Literature Review - Anomaly Detection in Computer Networks

Henry Clausen

November 7, 2018

Chapter 1

Introduction

Computer usage can be diverse, and human induced activity on a computer is not constant, but varies in correspondence to the particular task conducted on that computer. In this work, we present a Bayesian framework that models a personal computer's network traffic in order to quantify different states in its usage. For this, we will develop a new hierarchical model based on the *Markov Modulated Poisson Process* that identifies temporal patterns in the arrival of network flow events, and relates them to a latent discrete process which represents the device state. Motivation for this work stems primarily from current interests in cyber-defence, and our inference method is intended to be a critical building block on which a broader cyber-security system would be based. Moreover, this work is heavily related to current procedures in network modelling, for which it might be of future interest.

In the wake of devastating personal information leaks, concerns over cyber-security are at an all-time high. Sophisticated data breaches such as the attack on *JP Morgan Chase* in 2014 affect hundreds of million customers and inflicts tremendous financial, reputational, and logistic damage . Cyber-security incidents increased by 38% in 2017, and the global cost of cyber crime is estimated to reach $2 trillion by 2019 . The prevention of cyber crime is therefore a globally demanded necessity.

One reason for the recent rise of cyber crime is the increased use of sophisticated techniques for the attack of specific targets. Attackers use customised social engineering and custom-build malware to pass common security frameworks. Existing solutions to commercial intrusion detection in computer networks are often based on detecting signatures of previously uncovered and analysed attacks. Examples of such signatures include file hashes[1] of malicious software, blacklisted IP addresses and domain names, and characteristics of known Command-and-Control (C&C) protocols. Detection of a signature usually indicates an imminent intrusion and triggers investigation.

Adjusting existing attack procedures in order to shed previously identified signatures is simple: A file hash can be altered by minor modifications in the program and IP and domain addresses can be switched by changing servers. A sophisticated attack will employ new, customized protocols and software that is fitted to the targeted computer infrastructure, and thus will not show any previously identified signatures.

A different approach to Network Intrusion detection is based on the notion that network intrusions and malicious activity leave different traces in computer event logs[2] than

[1] A hash function encodes a file with a basic data structure into a number or string, which is known as the file hash. Every file is uniquely identifiable with its file hash.

[2] system calls, network traffic, authentication events, etc.

normal computer behaviour, and that these differences can be quantified using more data-oriented methods from the area of statics, machine learning, ...

A widely used source of such events are network traffic logs...

In this review, I will focus on methods

on quantitative models of event logs

the accurate reflection of normal behaviour in a computer network.

A relatively new approach to intrusion detection is based on the accurate reflection of normal behaviour in a computer network. When anomalous behaviour is observed, alternative hypotheses can be formed that reflects attack behaviour. An intrusion is therefore treated as a cumulation of improbable events. Such events can include previously unobserved edges between computers, new processes in combination with E-mail clicks, or failed network logins. To build such a framework, accurate models of several key characteristics of computer network dynamics are necessary in order to capture the aspects that separate regular from irregular behaviour.

1.1 Network Intrusion Detection

Anderson [3] define an intrusion attempt or a threat as an unauthorized and intentional attempt to either access or manipulate information, or to render a system unreliable or unusable. Such attacks can be very diverse in their nature: They can be used to achieve different goals, and correspondingly exploit different types of tools and vulnerabilities. From a network perspective, usually four classes of malicious traffic are distinguished:

1. *DoS-attacks*: A denial-of-service attack is an attempt to remove ability of a particular computer to communicate with other machines over an extended period of time. Such attacks are usually targeted at network servers in order to disrupt the service it is providing. All major types of DoS-attacks achieve this by overwhelming the target server with service requests, which are usually corrupted in a way that causes the server to bind resources unnessecarily long for each request, and thus losing its capability to process other requests. SYN-floods for example exploit the TCP protocol by sending many SYN-requests to the server while ignoring the SYN-ACK response packets sent by the server. This causes the server to keep waiting for a response by the attacker and thus binds resources while being computationally very cheap for the attacker.

2. *Network probing/Reconnaissance attack*: The purpose of network probing attacks is to gather information about computers in a network and possibly find vulnerabilities which can be exploited in further attacks. This typically involves sending specific service requests to other computers in the network, and gather information, such as open ports or operating system on a machine, contained in the corresponding response packets.

   A common type of network probing attacks is *port scanning*. Its aim is to gather knowledge of computers in the network than run vulnerable services, such as HTTP servers, mail servers, and so on. A port scan achieves this by sending queries to one or more network ports on one or more computers in the network. A computer on which the contacted network port is open will respond to the query and thus reveils himself. A port scan can either be vertical, during many ports on one computer are

scanned, or horizontal, where the attacker scans a small number of ports on many computers in the network.

3. *Access Attacks*: These are attacks that try to gain unauthorized access to U2R R2L
4. *Data Manipulation Attack*: Sometimes used to start access attack, for example by passing malware instead of software update (as Flame did)
5. *C&C traffic*: Botnet (extended communication and continuous), ransomware (limited traffic), any malware that communicates with outside, usually port 80 or 443 detection techniques very different, botnet detection using the fact that multiple hosts are having same behaviour
6. *Data exfiltration*

A lot of successful research has been addressed to detecting botnets and port scanning, so we won't address it in this literature review.... ...

1.2 Data Sets

In order to evaluate its ability to model the behaviour of a network and to identify malicious activity and network intrusions, new methodologies have to be tested using existing data sets of network traffic. We can generally distinguish four different types of data sets containing network traffic:

1. **Real network data containing known intrusions**:
2. **Real network data containing injected intrusions**:
3. **Real network data containing no intrusions**:
4. **Untruthed real network data**:
5. **Synthetic network data with/without injected intrusions**:

Write about privacy concerns and general problematic of getting data
Furthermore, we can also identify different formats of network data:

- **Raw packets**
- **Network flows**

1.2.1 Existing data sets

Los Alamos National Laboratory, 2015 - Comprehensive, Multi-Source Cyber-Security Events [19][18]

In 2015, the Los Alamos National Laboratory (LANL) released a large data set containing **network flow** traffic from teir corporate computer network, which contains about 17600 computers. The data was gathered over a period of 58 days with about 600 million events per day. The data only contains internal network connections, i.e. no flows going to or coming from computers outside the network are included. IPs and ports were de-identified

(with the exception of the most prominent port), but are consistent throughout the data. Since the data stems exclusively from one corporate network, it can be assumed that it shows more homogeneity in the observed traffic patterns than general network traffic.

Additionally, the data set also contains other event sources which were recorded in parallel in order to give a more comprehensive look at the network, and could be very useful when investigating a detection approach that correlates multiple event sources. These sources include process events and authentication events from Windows-based computers and servers, and DNS lookup events from the DNS servers within the network.

The dataset furthermore contains a labeled set of redteam events which should resemble intrusions. However, these events are not part of the network flow data and only contain information about the time of the attack and the attacked computer. These events apparently resemble remote *access attacks*, are not described further and appear to be artificial or injected into the data set. It is thus not certain how well they resemble actual network intrusions.

LANL released another data set containing network flow traffic from their network in 2017 [42]. This data set is similar to the one from 2015, but spans over a longer period of time, 90 days. Furthermore, it contains no labeled malicious activity, however that does not mean that the data is completely free of malicious activity.

CTU 2013 [1, 11]

The *Stratosphere Laboratory* in Prague released this dataset in 2013 to study botnet detection. It consists of more than 10 million labeled **network flows** captured on lab machines for 13 different botnet attack scenarios. Additionally, the raw packets for the botnet activity is also available for attack analysis.

The labelling in this dataset is different from other datasets as each flow in the list is labeled based on the source IP address. In the experiments, certain hosts are infected with a botnet and any traffic arising from such a host is labeled as Botnet traffic. Traffic from uninfected hosts is labeled as Normal. All other traffic is Background, as one cannot classify it.

A criticism of this dataset is the unrealistically high amount of malicious traffic contained in the dataset, which makes it easier to spot it while reducing false positives. Furthermore, the way normal or background traffic is generated is described only poorly and leaves the question how representative it is of actual network traffic.

UGR 2016 [28]

The UGR'16 data set was released by the University of Grenada and contains **network flow**[3] data from a spanish 3-tier ISP. This ISP is a cloud service provider to a number of companies, and thus the data comes from a much less structured network than the LANL data. It contains both client's access to the Internet and traffic from servers hosting a number of services. The data therefore contains a very wide variety of traffic patterns, an advantage emphasised by the authors. IP-adresses are consistently anonymised while network ports are unchanged. However, it is not ensured that the traffic capture is complete, i.e. that all traffic coming from and going to a particular machine is captured.

A main focus in the creation of the data was the consideration of long-term traffic evolution and observable periodicity in order to enable the testing of so called *cyclosta-*

[3] netflow v9

tionary traffic models. The data set correspondingly covers a very long period, spanning from March to August of 2016, and containing about 14 GB of traffic per week.

The data is split into a training set and a test set, with the latter containing labeled attack data. This attack data does not stem from rogue agents but is in part generated in controlled attacks on victim machines, and in part injected from previously observed malware infections. The attack data is therefore does not truly correspond to actual attacks, but achieves a high degree of similarity. The implemented attacks contain:

- DoS attacks (controlled attacks),
- Port scanning (controlled attacks),
- C&C traffic from a botnet (injected).

The authors also acknowledge that the background traffic is not necessarily free from further attacks. In fact, three real attacks have been observed and labeled, corresponding to IP-scanning and a spam mail campaign.

UNSW-NB 2015 [30]

The data set realeased by the *University of New South Wales* in 2015 contains real background traffic and synthetic attack traffic collected at the "Cyber Range Lab of the Australian Centre for Cyber Security". The data is collected from a small number of computers which generate real background traffic, and is overlayed with attack traffic using the *IXIA PerfectStorm tool*. The time span of the collection is in total 31 hours.

An advantage of the collected data set is the inclusion of both **raw packets** and **network flows** along with two other data formats containing newly engineered features. This allows a more detailed analysis of the data and possibly a better distinction between attack and benign traffic. In total, the data contains 260 000 events.

Another advantage of the data is the variety of attack data, containing a number of DoS, reconnaissance, and access attacks. However, due to the synthetical injection of these attacks, it is unclear how close they are to real-world attack scenarios.

Since this data set is collected from a relatively small number of machines and during a limited period of time, it is furthermore unclear how suitable for capturing both the temporal evolution and the heterogeneity of real background traffic.

CICIDS 2017 [12][38]

This data set, released by the *Canadian Institute for Cybersecurity* (CIC), contains 5 days of network traffic from 12 computers. These computers all have either different different operating systems such as Windows, OSX, or Ubuntu, or different versions of the same operating system in order to enable a wider range of attack scenarios. The network furthermore contains switches, routers, a web server, a modem, and a firewall in order to ensure a realistic network topology. The traffic data itself consists of **labeled benign and attack traffic**, and is available as 11 GB per day of **raw packets** with payloads, or as **network flows**.

It was ensured that the data contains all traffic coming and going from individual machines. However, in contrast to other data sets, the background traffic is not directly generated through user interactions on the machine, but by using a method to profile abstract user behaviour in different traffic protocol. The purpose of this is to make the

traffic more heterogenuous and to ensure that different types of behaviour are present in the data during the comparably short time span. This However, it is not completely clear how much of the underlying structure of real traffic is lost in the process, and therefore how suitable this data is to build models of benign user activity.

The attack data of this dataset is one of the most diverse among NID datasets, as it contains a variety of up-to-date attacks, such as different types of DoS attacks, SQL-injections and Heart-bleed attack, network scanning, or botnet activity. These are not always successful in order to reflect actual attack scenarios. However, the authors did not describe very well how the data from these attacks is generated and combined with the background traffic as it is also processed through a form of profiling engine.

The CIC released another very similar dataset to this one in 2012.

DARPA 1998 [27]

The *Defense Advanced Research Projects Agency* released the first major dataset to test network intrusion detection systems. The data stems from two experiments at the *MIT Lincoln Laboratory* were multiple victim hosts running Unix and Windows NT were subject of over 200 attacks of 58 different types. The data spans three weeks of training and two weeks of testing data and contains *raw packets* that are labeled. It was since then heavily used as a benchmark to test new detection methods.

Also due to its prominence, it was heavily scrutinised and received a lot of criticism for its lack of realistic background traffic, which was generated through simulation procedure, and the presence of artifacts from these simulations in the data that could heavily skew any model relying on benign traffic. Also, the high percentage of attack traffic in the data is described as unrealistic.

Furthermore, since the dataset is now more than 20 years old, it is remarked that both the benign and attack traffic does not resemble modern network traffic anymore.

KDD Cup 1999 [8, 9]/NSL-KDD 2012 [40]

The *MIT Lincoln Laboratory* created this dataset in 1999 by processing portions of the 1998 DARPA dataset with new labels for a competition at the conference on *Knowledge Discovery and Data Mining*, and is the most widely used dataset in intrusion detection. It contains 2 million connections summaries in a new format and in total 38 attack types. This new format is essentially a form of **network flows** with a greatly increased number of features, 46 in total, whichgive additional details to the origin of the connection. Since the KDD'99 data stems directly from the DARPA dataset, it faces the same problems and criticism.

The *Canadian Institute for Cybersecurity* postprocessed the KDD'99 data in order to address some of its shortcomings. This includes removing redundant records, balancing the size of the training and test data, and adjusting the proportion of attack traffic in the data. However, the biggest criticism from the KDD'99 and the DARPA data, the unrealistic generation of background data, still prevails.

LBNL 2013 [33]

This dataset released by the *Lawrence Berkeley National Laboratory* in 2005 is the first one to examine internal network traffic inside a modern enterprise. It contains more than 100 hours of *packet headers* from several thousand internal hosts.

This dataset contains no known attack traffic, and is therefore only suitable for traffic analysis and model fitting analysis. Furthermore, as being the first dataset containing enterprise traffic, privacy concerns caused the authors to remove any possibilities to identify individual IP addresses.

In 2011, Saad et al. [36] combined this dataset with existing botnet traffic to create a dataset containing both benign and attack traffic.

UNIBS 2009[43]

This dataset was collected on the campus network of the *University of Brescia* on three consecutive days in 2009. The dataset contains in total 79000 anonymised TCP and UDP *network flows*.

This dataset is not directed towards intrusion detection research, but was made as *ground truth data* for traffic classification. It therefore contains labels which indicate which of in total six applications generated the corresponding traffic flow. It might however still be of interest for model assessment in intrusion detection that is relying on traffic classification.

CAIDA 2016 [45]

The *Center for Applied Internet Data Analysis* started collecting network traces from a high-speed backbone link in 2008 with the collection still ongoing. The data is available in anonymised yearly datasets containing one hour of **packet headers** for each month.

Since the traffic is collected from a backbone link, it is very unstructured and heterogenuous. It is furthermore not necessarily free from attack traffic. Although this dataset has been used for intrusion detection before, it is more suitable for general internet traffic analysis.

MAWI 2000 [39]

Similarly to the CAIDA dataset, this dataset contains **packet headers** from the WIDE backbone. It is therefore similarly unstructured, anonymised, and not free from attack traffic. Since this dataset was already collected and released in 2000, it can also be remarked that the contained traffic is too old to represent modern traffic.

ADFA 2013/2014 [6, 7]

The ADFA datasets, released by the *University of New South Wales*, focuses on attack scenarios on Linux and Windows systems as well as **stealth attacks**. To create host targets, the authors installed web servers and database servers , which were then subject to a number of attacks.

The dataset contains both attack traffic and benign traffic. However, the dataset is directed more towards attack scenario analysis and is criticised as being unsuitable for intrusion detection due to its lack of traffic diversity. Furthermore, the attack traffic is not well separated from the normal one.

ICT datasets [2]

The *Impact Cyber Trust* releases cyber security oriented data. Its repository includes many data sets, synthetic as well as real captures, from different sources. Many datasets

focus on observed attack data and thus are not directly applicable to intrusion detection. Furthermore, there is in general very little information provided that describes a dataset's origin, which makes it hard to investigate the network topology.

Among the more useful datasets are the *USC datasets*[4], which contain network traffic (both **packet headers** and **network flows**) from academic networks in the US between 2008 and 2010. The datasets are very large, with the largest one covering 48 hours and containing 357 GB of packet headers.

[4]DS-062, LANDER Data, and DS-266

Chapter 2

Existing literature

Existing literature on intrusion detection can be divided into two approaches:

- **Misuse detection**
- **Anomaly detection**

In misuse detection, abnormal or malicious behaviour is defined first before developing a model to separate the defined behaviour from other traffic. This approach is often used to detect reoccuring patterns in known intrusion. Applications include botnet detection, where the behaviour of many machines connecting to one (or more) C&C servers is defined as malicious activity, or port scan detection, or stepping stone/relay detection.

In contrast, anomaly detection aims at building a model of normal system behaviour that is accurate enough to spot any malicious behaviour as traffic that deviates from the estimated model. Anomaly detection is principally more difficult than misuse detection since the traffic model has to incorporate potentially very heterogenuous traffic behaviours. However, it is generally acknowledged that anomaly detectionhas is more suitable to detect new and previously unseen malicious behaviour as it makes no definite assumptions on the anomalous behaviour.

In reality, anomaly and misuse detection are not necessarily mutually exclusive, and there is a fluent passage between the two. This is because many anomaly detection approaches choose a particular set of features to be modelled with a particular threat in mind. For instance, models for the number of connections of a machine are naturally suitable for detecting DoS attacks, port scans, or Worm attacks.

Chapter 3

Anomaly detection

Anomaly-based intrusion detection moved into the focus of researchers at the end of the 90s, with many advances and new ideas being implemented between 1998 and 2005.

3.0.1 Approaches based on Volume or Traffic Aggregation

Difficult to identify individual malicious flows and attack attribution.

Subspace projection/PCA-based

Principal Component Analysis is a statistical form of *orthogonal coordinate transformation* to convert a set of observations (or feature vectors) into a set of linearly independent variables[1]. These variables uncorrelated variables each account for differing amounts of the variation contained in the data. By projecting an observation only onto the components that account for the most variation, it is possible to retain most of the information while operating in much lower dimensions.

Lakhina et al. [25, 24] introduced a PCA-based anomaly detection method for network traffic in 2004. In their approach, they aggregated the network flows for each OD[2] pair into 5-minute bins, with the number of transferred bytes, packets, and flows being the features for each bin. Each 120 consecutive bins were then treated as a an observation (with $3 \cdot 120$ variables), and PCA was then applied to the collection of observations. The first 5 principal components are then identified as the dominant temporal patterns. Anomalies were then identified as observations that could only very poorly reconstructed using the first 5 principle components. Since then, this approach has been adopted to several other datasets without much methodological advances. Camacho et al. [5] proposed an improvement to the existing PCA-based approach with a more natural implementation of spotting anomalies.

This approach can be applied to individual OD pairs, or on a network-wide basis by using q-statistics to spot multivariate anomalies. The approach was tested on data from the Abilene backbone network, and worked well to identify significant episodes such as DoS attacks, fast spreading worms and other large-scale scanning activity, alpha-flows, or power outages.

Naturally, since the traffic is aggregated into bins and the temporal behaviour of these bins are examined, this approach is aimed towards identifying attacks with a comparably large volume of traffic, even if they are isolated in time. It is however unlikely that it

[1] the *principal components*

[2] Origin-Destination

is capable of spotting smaller U2R and R2L attacks or C&C traffic. Another possible criticism is that anomalies are not spotted immediately, but in the worst case after hours.

Ringberg and Rexford [35] provided a discussion of PCA-based approaches to traffic anomaly detection. They concluded that PCA is very sensitive to small differences in the number of used principal components and to the level of aggregation of the traffic measurements. Furthermore, the training data has to be absolutely free of any traffic anomalies, otherwise the projection onto the first principle components can change drastically.

Entropy-based

The entropy is a measure for the degree of disorder a system is in. Applied to network traffic, a popular quantity to measure is the dispersion of events onto different source or destination IP addresses. A high entropy would correspond to all events being evenly distributed among the all existing IPs while a low entropy corresponds to the majority of events being concentrated between a small number of IPs.

Entropy-based approaches can usually be attributed more towards misuse detection, but can also have reasonable applications in anomaly detection.

Wagner and Plattner [44] measured the entropy of network flow distribution across source and destination IP addresses and across source and destination ports on a network-wide basis. The entropy is measured in a sliding window of 5-minutes with 1-minute shifts and monitored continuously. Anomalies are flagged as sudden changes in one or more of the mentioned entropy sources with tresholds that were determined from empirical judgement. The described measures were applied to network data from a swiss internet backbone which contained data from two large-scale worm outbreaks[3]. The characteristics of these worms in terms of entropy changes[4] were then analysed as an evaluation of the technique.

Lakhina et al. [26] used entropy in a similar, albeit more sophisticated way to detect anomalies. Instead of using entropy on a network-wide basis, it used to monitor src and dst IP and port distributions on individual hosts over time. The obtained values are then bundled in a three-way matrix $H(t, p, k)$ where t is the current time, p is the particular host, and k is one of the four monitored traffic features. This data matrix is then converted into a two-way matrix and similar to other work by Lakhina, a PCA-like subspace projection is applied to mine temporal features as orthogonal components. However, here these components reflect correlations in simultaneous entropy changes on different hosts and features. Anomalies are again detected as poor reconstructions by via the most dominant components, and the performance is examined using untruthed data from the Abilene backbone network. Another clever addition of this paper is the use of unsupervised learning to identify different types of observed anomalies. This is done by applying hierarchichal clustering with a fixed number of clusters to the residual vector of H. However, contains some obvious flaws that in my opinion prevent a generalised grouping of anomalies.

[3] *Blaster worm* and *Witty worm*, both more than 50 000 infections

[4] Source IP and destination port entropy decreases drastically, destination IP entropy increases moderately

Other

Kind et al. [20] proposed a network-wide detection approach that is based on traffic histograms and clustering to identify and model substructures in normal traffic for better anomaly detection. A number of different traffic quantities are divided into a fixed number of subgroups, such intervals of IP-addresses or network ports, bins of packet sizes or connection durations, or the different TCP flags in a connection. For every different feature, authors create histograms measuring the number of events occuring in each subgroup during a 5-minute interval. Histograms are monitored over a period of time without any malicious activity in order to gather a collection of training histograms. After removing subgroups that remain, each traffic feature is then divided into clusters using the Mahalanobis-distance measure as a similarity metric between histograms and either hierarchical or k-means clustering. Anomalies in individual traffic features can then be detected as histograms with distances from the nearest cluster that exceed a certain treshold. Evaluation is conducted with an unnamed dataset containing 15 labeled attack, containing DoS attacks, worm propagation and network scanning, and network-wide system fingerprinting, and mail bombs, of which 13 were detected. Unsurprisingly, the two undetected attacks addressed fewer machines and thus consisted of less traffic. In general, this approach indivates a great improvement from the previous approaches as it is the first significant attempt at discovering substructures in aggregated traffic, which generally grants a better detection of non-trivial anomalies. An interesting improvement would be to correlate simultaneous outliers in different features using a variant of hypothesis tests as a lower anomaly treshold could be used.

Heard et al. [15, 14] recently proposed a rather different approach: The authors here monitor the number of network-internal connections and authentication events on each machine over 5-minute invervals. The authors observed network-wide a strong *power-law-like* distributions of the number of events per host, i.e. the number of events is very large for some hosts an declines proportional to cx^{-k} where x is the number of the host. The authors then model the behaviour using two related probabilistic methods, the *Dirichlet process* and *latent Dirichlet allocation*, where the model parameters were estimated from attack-free training data in a Bayesian fashion. Event numbers which deviate strongly from the estimated model were then scored according to their unlikeliness. Both approaches were tested using the LANL dataset and asigned high scores to known inected computers. Furthermore, the authors claim to have possibly detected an unlabeled machine as being infected through manual investigation after it was asigned the highest score of all machines. As this approach is solely looking at the number of incoming and outgoing connections and events, this approach similarly to many entropy-based approaches covers a narrow area of possible malicious activity, and it is concerning that the ratio of benign machines with high anomaly scores (which can be seen as false alerts) is very high. However, this approach marks a step into a more probabilistic anomaly asignement which is beneficial for a more adaptive approach to model estimation and a quantification of detection certainty.

Wavelet-based

Wavelet modelling is a frequency-based signal processing approach. Amplitudes of most signals can be described as a finite sum of wavelets with different frequencies. These frequency coefficients can then be used as a measure to describe the signal's generalised behaviour, and to compare with future data from the same signal. Three significant

papers applying wavelet modelling to intrusion detection can be found:

Both **Barford et al.** [4], and **Thottan and Ji** [41] introduced wavelets to network anomaly detection in 2002/2003. Both approaches are fairly simple, as they only look at the network flow numbers from multiple machines at different points in the network, aggregated into 5-minute intervals. Using enough anomaly-free training data[5], this volume signal can be described by a set of frequency-components. Barford et al. then compare the frequencies of any future traffic episodes against this set, and marked as anomalous if a treshold is exceed. The approach is directed towards detecting network-wide flash crowds, DoS attacks, and outages, which is evaluated using proprietary data. Thottan and Ji detected anomalies by looking at the reconstruction error of such traffic episodes using the estimated frequency-components instead comparing different estimates. The reconstruction error is assumed to follow a gaussian distribution, and anomalies can be detected using a hypothesis test.

Jian et al. [16] proposed a refined wavelet-based model in 2014 which is applied to individual OD pairs. All observed OD pairs in the network are grouped into q[6] groups. To each group, an S-transformation (a modified version of a wavelet-transformion) is applied. The signal for each OD pair is then reconstructed using only the estimates of the high-frequency components since these correspond to any bursty behaviour. Now, the reconstructed signal is free from any slow variation and contains only fast and bursty behaviour. The assumption of the authors this degenerated signal must be heavily correlated between individual OD pairs. Using a sliding window, the pairwise correlations of each OD pair is computed. If the correlation for any pair falls below a certain treshold, this pair is marked as an anomaly. The approach is designed to detect volume-intensive attacks on busy servers, the exact motivation for this approach however is not described very well by the authors. The evaluation is done using data from the *Abilene backbone*.

It is clear that any approach that looks exclusively at the traffic volume either between individual hosts or in a network-wide fashion will only detect attacks with sufficient attack volume, such as DoS attacks. Additionally, wavelet-based approaches do not provide a probabilistic framework for anomaly detection, which makes the seperation of true anomalies from false positives difficult, especially in larger networks.

3.0.2 Event-based

The majority of network anomaly detection approaches are based on point anomaly detection, in other word they identify individual events as malicious solely on the observed characteristics of this event. Such events are usually either individual network packets or flows. In contrast to approaches on aggregated traffic, which can usually only detect attacks with a certain traffic volume, an event-based approach is independent of the traffic volume and therefore more suitable to identify activities consisting of only a few events, such as data exfiltration, R2L attacks, or C&C communication.

Direct application of Machine Learning

Due to the availability of datasets such as DARPA'98 or KDD'99 which are labeled and rich in both benign and malicious traffic, it is tempting to apply existing machine learning approaches directly onto the event features provided in the data, and there exists a large

[5]It is crucial that this data is absolutely free of anomalies that are to be detected

[6]number depends on network topology

body of literature doing exactly that. Unfortunately, such approaches often lack the necessary understanding of the data and are overfitting, or do not learn any generalisable behaviour.

Also, a number of different classifier techniques were applied on labeled datasets. Although they often achieve good accuracy in the DARPA'98 or KDD'99 datasets, it is unclear how well they translate. Should I include papers using classifiers

I will briefly discuss some of the better ones here:

Statistics-based

Mahoney and Chen [29] were one of the first to develop statistical methods to identify anomalous events in network traffic. Their approach consists of two separate scoring stages.

The first stage is the *packet header anomaly detection* (PHAD). Here, the 33 different fields of an Ethernet-transmitted packet are converted from their one to four bytes to an integer value. The gathered values for each field are then clustered in a simple agglomerative fashion, and the clusters are updated each time a new packet arrives in order to keep the number of clusters below a treshold. The anomaly score of a packet is then proportional to the number of fields in which the clusters had to be updated.

At the second stage, the *application layer anomaly detection* (ALAD), scores are asigned to the packet according to a frequency table build using previously collected packets. These frequency tables address the several combinations of a variable conditional on another variable. These variables include source or destination IPs, destination port, TCP flags, or the first word of the payload. An interesting factor considered by the authors is the inclusion of the time since last observance for each of these frequency tables in the anomaly score.

The approach was tested on the DARPA'98 dataset and detected 70 out of 180 attacks while raising 100 false alerts. When the unrealistically high number of malicious packets in the data is considered, the number of false alerts is alarmingly high. Another issue with this publication is that the authors claim that they are building nonstationary models, yet give little how these models should adapt over time.

Kruegel et al. [23] developed an approach that is aimed fitting individual models for each of the different services generating network traffic. Their assumption is that by concentrating on only one type of traffic, statistical data with lesser variance can be collected.

The approach works as following: Once a connection is openened, the packet processing unit reads the first packets of a connection and extracts the specific service, such as a get request for a HTTP request. It is then asigned an anomaly score based on the different aspects: The type of service, the length of the request, and the payload contained in the request.

The anomaly score associated with the type of service is proportional to the negative logarithm of the service frequency observed in the training data. Thus, rare services receive a higher anomaly score.

To score the length l of the request, the mean μ and standard deviations σ of request lengths in the training data is estimated using maximum likelihood. The score is then proportional to $(l - \mu)/\sigma$.

Finally, the payload is scored according to a frequency distribution of the letters occuring in the training data. The deviation of a payload from the distribution can easily be estimated using a χ^2 test. By scoring the payload of a service request, the authors hope

to detect malicious requests that try to disrupt the reciver through a corrupt combination of non-printable or replaced characters. Kruegel et al. [22] later greatly improved the payload scoring specifically for HTTP traffic by using a *Markov model*.

In their evaluation, the authors only considered DNS traffic due to lack of resources and space. Testing was done after a calibration of the anomaly tresholds by attacking their own DNS servers with 5 different attacks, all of which have been detected. However, evaluation of other services and on independent data would shed more light on the actual performance. Another important issue not addressed by the authors is the possible temporal drift of the estimated distributions.

Both of these papers introduced introduced new concepts of how to model the distributions of individual event features which take into account the nature of network traffic. However, there is a lot to criticise about these approaches. The developed estimation and scoring methods lack a broader probabilistic foundation and can be improved greatly. Furthermore, these papers do not address any possible interdependence of features, which could lead to serious mismodelling of behaviour observed as anomalous. Also, it is unclear if these models will provide behaviour over time.

Clustering based

Representation-learning based

Representation learning, also called *feature learning* is a set of techniques aimed at automatically learning underlying structures in raw and noisy data, and are in a broader sense a form of density estimation. These techniques are often based on learning lower dimensional representations of the data, similar to subspace-projections, and are therefore suitable for data with highly correlated variables. Existing methods are often based on neural-networks and backpropagation. Learning of normal traffic behaviour can be done directly using representation learning instead of deriving probability distributions and correlations of individual traffic variables first. However, current methods are only suitable for numerical variables and not for categorical ones.

Ramadas and Ostermann [34] in 2003 proposed the use of *self-organizing maps* (SOM) to learn the representation of individual types of network services. A self-organizing map projects input data onto a two-dimensional lattice, which is why they are often used for data visualisation. The projection is learned using groups of competitive neurons. Each generation drops neurons which have different representations from the group, which makes this approach particularly computation-intensive. Since the projected data lies densely together, the authors train the map with normal traffic and detect anomalous events via their distance to the nearest neighbour. The authors however only evaluate their approach using 6 numerical features from DNS and HTTP network flows which they collected themselves. This makes a performance evaluation difficult and also leaves the question open how much knowledge is gained by using only 6 different flow features. Kayacik et al. [17] later extend this approach to all 41 numerical features of the KDD'99 data. The evaluation showed most success in the detection of DoS and probing attacks.

A more direct approach to outlier detection is provided by **Hawkins et al**. [13] in 2002. They applied a *replicator neural network*[7] to the numerical features of the KDD'99 data. A replicator network tries to accurately reconstruct any input data it receives after sending it through a lower-dimensional bottleneck. The difference to an

[7]Also called *Autoencoder network*

SOM is that learning is based on error-correction. By training it on normal traffic, the authors build a model that can reconstruct any normal traffic from its lower-dimensional representation with small errors. Anomalies are then detected as input data which is not reconstructed well and therefore deviates substantially from the learned data structures. Supposedly, a replicator network is robust agains small numbers of outliers in the training data. However, it requires careful examination how well this assumption translates onto network traffic.

Gao et al. [10] use a similar technique called *deep belief networks* (DBN) on the KDD'99 data. They have a similar structure to replicator networks, but training is more difficult since their hidden layers are probabilistic. The authors mainly focus on explaining the benefits of using probabilistic neurons and discussing possible ways how to train a DBN on network traffic while not providing a thorough discussion of their results.

3.0.3 Temporal correlation/Semantics-based

In 2003, **Krishnamurthy et al.** [21] propose a rather simple, yet efficient method for network-wide monitoring of individual key occurrences in an online fashion. For that, a sliding window approach is used to asign all keys (which here stand for individual IP addresses or network ports) a value containing the number of its occurrency. For each key, the collected values are then used to train either an *ARIMA* or a *Holt-Winters* method, both popular and powerful time-series forecasting models which can be trained in an online fashion. Anomalies are then identified as values with a forecasting error exceeding a certain treshold and thus indicating a sudden change in occurrency-behaviour of that particular key. This is an improvement to summarisation measures such as entropy or histograms in two ways: It provides a better resolution of individual traffic channels, enabling the detection of attacks with far less volume, and enabling the modelling of more complex temporal patterns which become apparent on a lower level. And it makes attack attribution far simpler by indicating directly the key which is subject to an anomaly.

Since traffic is usually arriving at a fast rate, it is a computationally hard and memory-consuming task to count the occurrencies of all keys simultaneously. Schweller et al. overcome this problem using a *sketch-based counting approach* which uses hash-function to direct the values of a key directly to a position in a hash table without the need to store actual key in the memory. As this process is not exact, the count value is subject to statistical variations and the authors propose an unbiased estimator. The inaccuracy of the count estimator is also preventing the authors from using a probabilistic approach to anomaly detection instead of simple tresholding. However, this approach is also only counting occurrencies and does not detect any correlated key behaviour.

A great problem of the proposed framework is the fact that the key translation works only in one direction, making it impossible to associate a detected change with the corresponding key. This problem is overcome by Schweller et al.[37] who propose a reversible sketch method.

Noble and Adams [31, 32] have recently proposed *ReTiNa*, a tool that measures temporal changes in the correlation between individual events in order to find intrusions on individual hosts. In their approach, they estimate the correlation between the time passed between two events, also called *interarrival time*, of an OD pair and the associated size or number of packets of the involved events. For this, interarrival time and the size/packet number are modelled as a bivariate gaussian distribution, and the covariance matrix is estimated using maximum-likelihood-estimation. The authors use a sophisticated online-

estimation method to adapt the estimates to changes in the correlation structure, which can then be identified by comparison to an offline estimate. Anomalies are then identified as a collection of changes happening across multiple OD pairs on one host or in the entire network by simple hypothesis testing, which decreases the false-positive rate. The assumption here is that different OD pairs are independent of each other.

A big advantage of this approach is that it is adaptive and does not need a training phase, i.e. it is not reliant on attack-free training data. The method was tested both on the LANL network flow data as well as internal data from the *Imperial College Academic network*. The method found several anomalies that coincide malicious activity in the network, but a definitive conclusion whether they are related is difficult to make.

Whitehouse, Evangelou and Adams [46] modeling the number of network flow and *user authentication* events on individual hosts as a polynomial function of the time and day and its rarity. Anomalies are then identified using Fisher's product test statistic and the reconstruction error. The method was tested on the LANL data using the auth and the flow sources and wa able to identify persistent structures in the data.

Semantic-based approaches using different data sources

hidden markov models for system calls - two papers

Bibliography

[1] The CTU-13 Dataset. A Labeled Dataset with Botnet, Normal and Background traffic. 00001.

[2] Impact cyber trust: USC DS-062, USC DS-266, USC LANDER. `https://www.impactcybertrust.org/dataset_view?idDataset=62/` `https://www.impactcybertrust.org/dataset_view?idDataset=75/` `https://www.impactcybertrust.org/dataset_view?idDataset=265`, 2010. Accessed on 05 Nov. 2018.

[3] J. P. Anderson. Computer security threat monitoring and surveillance. *Technical Report, James P. Anderson Company*, 1980.

[4] P. Barford, J. Kline, D. Plonka, and A. Ron. A signal analysis of network traffic anomalies. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 71–82. ACM, 2002.

[5] J. Camacho, A. Prez-Villegas, P. Garca-Teodoro, and G. Maci-Fernndez. PCA-based multivariate statistical network monitoring for anomaly detection. *Computers & Security*, 59:118–137, June 2016. 00027.

[6] G. Creech. *Developing a high-accuracy cross platform Host-Based Intrusion Detection System capable of reliably detecting zero-day attacks.* PhD thesis, University of New South Wales, Canberra, Australia, 2014.

[7] G. Creech and J. Hu. Generation of a new ids test dataset: Time to retire the kdd collection. In *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*, pages 4487–4492. IEEE, 2013.

[8] K. Cup. Data. knowledge discovery in databases darpa archive, 1999.

[9] K. Cup. Dataset. `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`, 1999. Accessed on 05 Nov. 2018.

[10] N. Gao, L. Gao, Q. Gao, and H. Wang. An Intrusion Detection Model Based on Deep Belief Networks. In *2014 Second International Conference on Advanced Cloud and Big Data*, pages 247–252, Nov. 2014. 00046.

[11] S. Garcia, M. Grill, J. Stiborek, and A. Zunino. An empirical comparison of botnet detection methods. *computers & security*, 45:100–123, 2014.

[12] A. Gharib, I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani. An evaluation framework for intrusion detection dataset. In *Information Science and Security (ICISS), 2016 International Conference on*, pages 1–6. IEEE, 2016.

[13] S. Hawkins, H. He, G. Williams, and R. Baxter. Outlier Detection Using Replicator Neural Networks. In Y. Kambayashi, W. Winiwarter, and M. Arikawa, editors, *Data Warehousing and Knowledge Discovery*, Lecture Notes in Computer Science, pages 170–180. Springer Berlin Heidelberg, 2002. 00451.

[14] N. Heard, K. Palla, and M. Skoularidou. Topic modelling of authentication events in an enterprise computer network. 2016.

[15] N. Heard and P. Rubin-Delanchy. Network-wide anomaly detection via the dirichlet process. In *Intelligence and Security Informatics (ISI), 2016 IEEE Conference on*, pages 220–224. IEEE, 2016.

[16] D. Jiang, Z. Xu, P. Zhang, and T. Zhu. A transform domain-based anomaly detection approach to network-wide traffic. *Journal of Network and Computer Applications*, 40:292–306, 2014.

[17] H. G. Kayacik, A. N. Zincir-Heywood, and M. I. Heywood. A hierarchical som-based intrusion detection system. *Engineering applications of artificial intelligence*, 20(4):439–451, 2007.

[18] A. D. Kent. Comprehensive, Multi-Source Cyber-Security Events. Los Alamos National Laboratory, 2015.

[19] A. D. Kent. Cybersecurity Data Sources for Dynamic Network Research. In *Dynamic Networks in Cybersecurity*. Imperial College Press, June 2015.

[20] A. Kind, M. P. Stoecklin, and X. Dimitropoulos. Histogram-based traffic anomaly detection. *IEEE Transactions on Network and Service Management*, 6(2):110–121, 2009.

[21] B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen. Sketch-based change detection: methods, evaluation, and applications. In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 234–247. ACM, 2003.

[22] C. Kruegel, G. Vigna, and W. Robertson. A multi-model approach to the detection of web-based attacks. *Computer Networks*, 48(5):717–738, 2005.

[23] C. Krügel, T. Toth, and E. Kirda. Service specific anomaly detection for network intrusion detection. In *Proceedings of the 2002 ACM symposium on Applied computing*, pages 201–208. ACM, 2002.

[24] A. Lakhina, M. Crovella, and C. Diot. Characterization of Network-wide Anomalies in Traffic Flows. In *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, IMC '04, pages 201–206, New York, NY, USA, 2004. ACM. 00439.

[25] A. Lakhina, M. Crovella, and C. Diot. Diagnosing Network-wide Traffic Anomalies. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '04, pages 219–230, New York, NY, USA, 2004. ACM. 01230.

[26] A. Lakhina, M. Crovella, and C. Diot. Mining anomalies using traffic feature distributions. In *ACM SIGCOMM Computer Communication Review*, volume 35, pages 217–228. ACM, 2005.

[27] R. P. Lippmann, D. J. Fried, I. Graf, J. W. Haines, K. R. Kendall, D. McClung, D. Weber, S. E. Webster, D. Wyschogrod, R. K. Cunningham, et al. Evaluating intrusion detection systems: The 1998 darpa off-line intrusion detection evaluation. In *DARPA Information Survivability Conference and Exposition, 2000. DISCEX'00. Proceedings*, volume 2, pages 12–26. IEEE, 2000.

[28] G. Maciá-Fernández, J. Camacho, R. Magán-Carrión, P. García-Teodoro, and R. Therón. Ugr 16: A new dataset for the evaluation of cyclostationarity-based network idss. *Computers & Security*, 73:411–424, 2018.

[29] M. V. Mahoney and P. K. Chan. Learning nonstationary models of normal network traffic for detecting novel attacks. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 376–385. ACM, 2002.

[30] N. Moustafa and J. Slay. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *2015 Military Communications and Information Systems Conference (MilCIS)*, pages 1–6, Nov. 2015. 00074.

[31] J. Noble and N. Adams. Real-Time Dynamic Network Anomaly Detection. *IEEE Intelligent Systems*, 33(2):5–18, Mar. 2018. 00000.

[32] J. Noble and N. M. Adams. Correlation-Based Streaming Anomaly Detection in Cyber-Security. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 311–318, Dec. 2016. 00004.

[33] R. Pang, M. Allman, M. Bennett, J. Lee, V. Paxson, and B. Tierney. A first look at modern enterprise traffic. In *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, pages 2–2. USENIX Association, 2005.

[34] M. Ramadas, S. Ostermann, and B. Tjaden. Detecting anomalous network traffic with self-organizing maps. In *International Workshop on Recent Advances in Intrusion Detection*, pages 36–54. Springer, 2003.

[35] H. Ringberg, A. Soule, J. Rexford, and C. Diot. Sensitivity of PCA for Traffic Anomaly Detection. In *Proceedings of the 2007 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '07, pages 109–120, New York, NY, USA, 2007. ACM. 00337.

[36] S. Saad, I. Traore, A. Ghorbani, B. Sayed, D. Zhao, W. Lu, J. Felix, and P. Hakimian. Detecting p2p botnets through network behavior analysis and machine learning. In *Privacy, Security and Trust (PST), 2011 Ninth Annual International Conference on*, pages 174–180. IEEE, 2011.

[37] R. Schweller, A. Gupta, E. Parsons, and Y. Chen. Reversible sketches for efficient and accurate change detection over network data streams. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 207–212. ACM, 2004.

[38] I. Sharafaldin, A. Gharib, A. H. Lashkari, and A. A. Ghorbani. Towards a reliable intrusion detection benchmark dataset. *Software Networking*, 2018(1):177–200, 2018.

[39] C. Sony and K. Cho. Traffic data repository at the wide project. In *Proceedings of USENIX 2000 Annual Technical Conference: FREENIX Track*, pages 263–270, 2000.

[40] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani. Nsl-kdd dataset. `http://www.unb.ca/research/iscx/dataset/iscx-NSL-KDD-dataset.html`, 2012. Accessed on 05 Nov. 2018.

[41] M. Thottan and C. Ji. Anomaly detection in ip networks. *IEEE Transactions on signal processing*, 51(8):2191–2204, 2003.

[42] M. J. M. Turcotte, A. D. Kent, and C. Hash. Unified Host and Network Data Set. *ArXiv e-prints*, Aug. 2017.

[43] UNIBS. Data sharing. `http://netweb.ing.unibs.it/~ntw/tools/traces/`, 2009. Accessed on 05 Nov. 2018.

[44] A. Wagner and B. Plattner. Entropy based worm and anomaly detection in fast ip networks. In *Enabling Technologies: Infrastructure for Collaborative Enterprise, 2005. 14th IEEE International Workshops on*, pages 172–177. IEEE, 2005.

[45] C. Walsworth, E. Aben, K. Claffy, and D. Andersen. The caida ucsd anonymized internet traces 2012,, 2015.

[46] M. Whitehouse, M. Evangelou, and N. Adams. Activity-based temporal anomaly detection in enterprise-cyber security. In *IEEE International Big Data Analytics for Cybersecurity computing (BDAC'16) Workshop, IEEE International Conference on Intelligence and Security Informatics*. IEEE, Nov. 2016. 00001.