Data Security Issues in MaaS-enabling Platforms

Franco Callegati, Saverio Giallorenzo, Andrea Melis, Marco Prandini

Data Security Issues in MaaS-enabling Platforms

Franco Callegati*, Saverio Giallorenzo†, Andrea Melis* and Marco Prandini†
*Department of Electrical and Information Engineering, Universitá di Bologna, Italy
†Department of Computer Science and Engineering, Universitá di Bologna, Italy
E-mail: {franco.callegati,a.melis,marco.prandini,saverio.giallorenzo2}@unibo.it

Abstract—Mobility as a Service takes the concept of XaaS to transportation: a MaaS provider shall merge transport options from different mobility providers, seamlessly handling the whole experience of traveling, from providing information, to travel planning, and payments handling. To effectively support the creation of a market of MaaS providers, we envision the creation of ICT infrastructures based on microservices, a modern and renowned development model that fosters the creation of an ecosystem of reusable components. The flexibility of such platforms is their key advantage, yet it poses many security issues. In this paper, we look at these problems through the lens of our experience on one of such platforms, called SMAll. We classify the most relevant vulnerabilities related to data reliability, integrity, and authenticity, and we investigate directions for their mitigation.

Index Terms—MaaS, Microservices, Integrity, Authentication, Provenance, Reputation

1 Introduction

The concept of Mobility as a Service (MaaS) was born in Finland and it is rapidly spreading worldwide [1]. Key-point of a MaaS provider is that it shall offer a unique and seamless interface to its users, aggregating heterogeneous transport options offered by different mobility providers (e.g., different agencies providing transportation by taxi, bus, train, plane, car-sharing, etc.), handling the whole experience of traveling, from providing information, to travel planning, and payments. To effectively support the creation of a market of MaaS providers, we envision the creation of ICT infrastructures based on microservices. This modern and renowned development model [2] fosters the creation of an ecosystem of reusable components. In the context of MaaS, microservices shall efficiently and flexibly combine heterogeneous data sources, such as available transport options, real-time data regarding vehicles and infrastructures, pricing, etc., to provide customized travel planning, information and ticketing to final users, as well as monitoring and strategic planning tools to policy-makers.

We are currently developing one of such infrastructures as a marketplace for mobility services, called *Smart Mobility for All* (SMAll). In our vision, SMAll is the enabling technology to solve the challenges of the MaaS market, from developing user-contributed, crowd-sourced applications, to launching a MaaS operator, to planning effective and sustainable transport policies for smart cities.

However, we recognize that such a promising platform as SMAll has a lot of security issues derived from its openness regarding the usage, deployment, and (above all) reliability of its services.

2 The SMAll architecture

The SMAll architecture revolves around the concept of service. SMAll is not simply a collection of services and it is rather an enabler for their deployment. Indeed, we already classified some macro-categories of services that we can expect to find in SMAll. Figure 1 outlines some of the most important ones, arranged in layers of increasing complexity — in this context, "complex" means the creation of functionalities on top of other "simple" services. Starting from the bottom, we find services that are either wrappers for legacy software, e.g., travel planners that do not include real-time functionalities, or services that provide basic data. The aim of this class of services is to standardize the data and the interfaces of legacy software to make them available to other services. Other, more complex services, found in the upper layers, orchestrate these basic ones to implement their behaviors, up to the very refined policies of MaaS operators.

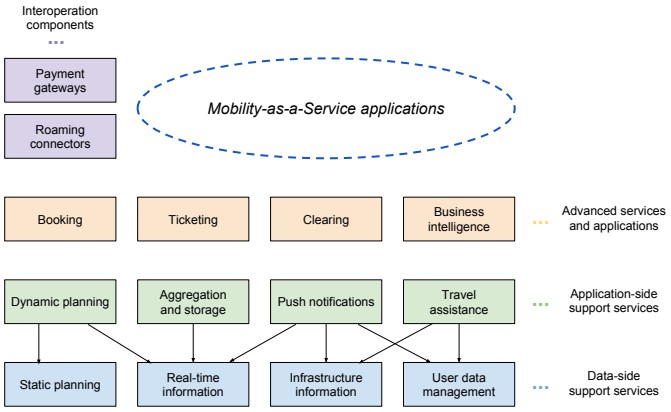

Fig. 1. Service categories in SMAll

There are already a plethora of mobility-related data sources and services. To fulfill our goal of seamless com-

position, SMAll must address the issues of standardization of data formats and of service invocation interfaces, foresee the implementation of infrastructural components which are likely to be needed by most services, and provide an orchestration framework to streamline the composition of available services into more complex applications. To provide all the needed components for MaaS operators, the SMAll platform (Figure 2) embraces the concept of microservices for offering:

- wrappers converting legacy data source into SMAll-Compliant Services (SCS);
- helper services (e.g., authentication, authorization, scheduling, routing, orchestration);
- The service registry storing the definition of all the services deployed on the platform;
- business intelligence providing auditing and KPI metering;
- the actual business logic deployed by operators or intermediaries, collecting, storing, and processing data to offer some data-related insight on the usage of services.

According to this model, there is no single actor responsible for data quality and service correctness, which poses serious security issues. However, the platform itself can play a crucial role in preventing abuses and in monitoring the correctness of transactions.

3 Security Issues

The SMAll platform is an example of cloud architecture, spanning the standard IaaS, PaaS, and SaaS layers. In particular, SMAll comprises the SaaS and PaaS layers respectively because *a*) it provides services on and for mobility and *b*) it supports the deployment of said services.

Indeed, the security issues known for these layers also apply to SMAll [3]. Hereinafter, we outline the specific problems that are most relevant for the context of mobility, followed by discussion of how the adoption of an integrated platform like SMAll can mitigate them, wherever deemed possible.

3.1 General cloud security considerations

Literature started to address the main security issues related to IaaS, PaaS, and SaaS since 2008, with the release of OpenNebula, the first true implementation of a Cloud service. According to [4] an essential list of most important issues encompasses:

- network security (spoofing, sniffing, DoS);
- data security (locality, integrity, segregation, authenticity, confidentiality, privacy, access control);
- authentication, identity management, sign-on process, and authorization;
- web application security;
- virtualization vulnerability;
- availability (high availability, disaster recovery).

SMAll enables the deployment of mobility services in the cloud. It also provides a set of helper services, orchestrated by a *dispatcher*, that take care of routing the invocation of services and the return of the results. Among the functionalites of helper services are countermeasures against some of the aforementioned security issues. *Authentication* and *user profile management* helpers act as a single, trusted interface towards identity providers, for single sign-on, and personal data vaults [5]. The *authorization* helper uniformly applies access control policies, which can be configured and managed as attributes of the deployed services (for example, following a MAC model similar to Flask/SELinux's Type Enforcement [6]). The invocation of a service through the SMAll *service caller* can control the data stream encryption (taking care of key management issues) to ensure its confidentiality. The *scheduler* assigns priorities to queries; during normal execution, this scheduler has the goal of making SMAll meet SLAs granted to different services and customers. In case of DoS attacks the scheduler can mitigate the saturation of resources.

3.2 Emerging threats

The PaaS layer in SMAll differs from most PaaS solutions because of its openness and flexibility. SMAll customers can access available data and services to build and deploy their own services, possibly making them available to themselves or other customers for the same purpose. A simple example to clarify this process: a one-stop ticketing application orchestrates: *a*) a dynamic planner service that provides the routing options; *b*) a user profile manager to sort them according to preferences; *c*) a real-time availability checking/seat reservation service for each operator; *d*) a set of gateway services for payment.

Every one of these services is available (and useful) as a standalone application. Moreover, the dynamic planner is not a simple service: it is the result of orchestrating a static planner with real-time information about delays, planned extraordinary events, and disruptions. The branches of this tree can be followed through many levels, until raw data is reached (yielded in a standard form by a wrapper service).

As the enabler, SMAll shall assume responsibility for the trustworthiness and reliability of the services; this is unusual for "classic" PaaS [7]. While it is unrealistic to expect that SMAll guarantees complete correctness of data sources and deployed services, especially under the assumption of being substantially open to any customer, providing at least a measure of their security is an important (and value-adding) service. In particular, it is necessary to define indicators for data quality and service behavior, and to devise a way to compute their values for complex data sources and services resulting from the aggregation and orchestration of existing ones [8], [9]. As it is made clear in the following discussion, these functions are an important component of a defense strategy against insider threats, which are likely and dangerous in the studied architecture.

3.3 Data Provenance

One of the most studied issues about data sources security is data provenance [10], [11]. Ascertaining provenance means ensuring that the source of data is verifiable, i.e., that it corresponds to the one declared in the process of creation. In a MaaS scenario, provenance protection is a defense against malicious operators claiming to expose data of a competitor, forging them to gain unfair advantage.

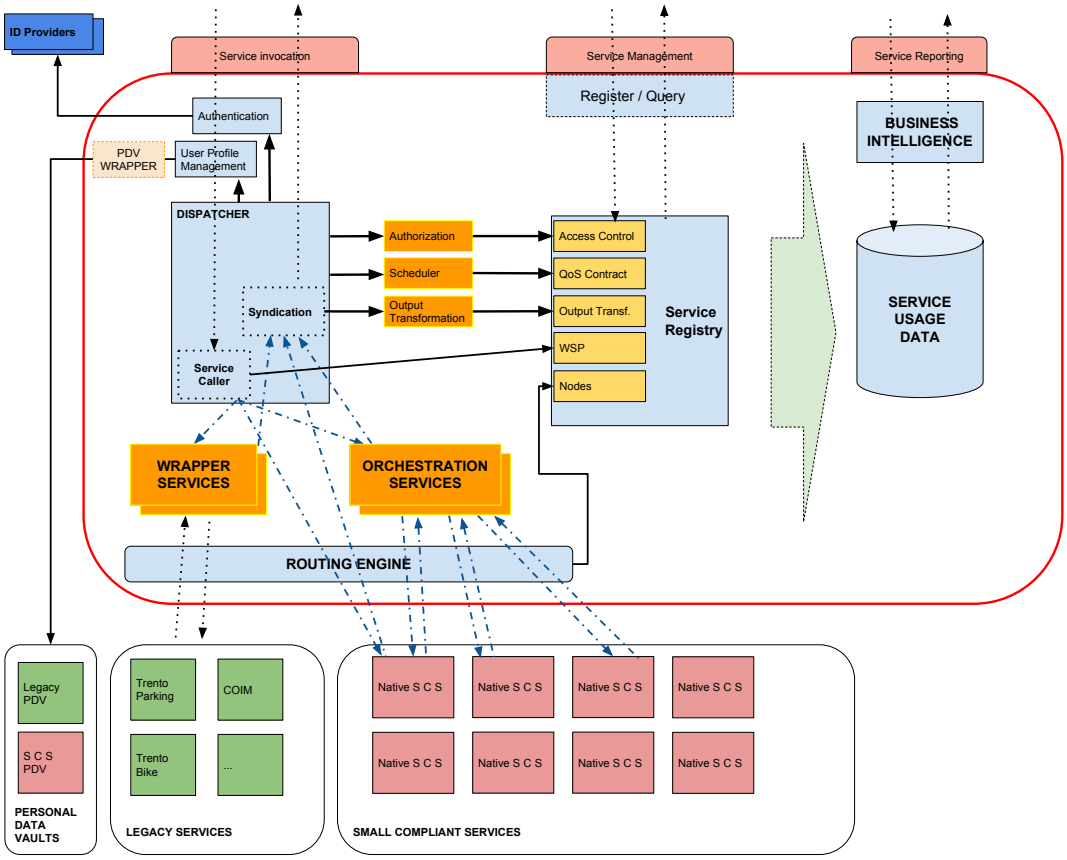

Fig. 2. SMAll platform architecture

3.4 Data Trustworthiness

Data trustworthiness, intended as the possibility to ascertain the correctness of the information provided by a data source, is loosely related to provenance [12]. Ideally, but infrequently, data samples can be independently measured by different users, thus allowing cross-checking and error correction. For original data, i.e., provided by its creator, the trustworthiness score is usually derived from the reputation of the creator. In this case, it is very difficult to block attacks in which, for example, the creator advertises a data source of given quality, but then exposes a degraded version, to keep the advantage of more precise/timely information for itself.

3.5 Service Maliciousness

The trustworthiness of a service is an easy concept to intuitively grasp, but difficult to formalize and to verify in an open environment. In the context of SMAll service trustworthiness can be associated with its compliance to a declared function, and shall be evaluated before the application is admitted to the platform and, ideally, again at every usage. If a service creates aggregated data, processing various sources, it is necessary to ensure that the computation is correct, that no useful results are hidden (completeness), and input data is not tampered with (soundness). These are all likely opportunities for a malicious insider that succeeds in registering a rogue service. For example, a tampered travel planner could slightly deflect routes to favor or damage certain businesses; a modified delay-checking application could hide or amplify violations of agreed service levels.

3.6 Service vulnerability to external attacks

There are applications that exhibit wrong behaviors not because of their maliciousness, but because of unexpected vulnerabilities to maliciously crafted invocations. In this case, there is an external threat in addition to the twofold insider threat: one in the loose meaning of an insider being so careless as to deploy a vulnerable application, the other in the case that another insider is in the best position to exploit it. For example, a service with extensive access privileges to private data could be tricked to leak it to a service with much more restrictive access rights. Another example, mixing service vulnerability with data provenance issues, is that of a service that provides crowd-sourced information about the status of the road transport network. Failure to implement an effective fraud-prevention mechanism could allow an attacker to inject fake reports to influence the behavior of users.

4 Directions in Insider Threat Mitigation

Many of the described problems are intrinsic to the concepts of cloud, SaaS, and data sharing. For this reason, before

being able to develop a solution for data quality and data provenance, there is a need for a preliminary analysis of all the metrics of these types of solutions.

4.1 Data Provenance

The first thing that we must consider when dealing with services that expose or elaborate data is to differentiate between data and information.

According to [13], information systems provide data in a certain business context. When data is used by human beings, it turns into information, and information finally turns into knowledge by being interpreted and linked for a given purpose.

As already mentioned, in the context of mobility, verified information is of paramount importance. SMAll supports the provision of different sources of data along with their associated metadata (e.g., used to verify their provenance). However, SMAll shall also provide techniques, embodied by helper services, to transform those data into verified information.

Data Provenance verification is a known problem in literature. Different approaches can be taken to support a solution for the problem of recognizing the source of a data stream. Literature agrees [14] that the requirements for a provenance management system are:

- Verifiability: a provenance system should be able to verify a process in terms of the actors (or services) involved, their actions, and their relationship with one another.
- Accountability: an actor (or service) should be accountable for its actions in a process. Thus, a provenance system should record in a non-repudiable manner any provenance generated by a service.
- Reproducibility: a provenance system should be able to repeat a process and possibly reproduce a process from the provenance stored.
- Preservation: a provenance system should have the ability to maintain provenance information for an extended period of time. This is essential for applications run in an enterprise system.
- Scalability: given the large amounts of data that an enterprise system handles, a provenance system needs to be scalable.
- Generality: a provenance system should be able to record provenance from a variety of applications.
- Customizability: a provenance system should allow users to customize it by setting metadata such as time, events of recording, and the granularity of provenance.

In a microservice architecture, an application is essentially a collection of workflows. These workflows can compose many levels of services, each processing and modifying the data before its final destination. What we need is a way to certify the metadata related to a data stream and manage its validity during time and re-elaboration [15]. Generally literature considers 4 different sets of techniques [16]:

- Subject of Provenance, in which provenance can either be available explicitly or be deduced indirectly.
- Representation of Provenance, in which the way provenance is represented follows from a tradeoff between the cost of recording it and its richness; it is typically implemented either with annotations or with inversion [17].
- Provenance Storage in which the design of metadata is also important to enable scalable storage.
- Provenance Dissemination where, in order to use provenance, a system should allow rich and diverse means to access it.

According to works like [18], this problem could be solved only with a creation of private and public key system for data stream certification. A good reference is the system developed in [19], describing a cryptographic provenance verification approach for ensuring data properties and integrity for single hosts. Specifically, the authors designed and implemented an efficient cryptographic protocol that enforces keystroke integrity. This kind of protocol can be integrated as a helper service in SMAll. However, public-key schemes are known for their significant computational load, thus existing techniques may not be suitable for high-rate, high-volume data sources. Moreover, there could be the need for an algorithm for the provenance of composed data.

In some cases, data originated from the composition of raw (or otherwise "lower ranked") sources should be accompanied by suitable metadata that allows to verify the provenance of the input values, in a cryptographically strong way. In the context of SMAll, it could be important and useful to capture and understand the propagation of data.

In [20], the authors exploit the propagation of the metadata on the various levels to create a general metadata storage and management layer for parallel file systems, in which metadata includes both file operations and provenance metadata.

Also Merkle hash trees could be a good candidate to build proofs for composed data pieces [21]. The combination of metadata propagation with key distribution propagation management can guarantee a good level of trust in provenance management systems. Works similar to [22] discuss how to support provenance awareness in spatial data infrastructure and investigates key issues including provenance modeling, capturing, and sharing that can be easily used to implement a key propagation system.

4.2 Data Trustworthiness

Rating systems are one possible implementation for trustworthiness evaluation mechanisms. Auditing services, integrated in the SMAll platform, attribute scores to the data-source services based on different criteria. For example, feedback from users of the data source, or a combination of reputation scores when the same data can be fed by many sources and cross-checked. Of course, reputation systems in turn introduce their fair share of problems [23], [24], [25], [26]. An alternative or complementary solutions are anomaly detection services based on machine learning and pattern recognition [27]. Then, as cited for provenance, SMAll should automatically compute the trustworthiness

score of data originated from the composition of other sources, based on their scores.

4.3 Service Maliciousness

In principle, the SMAll service deployment interface can verify the correctness of an application before accepting it. In practice, this operation is very hard to perform. One indicator of correctness is the compliance to a template of acceptable interfaces for the kind of service the application provides. However, it is very difficult to define templates strict enough to allow sensible compliance checks, but general enough to avoid hindering the deployment of legitimate services. Another way to check correctness is to look at the actual behavior of the application, as it is common in anti-malware checks, through static analysis, verifying the code to discover possible malicious behaviors. These techniques are far from infallible, and their scope falls much shorter than what is required in our context. Indeed, in this context a malicious behavior can be a subtle deviation from the correct calculation [28], which is far more difficult than the detection of traditional malicious behaviors (e.g., damaging or self-replicating ones).

A more promising technique is that of dynamic analysis of malware [29]. A way to discover a malicious behavior of a service is to implement a mechanism that could guarantee, in every moment, a reproducibility of the results. Taking advantage of the data provenance certifications of raw data, and of its propagation to results, it is possible to implement a reference monitor to verify the compliance of results to the expected values. In case of conflicts between the declared results and the actual ones, SMAll could discover what has been tampered with: the source data, or theservice logic. In the first case, this detection can also feed the data trustworthiness rating system.

4.4 Service vulnerability to external attacks

SMAll could provide input sanitization, or in general, Intrusion Prevention System / Intrusion Detection System as a service, to protect applications from most of the malicious invocations. Tracing cross-callings between SMAll services can thwart insider attacks aimed at privilege escalation. This can be done by taking into account the whole set of access control rules before allowing unauthorized data to leak in or out through a careless application.

5 Conclusion

In this paper, we have discussed security issues in MaaS-enabling platforms, one of the emerging scenarios made possible by developments in the field of microservices. The SMAll platform allows to conduct a specific case study. SMAll aims to be open to distributed service deployment from any customer, potentially giving way to insider threat scenarios, but at the same time it represents an opportunity to provide centralized security functions. From its architecture, general cloud-related, as well as specific security issues arise. This paper gives an overview of the most sensible approaches to mitigation of the illustrated threats.

References

[1] S. Pippuri, S. Hietanen, and K. Pyyhti, "Maas finland." http://maas.fi/.

[2] S. Newman, *Building Microservices*. " O'Reilly Media, Inc.", 2015.

[3] D. Chen and H. Zhao, "Data security and privacy protection issues in cloud computing," in *ICCSEE*, vol. 1, pp. 647–651, March 2012.

[4] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1 – 11, 2011.

[5] K. K. Antti Poikola and H. Honko, "Mydata a nordic model for human-centered personal data management and processing," tech. rep., Ministry of Transport Finland, 2010.

[6] RedHat, "Red Hat SELinux Guide." https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/4/html/SELinux_Guide/.

[7] S. E. Madnick, R. Y. Wang, Y. W. Lee, and H. Zhu, "Overview and framework for data and information quality research," *J. Data and Information Quality*, vol. 1, pp. 2:1–2:22, June 2009.

[8] C. Falge, B. Otto, and H. sterle, "Data quality requirements of collaborative business processes," in *HICSS*, pp. 4316–4325, Jan 2012.

[9] S. Dustdar, R. Pichler, V. Savenkov, and H.-L. Truong, "Quality-aware service-oriented data integration: Requirements, state of the art and open challenges," *SIGMOD Rec.*, vol. 41, pp. 11–19, Apr. 2012.

[10] Y. L. Simmhan, B. Plale, and D. Gannon, "A survey of data provenance in e-science," *SIGMOD Rec.*, vol. 34, pp. 31–36, Sept. 2005.

[11] P. Buneman, S. Khanna, and W.-C. Tan, *FST TCS 2000*, ch. Data Provenance: Some Basic Issues, pp. 87–93. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000.

[12] C. Dai, D. Lin, E. Bertino, and M. Kantarcioglu, *SDM*, ch. An Approach to Evaluate Data Trustworthiness Based on Data Provenance, pp. 82–98. Berlin, Heidelberg: Springer, 2008.

[13] J. Cordeiro, Y. Manolopoulos, J. Filipe, and P. Constantopoulos, *ICEIS*, vol. 3. Springer Science & Business Media, 2008.

[14] P. Groth, M. Luck, and L. Moreau, "A protocol for recording provenance in service-oriented grids," in *Principles of Distributed Systems*, pp. 124–139, Springer, 2004.

[15] W.-T. Tsai, X. Wei, Y. Chen, R. Paul, J.-Y. Chung, and D. Zhang, "Data provenance in soa: security, reliability, and integrity," *Service Oriented Computing and Applications*, vol. 1, no. 4, pp. 223–247, 2007.

[16] Y. L. Simmhan, B. Plale, and D. Gannon, "A survey of data provenance in e-science," *ACM Sigmod Record*, vol. 34, no. 3, pp. 31–36, 2005.

[17] D. Bhagwat, L. Chiticariu, W.-C. Tan, and G. Vijayvargiya, "An annotation management system for relational databases," *The VLDB Journal*, vol. 14, no. 4, pp. 373–396, 2005.

[18] Y. L. Simmhan, B. Plale, and D. Gannon, "A survey of data provenance techniques," *Computer Science Department, Indiana University, Bloomington IN*, vol. 47405, 2005.

[19] K. Xu, H. Xiong, C. Wu, D. Stefan, and D. Yao, "Data-provenance verification for secure hosts," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, pp. 173–183, March 2012.

[20] D. Zhao, C. Shou, T. Malik, and I. Raicu, "Distributed data provenance for large-scale data-intensive computing," in *CLUSTER*, pp. 1–8, IEEE, 2013.

[21] R. C. Merkle, *CRYPTO*, ch. A Digital Signature Based on a Conventional Encryption Function, pp. 369–378. Berlin, Heidelberg: Springer Berlin Heidelberg, 1988.

[22] L. He, P. Yue, L. Di, M. Zhang, and L. Hu, "Adding geospatial data provenance into sdia service-oriented approach," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, vol. 8, no. 2, pp. 926–936, 2015.

[23] A. Jsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618 – 644, 2007.

[24] R. A. Malaga, "Web-based reputation management systems: Problems and suggested solutions," *ECR*, vol. 1, no. 4, pp. 403–417.

[25] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation systems," *Commun. ACM*, vol. 43, pp. 45–48, Dec. 2000.

[26] H.-S. Lim, Y.-S. Moon, and E. Bertino, "Provenance-based trustworthiness assessment in sensor networks," in *IWDMSN*, pp. 2–7, ACM, 2010.

[27] A. DeOrio, Q. Li, M. Burgess, and V. Bertacco, "Machine learning-based anomaly detection for post-silicon bug diagnosis," in *Proceedings of the Conference on Design, Automation and Test in Europe*, DATE '13, (San Jose, CA, USA), pp. 491–496, EDA Consortium, 2013.

[28] A. Moser, C. Kruegel, and E. Kirda, "Limits of static analysis for malware detection," in *ACSAC*, pp. 421–430, Dec 2007.

[29] M. D. Ernst, "Static and dynamic analysis: Synergy and duality," in *WODA 2003: ICSE Workshop on Dynamic Analysis*, pp. 24–27, Citeseer, 2003.