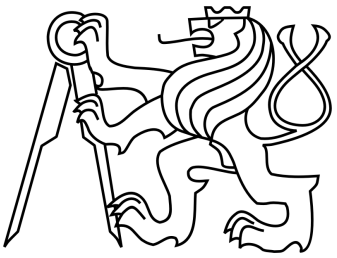
Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Cybernetics



Diploma Thesis [draft]

Network Anomaly Detection by Means of Spectral Analysis

Bc. Peter Boráros
(Supervisor: Ing. Martin Rehák, PhD.)

November 28, 2012

Abstract In the present work we study ... Uvede se anglický abstrakt v rozsahu 80 až 200 slov. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut sit amet sem. Mauris nec turpis ac sem mollis pretium. Suspendisse neque massa, suscipit id, dictum in, porta at, quam. Nunc suscipit, pede vel elementum pretium, nisl urna sodales velit, sit amet auctor elit quam id tellus. Nullam sollicitudin. Donec hendrerit. Aliquam ac nibh. Vivamus mi. Sed felis. Proin pretium elit in neque. Pellentesque at turpis. Maecenas convallis. Vestibulum id lectus.

Keywords Anomaly Detection, Spectral Analysis, Network Security

Contents

1 Introduction

1.1 Anomaly Detection

1.2 Computer Network Security Aspects

1.3 Related Work

Anomaly detection as an inportant problem has been researched within wide variety of reasearch areas and domains.

Chandola et al. [6] addressed anomaly detection in general and also identified various approaches and application domains. They described methods based on *classification*, *clustering*, *nearest neighbour*, *statistical*, *information theory* and *spectral analysis*. They covered several application domains such as *cyber-intrusion detection*, *fraud detection*, *idustrial damage detection*, *sensor networks* etc. Their contribution in respect to our work is mainly an exact definition of anomaly detection and deep, structured overview of the known techniques in various application domains. In the domain of our interest, the network intrusion detection, they depicted fact that although available data has an temporal content, known techniques typically do not exploit this aspect explicitly. The data is mostly high-dimensional with continuous as well as categorical attributes. The challenge faced by used techniques in this domain is the changing nature of anomalies as the intruders adapt to the existing intrusion detection solutions and the high dimensionality and hight ammount of the data. They showed that the existing methods in this domain are based on statistical analysis, classification, clustering, spectral and information theoretic. **Patcha** and **Park** [25] covered cyber-intrusion domain focusing on statistical, data-mining and machine learning anomaly detection techniques. They described works that are seminal for the cyber-intrusion detection and referenced number of research systems. **Davis** and **Clark** [11] focused on data preprocessing techniques for network intrusion detection. They described *dataset creation*, *feature construction* and *reduction* techniques. In this comprehensive review they grouped a related works according to the type of features and data preprocessing techniques they addressed. They identified aggregation of packets into flows as useful as it enforces contextual analysis and statistical measures to detect anomalous behavior. They notice that packet header based approaches are not sufficient as the use of defense against attacks forced attackers to use different attack vectors such as crafted application data. They suggest that there is need to use features derived from contents of packets but as there is little research in this area they expect that more results would emerge in future. **Onut** and **Ghorbani** [24] derived taxonomy of features used for anomaly detection. Futhermore they introduced anomaly network intrusion detection systems which use them. **Gogoi et. al** [14] focused on comparison of specific techniques used for network anomaly detection. They covered supervised and unsupervised approaches covering several techniques in detail, such as statistical, signal processing, graph teoretic, clustering or rule-based techniques.

X. He, et al. [16] explored an spectral analysis approach using fourier transform on features obtained from packet traces. They focused on link layer and showed that signature specific to link is observed in frequency spectrum after link is saturated. They applied the fourier transform to convert the packet arrival process to frequency domain. In addition they compared the signatures of different layers of the traffic - link layer and application layer. The work of **Chen** and **Hwang** [10], [9] used similar approach to analyze spectral characteristic of network protocols TCP, UDP and they were able to distinguish the traffic using statistical methods using features derived from freqency spectrum of the packet arrival process. In adition they introduced classification method to distinguish between legitimate and malicious TCP flows. In their work they focused on reduction-of-service (RoS) attack. The RoS attack unlike denial-of-service (DoS) attack don't attempt to completely deny the service by throttling the resources submittig a fake requests, but the attacker's focus is on reduction of the quality (e.g. prolong the response times) by using small ammount of the requests. Due to low traffic during attack, RoS attacks are hard to detect with volume-based methods.

Wright et al. [33] and **Dusi et al.** [12] investigated an detection of encrypted tunnels inside the application layer. They addressed the problem of bypassing an network-boundary security inspection by encapsulating of data subject to restrictions (peer-to-peer, chat, e-mail and others) into protocols that are considered safe and necessary (HTTP, HTTPS, SSH, DNS etc.).

Wright et al. [33] used features derived from packet headers agregating packets over protocols, and time span of arrival. They counted packets in categories during an epoch resulting in vector. Then they used k-nearest neighbor (kNN) and hidden Markov model (HMM) techniques. They constructed models for differnet kind of encrypted tunnels such as single- or multi-flow tunnels. They were able to infer application protocols even in multiplexed packet flows without need of demultiplexing.

Dusi et al. [12] brought an statistical approach to detect an tunnel inside application layer. In the paper they described diffenrent tunneling techniques and designed statistical pattern recognition classifier to identify them.

Estévez-Tapiador et al. Measuring normality in HTTP traffic for anomaly-based intrusion detection [13] In this paper, the problem of measuring normality in HTTP traffic for the purpose of anomaly-based network intrusion detection is addressed. The work carried out is expressed in two steps: first, some statistical analysis of both normal and hostile traffic is presented. The experimental results of this study reveal that certain features extracted from HTTP requests can be used to distinguish anomalous (and, therefore, suspicious) traffic from that corresponding to correct, normal connections. The second part of the paper presents a new anomaly-based approach to detect attacks carried out over HTTP traffic. The technique introduced is statistical and makes use of Markov chains to model HTTP network traffic. The incoming HTTP traffic is parameterised for evaluation on a packet payload basis. Thus, the payload of each HTTP request is segmented into a certain number of contiguous blocks, which are subsequently quantized according to a previously trained scalar codebook. Finally, the temporal sequence of the symbols obtained is evaluated by means of a Markov model derived during a training phase. The detection results provided by our approach show important improvements, both in detection ratio and regarding false alarms, in comparison with those obtained using other current techniques.

DARPA Intrusion Detection Data Sets (IDEVAL) are five weeks of packet trace data gen-

erated at *MIT Lincoln Labs* for Intrusion detection evaluation [1]. The data represent simulated traffic on fictional Air Force base. For each week there are five network trace files that represent nnetwork traffic from 8:00 to 17:00. The data is considered very inportant due to many network intrusion detection system papers used it for evaluation. **McHugh** [23] depicted an issues associated with design and execution of the IDEVAL datasets.. **Mahoney** and **Chan** [22] analyzed the IDEVAL dataset with data captured in their university server. They comapared the attributes of the packet headers and some inferred features. Furthermore they mixed the simulated data with real data and tested several anomaly detection systems. They ...

Sarasamma et al. Hierarchical Kohonenen net for anomaly detection in network security [27] A novel multilevel hierarchical Kohonen Net (K-Map) for an intrusion detection system is presented. Each level of the hierarchical map is modeled as a simple winner-take-all K-Map. One significant advantage of this multilevel hierarchical K-Map is its computational efficiency. Unlike other statistical anomaly detection methods such as nearest neighbor approach, K-means clustering or probabilistic analysis that employ distance computation in the feature space to identify the outliers, our approach does not involve costly point-to-point computation in organizing the data into clusters. Another advantage is the reduced network size. We use the classification capability of the K-Map on selected dimensions of data set in detecting anomalies. Randomly selected subsets that contain both attacks and normal records from the KDD Cup 1999 benchmark data are used to train the hierarchical net. We use a confidence measure to label the clusters. Then we use the test set from the same KDD Cup 1999 benchmark to test the hierarchical net. We show that a hierarchical K-Map in which each layer operates on a small subset of the feature space is superior to a single-layer K-Map operating on the whole feature space in detecting a variety of attacks in terms of detection rate as well as false positive rate.

Thomas et al. Usefulness of DARPA dataset for intrusion detection system evaluation [30]

McHugh et al. Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory In 1998 and again in 1999, the Lincoln Laboratory of MIT conducted a comparative evaluation of intrusion detection systems (IDSs) developed under DARPA funding. While this evaluation represents a significant and monumental undertaking, there are a number of issues associated with its design and execution that remain unsettled. Some methodologies used in the evaluation are questionable and may have biased its results. One problem is that the evaluators have published relatively little concerning some of the more critical aspects of their work, such as validation of their test data. The appropriateness of the evaluation techniques used needs further investigation. The purpose of this article is to attempt to identify the shortcomings of the Lincoln Lab effort in the hope that future efforts of this kind will be placed on a sounder footing. Some of the problems that the article points out might well be resolved if the evaluators were to publish a detailed description of their procedures and the rationale that led to their adoption, but other problems would clearly remain. [23]

Yamada et al. Intrusion Detection for Encrypted Web Accesses [35] As various services are provided as web applications, attacks against web applications constitute a serious

problem. Intrusion detection systems (IDSes) are one solution, however, these systems do not work effectively when the accesses are encrypted by protocols. Because the IDSes inspect the contents of a packet, it is difficult to find attacks by the current IDS. This paper presents a novel approach to anomaly detection for encrypted web accesses. This approach applies encrypted traffic analysis to intrusion detection, which analyzes contents of encrypted traffic using only data size and timing without decryption. First, the system extracts information from encrypted traffic, which is a set comprising data size and timing for each web client. Second, the accesses are distinguished based on similarity of the information and access frequencies are calculated. Finally, malicious activities are detected according to rules generated from the frequency of accesses and characteristics of HTTP traffic. The system does not extract private information or require enormous pre-operation beforehand, which are needed in conventional encrypted traffic analysis. We show that the system detects various attacks with a high degree of accuracy, adopting an actual dataset gathered at a gateway of a network and the DARPA dataset.

Undercoffer et al. Modeling Computer Attacks: An Ontology for Intrusion Detection [31] We state the benefits of transitioning from taxonomies to ontologies and ontology specification languages, which are able to simultaneously serve as recognition, reporting and correlation languages. We have produced an ontology specifying a model of computer attack using the DARPA Agent Markup Language+Ontology Inference Layer, a descriptive logic language. The ontology's logic is implemented using DAMLJessKB. We compare and contrast the IETF's IDMEF, an emerging standard that uses XML to define its data model, with a data model constructed using DAML+OIL. In our research we focus on low level kernel attributes at the process, system and network levels, to serve as those taxonomic characteristics. We illustrate the benefits of utilizing an ontology by presenting use case scenarios within a distributed intrusion detection system.

Wu et al. The use of computational intelligence in intrusion detection systems: A review [34] Intrusion detection based upon computational intelligence is currently attracting considerable interest from the research community. Characteristics of computational intelligence (CI) systems, such as adaptation, fault tolerance, high computational speed and error resilience in the face of noisy information, fit the requirements of building a good intrusion detection model. Here we want to provide an overview of the research progress in applying CI methods to the problem of intrusion detection. The scope of this review will encompass core methods of CI, including artificial neural networks, fuzzy systems, evolutionary computation, artificial immune systems, swarm intelligence, and soft computing. The research contributions in each field are systematically summarized and compared, allowing us to clearly define existing research challenges, and to highlight promising new research directions. The findings of this review should provide useful insights into the current IDS literature and be a good source for anyone who is interested in the application of CI approaches to IDSs or related fields.

Chebrolu et al. Feature deduction and ensemble design of intrusion detection systems [8] Current intrusion detection systems (IDS) examine all data features to detect intrusion or misuse patterns. Some of the features may be redundant or contribute little (if anything) to the detection process. The purpose of this study is to identify important input features in building an IDS that is computationally efficient and effective. We investigated the perfor-

mance of two feature selection algorithms involving Bayesian networks (BN) and Classification and Regression Trees (CART) and an ensemble of BN and CART. Empirical results indicate that significant input feature selection is important to design an IDS that is lightweight, efficient and effective for real world detection systems. Finally, we propose an hybrid architecture for combining different feature selection algorithms for real world intrusion detection.

Ingham and Ingham. Comparing anomaly detection techniques for http [18] Much data access occurs via HTTP, which is becoming a universal transport protocol. Because of this, it has become a common exploit target and several HTTP specific IDSs have been proposed as a response. However, each IDS is developed and tested independently, and direct comparisons are difficult. We describe a framework for testing IDS algorithms, and apply it to several proposed anomaly detection algorithms, testing using identical data and test environment. The results show serious limitations in all approaches, and we make predictions about requirements for successful anomaly detection approaches used to protect web servers.

Sommer and Paxson. Outside the closed world: On using machine learning for network intrusion detection [29] In network intrusion detection research, one popular strategy for finding attacks is monitoring a network's activity for anomalies: deviations from profiles of normality previously learned from benign traffic, typically identified using tools borrowed from the machine learning community. However, despite extensive academic research one finds a striking gap in terms of actual deployments of such systems: compared with other intrusion detection approaches, machine learning is rarely employed in operational "real world" settings. We examine the differences between the network intrusion detection problem and other areas where machine learning regularly finds much more success. Our main claim is that the task of finding attacks is fundamentally different from these other applications, making it significantly harder for the intrusion detection community to employ machine learning effectively. We support this claim by identifying challenges particular to network intrusion detection, and provide a set of guidelines meant to strengthen future research on anomaly detection.

Bajcsy et al. Cyber defense technology networking and evaluation [3] Creating an experimental infrastructure for developing next-generation information security technologies.

Goodall et al. Focusing on context in network traffic analysis [15] The time-based network traffic visualizer combines low-level, textual detail with multiple visualizations of the larger context to help users construct a security event's big picture. TNV is a visualization tool grounded in an under standing of the work practices of security analysts. We designed it to support ID analysis by giving analysts a visual display that facilitates pattern and anomaly recognition, particularly overtime. It also offers more focused views on packet-level detail in the context of the surrounding network traffic.

Jamdagni et al. Intrusion detection using GSAD model for HTTP traffic on web services [21] Intrusion detection systems are widely used security tools to detect cyber-attacks and malicious activities in computer systems and networks. Hypertext Transport Protocol (HTTP) is used for new applications without much interference. In this paper, we focus on

intrusion detection of HTTP traffic by applying pattern recognition techniques using our Geometrical Structure Anomaly Detection (GSAD) model. Experimental results reveal that features extracted from HTTP request using GSAD model can be used to distinguish anomalous traffic from normal traffic, and attacks carried out over HTTP traffic can be identified. We evaluate and compare our results with the results of PAYL intrusion detection systems for the test of DARPA 1999 IDS data set. The results show GSAD has high detection rates and low false positive rates.

Ingham. Anomaly detection for HTTP intrusion detection: algorithm comparisons and the effect of generalization on accuracy [19] Network servers are vulnerable to attack, and this state of affairs shows no sign of abating. Therefore security measures to protect vulnerable software is an important part of keeping systems secure. Anomaly detection systems have the potential to improve the state of affairs, because they can independently learn a model of normal behavior from a set of training data, and then use the model to detect novel attacks. In most cases, this model represents more instances than were in the training data set—such generalization is necessary for accurate anomaly detection. This dissertation describes a framework for testing anomaly detection algorithms under identical conditions. Because quality test data representative of today's web servers is not available, this dissertation also describes the Hypertext Transfer Protocol (HTTP) request data collected from four web sites to use as training and test data representing normal HTTP requests. A collection of attacks against web servers and their applications did not exist either, so prior to testing it was necessary to also build a database of HTTP attacks, the largest publicly-available one. These data were used to test nine algorithms. This testing was more rigorous than any performed previously, and it shows that the previously-proposed algorithms (character distribution, a linear combination of six measures, and a Markov Model) are not accurate enough for production use on many of the web servers in use today, and might explain the lack of their widespread adoption. Two newer algorithms (deterministic finite automaton induction and n-grams) show more promise. This dissertation shows that accurate anomaly detection requires carefully controlled generalization. Too much or too little will result inaccurate results. Calculating the growth rate of the set that describes the anomaly detector's model of normal provides a means of comparing anomaly detection algorithms and predicting their accuracy. Identification of undergeneralization locations can be automated, leading to more rapid discovery of the heuristics needed to allow an anomaly detection system to achieve the required accuracy for production use.

Zanero. Analyzing TCP traffic patterns using self organizing maps [39] The continuous evolution of the attacks against computer networks has given renewed strength to research on anomaly based Intrusion Detection Systems, capable of automatically detecting anomalous deviations in the behavior of a computer system. While data mining and learning techniques have been successfully applied in host-based intrusion detection, network-based applications are more difficult, for a variety of reasons, the first being the curse of dimensionality. We have proposed a novel architecture which implements a network-based anomaly detection system using unsupervised learning algorithms. In this paper we describe how the pattern recognition features of a Self Organizing Map algorithm can be used for Intrusion Detection purposes on the payload of TCP network packets.

Yu Chen. TCP Flow Analysis for Defense against Shrew DDoS Attacks The shrew or RoS attacks are low-rate DDoSattacks that degrade the QoS to end systems slowly but not todeny the services completely. These attacks are more difficultto detect than the flooding type of DDoS attacks. In this paper,we explore the energy distributions of Internet traffic flows infrequency domain. Normal TCP traffic flows present someform of periodicity because of TCP protocol behavior. Ourresults reveal that normal TCP flows can be segregated frommalicious flows using some energy distribution properties. Wediscover the spectral shifting of attack flows from that ofnormal flows. Combining flow-level spectral analysis withsequential hypothesis testing, we propose a novel defensescheme against shrew DDoS or RoQ (reduction-of-service)attacks. Our detection and filtering scheme can effectivelyrescue 99% legitimate TCP flows under the RoS attacks.

M. Zalewski. Silence on the Wire [38] [36] [37]

M. Iliofotou, et al. Exploiting dynamicity in graph-based traffic analysis: techniques and applications [17] Network traffic can be represented by a Traffic Dispersion Graph (TDG) that contains an edge between two nodes that send a particular type of traffic (e.g., DNS) to one another. TDGs have recently been proposed as an alternative way to interpret and visualize network traffic. Previous studies have focused on static properties of TDGs using graph snapshots in isolation. In this work, we represent network traffic with a series of related graph instances that change over time. This representation facilitates the analysis of the dynamic nature of network traffic, providing additional descriptive power. For example, DNS and P2P graph instances can appear similar when compared in isolation, but the way the DNS and P2P TDGs change over time differs significantly. To quantify the changes over time, we introduce a series of novel metrics that capture changes both in the graph structure (e.g., the average degree) and the participants (i.e., IP addresses) of a TDG. We apply our new methodologies to improve graph-based traffic classification and to detect changes in the profile of legacy applications (e.g., e-mail).

P. Smith, et al.: Network resilience: a systematic approach [28] The cost of failures within communication networks is significant and will only increase as their reach further extends into the way our society functions. Some aspects of network resilience, such as the application of fault-tolerant systems techniques to optical switching, have been studied and applied to great effect. However, networks - and the Internet in particular - are still vulnerable to malicious attacks, human mistakes such as misconfigurations, and a range of environmental challenges. We argue that this is, in part, due to a lack of a holistic view of the resilience problem, leading to inappropriate and difficult-to-manage solutions. In this article, we present a systematic approach to building resilient networked systems. We first study fundamental elements at the framework level such as metrics, policies, and information sensing mechanisms. Their understanding drives the design of a distributed multilevel architecture that lets the network defend itself against, detect, and dynamically respond to challenges. We then use a concrete case study to show how the framework and mechanisms we have developed can be applied to enhance resilience.

V. Chandola et al. Anomaly Detection for Discrete Sequences: A Survey [4] This survey attempts to provide a comprehensive and structured overview of the existing research for the problem of detecting anomalies in discrete/symbolic sequences. The objective is to provide a global understanding of the sequence anomaly detection problem and how existing techniques relate to each other. The key contribution of this survey is the classification of the existing research into three distinct categories, based on the problem formulation that they are trying to solve. These problem formulations are: 1) identifying anomalous sequences with respect to a database of normal sequences; 2) identifying an anomalous subsequence within a long sequence; and 3) identifying a pattern in a sequence whose frequency of occurrence is anomalous. We show how each of these problem formulations is characteristically distinct from each other and discuss their relevance in various application domains. We review techniques from many disparate and disconnected application domains that address each of these formulations. Within each problem formulation, we group techniques into categories based on the nature of the underlying algorithm. For each category, we provide a basic anomaly detection technique, and show how the existing techniques are variants of the basic technique. This approach shows how different techniques within a category are related or different from each other. Our categorization reveals new variants and combinations that have not been investigated before for anomaly detection. We also provide a discussion of relative strengths and weaknesses of different techniques. We show how techniques developed for one problem formulation can be adapted to solve a different formulation, thereby providing several novel adaptations to solve the different problem formulations. We also highlight the applicability of the techniques that handle discrete sequences to other related areas such as online anomaly detection and time series anomaly detection.

V. Chandola. Comparative evaluation of anomaly detection techniques for sequence data [5] We present a comparative evaluation of a large number of anomaly detection techniques on a variety of publicly available as well as artificially generated data sets. Many of these are existing techniques while some are slight variants and/or adaptations of traditional anomaly detection techniques to sequence data.

Alshammari et al. Machine learning based encrypted traffic classification: identifying SSH and skype [2] The objective of this work is to assess the robustness of machine learning based traffic classification for classifying encrypted traffic where SSH and Skype are taken as good representatives of encrypted traffic. Here what we mean by robustness is that the classifiers are trained on data from one network but tested on data from an entirely different network. To this end, five learning algorithms - adaboost, support vector machine, NaiÂ¿e Bayesian, RIPPER and C4.5 - are evaluated using flow based features, where IP addresses, source/destination ports and payload information are not employed. Results indicate the C4.5 based approach performs much better than other algorithms on the identification of both SSH and Skype traffic on totally different networks.

H. Ringberg, Sensitivity of PCA for traffic anomaly detection [26] Detecting anomalous traffic is a crucial part of managing IP networks. In recent years, network-wide anomaly detection based on Principal Component Analysis (PCA) has emerged as a powerful method for detecting a wide variety of anomalies. We show that tuning PCA to operate effectively in practice

is difficult and requires more robust techniques than have been presented thus far. We analyze a week of network-wide traffic measurements from two IP backbones (Abilene and Geant) across three different traffic aggregations (ingress routers, OD flows, and input links), and conduct a detailed inspection of the feature time series for each suspected anomaly. Our study identifies and evaluates four main challenges of using PCA to detect traffic anomalies: (i) the false positive rate is very sensitive to small differences in the number of principal components in the normal subspace, (ii) the effectiveness of PCA is sensitive to the level of aggregation of the traffic measurements, (iii) a large anomaly may in advertently pollute the normal subspace, (iv) correctly identifying which flow triggered the anomaly detector is an inherently challenging problem.

A. Chapanond. Graph Theoretic and Spectral Analysis of Enron Email Data [7]
Analysis of social networks to identify communities and model their evolution has been an active area of recent research. This paper analyzes the Enron email data set to discover structures within the organization. The analysis is based on constructing an email graph and studying its properties with both graph theoretical and spectral analysis techniques. The graph theoretical analysis includes the computation of several graph metrics such as degree distribution, average distance ratio, clustering coefficient and compactness over the email graph. The spectral analysis shows that the email adjacency matrix has a rank-2 approximation. It is shown that preprocessing of data has significant impact on the results, thus a standard form is needed for establishing a benchmark data.
[32]

1.4 Our Contribution

1.5 Organization

2 Anomaly Detection

In general an anomaly detection is the problem of finding patterns in data that do not conform to expected behavior. A term *anomaly* or *outlier* refers to these non-conforming patterns. Usually knowledge about non-conforming patterns is inportant due to fact that they may refer to singnificant information, in many cases also critical and actionable, e.g. a tumor presence may be indicated by anomalous magnetic resonance imaging (MRI) scan, network intrusion may cause anomalous signature of the packets to be observed.

The anomaly detection has been studied as early as the 19^{th} century by statisticians as a statistical method. Due now, several techniques have been developed, using domain-independent approach or developed specificaly for particular domain.

Apparently simple approach of anomaly detection is to define a region representing normal behavior and declare any patterns which does not conform to this region as anomaly. This approach is obfuscated by several factors:

- Definition of normal behavior must contain every possible normal behavior and it is difficultly achievable.
- The boundary between anomalies and normal behavior is not accurate and can introduce wrong interpretation of particular patterns laying near the boundary.
- Adaptation of malicious agents to make their outcomes appear like normal in given feature space.
- Normal behavior is evolving in time and thus an normal model defined in one time span can be inaccurate or invalid in future.
- An ammount of labeled data needed for derivation of the normal model is insuficient.
- Presence of the noise that can be similar as anomalies, and thus it can be difficult to suppress.
- Different application domains have different notion of an anomaly, thus development of domain-indepedent method is complicated.

In general the anomaly detection problem is difficult to solve. Most techniques solve a specific formulations of the problem, induced by a factors specific for a particular domain. The anomaly detection techniques itself were developed by adoption of the concepts from diverse disciplines such as *statistics*, *machine learning*, *data mining*, *information theory*, *spectral theory*.

2.1 Input Data

Input is generally a collection of data instances, referred as *pattern*, *sample* or *observation*. Each data instance is represented by non-empty set of attributes, also refered as *variable* or

feature. Attributes can be instances of different data types e.g. *continous*, *cathegorical*, or *binary*. Furthermore in case of each data instance consist of single attribute it is reffered to as *univariate* otherwise it is *multivariate*. For multivariate instances the data types of the attributes might be mixed as well as the domain of definition might be different.

2.1.1 Relationship Among Data Instances

Based on presence of the relationship in data, the input data can be further categorized as *point data*, *sequence data*, *spatial data*, and *graph data*. In point data no relationship is assumed among the instances. In sequence data, presence of the *total order relation*[1] among data instances is assumed. The sequence data can be time-series, protein sequences, etc. In *spatial data* presence of *metric*[2] is required. The metric determines an neighbourhood of each data instance. The examples of metrics are *Minkowski metric*[3] (e.g. *Euclidean* distance or *Manhattan* distance), *Levenshtein distance* (editation distance between strings of characters). Typical example of spatial data is the coordinate in geographic coordinate system or, asuming our definition, also textual data (notice that Levenshtein distance is *metric* among the strings of characters). The *graph data* instances are represented by graph structure[4]. As an example of the graph data is map of social social interactions.

In case context are mixed we refer to spatio-temporal (e.g. climate data) or graph-temporal data (computer network packet flows).

2.1.2 Data Labels

Labels associated with particular data instances denote if instance is *anomalus* or *normal*. Labeling is often done by human expert hence it is very expensive and requires huge effort. Obtaining labels for all possible normal behavior is often less difficult than obtaining labels for anomalous behavior. Moreover, anomalous behavior is dynamic so new types of the anomalies might originate. Newly formed anomalies might be then missing from models and hence might elude undetected in detection process.

2.2 Anomalies

Based on presence of the relationship between data instances and problem formulation, anomalies can be divided into *point anomalies*, *contextual anomalies* and *collective anomalies*.

[1] In set theory a *total order* is a binary relation on some set X. The relation of total order is defined by axioms of *antisymetry*, *transitivity* and *totatlity*. Total order is usually denoted as $\leq$.

[2] Metric, or distance function, is a non-negative function which defines distance or similarity between elements of the set. Metric is required to satisfy axioms of *coincidence*, *symmetry* and *triangle inequality*. A metric space is mathematical structure (X,d), where X is a set and function $d : X \times X \to \mathbf{R}$ is a metric.

[3] Minkowski metric, defined as $d(x,y) = (\sum_{i=1}^{n}(x_i - y_i)^k)^{\frac{1}{k}}$, is a distance between n-vectors x and y. By choosing value of parameter $k = 1$ we get a Mahattan or a Hamming distance, for $k = 2$ we get an Euclid distance, or for $k = \infty$ we get a Chebyshev distance.

[4] In most common sense, a *graph* G is mathematical structure $G = (V,E)$ comprising a set of vertices V with set of edges E. Edges can be two-element subsets of V (undirected graph) or ordered pairs of elements of V (directed graph). In addition if *weight function* – $w : E \to \mathbf{R}$ is defined, assigning a number (e.g. weight, price, etc.) to each edge, we call structure $G = (V,E,w)$ a *weighted graph*.

2.2.1 Point anomalies.

In the simplest case, if an individual data instance is considered as anomalous with respect to the rest of data. No information about relationship between data instances is assumed. This type of anomaly is target of most of the research studies.

2.2.2 Contextual anomalies.

In many cases, an context is present in data set. Context is induced by the structure of the data. In case a data instance is anomalous only whithin a given context, it is called *contextual anomaly*. The notion of the context has to be specified within problem formulation. By introducing the context in the data features are divided to *contextual features* and *behavioral features*.

The *contexutal features* are used to determine the context for particular data instance. As an examples of the contextual features are: a timestamp denoting temporal context in sequential data, a geographic coordinate denoting spatial context.

The *behavioral features* define non-contextual characteristics of an instance. For example, the number of arrived packets during network communication whithin a specific time span is considered as an behavioral attribute. Identical data instances (in terms of behavioral attributes) may be considered as anomalous or non-anomalous in a different contexts.

2.2.3 Collective anomalies.

If a collection of related data instances is anomalous with respect entire dataset it is called *collective anomaly*. The collective anomaly is defined only in data set where an relationship among instances are related, e.g. in sequence data, graph data or spatial data.

It is important to note that *point anomalies* can occur in any data set, while *contextual anomalies* depend on notion of the context and its definition in problem formulation, and *collective anomalies* are relevant for data where relationship among instances is defined (e.g. distance metric). So by taking in account the context information a point or collective anomaly detection problem can be converted into contextual anomaly detection problem.

2.3 Techniques

Typically, the outputs produced by anomaly detection techniques are one of the following two types:

- **Scores.** Assigning score to each data instance depending on the degree to which that instance is considered as anomaly.
- **Labels.** Techniques in this category assign a label (normal or anomalous) to each test instance.

Extent in which labels are present in data, can significantly affect mode of operation anomaly detection technique:

- **Supervised anomaly detection** sssumes availability of labeled data, for normal and also for anomaly class.

- **Semi-Supervised anomaly detection** assumes availability of data labels only for normal class.
- **Unsupervised anomaly detection** do not require training data. The unsupervised techniques are based on assumption that normal data are more frequent in data than anomalous.

2.4 Application domain

3 Computer Network Security Aspects

3.1 Techniques

3.2 Input Data

3.3 Problem Sets

4 Proposed Method

He et al. [16] showed that the different layers of the network protocols inprint distinct patterns in a frequency spectrum. Further work of *Chen* and *Hwang* [9] used the features derived from frequency spectrum to classify malicious and normal traffic. They noticed that transport protocols (transmission control protocol – TCP and user datagram protocol - UDP) have distinct frequency spectrum and pover spectral density. They exploited this property to identify low-rate attacks on TCP protocol. *Wright et al.* [33] researched methods based on hidden Markov models to classify different application protocols embeded in encrypted application layer. They developed classification method, able to classify different application protocols multiplexed in single encrypted packet flow. *Dusi et al.* [12] brought an statistical approach to detect an tunnel inside application layer. In the paper they described different tunneling techniques and designed statistical pattern recognition classifier to identify them.

Our goal is to involve statistical analysis of the frequency components of a time-domain signal (spectral analysis) to detection of the tunneled connection in application layer. Our idea is that legitimate connection, that is not missusing application protocol, imprints specific patterns in the power spectral density that are distinct from spectral density of the other unwanted protocols tunneled through application layer.

In experiments we focused on the application protocols that are most spread in computer networks and that are thus most widely miss-used. In particular we will analyze protocol HTTP and its encrypted version – HTTPS. Most of the network gateways allow only usage of network protocols HTTP or HTTPS. As protocol HTTPS uses encryption, the gateway is unable to detect miss-use by analyzing content and thus it is unable to enforce policy restrictions.

Even though it is not possible to analyze payload of particular packets in ecrypted connection, it is possible to observe the time of the packet transit, its size, direction, source and destination endpoint, etc. This data is denoted as packet traces and it is extracted from unencrypted part of the packet. The goal is to develop feature creation and pattern recognition method for the network packet traces involving spectral analysis. The method is suppoded to infer different application protocols of the network traffic only by observation of the packet traces.

4.1 Data Collection

The input data for our method consits of timestamped packet traces. Following information has been used from the captured packet data:

- packet *timestamp*, i.e. the time of the pass trought the capturing gateway,
- identification of source and destination endpoint (i.e. *source address* and *port*, *destination address* and *port* [1]) and tranmission *protocol* – we use this infomation (5-tuple) as a key to identify packet *flow*,

[1] in literature 4-tuple of the *source address* and *port* and *destination address* and *port* is denoted as *quad*

- *size* of packet's payload,
- packet *direction* [2],

4.1.1 Training Dataset

A `tcpdump` software [20] has been used to capture packet data in experimental testbed. Testbed consist of three or more peers A, B, C, D etc. Peer A (a client) is connected with each other using secure tunnel to the destination TCP port 443 (destination is on the peers B, C, ...). Peer B serves virtual private network service and provides network address translation from the tunnel to the internet. Peers C, D, etc. are able to serve single service on the end of tunnel (such as HTTP server, telnet, file tranfer service, VOIP, ...). The traffic tunneled trough peer B consist varying number of connections and different protocols, while the others are tunneling single application protocol.

The peer A generates traffic to other peers simultaneously, mimicking typical behavior of the client of particular application protocol (i.e. web browser). Packet capturing is set up on peer A intercepting and storing every packet. Captured packet flows are then labeled using name of known the application service.

4.1.2 Evaluation Dataset[3]

The data used for evaluation of proposed method, consists of public packet traces obtained at....*(to be discussed)* mixed with simulated and anotated packet traces captured in our testbed. Before mixing, statistical test has been performed. The test procedure first extracts and tansforms features (denoted here as $\vec{f}$) by using our proposed method and then compares distribution of features with respect to the label – $Pr\left[\vec{f}|label\right]$.

Labeling of the obtained dataset is obtained by inferring application protocol using destination port. Since the public dataset can contain malicious connections, the result of this statistical test will underestimate the real fitness.

4.2 Feature Creation and Pattern Definition

For a specified flow f the packet arrival process $x_f[t]$ (or simply *packet process*) is defined as a count of packet arrivals at given timespan $I = \left\langle \frac{t}{s}, \frac{t+1}{s} \right)$:

$$\forall t \in \mathcal{N} : x_f[t] = |\{p : f = flow(p) \wedge timestamp(p) \in I\}| . \tag{4.1}$$

where s is the sample rate, function $flow(p)$ yields the *flow* 5-tuple and function $timestamp(p)$ yields the *timestamp* of given packet p. [4]

[2] we embed *direction information* into the *size* parameter using negative size if packet travels from destination to source and otherwise positive

[3] Evaluation dataset is used during assesment of the method on real data. As we have some old packet trace data, we want to somehow compare it with training samples. My idea is to prove that exctracted features are epoch invariant by means statistical goodnes-of-fit test. On the other hand older protocols can behave in diferent way and thus the normal models derived from current data would be underestimated during assesment proces.

[4] This is very important formula, we need to focus on it; according to *Dusi et al.* [12] zero-length packets are ulikely to be induced by application, so we can exclude them here; in addition they extract incomming and

According to a Wiener–Khinchin theorem the power spectral density $S_{xx}(f)$ is obtained by application of discrete-time Fourier transform $\mathscr{F}_{\cdot}(f)$ on autocorrelation function of the packet process $R_{xx}[\tau]$:

$$R_{xx}[\tau] = E[x[t]\,x[t+\tau]]\,, \tag{4.2}$$

$$\begin{aligned} S_{xx}(f) = \mathscr{F}_{R_{xx}}(f) = \sum_{\tau=-\infty}^{\infty} (R_{xx}[\tau] \exp(-\imath 2\pi f \tau)) \\ \forall f \in \left\langle -\frac{s}{2}, \frac{s}{2} \right\rangle, \end{aligned} \tag{4.3}$$

where τ is the time-lag, $E[\cdot]$ is expected value of a random variable and f is the frequency. The autocorrelation function is capable of enforcing periodicity. Equations (4.2) and (4.3) hold under assumption that packet process is *wide-sense stationary random proces*.

As the last assumption seems to be false for infinite time span in network traffic, we involved an *windowing function* $w(n)$ and *discrete Fourier tranform* instead of discrete-time Fourier transform. Rectangular windowing is defined as follows:

$$w(n) = \begin{cases} 1, \text{ if } n \in \langle 0, M) \\ 0, \text{ otherwise} \end{cases}. \tag{4.4}$$

Windowing function is nonzero inside specified interval $\langle 0, M)$ otherwise it is zero. Parameter M is length of sub-sequence selected from packet arrival proces. We iteratively apply windowing function to the whole sequence generating non-overlapping adjancent sequence of windows. We identify the particular window in this sequence with upper index – e.g. S^i_{xx} is a power spectral density function of an i-th window. Thus, we rewrite equations (4.2) and (4.3) for non-overlapping windows as follows:

$$R^i_{xx}[m] = \frac{1}{M} \sum_{t=0}^{M} x[t+iM]\,x[t+m+iM]\,, \tag{4.5}$$

$$\begin{aligned} S^i_{xx}(k) = \mathscr{F}_{R^i_{xx}}(k) = \sum_{m=0}^{M-1} \left(R^i_{xx}[m]\, w(m) \exp\left(-\imath 2\pi m \frac{k}{M} \right) \right) \\ \forall k \in \{0, 1, 2, ..., M-1\}\,. \end{aligned} \tag{4.6}$$

Note that the windowing function is inherent in equations (4.5) and (4.6) by using limitted ranges of sumation, and domain of definition of the power spectral density.

By involving windowing function we introduced *spectral leakge*. Use of diferent windowing function, e.g. *Hann* (see equation 4.7), could be appropiate in decreasing leakage.

$$w(n) = \begin{cases} 0.5\left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right), \text{ if } n \in \langle 0, M) \\ 0, \text{ otherwise} \end{cases}. \tag{4.7}$$

outgoing stream separately – I think it is good idea to work with in- and out- streams separately and compute cross-correlation function instead of auto-correlation function. The idea of usage I/O cross-correlation function $R_{io}(\tau)$ instead of auto-correlation is that $R_{io}(\tau)$ (at the specific time-lag τ) would enforce the typical request-response round-trip. Questionable is, how to physically interpret the resulting spectral components and if the Wiener-Kitchin theorem is applicable.

Furthermore if the parameter M is too high the packet proces is unlikely to be stationary, on the other hand selecting too low value causes that the spectrum is sensitive to transient phenomena on the network.

The sampling rate s must be selected according to the Nyquist theorem. Too low value entails aliasing[5], while too high value incurs data storage and processing overhead.

By the application of the Fourier transform original features has been mapped into new space. We denote features in new space as *spectral components* (or *frequency components*). Temporal context of the features has been altered. In new feature space the temporal context is determined by sequence of the detection windows of size M. At the level of the detection window a temporal notion is decomposed into tuple of temporal functions parametrised by *spectral components*. Although temporal aspect is still present it has not been used in further analysis. Anomalies in new feature space are thus regarded to as *point anomalies*.

There are few parameters affecting quality of the features: the smaple rate s and the window length M. In addition, *spectral components* can be subject to further feature extraction or construction, comprising combination of extisting features or discarding irrelevant, redundant or noisy[6]ones based on the domain knowledge (e.g. sum of a lower resp. a upper half of the spectral components resulting in two features a low- and a high-frequency power densities; or retaining specific spectral features known for its relation to periodic stochatic processes to be in research interest).

This aspects and process of seeking of proper parameters are subject of further research and they are discussed in *chapter 5*.

4.3 Pattern Recognition

4.4 Assessment and Interpretation of Results

[5]The aliasing is caused by folding of the frequencies above Nyquist frequency $\frac{s}{2}$ symmetrically below this frequency. Thus this two frequencies are undistinguishable. To properly reconstruct the signal that contain no frequency higher than f_{max} the sample rate is bounded by $s > 2f_{max}$.

[6]In case of low signal-to-noise ratio, a feature is typically not usefull for discriminative outcome.

5 Experiments

6 Conclusion

Bibliography

[1] Darpa intrusion detection data sets (ideval). `http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/index.html`.

[2] R. Alshammari and A.N. Zincir-Heywood. Machine learning based encrypted traffic classification: identifying ssh and skype. In *Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on*, pages 1–8. IEEE, 2009.

[3] R. Bajcsy, T. Benzel, M. Bishop, B. Braden, C. Brodley, S. Fahmy, S. Floyd, W. Hardaker, A. Joseph, G. Kesidis, et al. Cyber defense technology networking and evaluation. *Communications of the ACM*, 47(3):58–61, 2004.

[4] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection for discrete sequences: A survey. *Knowledge and Data Engineering, IEEE Transactions on*, 24(5):823–839, 2012.

[5] V. Chandola, V. Mithal, and V. Kumar. Comparative evaluation of anomaly detection techniques for sequence data. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 743–748. IEEE, 2008.

[6] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, July 2009.

[7] A. Chapanond, M.S. Krishnamoorthy, and B. Yener. Graph theoretic and spectral analysis of enron email data. *Computational & Mathematical Organization Theory*, 11(3):265–281, 2005.

[8] S. Chebrolu, A. Abraham, and J.P. Thomas. Feature deduction and ensemble design of intrusion detection systems. *Computers & Security*, 24(4):295–307, 2005.

[9] Y. Chen and K. Hwang. Spectral analysis of tcp flows for defense against reduction-of-quality attacks. In *Communications, 2007. ICC'07. IEEE International Conference on*, pages 1203–1210. IEEE, 2007.

[10] Y. Chen and K. Hwang. Tcp flow analysis for defense against shrew ddos attacks. In *IEEE International Conf. on Communications*, pages 24–28. Citeseer, 2007.

[11] J.J. Davis and A.J. Clark. Data preprocessing for anomaly based network intrusion detection: A review. *Computers & Security*, 30(6):353–375, 2011.

[12] M. Dusi, M. Crotti, F. Gringoli, and L. Salgarelli. Tunnel hunter: Detecting application-layer tunnels with statistical fingerprinting. *Computer Networks*, 53(1):81–97, 2009.

[13] J.M. Estévez-Tapiador, P. Garcıa-Teodoro, and J.E. Dıaz-Verdejo. Measuring normality in http traffic for anomaly-based intrusion detection. *Computer Networks*, 45(2):175–193, 2004.

[14] P. Gogoi, DK Bhattacharyya, B. Borah, and J.K. Kalita. A survey of outlier detection methods in network anomaly identification. *The Computer Journal*, 54(4):570–588, 2011.

[15] J.R. Goodall, W.G. Lutters, P. Rheingans, and A. Komlodi. Focusing on context in network traffic analysis. *Computer Graphics and Applications, IEEE*, 26(2):72–80, 2006.

[16] X. He, C. Papadopoulos, J. Heidemann, and A. Hussain. Spectral characteristics of saturated links. *Under Submission*, 2004.

[17] M. Iliofotou, M. Faloutsos, and M. Mitzenmacher. Exploiting dynamicity in graph-based traffic analysis: techniques and applications. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 241–252. ACM, 2009.

[18] K. Ingham and H. Inoue. Comparing anomaly detection techniques for http. In *Recent Advances in Intrusion Detection*, pages 42–62. Springer, 2007.

[19] K.L.R. Ingham III. *Anomaly detection for HTTP intrusion detection: algorithm comparisons and the effect of generalization on accuracy*. PhD thesis, The University of New Mexico, 2007.

[20] V. Jacobson, C. Leres, and S. McCanne. The tcpdump manual page, 1989.

[21] A. Jamdagni, Z. Tan, P. Nanda, X. He, and R.P. Liu. Intrusion detection using gsad model for http traffic on web services. In *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*, pages 1193–1197. ACM, 2010.

[22] M. Mahoney and P. Chan. An analysis of the 1999 darpa/lincoln laboratory evaluation data for network anomaly detection. In *Recent Advances in Intrusion Detection*, pages 220–237. Springer, 2003.

[23] J. McHugh. Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. *ACM transactions on Information and system Security*, 3(4):262–294, 2000.

[24] I.V. Onut and A.A. Ghorbani. A feature classification scheme for network intrusion detection. *International Journal of Network Security*, 5(1):1–15, 2007.

[25] Animesh Patcha and Jung-Min Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Comput. Netw.*, 51(12):3448–3470, August 2007.

[26] H. Ringberg, A. Soule, J. Rexford, and C. Diot. Sensitivity of pca for traffic anomaly detection. *ACM SIGMETRICS Performance Evaluation Review*, 35(1):109–120, 2007.

[27] S.T. Sarasamma, Q.A. Zhu, and J. Huff. Hierarchical kohonenen net for anomaly detection in network security. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 35(2):302–312, 2005.

[28] P. Smith, D. Hutchison, J.P.G. Sterbenz, M. Scholler, A. Fessi, M. Karaliopoulos, C. Lac, and B. Plattner. Network resilience: a systematic approach. *Communications Magazine, IEEE*, 49(7):88–97, 2011.

[29] R. Sommer and V. Paxson. Outside the closed world: On using machine learning for network intrusion detection. In *Security and Privacy (SP), 2010 IEEE Symposium on*, pages 305–316. IEEE, 2010.

[30] C. Thomas. Usefulness of darpa dataset for intrusion detection system evaluation. 2010.

[31] J. Undercoffer, A. Joshi, and J. Pinkston. Modeling computer attacks: An ontology for intrusion detection. In *Recent Advances in Intrusion Detection*, pages 113–135. Springer, 2003.

[32] W.W.S. Wei. *Time series analysis*. Addison-Wesley Redwood City, California, 1994.

[33] C.V. Wright, F. Monrose, and G.M. Masson. On inferring application protocol behaviors in encrypted network traffic. *The Journal of Machine Learning Research*, 7:2745–2769, 2006.

[34] S.X. Wu and W. Banzhaf. The use of computational intelligence in intrusion detection systems: A review. *Applied Soft Computing*, 10(1):1–35, 2010.

[35] A. Yamada, Y. Miyake, K. Takemori, A. Studer, and A. Perrig. Intrusion detection for encrypted web accesses. In *Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on*, volume 1, pages 569–576. IEEE, 2007.

[36] M. Zalewski. Advanced sheep-counting strategies. In *Silence on the Wire: A Field Guide to Passive Reconnaissance and Indirect Attacks*, No Starch Press Series, pages 151–172. No Starch Press, 2005.

[37] M. Zalewski. In recognition of anomalies. In *Silence on the Wire: A Field Guide to Passive Reconnaissance and Indirect Attacks*, No Starch Press Series, pages 173–188. No Starch Press, 2005.

[38] M. Zalewski. *Silence on the Wire: A Field Guide to Passive Reconnaissance and Indirect Attacks*. No Starch Press Series. No Starch Press, 2005.

[39] S. Zanero. Analyzing tcp traffic patterns using self organizing maps. *Image Analysis and Processing–ICIAP 2005*, pages 83–90, 2005.

Appendix

Whatever