

Volume 16, Issue 4 • Fall 2017

ISSN 1445-3312 (Print)

ISSN 1445-3347 (Online)

JOURNAL OF INFORMATION WARFARE



# Journal of Information

# Warfare

Volume 16 Issue 4 Fall 2017



# Journal of Information Warfare

Volume 16, Issue 4  
Fall 2017

## Contents

<b>From the Editor</b>	i
<i>L Armistead</i>	
<b>Authors</b>	iii
Understanding Cyber Terrorism from Motivational Perspectives	1
<i>Z Yunos, S Sulaman</i>	
Password Recovery and Data Retrieval in the Android Operating System	14
<i>D Hintea, R Bird, J Moss</i>	
Preparation, Modelling, and Visualisation of Cyber Common Operating Pictures for National Cyber Security Centres	26
<i>T Pahi, M Leitner, F Skopik</i>	
Creation of Flow-Based Data Sets for Intrusion Detection	41
<i>M Ring, S Wunderlich, D Gründl, D Landes, A Hotho</i>	
Energy-Conscious Adaptive-Security Scheme: A Reliability-Based Stochastic Approach	55
<i>C Taramonli, MS Leeson, RJ Green</i>	
Ant Tree Miner Amyntas: Automatic, Cost-Based Feature Selection for Intrusion Detection	73
<i>FH Botes, L Leenen, R De La Harpe</i>	
Phobic Cartography: A Human-Centred, Communicative Analysis of the Cyber-Threat Landscape	93
<i>JKL Scott</i>	
Should 'RuNet 2020' Be Taken Seriously? Contradictory Views about Cyber Security Between Russia and the West	113
<i>M Ristolainen</i>	

## Authors

---



**Robert Bird** is a Senior Lecturer at Coventry University and the course director for a number of programs, including the master's program in Forensic Computing. Prior to joining Coventry, he was a superintendent with the West Midlands Police.



**Frans Hendrik Botes** is a postgraduate student at Cape Peninsula University of Technology. He is a hobbyist hacker and has research interests in artificial intelligence and cyber security.



**Dr. Retha De La Harpe** is the acting Head of the IT Department in the Faculty of Informatics and Design at Cape Peninsula University of Technology. She earned both a bachelor's degree in Informatics and a master's degree from Rand Afrikaans University. She earned D.Tech (IT) qualification at Cape Peninsula University of Technology in 2009. She is the South African Coordinator of the Informatics Development for Health in Africa (INDEHELA) international research network and was a National Research Fund grant holder for four years. She is author and co-author of several research funding proposals—including the South Africa Finland Partnership. Her main research interest concerns data quality implications in both business and healthcare contexts.



**Dr. Roger Green** is Emeritus Professor of Electronic Communication Systems at the University of Warwick. He earned a bachelor's degree in Electronics from the University of Manchester in 1973, and a doctorate in Video Communications from Bradford University in 1976.

After graduation, he worked for GEC Electro-Optical Systems in Essex until 1978 when he rejoined academic life at Bradford University, serving there from 1978 to 1999. He was appointed to the Chair in Electronic Communication Systems at Warwick University in September 1999 and led a research group there in optical communications. For five years during his time at Warwick he served as Head of the Division of Electrical and Electronic Engineering. He continues to be active in the area of optical wireless communications. In 2009, he was awarded the D.Sc. in Photonic Communications, Systems, and Devices by Warwick University for his research. He was appointed to fellowship with the Institution of Engineering and Technology, and the Institute of Physics, and became a Senior Member of the Institute of Electrical and Electronics Engineers. He is also a member of the European Society for Engineering and Medicine. He oversaw the successful completion of 66 research students—mainly at the doctoral level—during his career. He has published widely, with around 300 refereed research papers published and presented internationally. He holds several patents.



**Dominik Gründl** is a graduate student at Coburg University of Applied Sciences and Arts, where he serves on Dr. Dieter Landes' research team investigating Intrusion Detection Systems (IDSs). He completed undergraduate studies in Informatics at Coburg.



**Dr. Diana Hintea** is a Lecturer in Computer Science at Coventry University. There, she leads a series of modules on advanced programming, algorithms, and digital forensics. She earned a bachelor's degree in Engineering in 2010 from the Technical University of Cluj-Napoca and a doctorate from Coventry University in 2014. Her research interests focus on digital forensics, machine learning, and reinforcement learning-based applications.





**Dr. Andreas Hotho** is a Professor at the University of Würzburg. He earned a doctorate from the University of Karlsruhe, where he worked from 1999 to 2004 at the Institute of Applied Informatics and Formal Description Methods (AIFB) in the areas of text, data, and web mining; semantic web; and information retrieval. From 2004 to 2009 he was a Senior Researcher at the University of Kassel. In 2011, he joined the L3S. Since 2005, he has been leading the development of BibSonomy, the social bookmark and publication-sharing platform. He has published more than 100 articles in journals and at conferences, has co-edited several special issues and books, and has co-chaired several workshops. He has worked as a reviewer for journals and has been a member of international conferences and workshop program committees. His research focuses on Data Science—in particular, on the combination of data mining, information retrieval, and the semantic web.



**Dr. Dieter Landes** is a Professor of Software Engineering and Database Systems at Coburg University of Applied Sciences and Arts. He holds a diploma in Informatics from the University of Erlangen-Nuremberg, and a doctorate in Knowledge-Based Systems from the University of

Karlsruhe. After several years working in industry—including time with Daimler Research—he joined Coburg in 1999. He has published 70 papers in journals, books, and at conferences. His research interests include requirements engineering, software-engineering education, learning analytics, and data mining.



**Dr. Louise Leenen** is a Principal Scientist in the Cyber Defence Research Group at the Council for Scientific and Industrial Research. She earned a doctorate in Computer Science from the University of Wollongong. She is the Chair of the International Federation for Information Processing's

Working Group on ICT in War and Peace. Her research focuses on artificial-intelligence applications in cyber defence.



**Dr. Mark S. Leeson** is a Reader in Communication Systems in the School of Engineering at the University of Warwick. He earned bachelor's degrees in Electrical and Electronic Engineering from the University of Nottingham in 1986. He earned a doctorate in Engineering from the

University of Cambridge in 1990. He worked as a Network Analyst for National Westminster Bank in London prior to taking academic posts in London and Manchester. In 2000 he joined the School of Engineering at Warwick. He has more than 250 publications and has supervised seventeen successful research students. He is a Senior Member of the Institute of Electrical and Electronics Engineers, and a Fellow of both the UK Institute of Physics and the UK Higher Education Academy. His major research interests are optical-communication systems, molecular communications, and machine learning.



**Dr. Maria Leitner** is a Scientist in the Center for Digital Safety & Security at AIT Austrian Institute of Technology. She earned a doctorate in Computer Science in 2015 from the University of Vienna. Her thesis focused on the integration and life-cycle management of security policies in process-

aware information systems in order to ensure holistic security-policy management in organisations. Prior to joining AIT, she was a Researcher at SBA Research and a Research Assistant in the Workflow Systems and Technology Group in the Faculty of Computer Science at the University of Vienna. She is currently coordinating and working on national and international research projects in the areas of situational awareness, cyber defence, ICS security and identity management. She is representing AIT in the European Cyber Security Organisation (ECSSO) Working Group 5 (Education, training, awareness, exercise) and in the Cyber Security Platform Austria. She is a member of the ACM and has published more than 25 refereed articles, conference papers, and workshop papers.



**James Moss** is a Senior Penetration Tester who has been involved in leading engagements against UK government systems and commercial clients.



**Keith Scott** is Programme Leader for English Language at De Montfort University, where he is also a member of the Cyber Security Centre. His research is concerned with human factors in cyber security, with a particular interest in the fields of influence and perception management.



**Timea Pahi** is a Junior Scientist at the Austrian Institute of Technology and is working on several research projects focusing on national cyber security, the protection of critical infrastructures, and cyber situational awareness.



**Dr. Florian Skopik** is a Senior Scientist at the ICT Security Research Team at the Austrian Institute of Technology (AIT), where he is responsible for national and European research projects focusing on smart grid security, the security of critical infrastructures, and national cyber security and cyber defence. Before joining AIT, he worked with the Distributed Systems Group at the Vienna University of Technology as a Research Assistant and Postdoctoral Research Scientist from 2007 to 2011, where he was involved in a number of international research projects dealing with cross-organisational collaboration over the Web. In the context of these projects, he finished his doctoral studies. He also spent a sabbatical at IBM Research India in Bangalore for several months. In addition, he has worked for numerous small- and medium-sized enterprises as a Firmware Developer for microcontroller systems for about 15 years. He has published more than 100 scientific conference papers and journal articles, and is a member of various conference program committees and editorial boards, as well as standardisation groups, such as ETSI TC Cyber and OASIS CTI. He holds 20 industry-relevant security certifications, including Trusted Security Auditor, ISA/IEC 62443 Security Specialist, CCNA Security, and ISO27001 Information Security Manager. In 2017, he finished a professional degree in Advanced Computer Security at Stanford University. He is an Institute of Electrical and Electronics Engineers Senior Member and Member of the Association for Computing Machinery (ACM).



**Markus Ring** is a Research Associate at Coburg University of Applied Sciences and Arts where he is working on his doctoral thesis. He previously studied Informatics at Coburg. He has previously worked as a Network Administrator at T-Systems Enterprise GmbH. His research interests include the generation of realistic flow-based network data and the application of data-mining methods for cyber-security intrusion detection.



**Dr. Mari Ristolainen** is a Researcher at the Finnish Defence Research Agency. She has studied psychology at the Moscow State University and she earned a doctorate in Russian Language and Cultural Studies from the University of Joensuu in 2008. She has been conducting postdoctoral research in the field of Russian and Border Studies in several Academy of Finland- and EU-funded projects at the University of Eastern Finland and at the University of Tromsø. Her current research interests include cyber warfare as a phenomenon, Russian digital sovereignty, and the governance of cyber/information space.



**Sharifuddin Sulaman** is the International Engagement Executive at CyberSecurity Malaysia, an agency under the Ministry of Science, Technology, and Innovation, Malaysia. He earned a bachelor's degree in Information Systems Management from Universiti

Teknologi MARA (UiTM).



**Dr. Chrysanthi Taramonli** is a Lecturer in Cyber Security and Forensics at Coventry University. She earned a bachelor's degree in Computer Science and a master's degree in Networks Security. She earned a doctorate in Engineering from the University of Warwick. Her

research is focused on the use of stochastic methods in low-energy encryption and adaptive security, as well as on network forensics and stochastic log-file analysis.



**Sarah Wunderlich** is a Research Associate at the Coburg University of Applied Sciences and Arts. She earned a master's degree in Computer Science from Coburg in 2016. She has also worked as a Lecturer in Data Mining at Coburg. Her research interests include the generation of

realistic flow-based network data and the application of data-mining methods for cyber-security intrusion detection.



**Dr. Zahri Yunos** is the Chief Operating Officer of CyberSecurity Malaysia, an agency under the Ministry of Science, Technology, and Innovation, Malaysia. He earned a doctorate in Information Security from the Universiti Teknikal Malaysia Melaka (UTeM). He has

contributed to various publications related to cyber security. He also has been appointed as Adjunct Professor at UTeM.

# Creation of Flow-Based Data Sets for Intrusion Detection

M Ring<sup>1</sup>, S Wunderlich<sup>1</sup>, D Grüdl<sup>1</sup>, D Landes<sup>1</sup>, A Hotho<sup>2</sup>

<sup>1</sup>*Faculty of Electrical Engineering and Informatics  
Coburg University of Applied Sciences  
Coburg, Germany*

*E-mail: markus.ring@hs-coburg.de; sarah.wunderlich@hs-coburg.de;  
dominik.gruedl@stud.hs-coburg.de; dieter.landes@hs-coburg.de*

<sup>2</sup>*Data Mining and Information Retrieval Group  
University of Würzburg  
Würzburg, Germany*

*E-mail: hotho@informatik.uni-wuerzburg.de*

**Abstract:** *Publicly available labelled data sets are necessary for evaluating anomaly-based Intrusion Detection Systems (IDSs). However, existing data sets are often not up-to-date or not yet published because of privacy concerns. This paper identifies requirements for good data sets and proposes an approach for their generation. The key idea is to use a test environment and emulate realistic user behaviour with parameterised scripts on the clients. Comprehensive logging mechanisms provide additional information which may be used for a better understanding of the inner dynamics of an IDS. Finally, the proposed approach is used to generate the flow-based CIDDS-002 data set.*

**Keywords:** *Data-Set Generation, NetFlow, OpenStack, Intrusion Detection Systems (IDSs), Coburg Intrusion Detection Data Sets (CIDDS)*

## Introduction

The scientific community has pursued the idea of detecting novel attacks with anomaly-based Intrusion Detection Systems (IDSs) for decades. Buczak and Guven (2016) provide an overview of that community effort. Operational systems, however, almost exclusively rely on signature-based IDSs. Sommer and Paxson (2010) identify various challenges (such as the lack of publicly available evaluation data sets, fundamental evaluation difficulties, or problems stemming from false-positives) for the use of machine learning methods in the context of anomaly-based IDSs. This work tackles one challenge of this larger task.

In particular, this work focuses on the shortage of publicly available evaluation data sets for IDSs. New anomaly-based intrusion detection methods need to be evaluated and compared with existing methods. One way of evaluating and comparing these methods is to use labelled evaluation data sets. Yet, many existing data sets are not publicly available due to privacy concerns, and available data sets are often not up to date or do not reflect necessary attack scenarios. Likewise, the use of

real network traffic for evaluation is problematic since no ground truth (labels) is available. Furthermore, malware and other attack scenarios must not be run in operational environments, as it would compromise these environments. Consequently, no malicious network traffic can be explicitly generated when capturing the network traffic of operational environments.

This work tackles the shortage of publicly available, realistic flow-based data sets which are also properly labelled by proposing an approach for their generation. First, characteristics of good data sets (for example, the presence of normal user behaviour) are defined, and a promising approach for their generation is presented. The main idea of this approach is to rebuild company networks in a test environment. Normally, test environments are isolated and are used to evaluate new software, network configurations, and security breaches. Consequently, traditional test environments do not contain typical user behaviour like writing emails or browsing the web. In order to overcome this shortcoming, the proposed approach emulates normal user behaviour by executing scripts on the clients, which follow self-defined guidelines to simulate realistic behaviour. In contrast to operational environments, it is possible to render malware and other attack scenarios in this test environment. Similar to the emulation of normal user behaviour, attack scripts generate malicious network activities on predetermined clients. Further, the approach uses external servers which are exposed to real and up-to-date threats from the Internet. Since both normal and malicious activities are executed by scripts with logging mechanisms, the captured flow-based traffic can be properly labelled. Finally, this test environment is used to create the flow-based CIDDS-002 port scan data set. For this data set, a small company (consisting of various servers and clients) is simulated in the test environment, and the generated network traffic is captured in unidirectional NetFlow format.

The main contributions of this article are the detailed description of an approach for generating reliable intrusion-detection data sets and the generation of the flow-based CIDDS-002 port scan data set.

The remainder of this article is organised as follows: available, network-based evaluation data sets and related work on traffic generators are discussed in the next section. Then, requirements for good evaluation data sets are defined, and the proposed data-generation approach is explained in more detail. Next, the CIDDS-002 port scan data set, which is generated following the outlined simulation approach, is investigated, thus underlining its feasibility. The final section provides a summary.

## **Related Work**

This section provides an overview of publicly available, network-based evaluation data sets and traffic generators. Generally, intrusion detection data sets may be classified as network-based, host-based, and application-based. Since the proposed approach is network-based, the following review considers only network-based data sets.

The MIT Lincoln Laboratory's DARPA98 and DARPA99 data sets are the best known intrusion-detection data sets. Both data sets captured network traffic from a simulated environment in packet-based format. The widely used KDD CUP 99 data set is a modified version of the DARPA98 data set. The data points of the KDD CUP 99 data set contain 41 attributes and four distinct types of attacks. However, this data set suffers from several problems, such as the huge



number of redundant data points. To overcome these problems, Tavallaee *et al.* (2009) published a revised version of this data set (called NSL-KDD). Yet the mentioned data sets were created more than 15 years ago, and it is questionable whether they still reflect up-to-date scenarios of normal and malicious network traffic. The MAWI repository (Fontugne *et al.* 2010) provides recent, packet-based data by capturing network traffic from an Internet backbone. They use a taxonomy and combine various anomaly detectors to provide additional labels. However, the accuracy of these labels is questionable as manual supervision is missing. Recently, Beer *et al.* (2017) presented the NDSec-1 data set which should serve as an attack repository. This data set is available in packet- and flow-based formats. It contains almost exclusively malicious network traffic. The intention of Beer *et al.* (2017) is that users combine this data set with real network traffic to obtain intrusion-detection data sets.

Sperotto *et al.* (2009) presented one of the first flow-based data sets. The authors collected network traffic from a honeypot that offered several services and then analysed the log files to label the corresponding flow-based data. Nevertheless, since normal user behaviour is missing, most of the captured flows are malicious. CTU-13 malware (García *et al.* 2014) is another flow-based data set which contains normal and malicious network traffic. Overall, García *et al.* (2014) executed 13 different malware scenarios and labelled the data based on the source IP address of infected hosts. Recently, Moustafa and Slay (2015) published the UNSW-NB15 data set, which was created within a synthetic environment. The data set is labelled and contains normal network traffic as well as nine distinct types of attacks. However, both data sets (García *et al.* 2014 and Moustafa and Slay 2015) do not change after their creation and are limited in terms of the number and kind of attacks. Therefore, these data sets are called ‘static’ in the following analysis.

Besides these network-based data sets, there are several publicly available traffic generators. MACE (Sommers, Yegneswaran & Barford 2004) is a packet-based traffic generator for malicious network traffic. This framework provides a test environment for generating malicious network traffic caused by worms and other kinds of attacks. Similarly, FLAME (Brauckhoff, Wagner & May 2008) is a traffic generator that uses real network traffic as input and combines it with synthetically created malicious network traffic. The available implementation offers the inclusion of attacks such as Denial of Service (DoS) attacks. Vasilomanolakis *et al.* (2016) follow a similar approach. Their generator, ID2T, uses real network traffic as input data and combines it with malicious network traffic. The authors create malicious network traffic by using predefined scripts or by manipulating the input data.

Shiravi *et al.* (2012) present a more sophisticated approach. The authors have developed a systematic approach to generate labelled data sets for IDSs. Therefore, various profiles which describe normal user activities as well as attack scenarios are used to generate network traffic within a test environment. Otto *et al.* (2016) offer another approach for the generation of network traffic with normal user behaviour. The authors built a test environment where real users worked on the clients within this environment to cause normal user activities.

The proposed approach of this article does not synthetically create malicious network traffic as do Brauckhoff, Wagner, and May (2008) and Vasilomanolakis *et al.* (2016). Instead, it captures network traffic from a test environment with normal and malicious activities in a similar fashion as García *et al.* (2014) and Moustafa and Slay (2015) do. However, García *et al.* and Moustafa and

Slay provide static data sets while network behaviour changes over time. Similar to Shiravi *et al.* (2012), this work presents an approach for dynamic data-set generation. In contrast to Shiravi *et al.*, the proposed approach also integrates external servers which are exposed to real and up-to-date threats from the Internet and comprises an extensive labelling process for providing additional information.

## **Requirements of Good Data Sets**

The objective of this work is the presentation of an approach for generating good Intrusion Detection (ID) data sets. Hence, this section identifies requirements that a data set has to fulfil to be considered good.

- **Dynamic generation.** The best-known data sets (DARPA98 and DARPA99) were recorded over 15 years ago. Their fate exemplifies the major problem of static data sets. It is inevitable that a static data set will, at some point, no longer represent recent behaviour of network traffic. Hence, an approach allowing the continuous generation of new data sets reflecting current trends in user behaviour is necessary.
- **Real data.** Network-based data sets should be captured within network environments rather than simulated by models. The large number of influence factors (response times of servers, possible bottlenecks of Internet connections, and other kinds of noise) makes it difficult to simulate realistic network traffic through closed models. Also, only network traffic captured in the wild can contain potentially unknown attacks which are also needed for comprehensive evaluation. Thus, a good data set should contain captured network traffic from a real or a test environment.
- **Network topologies.** It is important to consider different deployment scenarios of an IDS. Networks of small and medium-sized companies are fundamentally different from networks of large-scale enterprises. Furthermore, business environments contain a variety of clients with different Operating Systems (OSs). While some business networks may exist that consist of Windows machines exclusively, most networks encompass at least a few Linux servers or actually a mix of Windows, Linux, Mac OS, and Android devices, each with special behaviour and each susceptible to different types of attacks. For a good data set, it is important to consider different network topologies with respect to the chosen deployment scenario of the IDS to be evaluated.
- **Normal user behaviour.** As previously mentioned, honeypot data sets consist almost exclusively of malicious network traffic. However, a good data set should also include normal user behaviour since most network traffic within a company is normal and the task of an IDS is to identify the malicious activities within the huge amount of network traffic.
- **Problem specific data.** Another requirement is the presence of anticipated attack scenarios. For example, if a port scan detection algorithm should be evaluated, the evaluation data set should primarily include scanning activities as malicious behaviour. Hence, a good data set needs to be adjusted to the main objective of the algorithm to be evaluated.

- **Labels.** Since content-related interpretation of network traffic is difficult for third parties, good data sets must be properly labelled and should provide additional information about the machines (IP addresses) within the data set. Furthermore, since many anomaly-based ID methods rely on data-mining approaches, the data sets are used for training and evaluation. Proper training of data-mining methods can only be achieved with precisely labelled data sets.
- **Public.** A data set should be able to serve as a basis for comparing different algorithms. This criterion can only be achieved by making the corresponding data set publicly available such that other researchers are able to evaluate the quality of the data set and test their algorithms on the same data set.

To summarise, a good data set should meet several requirements; namely it should

1. be recent and up-to-date,
2. contain real network traffic,
3. consider the network topologies,
4. contain normal user behaviour,
5. contain the desired attack scenarios,
6. be labelled, and
7. be publicly available.

## Data-Generation Approach

This section presents the proposed data-generation approach in more detail. It starts with an overview and outlines the underlying ideas. Then, the generation of normal and malicious network traffic is described in more detail. Finally, labelling and anonymisation of captured flow-based data are explained.

## Overview and underlying ideas

The objective of this work is the generation of labelled flow-based data sets which fulfil the requirements identified in the previous section. The proposed approach uses the software platform OpenStack. OpenStack allows the creation of virtual environments with virtual networks, virtual machines, and virtual network devices. Using a virtual environment as a test environment offers great advantages with regard to the generation of labelled ID data sets. First, a virtual environment offers full control over the network. For example, firewall rules can be configured to allow desired test scenarios. Such test scenarios are likely to be important for reconstructing and comprehending the increasing trend of insider attacks since attackers from inside the network do not have to overcome security mechanisms, such as firewalls (Ring *et al.* 2017a). Another major advantage of generating data sets within a virtual environment is the continuous generation of data sets with the opportunity of regular adjustments. This way, new attacks or new trends in user behaviour can be included easily to constantly generate current and up-to-date data sets, thus satisfying requirement 1, above. Also, new ideas for improving the quality can be easily integrated and tested. The proposed approach captures the generated network traffic at the virtual network devices in unidirectional NetFlow format (requirement 2). Generally, it would also be possible to configure OpenStack to capture the network traffic in other flow-based formats or in packet-based format. Further, different network topologies can be easily set up, thus satisfying requirement 3.

Randomised and parameterised Python scripts emulate a variety of network activities on the clients and can be adapted to specific scenarios, thus satisfying requirement 4. These scripts follow some guidelines as described in the next subsection. For generating malicious network traffic, different types of attacks can be executed within the virtual network (see requirement 5). To make the generated data even more realistic, this approach integrates external servers which are exposed to real and up-to-date threats from the Internet. Other than a honeypot, which only captures malicious network traffic, these servers are correctly used by the clients from the OpenStack environment. Consequently, normal and malicious network traffic is captured from the network cards of external servers.

### **Generation of normal user behaviour**

Normal user behaviour is emulated by executing Python scripts on the clients. These scripts follow two self-defined guidelines to ensure realistic user emulation. The first guideline takes the heterogeneity of OSs into account; the second guideline focuses on realistic emulation of user behaviour. The first guideline entails using the platform-independent language Python. Consequently, the user-emulation scripts can be used to cause normal user activities on different OSs such as Windows or Linux.

Meeting the second guideline is more challenging. First of all, the scripts have to deal with typical computerised activities of employees. Employees conduct a wide range of activities as part of their daily work, such as writing emails, creating documents and presentations, browsing (personal or business-related), printing, sharing files, and so on. For emulating such activities with respect to potential different characteristics of different employees, each client has an individual configuration file. The configuration file controls user activities and their frequency for each client. Thus, different user profiles may be assigned to different clients. For transferring files and printing documents, it is important to ensure that the corresponding files vary in terms of types and sizes. Further, when sending emails, the number of attachments should change. Additionally, realistic user behaviour must be free of period repetitions. Therefore, the scripts do not periodically execute a list of predefined activities. Instead, the user scripts use randomised periodic temporal sequences for executing user activities. However, these activities should not be totally random; they should follow a probability distribution based on typical work hours. Usually, employees are not continuously performing tasks which cause network traffic. Therefore, the scripts also contain offline activities such as meetings, offline work, or coffee breaks. Further, the scripts emphasise work hours and stop activities in breaks and in the evening as well as on weekends.

Following these guidelines results in data sets with a good approximation of realistic user behaviour. Since the configuration file is modular, new restrictions and ideas may be easily integrated.

The monitoring of clients is a crucial factor to consider when generating user behaviour with scripts. Since the scripts use several packages to execute the different user activities, errors may occur during runtime. Therefore, a monitoring dashboard was developed which is illustrated in **Figure 1**, below.



Client	Hours	Workdays							Browsing	Mailing	...	Breaks	Sum
		M	T	W	T	F	S	S					
dev-deb-1 (220.42)	9-20	x	x	x	x	x	x	-	19% 2017-08-10 10:49:43	1%	...	5%	80.56h 3.36d
dev-deb-2 (220.43)	7-16	x	x	x	x	x	-	-	23%	11%	...	15%	80.56h 3.36d
off-deb-1 (210.45)	9-20	-	x	x	x	x	-	-	7%	3% 2017-08-08 18:39:11	...	7%	66.25h 2.76d

Figure 1: Illustration of the monitoring dashboard

The monitoring dashboard is divided into a left and a right area (see **Figure 1**). The left area gives information about the configuration file of each client, such as client name, work hours, or workdays. The right area displays information about the states and duration of activities executed by this client. If clients execute no activities for a predefined period of time, the complete row is highlighted with diagonal background lines (see client ‘dev-deb-2’ in **Figure 1**). If a single activity fails, the affected cell is highlighted with diagonal background lines (see ‘Browsing’ activity in **Figure 1**). If an activity is successfully executed after it failed in a previous attempt, the background of the cell becomes dotted (see ‘Mailing’ activity in **Figure 1**).

### Generation of malicious traffic

A comprehensive ID data set consists of both normal and malicious network traffic. Therefore, the proposed approach uses three different methods for the generation of malicious network traffic. One method is that a user takes control of a virtual machine in the test environment and executes attacks manually.

The second method uses attack scripts from the CIDDS repository (Ring *et al.* 2017b). A flag within the client configuration file can tag a client as an attacker resulting in execution of additional attacking scripts parallel to the normal user behaviour scripts. One advantage of this method is that all attacks are automatically logged for later labelling processes. Currently, the list of attacks includes port scans, SSH Brute Force, and DoS attacks.

The third method for inclusion of malicious network traffic is the use of external servers. These external servers should be directly deployed in the Internet. As a consequence, the external servers are exposed to real and up-to-date threats from the Internet. In addition to that, these servers must offer services which are correctly used by the clients from the test environment in order to also record normal user behaviour. This approach was used for the CIDDS-001 data set (Ring *et al.* 2017c)

### Labelling

Labelling the captured flow-based data is an indispensable step (see requirement 6). Therefore, the proposed approach includes an extensive labelling process. All clients log their user activities (including attacks) in a predefined format. These log files are used in a four-step labelling process. The first label attribute is called class and classifies flows into five categories: ‘normal’, ‘attacker’,

‘victim’, ‘suspicious’, or ‘unknown’. All flows caused by attack scripts from the CIDDs repository or by manually executed attacks are labelled as ‘attacker’ or ‘victim’. Depending on the source IP address of the flow, the flow is labelled as ‘attacker’ if the source IP address is the origin of the attack or as ‘victim’ otherwise. The second label attribute is called `attackType` and provides the type of attack (for example, DoS), if the label attribute class contains the value ‘attacker’ or ‘victim’. A third label attribute called `attackID` assigns a unique identifier to all flows that belong to the same attack. The fourth label attribute is called `attackDescription` and gives more information about the attack (for example, the number of established connections for a DoS attack). This labelling process allows for a more detailed analysis. Besides these labels, additional user activity files for each host provide information about the user activities that have caused the network traffic.

The labelling process for the captured network traffic within the OpenStack environment differs from network traffic captured at network cards from external servers. Network traffic captured within the OpenStack environment can easily be labelled. Since origins, targets, and timestamps of executed attacks are known, attack traffic can be easily identified and assigned with the corresponding labels (‘attacker’ or ‘victim’). The remaining traffic is labelled as ‘normal’.

Labelling network traffic of external servers is more time consuming. All clients from the OpenStack environment communicate with the same public IP address to the external servers. This traffic can be labelled as ‘normal’ traffic, since the emulated clients should not attack the external servers. Further, additional machines that are directly connected to the Internet can be used to explicitly attack the external servers, as described in Ring *et al.* (2017c). For these machines, IP addresses and timestamps can be logged and used to label the corresponding flow-based network traffic with ‘attacker’ or ‘victim’. However, for the remaining traffic only unequivocal labels can be assigned. Here, the following two rules should be used: (1) Network traffic to services for public users should be labelled as ‘unknown’ since it is not clear if the request is ‘normal’ or an ‘attack’. (2) All other requested services on the external servers should be labelled as ‘suspicious’ since these services are not offered for public users.

## **Anonymisation**

For privacy reasons, all public IP addresses are anonymised according to the following approach: the IP address of the DNS server is replaced with ‘DNS’. For all other public IP addresses, the first three bytes of each IP address are replaced with a randomly generated number. The anonymisation process ensures that the same IP address is always mapped to the same generated number. This allows anonymisation of public IP addresses while preserving information about subnets. For example, possible transformations could include the following:

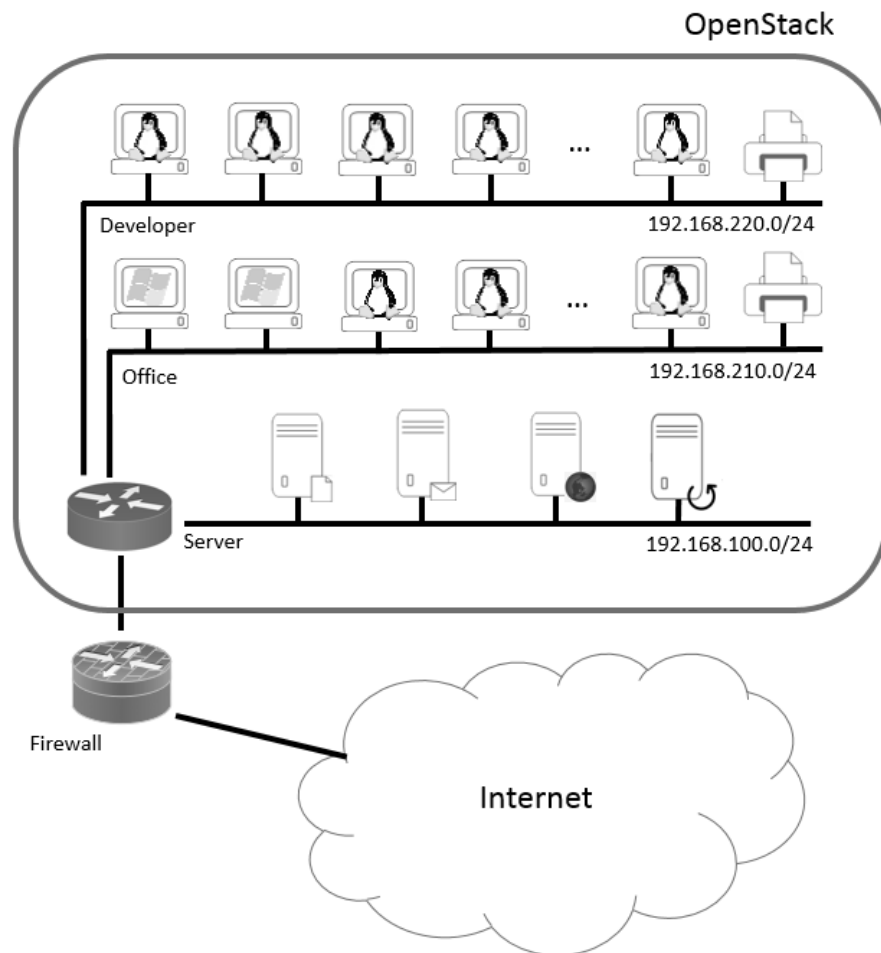
- 8.102.3.251 to 4711\_25
- 8.102.3.233 to 4711\_233
- 6.204.34.23 to 2342\_23
- 201.133.175.87 to 9721\_87

## Generation of the CIDDS-002 Data Set

A second major contribution of this work is the generation of the CIDDS-002 port scan data set, which is analysed in more detail below. This data set will be made publicly available to the community (Ring *et al.* 2017d).

## Testbed network architecture

For generating the CIDDS-002 port scan data set, a small company was rebuilt in the OpenStack test environment. **Figure 2**, below, provides an overview of the network architecture.



**Figure 2:** Testbed network architecture for the generation of the CIDDS-002 port scan data set

The network architecture of the emulated company includes four servers, two printers, and 23 clients. As can be seen in **Figure 2**, the emulated business environment contains three subnets reflecting organisational structure. The ‘Server’ subnet (192.168.100.0/24) contains the internal servers (file, email, web, and backup) while the other two subnets (‘Developer’ and ‘Office’) represent the departments of the company. The ‘Developer’ subnet (192.168.220.0/24) includes eleven Linux clients and one printer. The ‘Office’ subnet (192.168.210.0/24) consists of four Windows clients, eight Linux clients, and one printer. Port scans are executed by a client from the

‘Developer’ subnet. In contrast to the CIDDS-001 data set (Ring *et al.* 2017c), this setup does not include external servers that are directly deployed in the Internet, since the goal is the creation of a data set consisting exclusively of normal user behaviour and port scans.

### Analysis of the captured data

The above described testbed network architecture was used to capture two weeks of network traffic in unidirectional NetFlow format. **Table 1**, below, provides an overview regarding the labels of the captured flows.

	Week1	Week2	Overall
<b>Total number of flows</b>	<b>8,185,992</b>	<b>7,975,191</b>	<b>16,161,183</b>
Subset of normal flows	7,919,622	7,678,921	15,598,543
Subset of attacker flows	162,688	189,065	351,753
Subset of victim flows	103,682	107,205	210,887

**Table 1:** Overview of the captured flows within the CIDDS-002 data set

The resulting data set consists of two parts: ‘Week1’ and ‘Week2’. **Table 1** shows the number of flows and their ‘class’ label distribution. About 16 million flows were captured, from which around 15.6 million flows were normal with only a small portion of flows being labelled as ‘attacker’ or ‘victim’. Further, both weeks contain a similar number of flows. **Table 2**, below, shows the number of the executed port scans within the CIDDS-002 scan data set.

The Nmap tool was used to perform all port scans within the CIDDS-002 data set. **Table 2** shows that 20 port scans were executed in ‘Week1’ and 23 port scans in ‘Week2’. Further, the CIDDS-002 data set contains five different types of port scans: SYN Scans, ACK Scans, UDP Scans, FIN Scans, and Ping Scans. The parameter ‘T’ (see **Table 2**) controls the timing of the port scan. Generally, higher values of ‘T’ indicate faster port scans. For example, a port scan with parameter T=0 sends a probe packet every five minutes whereas a port scan with parameter T=1 sends a probe packet every 15 seconds (Lyon 2008).

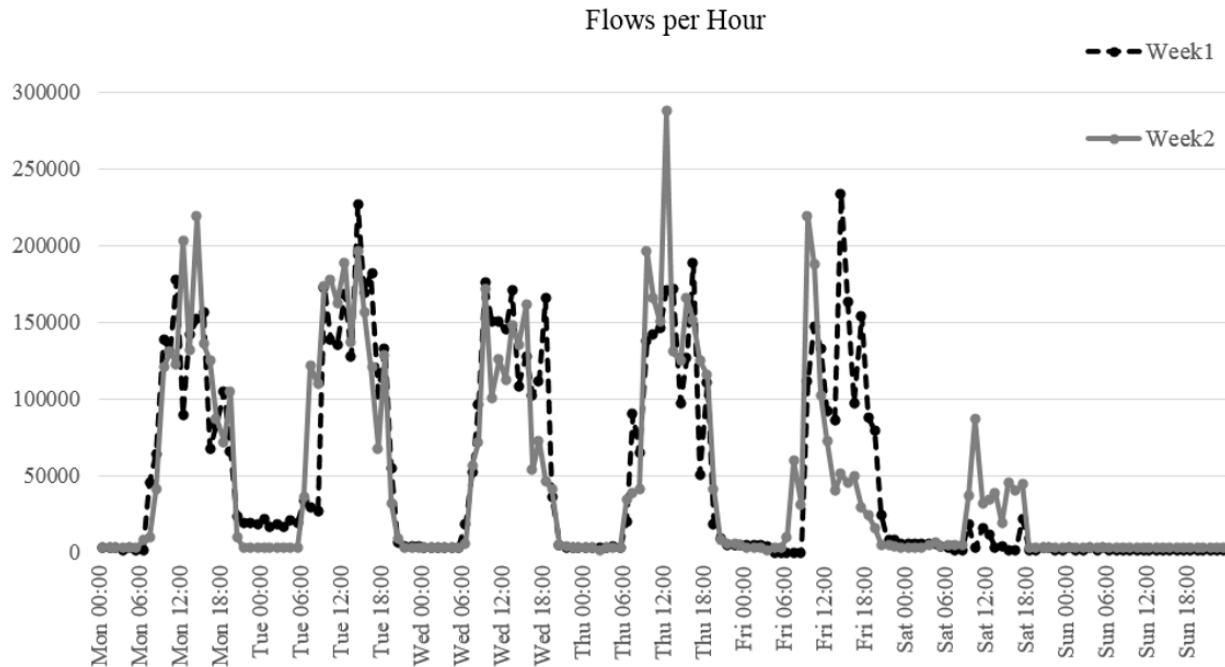


	Week1	Week2	Overall
SYN Scan -T 1	2	3	5
SYN Scan -T 2	1	2	3
SYN Scan -T 3	1	0	1
ACK Scan -T 1	2	1	3
ACK Scan -T 2	1	0	1
ACK Scan -T 3	2	1	3
UDP Scan -T 0	1	0	1
UDP Scan -T 1	2	1	3
UDP Scan -T 2	2	3	5
UDP Scan -T 3	0	1	1
FIN Scan -T 1	0	3	3
FIN Scan -T 2	2	3	5
FIN Scan -T 3	3	2	5
Ping Scan -T1	0	2	2
Ping Scan -T2	1	1	2
<b>Sum</b>	<b>20</b>	<b>23</b>	<b>43</b>

Note: The first column indicates the different types of executed port scans. The parameter 'T' controls the timing of the port scan.

**Table 2:** Number of executed port scans within CIDDS-002 data set

**Figure 3**, below, illustrates the temporal sequence of captured flow-based network traffic. Each line represents a week of network traffic, while the y axis indicates the number of flows per hour. Typical work hours can be easily recognised in **Figure 3**. Work days, such as Mondays, exhibit an increase of flows around 06:00 when the first employees start their work. Then, the number of flows decreases slightly at lunch time (around 12:00) and rises again one hour later. Between 16:00 and 19:00, when most employees leave work, the volume of network flows decreases. Further, typical work days (Monday to Friday) exhibit a greater number of network flows than on Saturdays and Sundays. The nightly backup of the servers causes only a small number of flows which is not recognisable in **Figure 3**. During non-work hours, an equal distribution of flows can be observed, which is primarily caused by the clients' integrated network drives and other default requests. The higher volume of flows in the night from Monday to Tuesday in 'Week1' is caused by port scans and browsing activities from a client. Besides that, a power failure was simulated in the test environment such that there are no flows available from Friday 03:52 to Friday 09:08 for 'Week1'.



**Figure 3:** Temporal sequence of the captured flow-based network traffic; the y axis counts the flows per hour, and each week is displayed as a line of different colour and structure

## Summary and Future Work

New anomaly-based ID methods have to be evaluated and compared against existing methods. For this task, the use of labelled data sets is a good approach. Existing data sets were analysed and several requirements of good data sets were identified. The main contribution of this work is the presentation of a novel approach for generating data sets which considers the identified requirements. The proposed approach was implemented in OpenStack, a software platform used to create virtual environments, which offers advantages compared to static data sets by easily generating new data sets. The main benefit of this approach is its ability to generate realistic data sets containing both normal and malicious traffic in a proper proportion that are flexible enough to be adapted easily to changing user behaviour and new types of attacks.

This approach was used to create the CIDDS-002 port scan data set. In particular, a small company was emulated with typical servers and several clients. Python scripts were used to emulate typical user activities on the clients. These scripts follow some self-defined guidelines to ensure high qualitative simulation of the user behaviour. For generation of malicious network traffic, one client used the tool Nmap to execute several port scans. The network traffic was captured in unidirectional NetFlow format and the flows were labelled during a four-stage labelling process. While the approach was only used to create a flow-based data set in this work, it is general enough to create packet-based or even host-based data sets as well.

In the future, additional services, such as repository servers and more sophisticated attack scenarios, should be integrated into this approach. Further, the quality of normal user behaviour should be improved by analysing distributions of real network traffic for even more realistic parameterisation of the user emulation scripts.

## Acknowledgement

This work is supported by the Bavarian Ministry of Economic Affairs through the WISENT project (Grant no. IUK 452/002). This paper is a revised and extended version of a contribution presented at the 16th European Conference on Cyber Warfare and Security, Dublin, Ireland, in June 2017. This source is cited in the References' section as: "Ring, M, Wunderlich, S, Grödl, D, Landes, D & Hotho, A 2017c".

## References

- Beer, F, Hofer, T, Karimi, D & Bühler, U 2017, 'A new attack composition for network security', *Proceedings of the 10th DFN-Forum Communication Technology, Gesellschaft für Informatik*, pp. 11-20.
- Brauckhoff, D, Wagner, A & May, M 2008, 'FLAME: A flow-level anomaly modeling engine', *Proceedings of the 3rd International Conference on Cyber Security Experimentation and Test (CSET)*, USENIX Association, pp. 1-6.
- Buczak, AL & Guven, E 2016, 'A survey of data mining and machine learning methods for cyber security intrusion detection', *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153-176.
- Fontugne, R, Borgnat, P, Abry, P & Fukuda, K 2010, 'MAWILab: Combining diverse anomaly detectors for automated anomaly labelling and performance benchmarking', *Proceedings of the 6th International Conference on Emerging Networking Experiments and Technology (Co-Next)*, ACM, pp. 8:1-8:12.
- García, S, Grill, M, Stiborek, J & Zunino, A 2014, 'An empirical comparison of botnet detection methods', *Computers & Security*, vol. 45, pp. 100-23.
- Lyon, G 2008, *Nmap network scanning: The official Nmap project guide to network discovery and security scanning*, Nmap Project, Insecure.com LLC, Sunnyvale, CA, U.S.A.
- Moustafa, N & Slay, J 2015, 'UNSW-NB15: A comprehensive data set for network intrusion detection systems', *Proceedings of the 2015th Conference on Military Communications and Information Systems (MilCIS)*, IEEE, pp. 1-6.
- Otto, F, Ring, M, Landes, D & Hotho, A 2016, 'Creation of specific flow-based training data sets for usage behaviour classification', *Proceedings of the 15th European Conference on Cyber Warfare and Security (ECCWS)*, ACPI, pp. 437-40.
- Ring, M, Wunderlich, S, Grödl, D, Landes, D & Hotho, A 2017a, 'A toolset for intrusion and insider threat detection', *Data Analytics and Decision Support for Cybersecurity*, eds. I Palomares, H Kalutarage & Y Huang, Springer International Publishing, Cham, Switzerland, pp. 3-31.
- 2017b, *Generation scripts for the Coburg Intrusion Detection Data Sets (CIDDS)*, Digital Bond, viewed 18 August 2017, <<https://github.com/markusring/CIDDS>>.

——2017c, ‘Flow-based benchmark data sets for intrusion detection’, *Proceedings of the 16th European Conference on Cyber Warfare and Security (ECCWS)*, ACPI, pp. 361-9.

——2017d, ‘Coburg Intrusion Detection Data Sets (CIDDS)’, Digital Bond, viewed 18 August 2017, <<https://www.hs-coburg.de/ciddds>>.

Shiravi, A, Shiravi, H, Tavallae, M & Ghorbani, AA 2012, ‘Toward developing a systematic approach to generate benchmark datasets for intrusion detection’, *Computers & Security*, vol. 31, no. 3, pp 357-74.

Sommer, R & Paxson, V 2010, ‘Outside the closed world: On using machine learning for network intrusion detection’, *Proceedings of the 2010th IEEE Symposium on Security and Privacy (SP)*, IEEE, pp. 305-16.

Sommers, J, Yegneswaran, V & Barford, P 2004, ‘A framework for malicious workload generation’, *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement (IMC)*, ACM, pp. 82-7.

Sperotto, A, Sadre, R, Van Vliet, F & Pras, A 2009, ‘A labelled data set for flow-based intrusion detection’, *Proceedings of the 9th IEEE International Workshop on IP Operations and Management (IPOM)*, IEEE, pp 39-50.

Tavallae, M, Bagheri, E, Lu, W & Ghorbani, AA, 2009, ‘A detailed analysis of the KDD CUP 99 data set’, *Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications*, IEEE, pp. 1-6.

Vasilomanolakis, E, Cordero, G, Milanov, N & Mühlhäuser, M 2016, ‘Towards the creation of synthetic, yet realistic, intrusion detection datasets’, *IEEE Symposium on Network Operations and Management Symposium (NOMS)*, IEEE, pp 1209-14.