Finding Classification Zone Violations with Anonymized Message Flow Analysis

Michael Meinig[1], Peter Tröger[2] and Christoph Meinel[1]
[1]*Hasso-Plattner-Institute (HPI), University of Potsdam, 14482 Potsdam, Germany*
[2]*Beuth University of Applied Science, 13353 Berlin, Germany*

Keywords: Anonymity, Log Analysis, Security Awareness, Threat Awareness, Threat Modelling.

Abstract: Modern information infrastructures and organizations increasingly face the problem of data breaches and cyber-attacks. A traditional method for dealing with this problem are classification zones, such as 'top secret', 'confidential', and 'unclassified', which regulate the access of persons, hardware, and software to data records. In this paper, we present an approach that finds classification zone violations through automated message flow analysis. Our approach considers the problem of anonymization for the source event logs, which makes the resulting data flow model sharable with experts and the public. We discuss practical implications from applying the approach to a large governmental organization data set and discuss how the anonymity of our concept can be formally validated.

1 INTRODUCTION

Information security is one of the crucial topics for modern industry and governmental organizations. Malicious attacks on industrial companies and governmental organizations can cause high annual damage. In 2017, the spending for IT security therefore increased dramatically, but at the same time, an all-high number of cyber-attacks and data breaches was reported (ENISA, 2018). Illegal knowledge transfer and economic sabotage are no longer a rare individual event, but a mass phenomenon. There are thousands of critical security incidents per year that produce an accumulated loss of billions of data records (RBS, 2018). Haystax Technology reported in 2017 that the largest threat for a company are managers with insider access rights, followed by consultants/suppliers and regular employees (Haystax Technology, 2017).

A common approach for dealing with sensitive data is the concept of *information classification* (European Commission, 2001). Different data items, such as paper documents and computer files, are classified according to their security sensitivity. The grouping takes place by the notion of *classification zones*, such as "TOP-SECRET", "SECRET", "RESTRICTED", and "UNCLASSIFIED". All data items in an organization, their storage facilities and the persons accessing them are categorized according to this scheme. The daily procedures and the IT infrastructure are now obligated to regulate the read and write access to data items in a way that their classification zone is taken into account. The necessary access control and monitoring can happen through software mechanisms, such as discretionary access control, electronic authentication procedures and automated security audits, or through more classical security means such as physical separation and entrance guards.

Given the fact that modern industries and institutions are forced to link their data silos and IT systems in complex networks, with some of them being constantly accessible from the Internet, it becomes increasingly harder to enforce classification zones and their protection. Several recent examples show that the detection and avoidance of *classification zone violations* become an increasingly relevant topic in the security community:

- In May 2015, the German federal parliament IT infrastructure was heavily attacked. Offices of parliament members were infiltrated and remotely searched for classified information (Zeit, 2015).
- A former external collaborator of the NSA was sued for a data theft of 50 Terabyte over a time period of 20 years, which is seen as the largest theft of classified governmental material of all times (WP, 2016).

Meinig, M., Tröger, P. and Meinel, C.
Finding Classification Zone Violations with Anonymized Message Flow Analysis.
DOI: 10.5220/0007352602840292
In *Proceedings of the 5th International Conference on Information Systems Security and Privacy (ICISSP 2019)*, pages 284-292
ISBN: 978-989-758-359-9

- In 2017, the credit office Equifax was attacked for stealing sensitive personal information of more than 147 million American people. The data breach leaked information such as name, driver license details and the social security number, which are important identity properties in the USA (Forbes, 2017).

1.1 Problem and Approach

Figure 1 shows the potential security threats for an infrastructure that relies on the idea of classification zones. Given the example of three protection categories (unclassified, restricted, secret), it can be seen that they include each other, e.g. every unclassified information can be treated the same way as a secret information, but not the other way around. There are 4 possible *information flows* that can now take place:

The first flow of information stores a lowly classified document on a lowly classified server, which is an allowed activity. The second possible flow of information stores a lowly classified document on a highly classified server. This flow of information is only allowed in one direction, an information backflow must be avoided. This is typically achieved with a data diode (Genua, 2016). The third flow of information attempts to transfer a highly classified document to a lowly classified server. This information flow is prohibited. The fourth flow of information stores a highly classified document on a highly classified server, which is again an allowed activity.

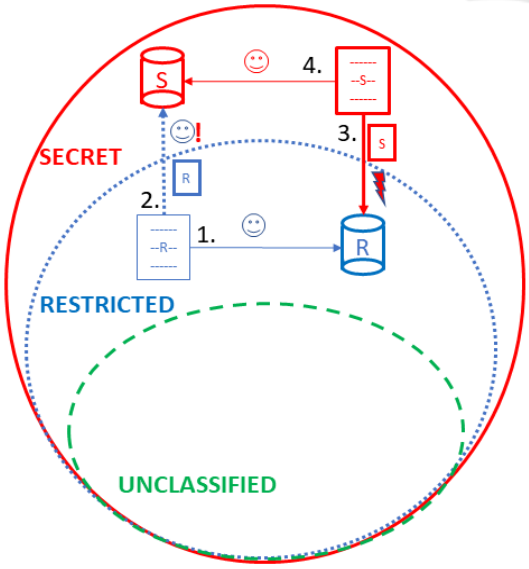

Figure 1: Classification zones - information flows and threats.

Given that general principle, the following security threats exist:

- Information flows from high to low zone
- Uncontrolled information flows from low to high zone (backflow problem - 2nd flow of information in figure 1)
- Information flows between uncategorized external stakeholders and internal servers

It should be noted that 'unclassified' is not equal to an unrestricted read or write access for external parties. The overall classification zone infrastructure is still treated as a closed ecosystem.
The problem scenario can be easily extended with human stakeholders, which also belong to a particular classification zone. They can be treated as a special kind of 'server' that gets read and write access to some classified document.

In this paper, we discuss our work on a practical approach for identifying *classification zone violation* as security threat. The investigation is performed based on protocols of past information flows in the organization, here given as simple textual log files from message exchange servers. The huge original raw data set is converted to an *information flow model*, which can be generated and investigated once, periodically, manually, or in an automated fashion. The goal of this investigation is to find weak security spots and unknown attack vectors in the infrastructure.

A unique property of our work is the anonymization of the source data. This is a common expectation in organizations with mandatory classification zones, but is generally the opposite of open security research. The anonymization demand also normally prevents the collaboration with third-party consultants, or forces them to sign strong legal non-disclosure agreement (NDA) contracts. In our attempt, we try to tackle both issues at the same time.

2 RELATED WORK

The detection of classification zone violations, or suspicious activities in general, is a classical topic in security research and practice. One early example from the 80s is the Haystack system (Smaha, 1988), which was intended to detect intrusions in Air Force systems based on atypical system usage. The authors already considered the problem of an "event horizon", meaning that the monitoring data needed to be reduced before further processing.

Modern computer systems constantly generate large amounts of event logs. These protocols are

intended for system administrators to understand the current system status and to pinpoint system downtime issues.

The usage of log files for finding dependability incidents is a classical topic in high-performance computing. In the past, there were attempts to use pattern discovery (Hellerstein et al., 2002), data mining (Stearley, 2004), (Vaarandi, 2004), (Yamanishi et al., 2005), clustering (Vaarandi, 2003), support vector machines (Bose et al. 2013), (Fronza, 2013), or decision tree learning (Reidemeister et al., 2013) for analyzing massive amounts of original log data. Oliner et al. (Oliner et al., 2008), for example, used the information entropy of message terms for finding failure-related events in HPC logs. Nearly all of these approaches focus on reliability events, such as hardware component failures, which are indicated through severity markers in the log files. This makes these approaches not directly applicable for our classification zone problem, but the available ideas for log information reduction can be directly applied (Cheng, 2015), (Azodi, 2014), (Sapegin, 2013), (Pantola, 2010). We also utilize security meta-data in the log analysis, something that is not commonly available in HPC-alike systems.

One part of our approach is the conversion of log data to data flow representations. The latter is a well-known tool in the security community. A classical approach for this is are data flow diagrams or similar techniques such as flowcharts (Gane, 1977), (DeMarco, 1978), (Yourdon, 1989). Data flow diagrams are well known for their application within the STRIDE threat modeling methodology by Microsoft (Swiderski and Snyder, 2004). Another example is the work by Schmidt et al. who used data flow analysis for ranking security issues in automotive software architectures (Schmidt et al., 2014). Meinig et al. described the notion of extended data flow diagrams, which can help to express the relation between infrastructure components and classification zones (Meinig and Meinel, 2018).

The resulting data flow information can be investigated by sophisticated algorithms, so that security threats can be found automatically and at run-time. Data flow analysis in the security context is already applied for system calls made by applications (Li et al., 2008), network traffic (Su et al., 2018), (Tzur-David et al., 2009), (Hofstede, 2018), virtual runtime engines (Feng et al., 2010), program binaries (Chen et al., 2011), or program code (Baca, 2010). For this publication, we focus on the generation of data flow diagrams that can be inspected by human auditors from inside and outside the organization.

3 FROM EVENT LOGS TO DATA FLOW ANALYSIS

Given the demand for both anonymity and useful security analysis results, we created an approach that combines the following steps (see also Figure 3):

1. Event log sanitizing
2. Evaluation of message paths
3. Model Generation (Anonymization of source data, while keeping relevant security information and translation into a data flow model)

We applied the approach to the following data set from a large governmental organization:

- Unstructured raw logs about network communication for a set of servers (see Figure 2)
  - Time range: 01/16 - 11/16
  - One folder per server
  - Multiple text files per folder, each covering a particular time span
  - Log entries per file in text column format
  - Per log entry: Timestamp, unique message identifier, send / receive operation, data size
- List of nodes and their security classification

Content and type of the exchanged data is a confidential information in itself and can therefore not be shared in the analysis result.

```
Folder Name: Server SA
File Name: 2016-01-01XXX.log
##########################################
01  00:02:12   Data received from   Server SB
    YYY Bytes XXXMessage-ID-ABC

01  00:02:12   Forwarded to         Server SC
                          XXXMessage-ID-ABC

01  00:02:12   Data send to         Server SD
    YYY Bytes         XXXMessage-ID-ABC
##########################################
```

Figure 2: Raw log data.

3.1 Event Log Sanitizing

We developed a set of Python scripts that parse the original log files linearly and generate a database of node communication activities. This step removes invalid and incomplete information items, such as log entries with missing message classification information or invalid timestamps. Both problems were discussed with the administrators of the infra-

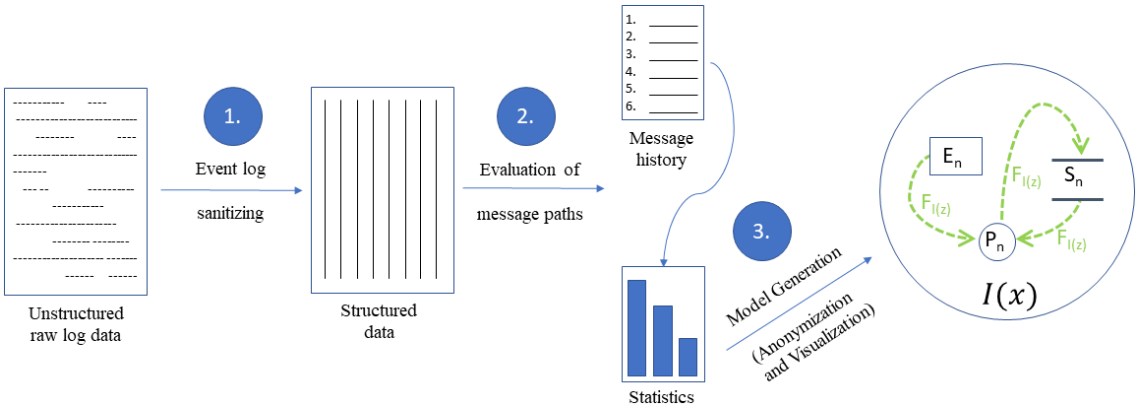

Figure 3: Steps to detection of classification zone violations.

structure and justified with implementation bugs, and not with a potential attempt for cleaning the logs from an earlier attack. The result of this normalization step is a set of records, as shown in Table 1, were each provides the following information:

- Column 1: Date
- Column 2: Time
- Column 3: Node generating the log entry
- Column 4: Data operation performed by the referenced node (send / relay / receive)
- Column 5: Data size of the message
- Column 6: Peer node
- Column 7: Unique message identifier

Table 1: Set of records.

Date	Time	Server Name	Data Direction	Byte Size	Server Name	Message ID
2016-01-01	00:02:12	S_A	Received from	YYY	S_B	ABC
2016-01-01	00:02:12	S_A	Relay to		S_C	ABC
2016-01-01	00:02:12	S_A	Send to	YYY	S_D	ABC
…	…	…	…			

It must be noted here that message sending comes in two flavors: Sending triggered by the node itself, and sensing triggered due to relaying of a received message. This leads to the fact that a single receive activity may be reasoned by one of them, which is not directly visible from the entry. For that reason, we need to consider the message path as a whole.

3.2 Evaluation of Message Paths

Given the set of records about direct node-to-node interaction, we are now detecting message paths that may cross relay nodes. This can be easily done by clustering single transfer activities based on the unique message identifier. This is again implemented with a set of Python scripts, since the amount of log data makes such an approach still feasible. For larger data sets, according scalable analysis tools based on NoSQL databases can be applied.

The result is a set message transfers, where each transfer has a set of participating nodes. This can be the starting point for a multitude of analysis steps. In the approach here, we continue by only relying on the pure node-to-node message frequency statistics. Together with the given node security classification, this leads to a dataset that describes the interaction frequency of nodes in different, or the same, classification zone. Together with the overall amount of messages, it is now possible to generate the graph representation for further analysis.

Figure 4 shows an example for the resulting data file. It allows to determine the path of every message through the network.

3.3 Model Generation

The next step is to create a representation for the message flow data that can be shared with external parties.

The visualization of the identified information exchange flows is done with the *secure data flow diagram (DFDSec)* approach (Meinig and Meinel, 2018). It supports the representation of classification zones in an extended data flow diagram. The model consists of five elements: processes, external entities, data stores, data flows and security classification zones.

Processes are entities that perform activities by operating on incoming data and potentially generating an output. In our example data set, this relates to the different servers in their role as message sender, receiver, or relay.

Message-ID:	[List of Server Connections	]
ABC:	['Server S_A → Server S_B',	
	'Server S_A → Server S_C',	
	'Server S_A → Server S_D'	]
…:	[…→ …	]

Figure 4: Message paths.

External Entities are actors outside the modeled system which are the source or destination of information. They may be part of different security classification zone or a complete own one.

Data stores can contain information but perform no further processing that may lead to new or forwarded messages. Given the fact that the raw log data only describes active entities, and not passive storage facilities, we do not utilize the concept of data stores with the experiment.

Data flows are information flows among classification zones, processes, external entities and data stores. The most interesting ones cross classification zones and must therefore be investigated more specifically with respect to security threats.

Classification zones are the specific security level that includes processes, data flows, data stores, and external entities. They are drawn as dashed circles around other elements in the data flow diagram.

We generate the DFDSec diagram automatically from the message exchange data generated in the last step. This is done by producing a single GraphML file per data set. This file is modified by an auto-layout algorithm for determining the model element positions in the final picture.

One of the targets of our work is the anonymized and pseudonymized representation of sensitive information in a security model. We therefore take the following anonymization steps:

- The timestamp of the message exchange is not used in the model generation (column 1 & 2).
- Server names are translated in pseudonyms (column 3).
- Message size and message identifier are not used in the model generation (column 5 & 7).
- Data flows between processes are attributed with the fraction of message exchanges they contributed to the overall amount of data exchange.

In order to reduce model complexity in practice, we omit data exchanges between nodes in the same classification zone that contributes less than 1% to the overall traffic. We argue here that for a potential attacker, high-traffic nodes are more valuable than sparsely used servers. One obvious reason is that successful cryptographic attacks mostly rely on the availability of larger message flow probes.

Message transfers that cross classification zones are not capped, since each single message transfer may indicate a potential security breach.

Figure 5 shows a DFDSec diagram generated from the real-world data set in a governmental organization. It can be seen that there are two different classification zones in real use, RESTRICTED and SECRET. Each classification zone contains several operational nodes (sender, receiver or relay servers). Between these nodes there are information flows. Within the utilized data set, it turned out that message exchanges crossing a classification zone border did not take place. For the given governmental system, this is the expected and correct mode of operation.

When a DFDSec analysis shows a zone crossing, it depends on the kind of source and destination zone, as discussed in Section 1.1, that it is a problem. A typical approach would be the repeated investigation of the issue over time, for example by generating multiple DFDSec models periodically and compare them. This would clarify if the classification zone crossing is a regular effect at normal business hours with an expectable message frequency, or if this is an anomaly in the system operation. The analysis can also consider specific attributes of the participating nodes, such as their role in the organization.

For the given data set, it turned out that the most important operational node is S_A. 26 percent of the traffic goes away from this node and 28 percent of the traffic goes in.

In the classification zone SECRET, the most important operational node is R_A. 36 percent of the traffic is leaving this node. By contrast, this node has hardly any incoming connection.

From an attacker point of view, these two nodes seem to be the most interesting. Most of the information flows start or end here, so they have the highest potential for gathering classified information. From the defender's point of view, IT security measures must be primarily taken on these two nodes. One idea would be to design these important nodes redundantly, so that, for example, in hierarchical organizations, they no longer act as single relay for messages. This would diffuse classified information over multiple paths in the organization, which makes it harder for an attacker to identify the relevant nodes in the system that are worth an attack.

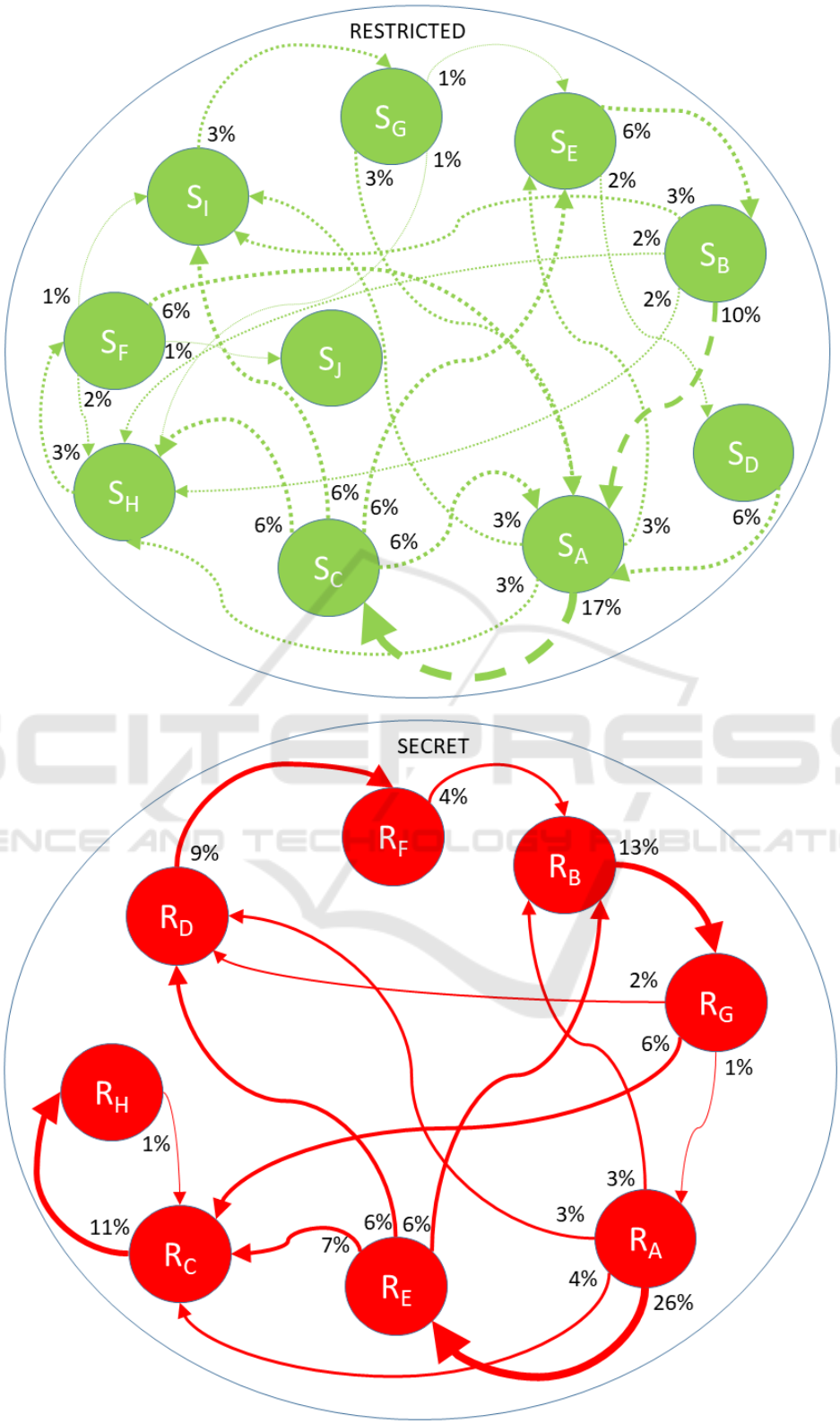

Figure 5: DFDSec.

The most important information flow happens in the classification zone RESTRICTED. The connection S_A - S_C is responsible for 17 percent of the message flows, followed by the connection S_B - S_A at 10 percent. The most important information flow in the classification zone SECRET is the connection R_A - R_E at 26 percent, followed by the connection R_B - R_G at 13 percent and the connection R_C - R_H at 11 percent.

From the point of view of an attacker, these connections are those that are eligible for a man-in-the-middle, e.g. by compromising switches and routers on the network path between the two nodes. From a defensive point of view, it would be therefore worth considering whether to redistribute traffic differently in order to avoid connections that are particularly busy, e.g. load sharing (Eager et al., 1986), or to harden the connection network accordingly.

4 ANONYMITY

Security analysis that relies on unfiltered classified data produces again a classified entity. It can rarely be made available directly to third parties. The challenge in the daily work is therefore to share such data without jeopardizing the disclosure of classified infrastructure details on the one hand, and not limiting the usefulness of the effort too much on the other hand.

While the DFDSec approach helps to make stripped results still useful, it remains an open question if the classified infrastructure knowledge is truly protected. It is not enough to reduce the published data set and hope for the best, instead, an objective and repeatable verification must be performed to validate this property.

One possibility for the evaluation of an anonymization approach was described by Samarati and Sweeney. Their model is called *k-anonymity*. A published data set is declared to have k-anonymity when identifying information for a single node, such as a server name, is indistinguishable from at least k-1 similar entities (Samarati, 1998), (Sweeney, 2002). A larger k means a higher degree of anonymization. The result is the confidence of 1 / k that a correct linking of correlating classified knowledge is not possible.

The model was later extended by the idea of l-diversity (Machanavajjhala, 2007) and t-closeness (Li, 2007). These extensions and the original approach prevent the possibility of de-anonymization, which could result from an unsorted matching attack, temporal attacks, or complementary release attacks (Ganta et al., 2008). These extensions fixed specific problems of k-anonymity such as the homogeneity attack or the background knowledge attack (Machanavajjhala, 2007).

For the approach described in this paper, we improve the *k-anonymity* by omitting 5 of the 7 original data columns. Only the sending server and receiving server are kept in their original meaning. k is determined here by the smallest number of server connections between two servers in the message flow data set. From the viewpoint of anonymization attacks, our approach covers the following security threats:

In *unsorted matching attacks*, the data columns are separated and disclosed in the same ordering. This allows the original data to be restored. This attack is prevented in our concept by only publishing two of the original columns. In addition, the sorting in the published graph representation is based on the node-to-node message frequency, which can vary. This automatically changes the order of the entries.

The *temporal attack* exploits the fact that data collections are dynamic. Adding, modifying, or removing entries can affect the k-anonymity of the analysis result. This attack is prevented by not changing the shared attributes (pseudonym server names) when publishing updated or extended analysis results. Subsequent versions of a DFDSec diagram therefore must re-use the same server pseudonyms as before.

In the case of the *complementary attack*, different releases of subsets of the base file may result in different anonymizations, each corresponding to k-anonymity. By combining the respective publications, the k-anonymity is canceled again. This attack is prevented by having all subsequent versions of an analysis result based on the original one. This is guaranteed by always using the same column attributes (inbound, outbound server) and column contents for unmodified data. The only changing variable for updated DFDSec should be the message frequency and the set of processes being modelled.

Homogeneity attacks rely on the idea that there are groups of data items, here model elements, that all have the same sensitive attribute. This attack is prevented in our concept by simply not using the sensitive attributes (e.g. message ID or server name) for the final published model. Additional diversity, as proposed in (Machanavajjhala, 2007), is therefore not needed.

The *background knowledge attack* exploits the fact that an attacker can unambiguously allocate data content despite k-anonymity through additional

knowledge. In our approach background knowledge is not made possible by omitting all sensitive attributes and by the physical separation of classified and anonymized public information. We publish only data necessary for external support, following the filter-in-principle described in (Pang and Paxson, 2003).

5 CONCLUSIONS AND FUTURE WORK

In this paper, we discussed an approach to identify classification zone violations by the means of automated message exchange analysis. The determination of critical information flows leads to the identification of sensitive operational nodes that are high-risk targets for security attacks. Since the raw data logs and the analysis results are classified information by themselves, we propose a straightforward anonymization approach for the original data. We discussed the potential for well-known anonymization attacks.

Our analysis of one year of real-world data showed that the system under investigation is in good health. Classification zone violations could not be identified. The visualization with reduced DFDSec diagrams turned out to be a feasible method for governmental organizations with high demands on data protection. The use of anonymized DFDSec models allows for sharing the analysis results with external security consultants and management personal that has a low security classification.

This paper is part of ongoing project work inside the governmental organization. Future work will focus on performing more advanced analysis, such as an anomaly detection that considers the time axis of the message exchange patterns or better consideration of message semantics. It is also planned to migrate these techniques from a pure offline analysis to online monitoring infrastructure.

REFERENCES

Azodi, A., Jaeger, D., Cheng, F., Meinel, C., 2014. Event Normalization Through Dynamic Log Format Detection, *ZTE Communications*.

Baca, D., 2010. Identifying Security Relevant Warnings from Static Code Analysis Tools through Code Tainting, *International Conference on Availability, Reliability and Security, IEEE*, Krakow, Poland, DOI: 10.1109/ARES.2010.108.

Bose, R. P. J. C., van der Aalst, W. M. P., 2013. Discovering signature patterns from event logs, *IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, IEEE, Singapore, Singapore, DOI: 10.1109/CIDM.2013.6597225.

Cheng, F., Sapegin, A., Gawron, M., Meinel, C., 2015. Analyzing Boundary Device Logs on the In-Memory Platform, *Proceedings of the IEEE International Symposium on Big Data Security on Cloud*.

Chen, Z., Wang, X. J., Zhang, X. X., 2011. Dynamic Taint Analysis with Control Flow Graph for Vulnerability Analysis, *First International Conference on Instrumentation, Measurement, Computer, Communication and Control*, IEEE, Beijing, China, DOI: 10.1109/IMCCC.2011.66.

DeMarco, 1978. T. Structured Analysis and System Specification, *Yourdon Press*, New York, NY.

Eager, D.L., Lazowska, E.D., Zahorjan, J., 1986. Adaptive Load Sharing in Homogeneous Distributed Systems, *IEEE Transactions on Software Engineering*, Vol. SE-12, pp. 662-675.

ENISA, 2018. ENISA Threat Landscape Report 2017, Final version 1.0 ETL 2017, www.enisa.europa.eu.

European Commission, 2001. Commission Decision of 29 November 2001 amending its internal Rules of Procedure (notified under document number C(2001) 3031), 2001/844/EC, ECSC, Euratom.

Feng, B., Zhang, M., Xu, G., Niu, X., 2010. Runtime protecting system for java applications with dynamic data flow analyzing, *2nd International Conference on Future Computer and Communication*, IEEE, Wuha, China, DOI: 10.1109/ICFCC.2010.5497460.

Forbes, 2017. Equifax's Enormous Data Breach Just Got Even Bigger, https://www.forbes.com/.

Fronza, I., Sillitti, A., Succi, G., Terho, M., 2013. Failure prediction based on log files using Random Indexing and Support Vector Machines, *Journal of Systems and Software*, Volume 86, Issue 1, p. 2-11, Elsevier Science Inc., New York, NY, USA, DOI: 10.1016/j.jss.2012.06.025.

Gane, C. and Sarson, T. 1977. Structured Systems Analysis and Design, *Improved Systems Technologies*, Inc., New York, NY.

Ganta, S., Kasiviswanathan, S., Smith, A. 2008. Composition attacks and auxiliary information in data privacy, in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge discovery and data mining (KDD 2008)*, ACM, New York, NY, USA, p. 265-273, DOI: 10.1145/1401890.1401926.

Genua gmbh, 2016. *Datendiode vs-diode*, Munich, Germany, webpage accessed 09.10.2018, https://www.genua.de/.

Haystax Technology, 2017. *Insider Attacks - Industry Survey*, http://haystax.com/.

Hellerstein, J. L., Ma, S., Perng, C.-S., 2002. Discovering actionable patterns in event data, *IBM Systems Journal*, Volume 41, Issue 3, p. 475-493, DOI: 10.1147/sj.413.0475.

Hofstede, R., Pras, A., Sperotto, A., Dreo-Rodosek, G., 2018. Flow-Based Compromise Detection: Lessons

Learned, *IEEE Security & Privacy*, Volume 16, Issue 1, p. 82 - 89, IEEE, DOI: 10.1109/MSP.2018.1331021.

Li, N., Li, T., Venkatasubramanian, S., 2007. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity, In: *International Conference Data Engineering*, Vol. 7, p. 106–115.

Li, P., Park, H., Gao, D., Fu, J., 2008. Bridging the Gap between Data-Flow and Control-Flow Analysis for Anomaly Detection, *Annual Computer Security Applications Conference (ACSAC)*, IEEE, Anaheim, CA, USA, DOI: 10.1109/ACSAC.2008.17.

Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M. 2007. l-diversity: Privacy beyond k-anonymity In: *ACM Transactions on Knowledge Discovery from Data*, Vol. 1, ACM.

Meinig, M. and Meinel, C., 2018. Securing the Flow - Data Flow Analysis with Operational Node Structures, In *Proceedings of the 4th ICISSP - Volume 1: ICISSP*, ISBN 978-989-758-282-0, p. 241-250. DOI: 10.5220/0006570302410250.

Oliner, A. J., Aiken, A., Stearley, J., 2008. Alert Detection in System Logs, *Eighth IEEE International Conference on Data Mining*, IEEE, Pisa, Italy, DOI: 10.1109/ICDM.2008.132.

Pang, R. and Paxson, V., 2003. A high-level programming environment for packet trace anonymization and transformation. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM'03)*. ACM, New York, NY, 339–351, DOI: 10.1145/863955.863994.

Pantola, V. A., Yatco, F. R., Pineda, J. D., 2010. Normalization of Logs for Networked Devices in a Security Information Event Management System, DOI: 10.13140/RG.2.1.4170.1202.

Reidemeister, T., Jiang, M., Ward, P. A. S., 2011. Mining unstructured log files for recurrent fault diagnosis, *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*, IEEE, Dublin, Ireland, DOI: 10.1109/INM.2011.5990536.

Risk Based Security, 2018. *Data Breach Quick View Report*, https://pages.riskbasedsecurity.com/.

Samarati, P., Sweeney, L. 1998. Protecting Privacy when Disclosing Information: k-Anonymity and Its Enforcement through Generalization and Suppression, *Tech. rep. SRI-CSL-98-04, SRI Computer Science Laboratory*, Palo Alto, CA.

Sapegin, A., Jaeger, D., Azodi, A., Gawron, M., Cheng, F., Meinel, C., 2013. Hierarchical Object Log Format for Normalisation of Security Events, *Proceedings of the 9th International Conference on Information Assurance and Security (IAS 2013)*. IEEE CS, Tunis, Tunisia.

Schmidt, K., Tröger, P., Kroll, H., Bünger, T. et al., 2014. Adapted Development Process for Security in Networked Automotive Systems, *SAE Int. J. Passeng. Cars Electron. Electr. Syst. 7*(2):516-526, DOI: 10.4271/2014-01-0334.

Smaha, S. E., 1988. Haystack: an intrusion detection system, [Proceedings 1988] *Fourth Aerospace Computer Security Applications, IEEE*, Orlando, FL, USA, USA, DOI: 10.1109/ACSAC.1988.113412.

Stearley, J., 2004. Towards informatic analysis of syslogs, *IEEE International Conference on Cluster Computing (IEEE Cat. No.04EX935)*, IEEE, San Diego, CA, USA, DOI: 10.1109/CLUSTR.2004.1392628.

Su, L., Yao, Y., Li, N., Liu, J., Lu, Z., Liu, B., 2018. Hierarchical Clustering Based Network Traffic Data Reduction for Improving Suspicious Flow Detection, *17th IEEE TrustCom/ 12th IEEE BigDataSE, IEEE*, New York, USA, DOI: 10.1109/TrustCom/ BigDataSE.2018.00108.

Sweeney, L. 2002. k-anonymity: a model for protecting privacy, *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10 (5), p. 557-570.

Swiderski, F. and Snyder, W., 2004. Microsoft Professional Threat Modeling, *Microsoft*, ISBN 0-7356-1991-3.

Tzur-David, S., Dolev, D., Anker, T., 2009. MULAN: Multi-Level Adaptive Network Filter, In: Chen Y., Dimitriou T.D., Zhou J. (eds) *Security and Privacy in Communication Networks, SecureComm 2009*, vol 19. Springer, Berlin, Heidelberg.

Vaarandi, R., 2003. A data clustering algorithm for mining patterns from event logs, *Proceedings of the 3rd IEEE Workshop on IP Operations & Management (IPOM 2003)* (IEEE Cat. No.03EX764), IEEE, Kansas City, MO, USA, DOI: 10.1109/IPOM.2003.1251233.

Vaarandi, R., 2004. A Breadth-First Algorithm for Mining Frequent Patterns from Event Logs, In: Aagesen F.A., Anutariya C., Wuwongse V. (eds) *Intelligence in Communication Systems, Lecture Notes in Computer Science*, vol 3283. Springer, Berlin, Heidelberg, DOI: 10.1007/978-3-540-30179-0_27.

Washington Post, 2016. Government alleges former NSA contractor stole 'astonishing quantity' of classified data over 20 years, webpage accessed 31.03.2018, https://www.washingtonpost.com/.

Yamanishi, K., Maruyama, Y., 2005. Dynamic syslog mining for network failure monitoring, *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, p. 499-508, ACM, Chicago, Illinois, USA, DOI: 10.1145/1081870.1081927.

Yourdon, E. 1989. Modern Structured Analysis, Yourdon Press, Upper Saddle River, NJ.

Zeit, 2015. Bundestags-Hack - Merkel and the Fancy Bear, webpage accessed 19.04.2018, https://www.zeit.de/.