SIDAR

SPRING 2016

Darmstadt, Germany

Proceedings of the
11th SPRING graduate workshop
of the special interest group
Security – Intrusion Detection and Response (SIDAR)
of the German informatics Society (GI)

SPRING 2016 was organized by

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

CASED

FLAMINGO
Project

Volume Editor:
Jessica Steinberger
University of Applied Sciences Darmstadt
Haardtring 100
D-64295 Darmstadt, Germany
E-mail: jessica.steinberger@cased.de

Published by:
Special interest group Security - Intrusion Detection and Response (SIDAR)
German Informatics Society (GI)
Wissenschaftszentrum
Ahrstr. 45
D-53175 Bonn

ISSN: 2190-846X

Cite article:
<Author>."<Title>". In *Proceedings of the 11th graduate workshop of the special interest group Security - Intrusion Detection and Response (SIDAR) of the German Informatics Society (GI) (SPRING 2016)* J. Steinberger, Eds, GI SIG SIDAR, Bonn, June, 2016, pp. xx-xx. [Online]. Available: `http://www.gi-fg-sidar.de/sidar-reports.html`, ISSN 2190-846X

Preface

SPRING 2016, 11th edition of the SPRING series, is a single-track event that was sponsored by the special interest group Security – Intrusion Detection and Response (SIDAR) of the German Informatics Society (GI). The purpose of SPRING is to provide young researchers the opportunity to discuss their work with other students and specialists in the research area of IT security. In particular, SPRING is a venue for presentation of early-stage research and solicits submission of scientific papers presenting novel research on malware analysis, intrusion detection, and related systems security topics. As per our tradition, SPRING encourages submissions from the following broad areas: Analysis of vulnerabilities, intrusion detection, malware, incident management and forensics.

This year the SPRING 2016 graduate workshop was held in Darmstadt, Germany, and was hosted at the University of Applied Sciences. SPRING took place from the 2nd to the 3rd of June 2016 and was the eleventh edition of the graduate workshop on IT security. It followed the successful events in Neubiberg in 2015, Bochum in 2014, Munich in 2013, Berlin in 2012, Bochum in 2011, Bonn in 2010, Stuttgart in 2009, Mannheim in 2008, Dortmund in 2007 and Berlin in 2006.

SPRING 2016 was organized in a 2-day program to encourage interactions between all participants. The program consists of a main track and opening research keynotes.

The presented volume includes all extended abstracts presented at SPRING 2016 as defined within the overall final program.

The editor would like to thank the many people who helped to make SPRING 2016 a successful event. Firstly, many thanks are extended to all authors who submitted their contributions to SPRING 2016 and to the keynote speakers Rüdiger Gad and Aiko Pras. Thanks are also addressed to Saed Alavi, Benjamin Kuhnert and Daniel Fischer for helping in organization and handling all the logistics and hosting the SPRING 2016 event. Additionally, special thanks go to the SPRING 2016 supporters, University of Applied Sciences Darmstadt, CASED, University of Twente and European FP7 Flamingo (ICT-318488).

June 2016 Jessica Steinberger
Darmstadt, Germany

Workshop Program

Day 1

Keynote 1

Session 1

Day 2

Keynote 2

Session 2

Keynote 1

CEP & Experiences after PhD

Rüdiger Gad

Abstract. This keynote focuses on Event-Driven Architectures (EDA), Complex Event Processing (CEP) in context of network traffic analysis. This includes the correlation of events and the derivation of complex events. Further, Rüdiger Gad reports about his "after PhD" experiences and describes his work as a senior software engineer at Terma GmbH in the field of space ground systems.

Fraud Detection in Voice over Internet Protocol Telephony

Sandra Kübler and Anton Wiens
University of Applied Sciences Darmstadt
D-64295 Darmstadt, Germany
{sandra.kuebler, anton.wiens}@h-da.de

According to the Communications Fraud Control Association (CFCA), losses of more than 38 billion USD worldwide are caused by telecommunication fraud in 2015 [CFCA15]. In most cases, expensive calls to overseas destinations are made by fraudsters after having gained illegal access to telecommunication devices. Such devices are, e.g., Voice over Internet Protocol (VoIP) phones or private branch exchange (PBX) systems connected to the internet. Fraudulent attacks can pose existencial damage to telecommunication providers, especially small and medium-sized enterprises. Most commonly, *fraud detection systems* are used to detect and react to fraud. There exist different approaches and challenges in the realm of fraud detection. At the latest when devising a fraud detection system, some nagging questions arise: how well does the system perform? Are all fraudulent calls detected? Are calls mistakenly labeled as fraud (false positives)? These questions are answered, at least as good as possible, through the help of a *detection rate*. In order to have profound results concerning a detection rate, the quality of the underlying data set is essential. The underlying analysis to detect fraud can be either *offline*, *online* or a combination of both. An offline analysis performs on collected data, in most cases on call detail record (CDR) data (text file containing parameters of a phone call like caller, callee, duration of the call; mostly used for billing purposes) and extracts features, for instance, mean duration and mean number of calls. Online analysis is performed on live datastreams, extracting features as well, but it is possible to react almost immediately to suspicious activities in contrast to offline analysis and algorithms. Especially concerning offline analysis, labeled data sets are needed. As these are rare, anomaly detection methods are used. [AD2009] The challenges lie in defining “normal” and “abnormal” behavior. Offline analysis needs training data which is classified as “normal”. Using online analysis requires self-learning algorithms without training data.

In this talk, we will address the challenge of having substantial data, as well as the problems which arise when using a fraud detection system ”in real life”. Moreover, common approaches to fraud detection and own personal experiences in devising a fraud detection system will be presented.

References

[CFCA15] Communications Fraud Control Association: *2015 Global Fraud Loss Survey*, October 2015. Available from `http://www.cfca.org/pdf/survey/2015_CFCA_Global_Fraud_Loss_Survey_Press_Release.pdf`

[AD2009] Varun Chandola, Arindam Banerjee and Vipin Kumar: *Anomaly Detection: A Survey.* In *ACM Computing Surveys (CSUR)*, Volume 41, Issue 3, July 2009, Article no. 15. Available from `http://delivery.acm.org/10.1145/1550000/1541882/a15-chandola.pdf`

Monitoring of the DNS Infrastructure for Proactive Botnet Detection

Christian Dietz[*][†], Anna Sperotto[†], Gabi Dreo[*] and Aiko Pras[†]

[*] Universität der Bundeswehr München
85577 Neubiberg, Germany
{Christian.Dietz, Gabi.Dreo}@unibw-muenchen.de

[†] University of Twente
7522 NB Enschede, Niederlande
{C.Dietz, A.Sperotto, A.Pras}@utwente.nl

1 Introduction

Botnets enable many cyber-criminal activities, such as DDoS attacks, banking fraud and cyber-espionage. Botmasters use various techniques to create, maintain and hide their complex C&C infrastructures. First, they use P2P techniques and domain fast-flux to increase the resilience against take-down actions. Second, botnets encrypt their communication payload to prevent signature based detection. However, botnets often use the domain name system (DNS), e.g., to find peers and register malicious domains. Since, botmasters manage a large distributed overlay network, but have limited personal resources, they tend to automate domain registration, e.g. using domain name generation algorithms (DGAs). Such automatically generated domains share similarities and appear to be registered in close temporal distance. Such characteristics can be used for bot detection, while their deployment is still in preparation. Hence, the goal of this research is early detection of botnets to facilitate proactive mitigation strategies. Using such a proactive approach prevents botnets from evolving their full size and attack power. As many end users are unable to detect and clean infected machines, we favour a provider-based approach, involving ISPs and DNS registrars. This approach benefits from its overview of the network that allows to discover behavioural similarities of different connected systems. The benefit of tackling distributed large-scale attacks at provider level has been discussed and demonstrated in previous studies by others. Further, initiatives to incentive ISPs centred botnet mitigation are already ongoing. Previous research already addressed the domain registration behaviour of spammers and demonstrated DGA based malware detection. In contrast, our approach includes the detection of malicious DNS registration behaviour, which we currently analyse for the .com, .net and .org top level domains. These domains represent half of the registered Internet domains. By combining DNS registration behaviour analysis with passive monitoring of DNS requests and IP flows, we are able to tackle botnets throughout their whole life-cycle.

2 Research Problem & Questions

The goal of this research is to enable early botnet detection in provider environments. Therefore, our approach is based on large-scale DNS registration behaviour analysis, as this will allow to discover botnet activity in the (pre-)deployment phase of its life-cycle. Thus, our novel approach can prevent the botnet from becoming deployed and actively used. Furthermore, the proposed approach takes into account the dynamics of botnet malware and the Internet infrastructure, high data rates, incompleteness of data and encrypted bot communication. In order to tackle the early botnet detection problem, we ask the following questions: (i) How do botnets interact with the domain name system? (ii) Can domain registration characteristics be used for botnet detection, and if yes, how?

3 Approach

The goal of this research is to allow faster botnet detection and mitigation. Current approaches are usually limited to detect bots after they already became active or while they are used in attacks. Our approach targets botnet detection in the pre-deployment phase. Therefore, our approach is based on two components: (1) passive monitoring of communication characteristics and (2) DNS registration behaviour analysis. DNS registration analysis allows to detect the preparatory actions of deployment of the C&C infrastructure and the bots. Therefore, our approach allows botnet early detection and consequently facilitates proactive botnet mitigation. In addition, our approach allows botnet detection in the subsequent phases of the bot life-cycle (preparation, infection, peer discovery, malware update, command propagation and attack) by using passive DNS and flow monitoring solutions. Figure 1 provides an overview of our novel approach.

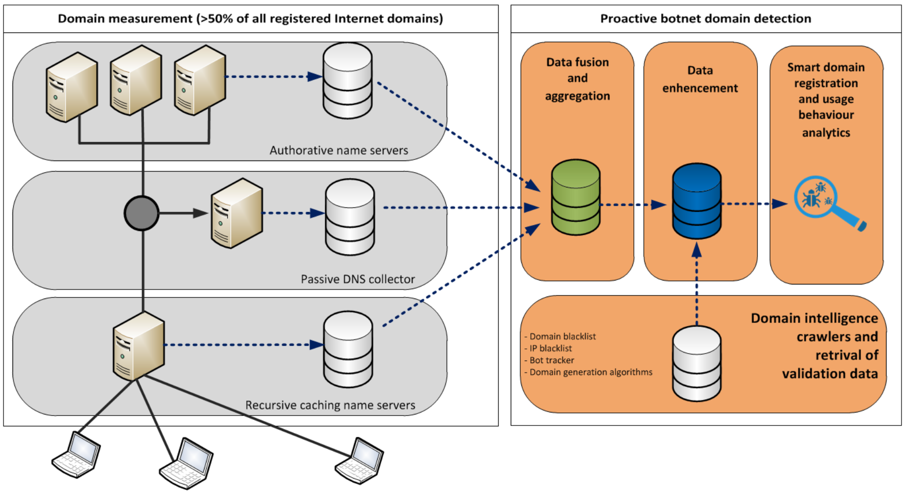

Domain measurement (>50% of all registered Internet domains)
Authorative name servers
Passive DNS collector
Recursive caching name servers
Proactive botnet domain detection
Data fusion and aggregation
Data enhencement
Smart domain registration and usage behaviour analytics
- Domain blacklist
- IP blacklist
- Bot tracker
- Domain generation algorithms
Domain intelligence crawlers and retrival of validation data

Figure 1: Components of the passive measurement and smart analytics infrastucture.

Research question (i) aims to get insight into the deployment and management of botnets. Therefore, we collect DNS registration data on a daily basis for the *.com*, *.net* and *.net* domains, representing half of the domains registered on the Internet. Second, we query different botnet tracking services and use DGAs to find botnet related records in the domain registration dataset. Research question (ii) aims to extract characteristics of botnets in their deployment phase to allow an early detection and mitigation. To answer this question, we use registration databases of top level domain registrars. Currently, our study involves the *.com, .net, and .org* top level domains. We will validate our novel approach based on simulations and real-live environments. Further, we compile different datasets. First, we crawl the registration database of multiple top level domains, different botnet domain and IP blocklists with time stamps. This allows us to measure the temporal difference between botnet deployment and detection. Second, we passively capture IP flow data and DNS requests in multiple provider networks to evaluate (a) how accurate our approach can detect the large-scale similarities between distributed bots and (b) determine the temporal delay between malicious domain registration and the first activity. This evaluation also uses IP address and DNS blocklists that our crawlers collect on a regular basis.

Acknowledgments This work is supported by the EU FP7 Flamingo Project (ICT-318488).

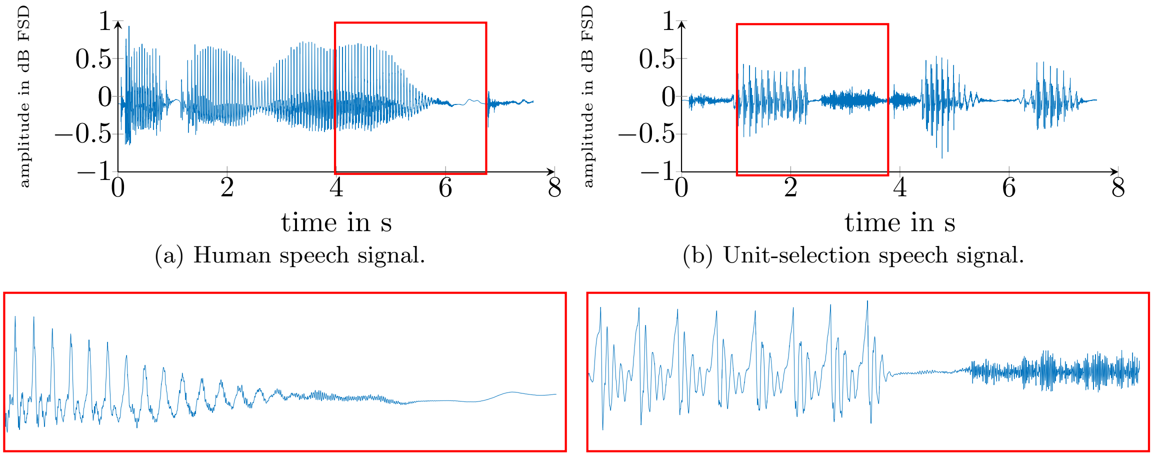
Unit-Selection Attack Detection Based on Unfiltered Frequency-Domain Features

Ulrich Scherhag, Andreas Nautsch, Christoph Busch

da/sec – Biometrics and Internet Security Research Group, Hochschule Darmstadt
D-64283 Darmstadt, Germany
{ulrich.scherhag, andreas.nautsch, christian.rathgeb, christoph.busch}@h-da.de

1 Introduction

Modern text-to-speech algorithms pose a vital threat to the security of speaker identification and verification (SIV) systems, in terms of subversive usage, i.e. generating presentation attacks. Voice presentation attacks are categorized in six attack types [1]: synthesis, voice conversion, mock-up, replay, unit-selection and mimicry. For unit-selection attacks, speech samples of the attacked subject are captured, segmented into parts, called units, and replayed in different sequence to the SIV system. In order to distinguish between presentation attacks and bona fide authentication attempts, presentation attack detection (PAD) subsystems are of utmost importance.
Most common features that analyze frequencies, such as MFCCs of CFCCs, aim at emulating the perception of humans. However, the human hearing is rather specialized for speech recognition, thus state-of-the-art presentation attack countermeasures are capable of yielding significantly better PAD performances compared to human observers [2]. Therefore, we utilize the complete frequency band without further filter-bank processing in order to detect non-smooth transitions in the full and high frequency domain caused by unit-selection attacks. In our frequency-domain analysis of unit-selection attacks, speech is interpreted as a concatenation of phonemes or likewise sound units, where concatenation points are referred to as transitions. In bona fide (human) speech, the phonemes are smoothly transferred into each other. The continuous transition of a bona fide speech signal is depicted in figure 1a.



(a) Human speech signal.

(b) Unit-selection speech signal.

(c) Close-up of transition in human speech signal.

(d) Close-up of transition in unit-selection speech signal.

Figure 1: Example of a bona fide/attack signal and transition.

Audio-signals which are compound of multiple voice fragments (phonemes, or other units) and not smoothed afterwards, show more abrupt changes of the frequency in the signal, as displayed in figure 1b. As the sudden changes in time domain, caused by non-natural transition, yield higher amplitudes for higher frequencies in frequency domain, a Fourier-based feature vector is motivated. The resulting vector of the Fourier transform represents the amplitude as natural value a and phase as imaginary values bi. For the purpose of compatibility with machine learning algorithms, the magnitude, $|a + bi|$ of the signal is calculated as: $|a + bi| = \sqrt{a^2 + b^2}$.
The Discrete Wavelet Transform (DWT) can be understood as bandpass filter decomposing the signal in iterative steps. Earlier iterations provide higher frequencies bands, later iterations lower. Assuming the discriminativity of higher frequencies, a feature vector extracting the fifth detail level is examined. This choice was elaborated based on experimental results employing 10 343 bona fide and 10 461 attack samples. In order to cover multiple frequency bands establishing more discriminative robustness, the proposed DTW feature comprises information fused from third to fifth iteration. As the DWT represents a bandpass filter, the dimension of the result depends on the length of the analyzed signal. In order to obtain features with a fix dimension, a Fourier transformation is applied.

Table 1: Best configurations evaluated with evaluation set and ASVspoof.

Feature	**Comparator**	**EER Eval-set**	**EER ASVspoof**
DWT-fusion+FFT	SVM	7.1%	11.7%
	GMM	15.0%	24.6%
FFT	SVM	8.5%	22.6%
	GMM	9.5%	27.7%
DWT-5+FFT	SVM	27.0%	11.7%
	GMM	40.1%	45.7%
CFCCIF [3]	GMM-UBM	–	8.5%

The proposed features are able to detect unit-selection attacks with an EER of 7.1% on the GSDC and 11.7% on the ASVspoof unit-selection attacks. Compared to the algorithms proposed at ASVspoof 2015, e.g. [3], the introduced features DWT-5+FFT (SVM) and DWT-fusion+FFT (SVM) yields competitive results with EERs of 11.7%, as depicted in table 1, operating on comparatively low computational costs. Our proposed features space and classifiers represent a contrastive PAD system, knowing the unit-selection attack scheme to face, which is unknown for the countermeasures presented in [3]. However, our analysis comprise data shifts in terms of capture environments, the experimental set-up, and the examined language.

References

[1] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, "Spoofing and countermeasures for speaker verification: A survey," *Speech Communication*, vol. 66, pp. 130–153, feb 2015.

[2] M. Wester, Z. Wu, and J. Yamagishi, "Human vs Machine Spoofing Detection on Wideband and Narrowband Data," in *Proc. 16th Annual Conference of the International Speech Communication Association*, 2015, pp. 2047–2051.

[3] T. B. Patel and H. A. Patil, "Combining Evidences from Mel Cepstral, Cochlear Filter Cepstral and Instantaneous Frequency Features for Detection of Natural vs. Spoofed Speech," in *Proc. 16th Annual Conference of the International Speech Communication Association*, 2015, pp. 2062–2066.

Simulation and Detection of Network Breaches

Rafael Uetz
Fraunhofer FKIE
53177 Bonn, Germany
rafael.uetz{at}fkie.fraunhofer.de

The last few years have seen a large number of publicly perceived network breaches. Even mainstream media reported on several massive incidents, affecting e.g. Sony Pictures Entertainment, the US Office of Personnel Management, and the German Bundestag. According to Verizon's Data Breach Investigations Report [1] most reported breaches go undetected for weeks or months after the initial compromise, and many of them are discovered not by the affected organization but by external parties like business partners or law enforcement. Looking at these facts, it appears likely that a large number of breaches are never detected at all. This is especially true for cases of cyber espionage or intellectual property theft where attackers obviously attempt to stay under the radar (as opposed to e.g. theft of money, blackmailing, or publication of exfiltrated data).

Why is it so hard for organizations to detect major breaches of their network? From a defender's point of view, it is very costly to detect evidence of breaches in the first place and to distinguish it from everyday minor security incidents due to a number of reasons, some of them being:

- Most malware today exists in numerous, quickly changing mutations and is thus rarely detected by signature-based anti-virus products [1].
- Even highly sophisticated attackers have repeatedly used commodity malware either to conceal their identity or simply because it sufficed to attain their goals.
- Anti-malware products with dynamic analysis capabilities (i.e., sandboxes) are becoming less effective due to an increasing prevalence of "anti-anti-malware" techniques (e.g., intentional server-side delays).
- Threat Intelligence feeds like IP or URL blacklists were shown to be rather ineffective [1] due to quickly changing threat indicators (attackers can easily change their IP addresses, URLs, file names, e-mail addresses, etc.).

Altogether, it is very difficult to detect breaches from typical low-level events sources like the ones mentioned above. The situation gets worse when considering the sheer number of security-related events in a company network – typical SIEM (Security Information and Event Management) deployments process several thousand events per second, sources being e.g. intrusion detection systems, anti-virus software, operating system logs, and firewall events. It is obviously not possible to sift all these events manually, nor are manually created correlation rules capable of detecting breaches satisfyingly. On the other hand, according to Verizon [2] for most reported breaches there was actually good evidence contained in existing log files. Unfortunately, this information usually got figured out in forensic investigations following a breach that was initially discovered by other means.

Up to now, there hasn't been much academic research on breach detection from heterogeneous event logs. We believe that one reason for that is a lack of suitable data for evaluating breach detection methods. We are therefore currently developing a breach simulation framework as well as novel breach detection methods.

Our simulation framework consists of a small number of virtual machines and a console for controlling and monitoring the simulation from the host. The virtual machines represent machines of a typical small company network as well as an external attacker. They are partitioned into three network zones, namely an internal network, a DMZ and the (simulated and real) internet. In order to create a realistic “noise floor” of legitimate events the clients in the network simulate typical employee activities. They surf the web, receive and send e-mails, and create, modify and delete documents.

To simulate breaches, we researched attack steps often seen in the real world. We then created an *attack taxonomy* from these attack steps consisting of 25 *attack modules*, divided into eight *attack phases*. Attack modules can be combined in different orders to create so-called Intrusion Kill Chains [3]. We employ a directed graph to formalize all possible attack sequences. Each vertex represents an attack module; edges denote possible next steps after an attack module was executed. This graph can be used to pseudo-randomly generate a very large number of attack sequences. Additionally, parameters like the delay between consecutive attack steps can be set. The resulting *attack scenarios* can be written to a *scenario script* so that simulation runs are both reproducible and cover an adequate fraction of the attack space in order to be significant for an evaluation. Events triggered by the attacks and the user activities are collected and normalized on a log server within the simulated company network using open source tools.

We believe that it is necessary to correlate events of multiple sources in order to recognize the high-level *tactics, techniques and procedures* of an attacker instead of detecting and displaying only low-level events (compare the Pyramid of Pain [4]). We are therefore investigating methods that specialize in the detection of typical attack step sequences as well as in the recognition of illegitimate data exfiltration from company networks. Our simulation framework serves as a first stage in the evaluation of these methods as it allows for reproducible experiments in a controlled environment. Subsequent evaluation stages will then incorporate event logs collected from real company networks in order to assess the real-world value of the methods.

References

[1] Verizon: *2015 Data Breach Investigations Report*, 2015. Online: `http://www.verizonenterprise.com/resources/reports/rp_data-breach-investigation-report-2015_en_xg.pdf`, received 13.4.2016.

[2] Verizon: *2011 Data Breach Investigations Report*, 2011. Online: `http://www.verizonenterprise.com/resources/reports/rp_data-breach-investigations-report-2011_en_xg.pdf`, received 13.4.2016.

[3] Eric M. Hutchins, Michael J. Cloppert and Rohan M. Amin: *Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains*, in *Leading Issues in Information Warfare & Security Research*, 2011.

[4] David Bianco: *The Pyramid of Pain [Update]*, 2014. Online: `http://detect-respond.blogspot.de/2013/03/the-pyramid-of-pain.html`, received 13.4.2016.

A Traffic Merging and Generation Framework for Realistic Synthesis of Network Traffic

Cem Gündogan, Sandro Passarelli, Peter Hillmann, Christian Dietz, and Lars Stiemert

Universität der Bundeswehr München
85579 Neubiberg, Germany
{Cem.Guendogan, Sandro.Passarelli, Peter.Hillmann, Christian.Dietz, Lars.Stiemert}
@unibw-muenchen.de

Introduction:
The Internet is steadily growing and is of increasing importance for our economy and society. Due to this increased importance it is also in the focus of attacks, e.g. distributed denial of service (DDoS) attacks. As attackers dynamically change their attack behavior, novel detection approaches that are able to automatically adjust to these dynamic attacks are needed. To train and test such network anomaly detection systems, it is necessary to provide realistic data. As of today, this area of research suffers from the lack of publicly available datasets that can be used to train and test anomaly detection systems and are exchangeable to allow reproducible research. Therefore, we propose a novel framework that enables researchers and developers to generate customizable synthetic datasets. It not only allows to generate fully-synthetic network traffic, but also to generate semi-synthetic network traffic by merging of multiple network captures from real-live environments. Further, it allows the mapping of IP addresses as well as the modification of other header fields, if desired. This enables researchers and developers to exchange network traces from sensitive environments without revealing any sensitive end-user related information, while perceiving the relevant characteristics of the network(s) and attack(s). In the following, we provide a description of, the problem, our concept and the features of our solution, the architecture and functional model and finally provide a short summary together with an outlook for future work.

Problem:
Testing of IDS and IPS often suffers from the lack of publicly available ground truth datasets that are derived from real-life environments. Such ground truth data includes labels for each sample, based on which one can analyse the accuracy of a detection system. Due to privacy constraints and the overhead to derive sufficient ground truth datasets, such datasets are usually either shared under non disclosure agreements or not at all. Thus, many researchers create or use syntehtic datasets to make their research reproducible, even though it is known that synthetic data can easily lead to false conclusions.

Concept and Functionality:
Our framework provides a novel approach, by combining the benefits of real live data captures and synthetic data generation. It provides multiple ways to generate and manipulate network traffic captures, one of which is simple, random based generation. Users are able to specify the generation process with parameters, i.e. source and destination IP-addresses and ports, amount and types of headers and payload. Additionally, it is possible to produce, e.g., uncompleted TCP handshakes which allows to perform packet manipulations across datasets based on user defined distributions. As Figure 1 shows, users can also merge random generated and live recorded traffic to ensure an

even better, more realistic outcome. The merging will automatically keep any given uncompleted handshakes in order, private addresses will be mapped and masked with a user specified super IP.

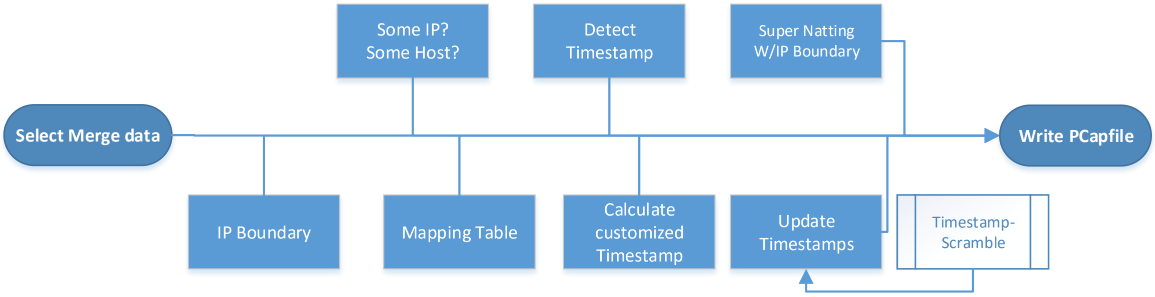


Figure 1: *PCap merging*

Model and Implementation:

First, to generate network traffic with user defined parameters, it is important to represent the first rudiments. Secondly, the framework provides an application programming interface to read and generate network traffic by using the open source java library jNetPcap. Figure 2 describes the different options to dump a PCap. The first attempt uses PCap-Templates by replacing or adding more information, like which protocol is used for packet switching. Currently, our reference implementation supports the UDP- and TCP-protocol. When the user decided to select TCP and to give a quantity of TCP packets with completed and uncompleted TCP 3-Way-Handshakes, this framework can order these PCap packets on his own certain handshake order. A distribution function handles the network data dissemination across the final output file. To build up a PCap it is essential to check the headers and proofing the validation by the right checksum as last step before dumping PCaps.

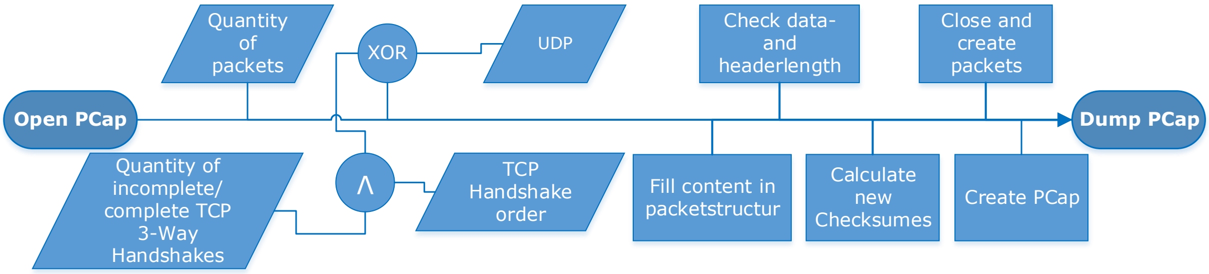


Figure 2: *Creating PCap*

Conclusion and Future Work

The whole framework is planned to be open sourced and available for the community. Features like a build in traffic transmission to instantly test any IDS and IPS are included. The validation of our approach is still in its initial state and planned to be extended in future work. After a successful validation we plan to implement the support of more different network traffic content by adding more protocols. Furthermore, we connect the databases of IDS and IPS with signatures of attacks and use this information to include packets based on attack patterns into the generated Pcap.

This work is supported by the EU FP7 Flamingo Project (ICT-318488).

Proceedings of the 11th graduate workshop SPRING of the special interest group Security - Intrusion Detection and Response (SIDAR) of the German Informatics Society (GI) (SPRING 2016), Darmstadt, Germany, June 02-03, 2016

Data-Analytics Engine for Security Monitoring

Jonathan Arnault[*] [†], Jérôme François[†], and Isabelle Chrisment[*]

[*] University of Lorraine
34 Cours Léopold
54000 Nancy, France

[†]Inria Nancy - Grand Est
615 Rue du Jardin botanique
54600, Viller-lès-Nancy, France

1 Context

The huge growth of the Internet and the large deployment of new devices over the internet represent a tremendous playground for attackers. Furthermore, attacks become more distributed and complex including Advanced Persistent Threats (APT) which are composed of various attack steps during a long period of time. Many techniques have been proposed to detect and prevent attacks. However, these techniques, including traffic monitoring or network analysis, are helpful, but not sufficient because they focus on a restricted set of common attacks making them impracticable for fighting complex attacks, which cannot be detected by a single type of detector. Therefore fighting such threats requires to collect, analyze and correlate heterogeneous sources of data (not only well-formatted logs, but also unstructured data) by applying multiple data processing techniques like aggregation, machine learning, visualization... This makes data-analytics technologies good candidates for enhancing these techniques by allowing to collect and analyze multiple sources of relevant data. Yet, existing approaches are based on few ones individually or are limited to simple correlations when relying on numerous sources of data.

2 Our approach

Nowadays, data-analytics in the context of security is a vast topic of research. In [1, 2, 3], the authors propose big data architectures dedicated to security monitoring. However, these methods rely on specific software stacks, for example, only Hadoop is used in [2]. Furthermore, the usage of data-analytics requires an additional effort to adapt existing security functions. To overcome such a difficulty, our goal is to propose a security engine with a high level of abstraction, which can automatically instanciate an abstracted security workflow to available resources and without assuming specific analytics services (a piece or a stack of software) a priori, like a traffic probe or a Hadoop stack.

The security workflow is assumed to be an input of our engine. More specifically, this workflow is composed of multiple functions. Each of them has to be described by the operations to be realized (e.g. algorithms), the associated data and data models to be used as input. Expected performances can also be given to guide our engine in making the right recommendation decision. Indeed, one primary goal of our engine is to select candidate services, including data-analytics ones, to execute the workflow.

Once services have been selected, the engine is also in charge of deploying them using proper available resources. Hence, those have to be described and can be of various types: machines (physical or virtual), network devices, cloud services... Once the services are deployed, the security engine orchestrates them and is able to use available resources to make the system scale by allocating new instances of security functions.

3 Challenges and methodology

Two major challenges have to be adressed by our engine: (1), the automated mapping between the security workflows and considered services, especially data-analytics (Map-Reduce Framework, NoSQL databases...); (2), the deployement and the orchestration of the security functions using selected services and available resources.

The engine can be seen as a recommendation system as it suggests services depending on security functions requirements and performance expectations. To achieve such recommendations, we will abstract security functions as signatures being compositions of operations. Once they are abstracted, each individual operation will be benchmarked a priori against services to predict the performances of the various compositions on demand. An underlying challenge is to keep the description of functions relevant to our context while being as smallest as possible.

Relying on the recommendation results, services are selected and the engine orchestrates and monitors them. Orchestration faces two issues: first, it must maximize the use of highly recommended services while considering the constraints on available resources. Indeed, such constraints could avoid the usage of ideal services, thus, the instanciation must take them into account. To optimize the performances of the security functions, we will rely on advances about VM placement and resource allocation [4, 5]. Secondly, instanciation and deployement of the functions based on the recommendations face the fact that some functions may require to be implemented within a specific service stack. To resolve this, manual adjustments can be performed; or algorithms can be described in a Domain Specific Language (DSL) to be compiled automatically to various programming languages. We could rely on work such as Data-Flow Graphs [6] to express security workflow and functions. We will validate our approach using existing algorithms such as NetFlow based approaches [7].

References

[1] Cisco: Open Security Operations Center (2014), `http://opensoc.github.io/`

[2] Lee, Y., Lee, Y.: Toward Scalable Internet Traffic Measurement and Analysis with Hadoop. SIGCOMM Comput. Commun. Rev. 43(1), 5–13 (2012)

[3] Marchal, S., Jiang, X., State, R., Engel, T.: A Big Data Architecture for Large Scale Security Monitoring. In: Proceedings of the 2014 IEEE International Congress on Big Data. (BIGDATACONGRESS), IEEE, Washington, DC, USA (2014)

[4] Palanisamy, B., Singh, A., Liu, L., Jain, B.: Purlieus: Locality-aware Resource Allocation for MapReduce in a Cloud. In: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis. pp. 58:1–58:11. SC '11, ACM, New York, NY, USA (2011)

[5] Warneke, D., Kao, O.: Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud. IEEE Trans. Parallel Distrib. Syst. 22(6), 985–997 (2011)

[6] Tran, N., Skhiri, S., Lesuisse, A., Zimanyi, E.: AROM: Processing big data with Data Flow Graphs and functional programming. In: 2012 IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom) (2012)

[7] Sperotto, A., Schaffrath, G., Sadre, R., Morariu, C., Pras, A., Stiller, B.: An Overview of IP Flow-Based Intrusion Detection. Communications Surveys Tutorials, IEEE 12(3), 343–356 (2010)

Keynote 2

How to write a paper - A reviewer perspective

Aiko Pras

Abstract. This keynote focuses on how to write a scientific paper and get it accepted at a journal or conference. In particular, the keynote gives practical tips before start writing and provides information about the writing style itself. Within the talk, the characteristics of a good manuscript and the basic structure of a scientific paper are presented. To overcome paper rejects, common mistakes of submitted papers are presented and discussed. In addition, this talk provides insight into the review and editorial process of a scientific paper.

Whom do we trust - Booters and SSL/TLS certificates

Jessica Steinberger[*][†], Benjamin Kuhnert[*], Saed Alavi[*], José Jair Santanna[†], Anna Sperotto[†], Harald Baier[*] and Aiko Pras[†]

[*] da/sec - Biometrics and Internet Security Research Group
University of Applied Sciences Darmstadt
Darmstadt, Germany
Email:{Jessica.Steinberger, Benjamin.Kuhnert, Saed.Alavi, Harald.Baier}@crisp-da.de

[†] Design and Analysis of Communication Systems (DACS)
University of Twente
Enschede, The Netherlands
Email:{J.Steinberger, J.J.Santanna, A.Sperotto, A.Pras}@utwente.nl

Nowadays, DDoS attacks still remain the top cause of network and service outages. The reason is that these attacks are getting more sophisticated and frequent whereas the required technial skills to perform these attack are not required anymore [**JS15**]. Currently, DDoS attacks are offered as a service, namely Booters, for less than 10 US dollars [**JS16**]. As Booters offer a service that a customer is required to pay for, Booters make use of SSL/TLS certificates. The use of SSL/TLS certificates is used to ensure secure credit card transactions, data transfer and logins.

In this talk, we present the early-stage results of the analysis of the used certificate chains of Booter websites. In particular, we present the common used certificate chains, the used cryptography and cipher suites, protocol use within SSL/TLS for purpose of security parameters negotiation, the issuer and the validity of the certificate. Our analysis revealed that there is a tyical certificate chain used by Booter websites. In our future work, we investigate if the SSL/TLS certificates and their certificate chains could be used to mitigate DDoS attacks performed by Booter websites.

References

[JS15] J.J. Santanna, R. van Rijswijk-Deij, R. Hofstede, A. Sperotto, M. Wierbosch, L. Zambenedetti Granville, and A. Pras. Booters - an analysis of DDoS-as-a-service attacks. In *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, pages 243–251, May 2015. DOI: 10.1109/INM.2015.7140298.

[JS16] J. Steinberger, J.J. Santanna, E. Spatharas, H. Amler, N. Breuer, K. Graul, B. Kuhnert, U. Piontek, A. Sperotto, H. Baier and A. Pras. "Ludo" - kids playing Distributed Denial of Service In *Proceedings of TERENA Networking Conference (TNC16), Prague (Czech Republic), June 2016, to appear.*

Adaptable Rules for Security Event Prioritization

Leonard Renners†

†Hochschule Hannover
D-30459 Hannover
leonard.renners@hs-hannover.de

Companies and organizations of all sizes are exposed to multiple security threats, ranging from random attacks to targeted threats. Intrusion detection and prevention systems (IDS/IPS) as well as advanced security systems, such as security information and event management systems (SIEM) are often applied to counter these tendencies. They are used to continuously monitor the network and its participants behavior to detect misuse and unwanted behavior.

Many attack scenarios comprise additional steps after an initial security breach, for example stealing data after gaining access to the internal network or escalating privileges. Therefore, near real-time capabilities are a major focus in research and development regarding these security systems. Nevertheless, the response of the administrator is a part that also has to be accounted for. It is necessary to evaluate the generated events, interpret the reports and choose appropriate counter measures. This should preferably also be done in a timely manner to prevent further loss of confidential and valuable information or further harm due to spreading to other systems and privileges.

This becomes a major task, due to the fact that most security mechanisms create potentially huge amounts of alerts. Particularly since many companies do not have the financial possibilities to employ security administrators for a 24/7 surveillance and response. Hence, incidents queue up and lead to the problem of choosing the *right* alert to handle first. It is quite obvious that a first come first serve principle is rather naive and can result in a delayed response to important and time-critical alerts.

We are arguing that beside a general reduction of the alerts, as tackled by alert verification and correlation techniques, the administrator needs sophisticated tools to reasonably and meaningfully make these choices, i.e. be presented with a comprehensible and modifiable arrangement of occurring alerts. Consequently, incidents need be evaluated with regard to the environment and context - by the means of a prioritization.

First approaches have been presented tackling this issue, proposing contextualized IDS frameworks or defining event prioritization calculation rules. We believe that some requirements are necessary to employ such a system in a useful way and are not yet fully satisfied.
(a) First, the rules need to be understandable in order to correctly reflect the company policies and enable the security administrator to verify the rules and react on incorrect evaluation.
(b) Second and consequently, the administrator and thus the rules need to be able to explicitly express relationships and dependencies between alerts, information and the evaluation results.
(c) The user/administrator should be supported in creating and maintaining security policies.
The last point is particularly important in small and medium enterprises, where the aforementioned 24/7 security administration is rarely implemented and budget for security aspects is a delicate topic.

In our research, we target these requirements and their implications on a rule and knowledge system. We are mainly focusing on three interdependent areas:

1. Rule representation to explicitly model concepts, conditions and weights
2. Structured background knowledge and definition of more abstract concepts as a kind of pre-processing
3. (Semi-)Automated approaches for learning and recommending concepts, rules and adaptations

The first part aims at a novel and specific rule representation to particularly target the needs of prioritization rules. More precisely the possibilities to distinguish between the used background knowledge, included conditions and the weights with regard to different rating parameters.

The second area comprises ideas from ontologies or attribute-oriented induction. The main idea is to abstract from singular values and conditions towards generalized and reusable concepts. E.g. the membership relationship of singular ip addresses in particular subnets or generalizing specific timestamps to working hours. These abstractions lead to more general rules and allow for an easier understanding and adaptation.

Lastly, we want to investigate how concepts from the machine learning and data mining domain can be applied to (semi) automatically induce the aforementioned parts.
On the one hand we want to propose recommendations regarding a meaningful structuring of the attributes by the means of grouping attribute values with regard to the rating process. On the other hand we want to develop a system to automatically induce rules from already rated or at least sorted events. That is, to learn which attributes and concepts the analyst payed most attention to and generate rules from this basis.

Both learning mechanisms should also be applied in a continuous manner in order to cope with environmental changes, i.e. in the network infrastructure, new and adapted attacks and adaptations of the company policies.

The Security Attack Experimentation Framework: An approach to test network mitigation strategies in compliance.

Benjamin Kuhnert*, Jessica Steinberger*†, Hendrik Amler*, Niklas Breuer*, Kristian Graul*, Ulrike Piontek*, Harald Baier*

* da/sec - Biometrics and Internet Security Research Group, University of Applied Sciences Darmstadt, Darmstadt, Germany

† Design and Analysis of Communication Systems (DACS) University of Twente, Enschede, The Netherlands

Distributed attacks are one of the major threats that can cause catastrophic events to network infrastructures. By misusing fundamental network technologies, an attacker can saturate resources on networks and services by using botnets or web based services like Booters [**JS15**], which offers DDoS-as-a-service for anyone. Often UDP-based protocols like NTP (Network Time Protocol) or DNSSEC (Domain Name System Security Extensions) are being misused. Sophisticated attackers use amplification and reflection techniques to achieve high bandwidth consumption on the target up to 500 Gbps, as seen in recent attack reports [**KS16**].

System operators of centralized services are challenged to verify their mitigation strategies to prevent damage that are caused by such attacks. Moreover, running reproducible tests in order to learn the effectiveness of countermeasures can be a difficult task, since the integration of existing tools often introduces different application runtime semantics and thus, do not work as a seamless system. Running such tests on an operative network infrastructure is not an easy task, since legal restrictions and enterprise policies can restrict or even forbid the use of security penetrating tools.

We introduce a work-in-progress framework that complies to an operator's requirements and could help to understand the possible outcome of a distributed attack within a network as a controllable and reliable process: The Security attack experimentation framework (STORM) [**JS16**]. In our work we discuss about the fundamental problems in developing such a framework and how it could be integrated into existing network infrastructures.

References

[JS15] J.J. Santanna, R. van Rijswijk-Deij, R. Hofstede, A. Sperotto, M. Wierbosch, L. Zambenedetti Granville, and A. Pras. Booters - an analysis of DDoS-as-a-service attacks. In *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, pages 243–251, May 2015. DOI: 10.1109/INM.2015.7140298.

[JS16] J. Steinberger, J.J. Santanna, E. Spatharas, H. Amler, N. Breuer, K. Graul, B. Kuhnert, U. Piontek, A. Sperotto, H. Baier and A. Pras. "Ludo" - kids playing Distributed Denial of Service In *Proceedings of TERENA Networking Conference (TNC16), Prague (Czech Republic), June 2016, to appear.*

[KS16] Kaspersky Lab Kaspersky DDoS Intelligence Report for Q1 2016 `https://securelist.com/analysis/quarterly-malware-reports/74550/kaspersky-ddos-intelligence-report-for-q1-2016/`, Feb 2016. Accessed: 07.05.2016

Index of Authors