H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 257

Introducing DINGfest: An architecture for next generation SIEM systems

Florian Menges[1] Fabian Böhm[1] Manfred Vielberth[1] Alexander Puchta[2] Benjamin Taubmann[3] Noëlle Rakotondravony[3] Tobias Latzo[4]

Abstract: Isolated and easily protectable IT systems have developed into fragile and complex structures over the past years. These systems host manifold, flexible and highly connected applications, mainly in virtual environments. To ensure protection of those infrastructures, Security Incident and Event Management (SIEM) systems have been deployed. Such systems, however, suffer from many shortcomings such as lack of mechanisms for forensic readiness. In this extended abstract, we identify these shortcomings and propose an architecture which addresses them. It is developed within the DINGfest project for which we seek feedback from the community.

Keywords: Forensics; Virtual Machine Introspection; Visual Analytics; Security Incident and Event Management; Identity and Access Management

1 Motivation and Problem Statement

During the last few years, IT infrastructures have evolved to heterogeneous and complex structures which are becoming increasingly harder to control and protect. To reduce the complexity of securing those systems, companies have integrated centralized and holistic protection and incident management measures like Security Information and Event Management (SIEM) systems [Sh16]. SIEM systems mainly cover log collection, normalization, analyses, storage and monitoring [Mi11]. However, today's SIEM systems come with a lot of deficiencies which we address in the DINGfest (DetektIon, VisualisieruNG, ForEnsische Aufbereitung von SicherheiTsvorfällen) project for which we seek feedback from the community.

First, SIEM systems are quite expensive and proprietary. In DINGfest, we plan to show that it is possible to build an effective SIEM system based on open source software. This lowers the barriers for practical usage, especially for small and medium-sized businesses. Second, common SIEM systems are not prepared for forensic analysis, which is required if a legal

[1] University of Regensburg, Chair of Information Systems, surname.name@ur.de
[2] Nexis GmbH, alexander.puchta@nexis-secure.com
[3] University of Passau, Assistant Professorship of Security in Information Systems, (bt|nr)@sec.uni-passau.de
[4] Friedrich-Alexander University Erlangen-Nürnberg, Department of Computer Science, Security Research Group tobias.latzo@fau.de

doi:10.18420/sicherheit2018_21

dispute is expected. To improve the trustworthy collection of evidence, data is acquired via Virtual Machine Introspection (VMI) and the integrity of extracted data is protected to support a clean chain of custody. Third, DINGfest has built-in common forensic tools for an extensive and fast forensic investigation and will be able to normalize gathered evidences and report incidents to third parties, e.g., the German Federal Office for Information Security. In addition to common SIEM systems, we use further data sources for analysis, e.g., from an Identity and Access Management (IAM) system. Furthermore, users spend an extensive amount of time to configure detection heuristics of SIEM systems. In DINGfest we provide Visual Security Analytics that aim to combine automated analysis with domain knowledge of security experts. In the following we introduce the more general DINGfest architecture to overcome these shortcomings.

2 Conceptual Architecture

The DINGfest architecture consists of three main components (see Fig. 1). **Data Acquisition** collects data in real time from various sources. **Data Stream** represents the central data hub used by the subsequent component to query data, which is implemented using Apache Kafka. **Data Analysis** leverages event processing and identity behavior analytics (IBA) for automated incident detection. Additionally visual security analytics for integrating expert knowledge is incorporated. Finally, **Digital Forensics & Incident Reporting** transforms identified incidents into a structured format and reports them. The main components of the proposed architecture are currently under active development in the ongoing project.

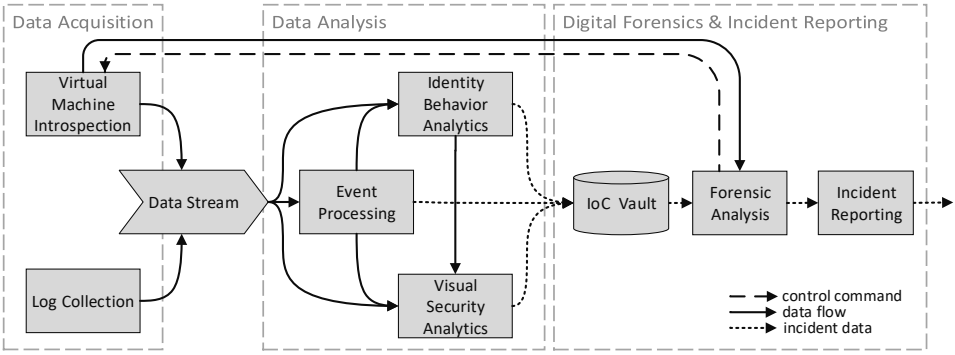

Fig. 1: DINGfest conceptual architecture and data flow

Data Acquisition: **Virtual machine introspection** (VMI) allows the acquisition of an untampered view on the system state from the outside without using an in-guest agent. Due to the strong isolation between the monitoring system and the monitored system, VMI is a valuable method for intrusion detection systems as well as for forensic analysis [Ja14]. Since VMI is resource intensive, we only gather a limited amount of information for intrusion detection. In the proposed architecture, we use VMI to periodically extract information

such as the process list of a monitored VM and trace system and function calls. In case of an incident, we collect additional information for forensic analysis. Besides that, we feed all system logs into the DINGfest data stream using the **log collection** module. Our VMI component is based on the libvmi library and supports the Xen hypervisor.

Data Analysis: The analysis within DINGfest is divided into different modules namely an event processing module, an identity behavior analytics module and a visual security analytics module. DINGfest's **event processing** module monitors the data stream, which consists of different sources like log data or system call traces. It recognizes the occurrence of pre-defined events using an event fingerprint database. Thereby, not only malicious events are detected but also benign ones. Afterwards, these events are passed to the following modules for further processing. To extend the event database, new fingerprints are computed based on the multiple execution of pre-defined events.
If data is generated via user actions (e.g. in applications) an additional **identity behavior analytics** (IBA) module can be applied. The term IBA was developed during the work of DINGfest and is based on the field of behavior analysis [SSW16]. Within this module, suspicious data packets can be identified based on the usage behavior of the underlying user permissions. Traditionally, IAM systems can be found in medium to large-sized organizations containing various information about users as well as their permissions and other attributes (e.g. department) [Hu16]. This data can be exploited for further analysis in the IBA module. Therefore, DINGfest maps data packets to an identity and its corresponding permission set. Thus, the module can compare the current behavior of users and their permissions with already known behavior (e.g. based on peer-group or history analysis).
The **visual security analytics** module combines automated analyses with interactive visualizations for an effective reasoning on large and complex data sets [Ke08]. In DINGfest, visual analytics is applied to include the domain knowledge of security experts into the analysis processes. Therefore, the two main analysis steps (event processing and identity behavior analytics) have to be supported by interactive knowledge-assisted visual interfaces. A central challenge here is to identify appropriate visual metaphors which allow appropriate conversion of knowledge between human and machine. This enables the continuous integration of expert knowledge and improves the detection rate of automated incident detection. Besides the support of the two analysis steps, DINGfest also provides an interface to visualize detected security incidents. This improves awareness for experts on the security situation, generates trust in the automated detection mechanisms, and allows to understand the incident. These components are based on open-source technologies (Angular 5, D3.js).

Digital Forensics & Incident Reporting: This component is responsible for preserving evidence that has been detected by the DINGfest analysis and to prepare this data to be reported to governmental institutions. The detected data is converted into Indicators of Compromise (IoC), which represent indications for incidents in a structured manner. These are stored into a graph database (**IoC vault**) which serves as information source for the

modules forensic analysis and incident reporting.
DINGfest's **forensic analysis** comes with an event fingerprint database which stores a large number of malicious and benign event fingerprints (as described above). Since only unique patterns are part of a fingerprint, the detection of a specific event can be considered as "forensically clean". To support the process of a chain of custody, extracted data is signed and hashed with the corresponding timestamp. For more elaborate forensic analysis, the hypervisor is extended with forensic tools. That allows the use of classic open source forensic analysis software.
The **incident reporting** module is responsible for reporting detected threats, attacks, and vulnerabilities. As soon as new incidents are identified and all relevant information is preserved by the forensics module. The information will be unified and transferred into a structured reporting format. It provides reporting capabilities for incident data to governmental institutions being compliant with applicable laws and directives such as the General Data Protection Regulation. Additionally, it enables the exchange of data within a cyber threat intelligence community, to quickly inform all participants about approaching threats. The module also provides pseudonymization capabilities to protect the reporting company's identity. In addition, procedures to create incentives for the exchange of threat information are researched and developed to increase the acceptance amongst companies.

Acknowledgment This research was supported by the Federal Ministry of Education and Research, Germany, as part of the BMBF DINGfest project (https://dingfest.ur.de). We wish to thank Günther Pernul, Hans Reiser and Felix Freiling for comments on an earlier version of this extended abstract.

References

[Hu16] Hummer, Matthias; Kunz, Michael; Netter, Michael; Fuchs, Ludwig; Pernul, Günther: Adaptive identity and access management—contextual data based policies. EURASIP Journal on Information Security, 2016(1):19, 2016.

[Ja14] Jain, B.; Baig, M. B.; Zhang, D.; Porter, D. E.; Sion, R.: SoK: Introspections on Trust and the Semantic Gap. In: 2014 IEEE Symposium on Security and Privacy. pp. 605–620, May 2014.

[Ke08] Keim, Daniel A.; Andrienko, Gennady L.; Fekete, Jean-Daniel; Görg, Carsten; Kohlhammer, Jörn; Melançon, Guy: Visual Analytics: Definition, Process, and Challenges. In: Information Visualization - Human-Centered Issues and Perspectives, volume 4950 of Lecture Notes in Computer Science, pp. 154–175. Springer, 2008.

[Mi11] Miller, David; Harris, Shon; Harper, Allen; VanDyke, Stephen; Blask, Chris: Security information and event management (SIEM) implementation. Network pro library. McGraw-Hill, New York, NY, 2011.

[Sh16] Shackleford, Dave: , SANS 2016 Security Analytics Survey, 2016. https://www.sans.org/reading-room/whitepapers/analyst/2016-security-analytics-survey-37467.

[SSW16] Shashanka, Madhu; Shen, Min-Yi; Wang, Jisheng: User and entity behavior analytics for enterprise security. In: 2016 IEEE International Conference on Big Data (Big Data). IEEE, pp. 1867–1874, 2016.