Digital Investigation 20 (2017) S56–S65

Contents lists available at ScienceDirect

Digital Investigation

journal homepage: www.elsevier.com/locate/diin

DFRWS 2017 Europe — Proceedings of the Fourth Annual DFRWS Europe

Evidence gathering for network security and forensics

Dinil Mon Divakaran*, Kar Wai Fok, Ido Nevat, Vrizlynn L.L. Thing

*Cyber Security Cluster, A*STAR Institute for Infocomm Research (I²R), 1 Fusionopolis Way, #21-01 Connexis (South Tower), 138632, Singapore*

ARTICLE INFO

Article history:
Received 31 January 2017
Accepted 31 January 2017

Keywords:
Forensics
Security
Network
Traffic
Regression

ABSTRACT

Any machine exposed to the Internet today is at the risk of being attacked and compromised. Detecting attack attempts, be they successful or not, is important for securing networks (servers, end-hosts and other assets) as well as for forensic analysis. In this context, we focus on the problem of evidence gathering by detecting fundamental patterns in network traffic related to suspicious activities. Detecting fundamental anomalous patterns is necessary for a solution to be able to detect as many types of attacks and malicious activities as possible. Our evidence gathering framework correlates multiple patterns detected, thereby increasing the confidence of detection, and resulting in increase in accuracy and decrease in false positives. We demonstrate the effectiveness of our framework by evaluating on a dataset consisting of normal traffic as well as traffic from a number of malwares.

Introduction

End-hosts and networks become prone to attacks and compromises once they get connected to the Internet. The types and sophistication of attacks have only been increasing with time, one of the motivations being the monetary profits that cyber-criminals obtain from such malicious and cyber–criminal activities. To quantify, in its annual security report of 2016, Cisco's estimation of the average annual income for ransomware per campaign is $34 million (Cisco Annual Security Report, 2016).

Detection of anomalies related to attacks is imperative for securing a network as well as for forensic analysis. Timely detection can thwart damages due to the attack; forensic analysis of the anomalies aid not only in investigations but also in profiling of attacks. There exist a plethora of research works on detecting anomalies and attacks in networks (Carl et al. (2006), Feily et al. (2009), Chandola et al. (2009), Vincent Zhou et al. (2010) and Bhuyan et al. (2014) are some relevant surveys). Often, different solutions are developed for different attacks, leading to complex management and higher costs for the users. Besides, individual anomalies are usually detected independently; and without having a mechanism to check if some of the detected anomalies are correlated, solutions could easily miss the forest for the trees. Worse, the detected anomalies might be assigned a low risk, and therefore ignored by an analyst.

Differing from the previous works, our aim is to develop a framework that can gather as many evidences as possible for different types of attacks and malicious activities. In this work, we focus on the problem of evidence gathering by analyzing network traffic. The evidences are the fundamental patterns related to suspicious activities. Detecting such patterns allows for detection of anomalies that are common to a number of attacks. We take a pragmatic approach in detecting anomalous patterns; we apply regression models to analyze network traffic data and detect patterns related to suspicious activities. The anomalous patterns, which form evidences, are correlated and analyzed, to increase the performance of detection of traffic related to attacks and malicious activities, where performance is expressed in terms of detection accuracy and false positive rate.

The advantages of our proposed system are:

1. No learning of normal behavior using benign data: Given that Internet traffic is always evolving and changing, a solution that models static data will not be adaptive. Besides, it is extremely difficult to obtain a real-life up-to-date traffic dataset of meaningful size that consists of only benign flows and captures the dynamic characteristics of benign traffic. Our approach, on the other hand, uses information of only attack traffic. Characteristics of attack traffic are very specific to the kind of attack, and also differs significantly from that of normal traffic.
2. Detection of multiple anomalous patterns: By detecting anomalous patterns in network traffic, our solution essentially gathers evidences related to the attack. For a single attack, there might be multiple patterns that are detected, some due to different

* Corresponding author.
E-mail addresses: divakarand@i2r.a-star.edu.sg (D.M. Divakaran), fokkw@i2r.a-star.edu.sg (K.W. Fok), ido.nevat@tum-create.edu.sg (I. Nevat), vriz@i2r.a-star.edu.sg (V.L.L. Thing).

http://dx.doi.org/10.1016/j.diin.2017.02.001

features used for analysis, while some due to the different stages of an attack (such as, reconnaissance, penetration and exploit). More the number of patterns, higher the confidence of detection.
3. Detection of multiple attacks: As our solution searches for different patterns related to attacks, it is not tailored for detection of any particular attack; in other words, it is designed to detect different kinds of attacks. As we demonstrate in Section "Performance evaluations", our solution is able to detect traffic generated by a number of different malwares.

We evaluate our solution on a dataset compiled from different sources, and consisting of both normal traffic as well as traffic generated by malwares. The evidence-gathering framework we developed is able to detect malware traffic sessions with high accuracy, while maintaining low false positive rate. Different from past works, our evaluation also provides insights into the number of evidences detected for malicious traffic sessions, thereby aiding operators to decide on the configurations.

We discuss related works in the next section. In Section "Framework for evidence gathering", we present the framework for evidence gathering. The techniques based on regression modeling and analysis that we use for detection of anomalous patterns are presented in Section "Regression analysis for detection of anomalous patterns". The eventual decision making criteria, on whether a (set of) traffic flows anomalous or not, is based on the evidences gathered, and this is presented in Section "Decision making based on evidences gathered". We present the experiments performed and discuss the results obtained in Section "Performance evaluations".

Related works

Tremendous amounts of efforts have been invested by the research community to develop solutions for detection of network attacks and anomalies. Even the literature on one specific set of attacks, for instance DoS attacks (Carl et al., 2006; Peng et al., 2007), is vast. This section provides a brief discussion on some of the important related works, without attempting to be exhaustive.

Although the terms anomalies and attacks are highly related, there is also a difference. An anomaly is anything that deviates from what is defined and observed as normal behavior. The cause of such deviations might be network faults (e.g., link failure), abrupt changes in network—for example route changes, or even changes in CDN (Content Delivery Network) caches (Fiadino et al., 2014), attacks, etc. While attacks are all activities that (attempt to) breach the security of any given system, not all anomalies are attacks. Examples of such anomalies (that are not attacks) range from sudden peak in traffic at a web server to receiving large number of scanning packets on a network. Yet, a port scan on a machine might be related to an impending penetration attempt or vulnerability exploitation at a specific port. Therefore, while individual anomalies may well turn out to be evidences of attacks and compromises, analyzing them independently might lead to unacceptable false-positive rate, thereby becoming counterproductive.

To solve the challenging problem of anomaly detection in networks, researchers have applied knowledge from different (overlapping) spheres such as expert system (Ilgun et al., 1995), information theory (Gu et al., 2005; Nychis et al., 2008), machine learning and data mining (Lee and Stolfo, 1998; Portnoy et al., 2001; Lakhina et al., 2005), signal processing (Barford et al., 2002; Thottan and Ji, 2003), statistical analysis (Kruegel et al., 2003; Simmross-Wattenberg et al., 2011; Thatte et al., 2011) and pattern recognition (Fontugne and Fukuda, 2011). We refer to a recent survey for discussions on the applications of some of these approaches (Bhuyan et al., 2014).

A rule-based system to detect penetrations by modeling them as state transitions was proposed in Ilgun et al. (1995). But, this was developed for the Unix system where actions or state changes occur on the execution of commands by the attacker. Besides, rule-based systems do not adapt to the fast evolving nature of network traffic.

Entropy has been used to evaluate features for network anomaly detection, in particular to understand the correlation of different features (both header features and behavioral features), emphasizing the need to select features with care (Nychis et al., 2008). One of the well-known works which builds on the principle of maximum entropy to detect anomalies in network is developed in Gu et al. (2005). Features are added one by one iteratively, such that in each step, the right weight for each feature is estimated using KL (Kullback–Leibler) divergence. Learning the parameters and thereby building a model from a given data, KL divergence is again used to detect anomalies. Clustering based anomaly detection solutions that used unlabeled data have also been proposed in the past (Portnoy et al., 2001; Leung and Leckie, 2005; Jiang et al., 2006). For example, Portnoy et al. (2001) develops a variant of single-linkage clustering to build normal clusters in unlabeled data, and subsequently use the clusters to detect anomalies.

A number of statistical techniques have also been explored in the past. In Simmross-Wattenberg et al. (2011) traffic was modeled using α-stable distributions, and anomalies such as flash and flash crowds are detected by comparing trained traffic windows with the test windows using the Generalized Likelihood Ratio Test (GLRT). This solution depends on labeled data; in addition, it is also computationally expensive.

More recent works explore traffic at multiple time resolutions and also deploy an ensemble of techniques. One such work is the unsupervised anomaly detection system developed in Casas et al. (2012); time-series are build at different time scales, and outlier traffic flows are ranked after they are identified using a combination of cluster techniques (namely, Sub-Space Clustering, Density-based Clustering and Evidence Accumulation Clustering). While the solution is shown to detect anomalies and attacks, the ability to detect more sophisticated attacks beyond DoS and DDoS attacks is not known.

Different from the above works, our focus is explicitly on gathering evidences by modeling and analyzing network traffic. By gathering multiple evidences, the solution is able to reduce false positives to acceptable levels. The framework we develop also does not require learning or building models based on normal traffic. Besides, our solution is more general in design (and not tailored for any specific attack), as the patterns being detected are fundamental to different malicious activities.

Framework for evidence gathering

Fig. 1 illustrates the framework that we develop for gathering evidences of and related to attacks in network traffic. Our system takes the following three-stage approach to detect attack sequences:

1. Model sessions of traffic flows, and detect anomalous patterns.
2. Detect network and port scans, as well as illegitimate TCP state sequences.
3. Gather and correlate anomalous patterns, and make final decision on traffic (whether it is anomalous or not).

We explain each of the above stages below.

Stage 1: Modeling and analyzing sessions to detect anomalous patterns

To start with, we define flows and sessions. A flow is a set of packets, localized in time, with the same five tuple of source and

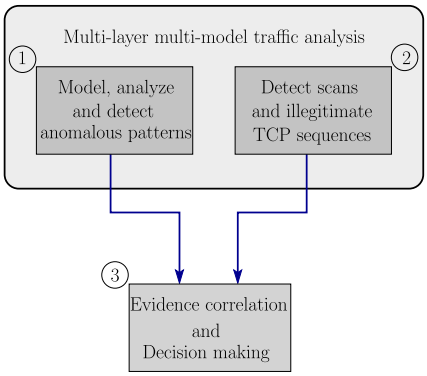


Fig. 1. Illustration of evidence-gathering framework.

destination IP addresses, source and destination ports, and protocol. Two flows with the same five tuple are identified as different flows if there is sufficient inactivity period separating them. Every TCP connection is a different flow.

A session is usually defined as a set of flows such that the inter-arrival time between any two subsequent flows is less than a given value (which is obviously higher than the inter-arrival time distinguishing two consecutive flows with the same five-tuple). Therefore, two similar sessions are distinguished as separate if there is sufficient inactivity time-period between the two.

Often, it is meaningful to define session at a coarser level, say using a three tuple instead of a five tuple. For example, to analyze traffic at a popular web server, session is defined by the three tuple of destination IP address, destination port and protocol. Thus, a session can be refined by identifying a common IP address. Depending on the features of interest that help in the analysis, the IP address might be that of a source (for example, a bot trying to contact its master by attempting to establish connections to a number of different destinations) or that of a destination (a victim being flooded with attack traffic, for example).

Features for traffic representation

For any end-host in an enterprise network, we analyze network traffic in flows and sessions. Each traffic flow/session of an end-host can be analyzed using a number of features. We use the following three features in our work here:

1. Inter-arrival times of flows in a session: Irrespective of whether a session is identified using a common source IP address or destination IP address, the inter-arrival times (IAT) of flows in a session is an important feature that can be used to detect anomalies. Consecutive flows generated by bots are likely to arrive at fixed time intervals, or in other words, the randomness in the arrival times between flows is likely to be lesser than in a normal scenario. For HTTP flows, we also find it useful to derive a feature based on IAT, that may help to differentiate normal users from bots. We define activity measure,

$$A = \frac{\text{Median of IATs of flows} \times \text{Number of flow}}{\text{total duration of the session}}$$

2. Sizes of flows: The size of a flow can be measured in the number of packets or the number of bytes that constitute the flow. We refer to the former as flow-size in packets or FSP, and the latter as flow-size in bytes or FSB. For a given session of flows in a measurement interval, the distribution of flow-sizes may vary significantly between an anomalous session and a normal session. Analyzing these features at both the source and destination levels are useful for detecting anomalies and attacks.
3. Degree of an end-host: We define 'degree of an end-host' as the number of distinct IP addresses that an end-host communicates to, within an interval. This feature is obtained per session, and needs to be extracted at both source level and destination level. The example mentioned previously, of a bot trying to communicate to unusually higher number of destinations, could possibly be detected if the session is modeled using the out-degree of the source IP address. Similarly, a DoS attack can often be detected by modeling session using the of destination IP address.

Though the last two features have been shown to be effective in previous works (for example, Lee et al. (1999), Lakhina et al. (2004), Nychis et al. (2008) and Silveira et al. (2010)); the inter-arrival times between flows (and not between packets) have not been analyzed for anomaly detection (to the best of our knowledge). We realize that there are also other features that might be useful in detecting anomalies, an example being the traffic rate at different granularities. However, our aim here is to demonstrate the effectiveness of the evidence-gathering framework and the techniques developed. As we will see later, the features listed above are able to detect malicious traffic flows with high accuracy; and therefore in this work, we limit the scope of the traffic features to the above three. In Section "Regression analysis for detection of anomalous patterns", we describe the techniques used to detect anomalous patterns using the above features.

Stage 2: Detecting scans and illegitimate TCP state sequences

Attackers scan networks and servers to determine ports that are open, which also reveals the services running on the machines. Depending on the services, and most often the versions too, there might be possibilities of security breaches. For example, an unpatched or zero-day vulnerability of a particular service might be exploited by sending the right set of packets.

Instead of detecting only scans, we take a broader view and detect *illegitimate* TCP state sequences. A state sequence of a TCP flow gives the path taken by the flow in the TCP state transition diagram (finite state machine). Consider a flow that conforms to TCP's finite state machine (FSM), taking a common path in the state machine. Then a corresponding state sequence, for instance, could be `ShA{Da}*FafA`, where `S` denotes a `SYN` packet, `h` a `SYN+ACK` packet, `A` an acknowledgment packet (without data), `D` a data packet (with acknowledgment), and `F` a `FIN` packet (lower case represents one direction, and upper case the reverse direction). The sequence above represents a TCP flow with zero or more data transfers; but more importantly, it conforms to the connection establishment process and also terminates properly, and therefore can be called a legitimate flow. We refer to a sequence of states that do not conform to the FSM as illegitimate. Observe that all kinds of TCP based scans that result in incomplete connections are illegitimate sequences. By detecting illegitimate TCP sequences, we detect TCP flows that terminate without proper connection termination, SYN scans, SYN attacks, SYN + ACK scan, RESET attack, etc.

To identify the state sequence of a flow, we check the TCP information (in the header) of the concerned packet of a flow being tracked; and based on the current state, make transition to the next state of the flow. Essentially, our algorithm implements the finite state machine to record and identify the state sequence of TCP flows.

We note that, Stage 2 operates differently from Stage 1, primarily because, in Stage 2, the method used to detect an anomaly—an illegitimate TCP flow—is deterministic. That a TCP flow is

legitimate or illegitimate is a binary decision depending only on the FSM. On the other hand, the approaches (described later, in Section "Regression analysis for detection of anomalous patterns") used to detect anomalous patterns in Stage 1 are probabilistic in nature.

Stage 3: Evidence correlation and decision making

Anomalous patterns are detected in the first stage (Section "Stage 1: Modeling and analyzing sessions to detect anomalous patterns"), using multiple features and multiple techniques (Section "Regression analysis for detection of anomalous patterns"). To elaborate, the multiple anomalous patterns detected independently but related to one attack, might be due to i) the different stages of the attack, such as, reconnaissance (port scan), penetration (brute force) and exploit (successful intrusion), ii) manifestation of the attack in different features used for detection, or iii) the different techniques that detect the attack.

There are a few ways in which the relation between different detected anomalous patterns can be found out. Each anomalous pattern is identified by its space (IP address) and time. Therefore, an obvious way to find related anomalies (including illegitimate flows and scans) is to search for flows with the same IP addresses. Though correlation in time can also be used, it is not always the case that the related anomalies have temporal proximity. For example, after a successful dictionary attack, an attacker might decide to exploit the system after a few hours or days. At the other extreme, it is also possible to find related anomalies by finding the similarity of RTTs (round trip times) of the packets involved. On the other hand, correlation of gathered evidences in space is a more natural option, as this means using IP addresses to correlate the detected anomalous pattern. For example, a bot might scan a number machines, before establishing a session with its master; this would result in two sessions with same source IP address. In our current work, we correlate anomalous patterns using IP addresses.

We highlight that, evidences might also appear legitimate when analyzed in isolation. For example, a buffer overflow attack might be followed by the opening of a new port, to which the attacker establishes a new connection to login and perform the necessary tasks. Such a connection, though normal, both by protocol definition and statistically, is a malicious one that needs to be filtered out and analyzed.

Once multiple evidences are gathered, we need to take decision on a given set of traffic flows or sessions. In Section "Decision making based on evidences gathered", we define a meta-decision maker for this purpose.

Regression analysis for detection of anomalous patterns

Before discussing the techniques for detecting anomalous patterns, we list down important types of anomalies that need to be detected and also provide examples.

1. Outliers: A source that tries to access a relatively large number of destinations raises suspicion; and so does connections to a number of ports in a short interval. Depending on factors, such as the end-user at the source, the source network, etc., as well as temporal relations, the number of destination IP addresses or ports that a source connects to might vary. Therefore, having a static threshold for detecting outliers would fail, leading to either high false positives or low recall.
2. Constant and periodic count: Almost the same, if not exact, number of IP addresses, being accessed is a suspicious pattern. Another example is the presence of new periodic communication to a host; i.e., the inter-arrival times between flows is a constant. Such behaviors might be due to an infected machine or a bot. Visually, these kind of anomalies, when plotted using the appropriate feature, appear as a line with near-zero slope.
3. Increasing count: An example of increasing count is the number of machines an end-host is communicating with. This might be an attack starting at a slow rate, but with the rate increasing steadily with time, so as to defeat systems that learn models online. If we fit a line, the line would have steep slope. Such a behavior might be seen in both cases, one in which sessions are grouped by source IP address and the other in which sessions are grouped by destination IP address.

To detect anomalous patterns in network traffic, we use regression models. The primary reason for using regression models is that they help in analyzing and explaining the relationship between the response variable and the independent variable, an approach fundamental to data analysis that is essential while building evidences.

Our analysis here is based mainly on linear regression. Let $\{(X_1, Y1), (X_2, Y_2),\ldots,(X_n, Y_n)\}$ be a set of n data points of interest. $\{Y_1, Y_2,\ldots,Y_n\}$ may denote the observed data quantities (say, number of destination IP address accessed by a given source), and $\{X_1, X_2,\ldots,X_n\}$ may denote the discrete time points or sample index. If we assume a linear relationship between X_i and Y_i, then a first-order linear model is,

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon, \qquad i = 1, 2, \ldots, n; \tag{1}$$

where β_0 denotes the intercept and β_1 the slope. The random error term ε accounts for the fact that observed values for Y_i's may not fall exactly on a straight line. ε is modeled using Normal distribution, specifically with zero mean and constant variance; $\varepsilon \sim \mathcal{N}(0, \sigma)$. The classical method used for fitting a regression line is using the method of *least squares* (refer Kutner et al. (2003) for detailed description). Let $\widehat{Y}$ denote the estimated regression function for the covariate X,

$$\widehat{Y} = \widehat{\beta}_0 + \widehat{\beta}_1 X. \tag{2}$$

The difference between the observed value and the predicted value is the residual, $r_i = Y_i - \widehat{Y}_i$. The regression coefficients $\widehat{\beta}_0$ and $\widehat{\beta}_1$ are solutions obtained by minimizing the *sum of squared errors*,

$$\mathrm{SSE} = \sum_{i=1}^{n} r_i^2. \tag{3}$$

An important property of the least squares (LS) method is that, $\widehat{\beta}_0$ and $\widehat{\beta}_1$ are unbiased estimators with minimum variance among all linear estimators (Gauss–Markov theorem).

Based on the above model, we develop techniques to answer the following questions, that will allow us to detect different kinds of anomalies in the data extracted from network traffic.

1. Are there outliers in the data?
2. Does the regression line fit too good to raise suspicion?
3. Does the estimated slope indicate a clear trend of increasing activity?
4. Is the LS regression not the right fit; and is a higher order model a better fit?

Fig. 2 gives examples of different anomalous patterns corresponding to the questions listed above. These examples were extracted from real network traffic obtained from the 24-h-long trace of 12 October 2014 from the MAWI repository of the WIDE project (The WIDE Project). Next, we present the corresponding techniques for analyzing data using the regression model.

Detection of outliers

While an outlier is in itself an anomaly in traffic, it is important to model and analyze the data again after removing the outlier, as further insights might be obtained. However, LS regression is known to be highly sensitive to the presence of outliers; indeed, its breakdown point is $1/n$ for finite sample size of n, and zero asymptotically. Breakdown point is the smallest percentage of outliers that can cause the estimator to deviate arbitrarily farther (we refer readers to the book by Rousseeuw and Leroy (1987) and references therein for detailed study of well-known results related to this section).

The literature has a number of robust regression techniques that deal with outliers in data. One such robust regression is the Theil-Sen (TS) estimator (Theil, 1950; Sen, 1968), which has a high breakdown point of $\approx$29.3%, as well as high asymptotic efficiency. Like the LS estimator, TS estimator is also an unbiased estimator. In TS regression, the slope is estimated as the median of all the combinations of slopes in the given dataset; that is,

$$\widehat{\beta_1} = \underset{1\le i<j\le n}{\text{median}} \left\{ \frac{Y_j - Y_i}{X_j - X_i} \right\}$$

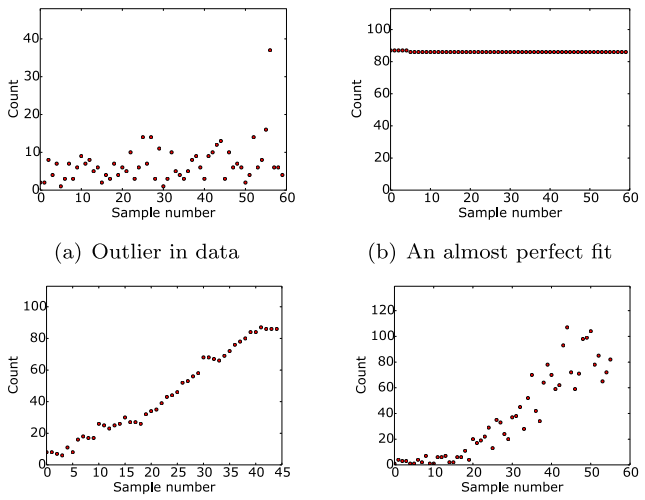
The computational cost of the intuitive algorithm for slope computation is $\mathcal{O}(n^2)$; however there are ways to compute the median in $\mathcal{O}(n \log n)$ time (see Katz and Sharir (1993)). The intercept is then estimated as,

$$\widehat{\beta}_0 = \underset{i}{\text{median}} \ \{Y_i\} - \widehat{\beta}_1 \ \underset{i}{\text{median}} \{X_i\}.$$

To detect outliers, we measure how far the residuals r_i's are based on the TS estimator. As TS estimator has a breakdown point of $b = 1 - \frac{1}{\sqrt{2}} \approx = 0.293$, a strict approach is to consider all points beyond the jth quantile of residuals $Q(j)$ as outliers, where $j = 1-b$. We define the following hypothesis test:

$$\begin{cases} \mathcal{H}_0 & : r_i^{\text{TS}} < Q(j(1+\kappa)), \text{ inlier} \\ \mathcal{H}_1 & : r_i^{\text{TS}} \ge Q(j(1+\kappa)), \text{ outlier} \end{cases} \tag{4}$$

where $0 \le \kappa \le b$ is a control parameter available for deciding how strict the system should be in marking outliers—higher the value of κ, lesser the number of points marked as outliers.



(a) Outlier in data (b) An almost perfect fit

(c) Linear relationship (d) Steep rise in count

Fig. 2. Examples for different anomalous patterns present in network traffic.

Goodness of fit

Once the outliers are detected, they are removed, and the remaining data $(\acute{\mathbf{X}}, \acute{\mathbf{Y}})$ is fitted using LS regression; that is the estimator minimizes the sum of the squared residuals. There are a few ways to measure the goodness of fit of the regression line; we use the sum of the squared errors, SSE given in Eq. (3), as the measure.

Observe that, as the sum of squared errors approaches zero, the line fits almost perfectly. If SSE is indeed zero, then there exists a functional relationship between the two variables, of the form $Y = f(X)$. In our case, this in fact indicates some suspicion—in network traffic, we expect a statistical relationship rather than a functional relationship between X and Y. A functional relationship arises due to automated communications, and hardly due to manual communications. Therefore, we use a threshold for SSE, to detect a perfectly correlated set of points. Though this method of testing using SSE would also detect a regression line of zero slope, such a set of points (see Fig. 2(b)) is also suspicious in network traffic.

Inference on the slope

A steep slope in the data denotes the third type of anomalies. Let θ denote the threshold of the estimated slope beyond which the fitted line indicates a suspicious behavior. So, we have two hypotheses:

$$\begin{cases} \mathcal{H}_0 \ : \ |\beta_1| \le \theta; & \text{not an anomaly} \\ \mathcal{H}_1 \ : \ |\beta_1| > \theta; & \text{anomaly} \end{cases}$$

As the error ε follows Normal distribution, the least squares estimator of the slope, $\widehat{\beta}_1$ has a normal distribution with mean β_1 and variance $\sigma^2_{\widehat{\beta}_1}$, where,

$$\sigma^2_{\widehat{\beta}_1} = \frac{\sigma^2}{\sum_{i=1}^{n} (X_i - \overline{X})^2}.$$

That is,

$$\sqrt{s_{xx}} \ \frac{\widehat{\beta}_1 - \beta_1}{\sigma} \sim \mathcal{N}(0,1) \tag{5}$$

where $s_{xx} = \sum_{i=1}^{n} (X_i - \overline{X})^2$. However, σ is unknown; therefore we replace it with its estimator s:

$$s = \sqrt{\frac{\text{SSE}}{n-2}}.$$

Due to the estimation of σ, the statistic follows Student's t distribution, admitting $n-2$ degrees of freedom (as regression coefficients themselves need to be estimated); i.e.,

$$\sqrt{s_{xx}} \ \frac{\widehat{\beta}_1 - \beta_1}{s} \sim t_{n-2} \tag{6}$$

At significance level α%, the rejection criterion for the null hypothesis $\mathcal{H}_0$ is:

`reject` $\mathcal{H}_0$ `if`

$$\widehat{\beta}_1 \notin \left[-t_{n-2,1-\alpha/2} \frac{s}{\sqrt{s_{xx}}} - \theta, t_{n-2,1-\alpha/2} \frac{s}{\sqrt{s_{xx}}} + \theta \right]. \tag{7}$$

We will use the above test to make inference on the slope.

Quadratic regression

Our final test is to check if the simple linear model (used above) is a good and sufficient fit. Even if a statistical test concludes that

the estimated slope $\widehat{\beta}_1 \neq 0$, it does not mean that a linear relationship exists between the variables; for instance, an exponential curve might fit be a better fit. Therefore, we find it necessary to compare the LS fitted model with, say, polynomial models. We consider a polynomial of degree two for fitting, and compare against the linear model of Eq. (1); our aim here is only to test if a higher order polynomial is a better fit for the data.

A quadratic model is represented by the following equation:

$$Y_i = \alpha_0 + \alpha_1 X_i + \alpha_2 X_i^2 + \varepsilon, \qquad i = 1, 2, \ldots, n; \tag{8}$$

where ε denotes the random error, and is modeled using a Normal distribution with zero mean and constant variance. Note that the response variable is still a linear function of the parameters, α_0, α_1, and α_2.

To determine which of the two models better estimates Y, we compute the coefficient of determination, R^2. The coefficient of determination quantifies the proportion of variation in data captured by the regression model.

$$R^2 = \frac{\sum_i \left(\widehat{Y}_i - \overline{Y}\right)^2}{\sum_i (Y_i - \overline{Y})^2} = 1 - \frac{\sum_i \left(Y_i - \widehat{Y}_i\right)^2}{\sum_i (Y_i - \overline{Y})^2}, \tag{9}$$

where $\overline{Y}$ denotes the mean of the response variable. However, as we are comparing two models with different number of regression coefficients, we use the adjusted coefficient of determination, R_a^2.

$$R_a^2 = 1 - \left(\frac{n-1}{n-p}\right) \frac{\sum_i \left(Y_i - \widehat{Y}_i\right)^2}{\sum_i (Y_i - \overline{Y})^2}, \tag{10}$$

where p is the number of regression coefficients. A given set of point is considered anomalous if the quadratic regression provides a better fit than the linear regression. Let R_{LS}^2 and R_{QR}^2 denote the adjusted coefficient of determination obtained using, the LS regression and the quadratic regression, respectively. For detecting anomalies, we define the following hypothesis test:

$$\begin{cases} \mathcal{H}_0 & : R_{\text{QR}}^2 - R_{\text{LS}}^2 \leq \theta_r, \text{ normal} \\ \mathcal{H}_1 & : R_{\text{QR}}^2 - R_{\text{LS}}^2 > \theta_r, \text{ anomaly} \end{cases} \tag{11}$$

where θ_r is a user-defined threshold.

Decision making based on evidences gathered

Each of the techniques presented in the previous section is potentially useful in detecting multiple anomalous patterns present in network traffic, as the traffic is represented using different features. However, if the decision on detection of traffic related to attacks is based on isolated and independent analyses of patterns, i.e., if we decide to raise an alert whenever there is any anomalous pattern detected using the techniques employed, we are likely to experience large number of false positives. False positives in this context are those normal traffic flows/sessions that are incorrectly classified as being anomalous. High false-positive rate makes a solution inefficient to use.

Observe that, for each of the features that we want to use for analysis, it could lead to the detection of anomalous patterns using any one of the techniques described in Section "Regression analysis for detection of anomalous patterns". Therefore, an anomalous pattern relates to a unique feature and a unique detection technique. Given this premise, we define a *meta-decision maker* or MDM for short. The MDM decides whether a traffic session (i.e., a set of flows) is anomalous or not, based on the following: *a session is classified as anomalous, if at least three anomalous patterns related to this session are detected; moreover, at least two of such patterns should have 'high' scores*. Recall that, we perform spacial correlation to find related sessions (refer Section "Stage 3: Evidence correlation and decision making").

Each technique for detecting anomalous pattern has a threshold, with respect to which, we make decision on whether an individual pattern is anomalous or not. For outlier detection (Section "Detection of outliers"), there is a threshold that is used in the hypothesis testing for deciding if the given set of points form an anomalous pattern. In the second technique (Section "Goodness of fit"), the SSE is expected to be close to zero for the pattern to be classified as anomalous. The slope, in the third technique (Section "Inference on the slope"), if greater than a given threshold raises suspicion. The difference in the adjusted coefficients of regression is compared with a threshold for hypothesis testing, to detect anomalous patterns in the fourth technique (Section "Quadratic regression"). For each technique, its corresponding threshold has a range such that, the probability of anomaly is the lowest at one end and the highest at the other. To compare them on the same scale, we normalize the threshold range for each technique to be [0,1] (irrespective of the actual values for these parameters). The normalized value of the test output is what we define as the *score*. The higher the score, the higher the probability of the corresponding anomalous pattern being a reliable evidence.

We point out that, the activity measure defined in Section "Features for traffic representation", is compared against a threshold for evidence detection in the associated traffic, as there are not multiple values to apply a regression model. We form a range for this threshold as well, so as to define an evidence score.

In essence, the MDM in our framework, classifies a traffic session as malicious if there are sufficient number of promising evidences.

Performance evaluations

In this section, we evaluate the ability of our evidence-gathering framework in detecting activities related to malicious activities. In the following, we describe the data used for the purposes of this work. In Section "Settings", the experimental settings are given. We present the results in Section "Results".

Data

For performance evaluations, we constructed a dataset from multiple sets of benign and malicious traffic. The reason for selecting malware traffic instead of any specific traffic, is because, malwares are involved in different kinds of attacks, such as DoS attack, hosting of criminal/malicious websites, spreading of worms and viruses, etc. Besides, different malware have different kinds of activities; therefore, it is meaningful to test our evidence-gathering framework on the traffic generated by malware.

We limit our study to only HTTP traffic, as the number of sessions due to other traffic is low. We describe the datasets below.

Normal traffic

The normal traffic data is obtained from three sources. The first source is the Internet traffic of two of the co-authors, who used secured Linux machines with common applications. We captured the Internet communications from these two users for a period of ten days. This traffic is used as part of the normal dataset. Most of these traffic flows are from web browsing activity; we ignore non-HTTP sessions (as mentioned previously). The remaining two sets are those that are publicly available for download. We use part of

the ISCX Intrusion Detection Evaluation DataSet (UNB ISCX, 2016), which contains simulated traffic mimicking normal user behavior. We also use normal datasets from the Lawrence Berkeley National Lab (LBNL) (LBNL Enterprise Trace Repository, 2005) which presents real traffic of a medium-sized enterprise network. The traffic is assumed to be non-malicious.

To not be biased by traffic data (and the related inaccuracies) from any one particular source, we mix traffic from all the sources to construct the normal traffic dataset.

Malware traffic

We obtain our malware traffic dataset from Stratosphere IPS Project (2016). The Stratosphere IPS Project provides a repository of malware traffic obtained via infecting host machines with malware. This repository consists of a variety of malware traffic related to different botnets. The botnets that are related to the malware traffic we use in our dataset include, Andromeda, Barys, Emotet, Geodo, Htbot, Miuref, Necurse, Sality, Vawtrak, Yakes and Zeus. Zeus is reported to be one of the top threats in the world from various sources (Top 10 Botnet Threats in US, 2012; Top 5 Scariest Zombie Botnets, 2014). Andromeda is one of the longest running and most prevalent malwares to have existed (Andromeda under the microscope, 2016). Sality is also one of the longest living threats and has more than two million infections per day (Sality, 2015). The traffic datasets provided by the Stratosphere IPS Project are not limited to the above. However, we selected the above based on the criteria that traffic due to one malware should be for at least one day; besides, we ignore non-HTTP sessions in this study.

For further details on the malware traffic, refer Stratosphere IPS Project (2016).

Settings

Regarding the configuration of parameters of the regression techniques, observe that, for each technique, there is one parameter range corresponding to each feature; in other words, there is one parameter range for each pattern (including A defined above). While these parameters do affect the detection of anomalous patterns, because of the higher level MDM defined in Section "Decision making based on evidences gathered", we have the luxury in setting a conservative range for each. This allows us to control the detection accuracies using the scores. Therefore, instead of presenting the range values for the parameters, we present the score values used for detection.

We define a score greater than or equal to 0.7 as 'high score' for each pattern (and, values less than 0.7 are considered as 'low scores'). Recall that, the score is in range [0,1]. Coming to the traffic due to illegitimate TCP state sequences, we set three score categories, depending on the number of such state sequences. If the number is less than 100, the score is 0, and if the number is between 100 and 1000, the score is 0.5, otherwise the score is 1.0.

In the following, traffic data is analyzed in sessions, where a session is identified as an aggregation of flows with the same three-tuple of source IP address, destination IP address and destination port number. Multiple sessions may have the same three-tuple values if they are sufficiently separated in time (24 h in the experiments here).

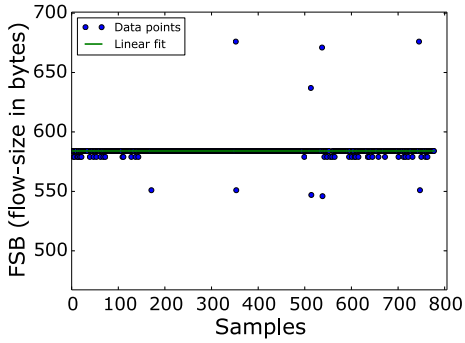
Results

The results presented here are the best of the three sets of range parameters we considered. However, we also highlight that the differences in the performance metrics are not significant; we attribute this behavior to the specific criteria we used for detection, that is based on the number of evidences and the scores.

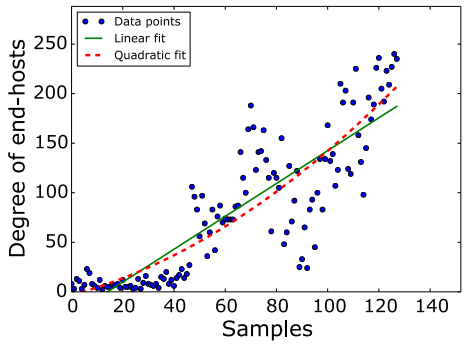
In the dataset, there were 969 benign sessions due to normal traffic, and 1397 malware generated traffic sessions. The overall detection rate of malware generated traffic sessions is 82.6%, whereas the false positive rate is a low 7.9%. Fig. 3 presents two examples of anomalous patterns that were detected in the malware traffic dataset. In Fig. 3(a), the FSB (flow-size in bytes) of an Andromeda bot traffic session is shown. Each data point in the graph represents a flow within the traffic session. This session is detected as anomalous because the SSE (Section "Goodness of fit") is zero after removing the outliers (Section "Detection of outliers"). Fig. 3(b) gives an example of the outdegree values of an end-host infected with Geodo-related malware. Each data point in the graph is the outdegree of the end-host in a 10-min interval. A quadratic model has a better fit than a linear model in the above figure (Section "Quadratic regression"), indicating an abnormal change in traffic. We highlight that, test was defined to check if a higher-order model fits better than the linear model, and not to strictly check if a quadratic model fits the data.

In Table 1, we present the detection results for malware traffic sessions belonging to each botnet category. Observe that, for four botnet categories, the detection rate is 100%. Besides, the traffic sessions related to Andromeda and Sality botnets, have high detection accuracies of, 89.2% and 98.9%, respectively. Note that, the counts of traffic sessions related to these botnets are also high. There are also a few botnet categories such as, Htbot, Miuref, Geodo and Yakes which have relatively lower detection rates. We will discuss the possible reasons for the lower detection rates on these bots below.

Next, we present histograms showing the number and percentage of sessions having 0–5 evidences, for both the normal traffic sessions and malware traffic sessions, in Figs. 4 and 5, respectively. For each of these figures, the Y-axis on the left represents the number of traffic sessions and the Y-axis on the right represents the percentage of total number of sessions. The X-axis



(a) An Andromeda bot traffic session



(b) Outdegree of a Geodo bot end-host

Fig. 3. Examples of anomalous patterns.

Table 1
Malware traffic detection results.

Related botnet	Total number of sessions	Detected sessions	Detection rate
Andromeda	148	132	89.2%
Barys	16	16	100.0%
Emotet	95	95	100.0%
Geodo	63	44	69.8%
Htbot	287	171	59.6%
Miuref	82	44	53.7%
Necurse	19	19	100.0%
Sality	440	435	98.9%
Vawtrak	40	40	100.0%
Yakes	39	25	64.1%
Zeus	168	133	79.2%

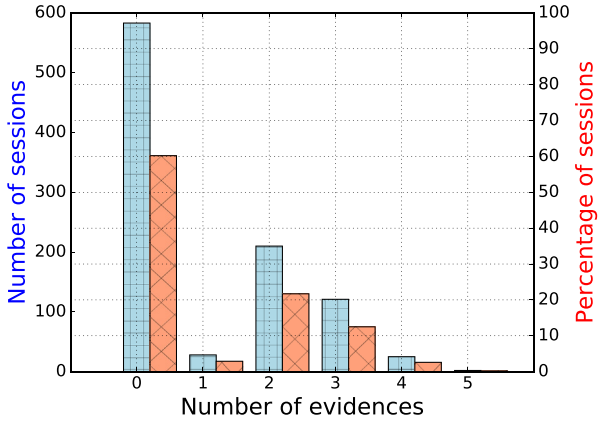


Fig. 4. Histogram of evidences for normal traffic sessions.

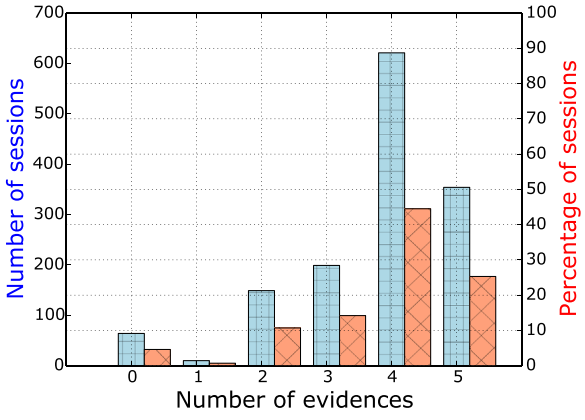


Fig. 5. Histogram of evidences for malware traffic sessions.

indicates the number of evidences found for a session. Thus, for each number of evidence x, the left bar represents the number of traffic sessions for which exactly x evidences were detected. The right bar for the same x gives the percentage of total sessions (of normal traffic in Fig. 4 and of malware traffic in Fig. 5) with exactly x evidences detected. A non-zero score in each anomalous pattern counts as an evidence. Recall that our detection criteria is that a session must have least three evidences in total and at least two evidences with a high score.

From Fig. 4, it can be observed that the number of sessions having two or less evidences, and hence are not falsely detected, make up most of the sessions for the normal traffic sessions. In addition, the percentage of sessions with zero evidences is greater than 60%. This shows that each individual pattern itself provides good measure for detecting anomalous traffic sessions. On the other hand, for the malware traffic sessions, Fig. 5 shows that most sessions have least three evidences—sessions having three or more evidences form 84% of the total malicious traffic sessions. Sessions that have four and five evidences make up more than half of the total sessions which indicates that all of our patterns are very relevant in detecting anomalous sessions. Thus, these correspond to the overall detection rate and false positive rate we presented earlier.

Earlier, we mentioned about the lower detection rates for a few bot categories (Htbot, Miuref, Geodo, Yakes). The undetected sessions belonging to these bot categories amount to approximately 14% of the total malware sessions. Among these undetected sessions, at least half had two or more evidences of anomalous patterns. While we could change our criteria to detect these missed out malware traffic sessions, the trade-off would be a higher false positive rate. Instead, our current detection criteria allowed us to maintain a low false positive rate on the normal dataset. To validate, we carried out another set of experiments, where we changed the decision criteria to detect any session that has two or more evidences, with at least one of them having high scores (greater than on equal to 0.7). We obtained detection accuracy of 93.9%, but with considerably high false positive rate of 26.8%.

We also observed that, for each bot category in our evaluation dataset, there exist traffic sessions generated by different malwares. For example, sessions belonging to the Htbot category are generated by three malwares with different hash identifiers. Thus for botnets like Htbot and Miuref, there might be specific versions of malware which implement certain evasion techniques, and hence have lead to lesser number of evidences found. This can be possibly be improved by modeling and analyzing the traffic with additional features, consequently increasing the patterns that could be detected.

Effectiveness of features

Table 2 gives the contribution of each feature as well as illegitimate TCP flows and scans to the detection of malware traffic sessions. Of the 1154 sessions detected by our framework, the first row gives the count of sessions to which a particular feature contributed. The second row gives the percentage of malware traffic sessions for which the feature contributed an evidence. Notice that, while all the features were useful in the detection of most of the malware traffic sessions, the illegitimate TCP flows and scans were present only for half the sessions.

Effectiveness of techniques

We analyzed the effectiveness of various regression techniques in detecting the malware traffic sessions; and this is summarized in Table 3. (In the table, the corresponding sections, where the

Table 2
Contribution of different features to detected sessions.

Detected sessions		Illegitimate TCP flows	IAT	FSP	FSB	Degree
#	1154	609	1129	1090	969	978
%	100	52.8	97.8	94.5	84	84.7

Table 3
Effectiveness of different regression techniques in detecting malware traffic sessions.

Detected sessions		Outlier (Section "Detection of outliers")	Goodness of Fit (Section "Goodness of fit")	Linear or Quadratic (Sections "Inference on the slope" and "Quadratic regression")
#	1154	1008	1154	94
%	100	87.3	100	8.1

techniques are defined, are indicated in parenthesis.) Note that, curve fitting can be either linear or quadratic, but not both; i.e., these two are mutually exclusive and hence grouped together in the table. We observe that the linear and quadratic model fitting provided the least number of evidences; yet they resulted in the detection of 94 sessions due to malwares. We also expect curve fitting regression techniques to be useful in more sophisticated attacks, where the rate is slowly increased to outwit anomaly detection techniques based on abrupt-change detection or even naive prediction.

Computational time

We estimated the time taken to process the dataset used in our experiments. The dataset consist of ≈1.6 million flows. On a desktop machine running Intel Xeon W3690 CPU @ 3.47 GHz and 12 GB RAM, the total time taken by our framework (a single-threaded implementation) to process the dataset is approximately 550 s. In other words, close to 3000 flows were processed per second, which is good enough to meet the requirements of a large enterprise network. However, the framework needs to be optimized for guaranteeing performance at packet level; we consider this as part of our future work.

Conclusions

In this paper, we proposed a framework for gathering evidences to detect traffic sessions related to attacks and malicious activities. Our framework does not require to learn characteristics of normal traffic. Instead, we applied regression models to detect fundamental anomalous patterns. Experiments were performed on a dataset from different sources, consisting of both normal traffic sessions as well as malware traffic sessions. Our solution achieved high accuracy in detecting malware traffic sessions, while maintaining acceptable low false positive rate.

As a next step, we are looking forward to extending our solution to identify evidences for different stages of attacks, such as reconnaissance, penetration and (successful) exploit. Such an enhanced solution will have the ability to not only detect attacks, but also to identify and profile attacks. In future, we will also experiment with more features as well as other information for correlation, for example proximity in IP addresses (Invernizzi et al., 2014), malicious domains being accessed, etc.

We also plan to enhance our framework so that it works real-time on live traffic. To achieve this, there are multiple challenges to be addressed. One, packets need to processed in real-time at the rate of link capacity, and their corresponding sessions (and flows) need to be updated. While older sessions are stored in the database, recent and ongoing ones need to be stored in RAM using efficient caches. Two, real-time analysis of multiple sessions, at different granularities, using different features, need to be performed. Fast algorithms for analysis are required as the number of sessions can be high and the session size (in terms of number of data points) can be large.

Acknowledgment

This material is based on research work supported by the Singapore National Research Foundation under NCR Award No. NRF2014NCR-NCR001-034.

References

Andromeda under the microscope, 2016. https://blog.avast.com/andromeda-under-the-microscope.
Barford, Paul, Kline, Jeffery, Plonka, David, Ron, Amos, 2002. A signal analysis of network traffic anomalies. In: Proc. of the ACM SIGCOMM Workshop on Internet Measurment, IMW '02, pp. 71–82.
Bhuyan, M.H., Bhattacharyya, D.K., Kalita, J.K., 2014. Network anomaly detection: methods, systems and tools. IEEE Commun. Surv. Tutor. 16 (1), 303–336.
Carl, G., Kesidis, G., Brooks, R.R., Rai, Suresh, Jan 2006. Denial-of-service attack-detection techniques. IEEE Internet Comput. 10 (1), 82–89.
Casas, Pedro, Mazel, Johan, Owezarski, Philippe, April 2012. Unsupervised network intrusion detection systems: detecting the unknown without knowledge. Comput. Commun. 35 (7), 772–783.
Chandola, Varun, Banerjee, Arindam, Kumar, Vipin, 2009. Anomaly detection: a survey. ACM Comput. Surv. (CSUR) 41 (3), 15.
Cisco Annual Security Report, 2016. http://www.cisco.com/c/m/en_us/offers/sc04/2016-annual-security-report/index.html.
Feily, Maryam, Shahrestani, Alireza, Ramadass, Sureswaran, 2009. A survey of botnet and botnet detection. In: Proc. of the 2009 Third International Conf. on Emerging Security Information, Systems and Technologies, SECURWARE '09, pp. 268–273.
Fiadino, P., D'Alconzo, A., Bär, A., Finamore, A., Casas, P., Sept 2014. On the detection of network traffic anomalies in content delivery network services. In: Proc. 26th International Teletraffic Congress (ITC), pp. 1–9.
Fontugne, Romain, Fukuda, Kensuke, August 2011. A Hough-transform-based anomaly detector with an adaptive time interval. SIGAPP Appl. Comput. Rev. 11 (3), 41–51.
Gu, Yu, McCallum, Andrew, Towsley, Don, 2005. Detecting anomalies in network traffic using maximum entropy estimation. In: Proc. of the 5th ACM SIGCOMM Conf. on Internet Measurement, IMC '05, pp. 345–350.
Ilgun, Koral, Kemmerer, R.A., Porras, Phillip A., March 1995. State transition analysis: a rule-based intrusion detection approach. IEEE Trans. Softw. Eng. 21 (3), 181–199.
Invernizzi, Luca, Miskovic, Stanislav, Torres, Ruben, Kruegel, Christopher, Saha, Sabyasachi, Vigna, Giovanni, Lee, Sung-Ju, Mellia, Marco, 2014. Nazca: detecting malware distribution in large-scale networks. In: Proc. 21st Annual Network and Distributed System Security Symposium, NDSS.
Jiang, ShengYi, Song, Xiaoyu, Wang, Hui, Han, Jian-Jun, Li, Qing-Hua, 2006. A clustering-based method for unsupervised intrusion detections. Pattern Recognit. Lett. 27 (7), 802–810.
Katz, Matthew J., Sharir, Micha, 1993. Optimal slope selection via expanders. Inf. Process. Lett. 47 (3), 115–122.
Kruegel, Christopher, Mutz, Darren, Robertson, William, Valeur, Fredrik, 2003. Bayesian event classification for intrusion detection. In: Proc. 19th Annual Computer Security Applications Conf., ACSAC '03.
Kutner, Michael, Nachtsheim, Christopher, Neter, John, 2003. Applied Linear Regression Models, 4 edition. McGraw-Hill Higher Education.
Lakhina, Anukool, Crovella, Mark, Diot, Christiphe, 2004. Characterization of network-wide anomalies in traffic flows. In: Proc. ACM SIGCOMM Conference on Internet Measurement (IMC), pp. 201–206.
Lakhina, Anukool, Crovella, Mark, Diot, Christophe, 2005. Mining anomalies using traffic feature distributions. In: Proc. ACM SIGCOMM '05, pp. 217–228.
LBNL Enterprise Trace Repository, 2005. http://www.icir.org/enterprise-tracing.
Lee, Wenke, Stolfo, Salvatore J., 1998. Data mining approaches for intrusion detection. In: Proc. 7th Conf. on USENIX Security Symposium – Volume 7, SSYM'98.
Lee, Wenke, Stolfo, S.J., Mok, K.W., 1999. A data mining framework for building intrusion detection models. In: Proc. IEEE Symposium on Security and Privacy, pp. 120–132.
Leung, Kingsly, Leckie, Christopher, 2005. Unsupervised anomaly detection in network intrusion detection using clusters. In: Proc. 28th Australasian Conf. on Computer Science – Volume 38, ACSC '05, pp. 333–342.
Nychis, George, Sekar, Vyas, Andersen, David G., Kim, Hyong, Zhang, Hui, 2008. An empirical evaluation of entropy-based traffic anomaly detection. In: Proc. 8th ACM SIGCOMM Conf. on Internet Measurement, IMC '08, pp. 151–156.
Peng, Tao, Leckie, Christopher, Ramamohanarao, Kotagiri, April 2007. Survey of network-based defense mechanisms countering the DoS and DDoS problems. ACM Comput. Surv. 39 (1).
Portnoy, L., Eskin, E., Stolfo, S., 2001. Intrusion detection with unlabeled data using clustering. In: Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA).
Rousseeuw, P.J., Leroy, A.M., 1987. Robust Regression and Outlier Detection. John Wiley & Sons, Inc.
Sality, 2015. https://www.botconf.eu/2015/sality/.
Sen, Pranab Kumar, 1968. Estimates of the regression coefficient based on Kendall's Tau. J. Am. Stat. Assoc. 63 (324), 1379–1389.
Silveira, Fernando, Diot, Christophe, Taft, Nina, Govindan, Ramesh, 2010. ASTUTE: detecting a different class of traffic anomalies. In: Proc. ACM SIGCOMM, pp. 267–278.
Simmross-Wattenberg, F., Asensio-Perez, J.I., Casaseca de-la Higuera, P., Martin-Fernandez, M., Dimitriadis, I.A., Alberola-Lopez, C., July 2011. Anomaly detection in network traffic based on statistical inference and alpha-stable modeling. IEEE Trans. Dependable Secure Comput. 8 (4), 494–509.
Stratosphere IPS Project, 2016. https://stratosphereips.org/category/dataset.html.
Thatte, Gautam, Mitra, Urbashi, Heidemann, John, April 2011. Parametric methods for anomaly detection in aggregate traffic. IEEE/ACM Trans. Netw. 19 (2), 512–525.
The WIDE Project. http://www.wide.ad.jp.
Theil, Henri, 1950. A rank-invariant method of linear and polynomial regression analysis. Indag. Math. 12, 85–91.

Thottan, M., Ji, Chuanyi, Aug 2003. Anomaly detection in IP networks. IEEE Trans. Signal Process. 51 (8), 2191–2204.
Top 10 Botnet Threats in US, 2012. http://www.enigmasoftware.com/top-10-botnet-threats-in-the-united-states/.
Top 5 Scariest Zombie Botnets, 2014. http://www.welivesecurity.com/2014/10/23/top-5-scariest-zombie-botnets/.
UNB ISCX, 2016. Intrusion Detection Evaluation DataSet. http://www.unb.ca/research/iscx/dataset/iscx-IDS-dataset.html.
Vincent Zhou, Chenfeng, Leckie, Christopher, Karunasekera, Shanika, 2010. A survey of coordinated attacks and collaborative intrusion detection. Comput. Secur. 29 (1), 124–140.