As an example, reports show that the the cost of downtime in industry ranges from \$100K to \$540K per hour on average. These undesired consequences can be avoided with better monitoring tools that can inform system administrators of the presence of anomalies in the system in a timely manner. However, key challenges remain, such as the difficulty in processing large amounts of information, the huge variety of anomalies that can appear, and the difficulty in characterizing these anomalies.

This thesis contributes a novel framework for the online detection and identification of anomalies in large-scale distributed systems that addresses these challenges. Our framework periodically collects system performance metrics, and builds a behavior characterization from these metrics in a way that maximizes the distance between nor-

mal and anomalous behaviors. Our framework then uses machine learning techniques to detect previously unseen anomalies, and to identify the type of known anomalies with high accuracy, while overcoming key limitations of existing works in the area. Our framework does not require historical data, can be employed in a *plug-and-play* manner, adapts to changes in the system behavior, and allows for a flexible deployment that can be tailored to numerous scenarios with different architectures and requirements.

In this thesis, we employ our framework in three anomaly detection application domains: distributed systems, large-scale systems, and malicious traffic detection. Extensive experimental studies in these three domains show that our framework is able to detect several types of anomalies with 0.80 *Recall* on average, and 0.68 mean *Precision* or 0.082 mean *FPR* depending on the domain. Moreover, our framework achieves over 0.80 accuracy in the identification of various types of complex anomalous behaviors. These results significantly improve similar works in the three explored research areas.

Most importantly, our approach achieves these detection and identification rates with significant advantages over existing works. Specifically, our framework does not rely on historical anomalous data or on assumptions on the characteristics of the anomalies that can make anomaly detection easier. Moreover, our framework provides a flexible and highly scalable design, and an adaptive method that can incorporate new system information at run time.

Table of contents

List of figures	xiv
List of tables	xvii
1 Introduction	1
1.1 Contributions	7
1.2 Thesis Organization	9
2 Literature Review	12
2.1 Anomaly Detection in Distributed Systems	13
2.1.1 Key Issues	17
2.2 Anomaly Detection in Large-Scale Systems	19
2.2.1 Key Issues	22
2.3 Malicious Traffic Detection	23
2.3.1 Key Issues	26
2.4 Analysis of Anomaly Detection Approaches	29
2.4.1 Definition of Anomalies	29
2.4.2 Normality Boundary	31
2.4.3 Scalability	31
2.5 Summary	34
3 A Framework for Behavior Identification without Historical Data	35

3.1	Overview	36
3.2	Characterizing System Behavior	37
3.3	Design Considerations	40
3.4	Behavior Identification Framework	42
3.4.1	Behavior Extractor	44
3.4.2	Behavior Identifier	46
3.4.3	Feedback Provider	58
3.4.4	Addressing Multiple Behavioral Perspectives	60
3.5	Summary	62
4	A Synthetic Distributed System to Model Complex Anomalous Behaviors	64
4.1	Overview	65
4.2	Synthetic Distributed System	67
4.2.1	System Design	67
4.2.2	Modeling Anomalous Behaviors	74
4.2.3	Testbed	75
4.3	Behavior Characterization	76
4.4	Parametric Study	82
4.4.1	Framework Deployment	84
4.4.2	Experimental Setup	84
4.4.3	Features	86
4.4.4	Initial Training	87
4.4.5	Alert Grouping	89
4.4.6	Window Size	90
4.4.7	Optimal Parameters	94
4.5	Summary	96
5	Feature Selection for Anomaly Detection and Identification	98
5.1	Detecting Change Point Anomalies	99

5.1.1	Results	101
5.2	Identifying Anomaly Types	102
5.2.1	Experimental Setup	102
5.2.2	Evaluation Metrics	103
5.2.3	Results	103
5.3	A Comprehensive Feature Selection Analysis	105
5.3.1	Experimental Setup	105
5.3.2	Evaluation Metrics	105
5.3.3	Features	106
5.3.4	Anomaly Detection	110
5.3.5	Anomaly Identification	114
5.4	Discussion	116
5.5	Summary	117
6	Decentralized Anomaly Detection in Large-Scale Systems	119
6.1	Overview	120
6.2	The Curse of Dimensionality	121
6.3	A Decentralized Deployment	122
6.4	Data Generation	124
6.5	Experimental Analysis	126
6.5.1	Experimental Setup	127
6.5.2	Results	128
6.5.3	Discussion	130
6.6	Summary	133
7	An Online Approach for the Detection of Novel Botnets	134
7.1	Overview	135
7.2	Botnet Characteristics	137
7.2.1	Bot Life Cycle	138

7.2.2	Command & Control Communication	139
7.2.3	Malicious Activities	140
7.3	Botnet Detection	141
7.3.1	Communication using IRC	142
7.3.2	DNS Lookups	142
7.3.3	P2P	143
7.3.4	SMTP	143
7.3.5	Group Activities	143
7.3.6	Transport Layer	144
7.4	Detection of Novel Botnets	144
7.4.1	Behavior Representation	147
7.5	Experimental Analysis	148
7.5.1	Dataset	148
7.5.2	Experimental Setup	149
7.5.3	Evaluation Metrics	150
7.5.4	Results	151
7.5.5	Comparison with Existing Work (Beigi et al. [1])	154
7.5.6	Discussion	154
7.6	Summary	156
8	Conclusions and Future Work	157
8.1	Directions for Future Work	160
	References	163

List of figures

1.1	The relationship between faults, errors and failures	3
3.1	Different perspectives that can be employed when modeling the behavior of distributed systems	39
3.2	Framework's design	43
3.3	Feature extraction process	46
3.4	2-dimensional support vector machine	47
3.5	Multi-class classifier as a DAG	56
3.6	Two-step classification process	57
3.7	<i>Feedback Provider's</i> internal behavior	58
3.8	Default setting used in global distributed system behavioral analysis .	61
3.9	Setting employed to analyze distributed systems from a system of systems perspective	62
3.10	Shared BM setting employed to analyze distributed systems from a replicated behavior perspective	63
4.1	Interfaces of the components in our synthetic distributed system . . .	68
4.2	Process modeling Clients' actions in the SDS	70
4.3	Process modeling Clients' voting in the SDS	70
4.4	Process modeling Clients' access to a Database in the SDS	71
4.5	Process modeling Clients' communication management in the SDS . .	72

4.6	Process modeling Databases' behavior in the SDS	73
4.7	Process modeling Clients' transfers in the SDS	73
4.8	Synthetic distributed system metrics under normal behavior	77
4.9	Hardware metrics collected from one of the nodes in a time span of two hundred seconds with the SDS running in deadlock mode.	78
4.10	Hardware metrics collected from one of the nodes in a time span of two hundred seconds with the SDS running in livelock mode.	78
4.11	Hardware metrics collected from one of the nodes in a time span of two hundred seconds with the SDS running in synchronization mode.	79
4.12	Network input of the five nodes running the SDS during an unwanted synchronization.	79
4.13	Hardware metrics collected from one of the nodes in a time span of two hundred seconds during a memory leak.	80
4.14	Messages sent per Client right after the SDS enters starvation mode.	80
4.15	Setting used in global distributed system behavioral analysis	84
4.16	Mean <i>Recall</i> and <i>Precision</i> obtained using different <i>Tr</i> values. Anomaly duration is 10 minutes, $z = 128$, and $Gr = 1$	88
4.17	Mean <i>Recall</i> , <i>Precision</i> and DT obtained using different <i>Gr</i> values. Anomaly duration is 10 minutes, $z = 128$, and $Tr = 200$	89
4.18	Mean F-score for different combinations of window size and anomaly duration ($Tr = 200$ and $Gr = 1$)	91
4.19	Mean detection time in seconds for different combinations of window size and anomaly duration ($Tr = 200$ and $Gr = 1$)	92
4.20	Mean F-score for different combinations of window size and anomaly duration ($Tr = 600$ and $Gr = 20$)	94
5.1	Change point anomalies	100

5.2	Mean <i>Recall</i> and <i>Precision</i> obtained in the Yahoo! dataset for a varying number of anomalous time series ($Gr = 10$ and $Tr = 50$) . . .	101
5.3	Mean <i>Recall</i> and <i>Precision</i> obtained with the different individual features	111
5.4	Mean $Recall_t$ and $Precision_t$ obtained with the different individual features	115
6.1	Setting employed to analyze distributed systems from a system of systems perspective	123
6.2	Network output and a time series replica for a period of 200 seconds .	125
6.3	Mean <i>Recall</i> for different anomalous profiles	129
6.4	Mean <i>Precision</i> for different anomalous profiles	130
6.5	Mean <i>Recall</i> when requiring three consecutive alerts to notify the system administrator	131
6.6	Mean <i>Precision</i> when requiring three consecutive alerts to notify the system administrator	132
7.1	Typical botnet architecture	138
7.2	Bot life cycle	139
7.3	Setting employed to analyze distributed systems from a replicated behavior perspective	146
7.4	<i>TPR</i> and <i>FPR</i> for different window sizes (with $Gr = 1$ and $Tr = 10\%$ of the training set)	151
7.5	<i>TPR</i> and <i>FPR</i> for different Gr values (with $z = 64$ and $Tr = 10\%$ of the training set)	152
7.6	Mean <i>TPR</i> and <i>FPR</i> for different train sizes	153
7.7	ROC curve obtained for different framework settings	153

List of tables

2.1	Comparison of anomaly detection methods in the area of distributed systems.	19
2.2	Comparison of anomaly detection methods for large-scale systems. . .	22
2.3	Comparison of anomaly detection methods in the area of botnet detection.	28
2.4	Comparison of anomaly detection methods in the domains covered by this thesis	33
4.1	Summary of the different parameters used to tune our framework. . .	83
4.2	Optimal parameter values.	95
4.3	Results obtained when using the best overall parameters ($Tr = 100$, $Gr = 50$, and $z = 512$).	96
5.1	Confusion matrix using a window of 64 seconds.	103
5.2	Confusion matrix using a window of 128 seconds.	104
5.3	Confusion matrix using a window of 256 seconds.	104
5.4	Mean detection F-score of the set of features $\{CO, ME, B0, S1\}$ versus ME	113
5.5	Average <i>Recall</i> , <i>Precision</i> and DT using the set of features $\{CO, ME, B0, S1\}$ in the SDS-Dataset-2 ($Tr = 400$, $Gr = 20$, $z = 128$).	113
5.6	Mean identification F-score of the set of features $\{LH, IN\}$ versus $\{LH, IN, EP\}$ and LH	116

5.7	Confusion matrix obtained using the feature set $\{LH, IN\}$ and a window of 128 seconds.	116
6.1	Different profiles evaluated.	126
6.2	Summary of the results obtained for the different profiles.	132
7.1	Results comparison with Beigi et al.	154