Feature trade-off analysis for reconnaissance detection.

KALUTARAGE, H.K. and SHAIKH, S.A.

2018

Chapter 1

Feature Trade-off Analysis for Reconnaissance Detection

Harsha Kumara Kalutarage† and Siraj Ahmed Shaikh‡

†*School of Computing Science & Digital Media, Robert Gordon University, Aberdeen, UK*
h.kalutarage@rgu.ac.uk

‡*Research Institute for Future Transport and Cities, Coventry University, CV1 5FB Coventry, UK*
s.shaikh@coventry.ac.uk

An effective Cyber Early Warning System (CEWS) should pick up threat activity at an early stage, with an emphasis on establishing hypotheses and predictions as well as generating alerts on (unclassified) situations based on preliminary indications. The design and implementation of CEWSs involve numerous challenges, such as a generic set of indicators, intelligence gathering, uncertainty reasoning, and information fusion. This chapter begins with an understanding of the behaviours of intruders, and then related literature is followed by a Bayesian-based methodology. It also includes a carefully deployed empirical analysis. Finally, the chapter concludes with a discussion on results, research challenges, and necessary suggestions to move forward in this research line.

1. Introduction

Traditional security solutions such as firewalls cannot assure the data, objects and resources restricted to unauthorised subjects. Such defensive approaches are increasingly insufficient for modern-day attacks as threat actors circumvent perimeter-based defences with a creative, stealthy, targeted and persistent manner that often goes undetected for significant periods. These attacks are multistage. If we can detect them *early* and respond quickly, then high impact cyber incidents can be avoided. An active approach for cyber defence is needed, which in turn needs to detect early stages of threat activities to deploy effective responses. A CEWS should serve such a goal.

One definition for a CEWS is that it "aims at alerting *unclassified* but potentially harmful system behaviour based on *preliminary indications* before possible damage occurs, and *contribute* to an integrated and aggregated situation report" [1]. Although there can be many overlaps between a typical intrusion detection system (IDS) and a CEWS, a particular emphasis for a CEWS is to establish hypotheses and predictions as well as to generate advice on unclassified activity based on preliminary indications [1].

This chapter sets off by attempting to present a generic attack model in Section 2. Section 3 presents an analytical approach building over prior work to threat monitoring [2–5]. Section 4 delves into the dataset and experimental setup used for this effort, and Section 5 describes the results of the proposed method for a lightweight CEWS. Section 6 provides an overview for related work. Section 7 concludes this chapter with some thoughts on open challenges in this area.

2. Cyber attack lifecycle

Modern attacks are multistage and producing evidence at each stage of the attack lifecycle. In principle, this evidence can be collected and analyse to alarm the attack, but difficult in practice as discussed in 7. The typical stages of a cyberattack lifecycle can be summarised as below.

Stage 1 **Reconnaissance:** The attacker conducts initial surveys on potential targets which can be either systems or people. Once the target identified, she starts to search for more specific information such as internet-facing services and individuals deciding which weapon to use. It could be a zero-day exploit, social engineering, spear-phishing, or even bribing an insider.

Stage 2 **Initial compromise:**: The attacker successfully executes malicious code on one or more systems on the target. She bypasses the perimeter defences and gains access to the internal network. It can be through a compromised system or user account, and attackers often use phishing for this purpose. Because exploiting user vulnerability is easier than exploiting software/hardware vulnerabilities.

Stage 3 **Command & control:** The attacker maintains continued control over the compromised system by installing a persistent backdoor. It could be via downloading additional utilities such as remote access Trojan (RAT) on to the victim system.

Stage 4 **Escalate privileges:** The attacker attempts to escalate privileges to gain enhanced access to the target systems and data. It could be via a technique like a pass the hash, keylogging, obtaining public key infrastructure (PKI) certificates, leveraging privileges held by an application, exploiting a vulnerable piece of software on the victim node or using any other method.

Stage 5 **Internal reconnaissance:** The attacker gains a better understanding of the environment, security at place, asserts and the roles and responsibilities of key subjects.

Stage 6 **Lateral movement:** The attacker compromises more systems and user accounts by moving between systems by using the access gained from previous stages. It could be accessing network shares, using task schedulers, remote desktop clients or virtual network computing. Since the attacker is often impersonating a legitimate user, evidence of their existence can be hard to find at this stage.

Stage 7 **Maintain presence:** The attacker installs multiple remote access entry points (e.g. malware back doors) and may have compromised several internal systems and user accounts to ensure that continued access to the environment. At this stage, she deeply understands the target environment, and within a proximity to reach her target at any time.

Stage 8 **Complete mission:** The attacker executes the final aspects of her mission. It could be stealing intellectual property, financial data or any other sensitive information, corrupting mission-critical systems in the business or disrupting the entire operations of the target business. Once the mission has completed, she might either leave the environment (without leaving evidence) or maintain access for returning in the future.

The notion of *early* for a CEWS can be explained using the above attack lifecycle. Notice the final stage (stage 8), it is the stage where exfiltration, corruption, and disruption happening. As a result, the cost to the business rises exponentially at this stage if the attack is not defeated. Any system that can alert the ongoing malicious attempt using preliminary indications before it reaches the final stage (i.e. mission completion) can be considered as a CEWS.

2.1. *Automated threat detection*

Modern cyber attacks don't just happen but evolve in a phased approach that includes early stages of reconnaissance and planning. It can take too long by humans to notice these activities, tipping the scales in favour of the criminals who want to break into our systems. Automated CEWSs are necessary as more devices rapidly get connected to the "Internet of Things" (IoT), including cars and homes. Automation increases the scalability and effectiveness of security monitoring. Due to the targeted nature, attack vector varies from one entity to another. Hence signature-based pattern matching techniques would not be useful in the detection of these attacks, and more sophisticated techniques are required. As shown in Figure 1, our ultimate goal is to employ cognitive technologies to "early detect". However, at this stage of the work, we propose a simple, but systematic behaviour analytic model suitable for early detection.

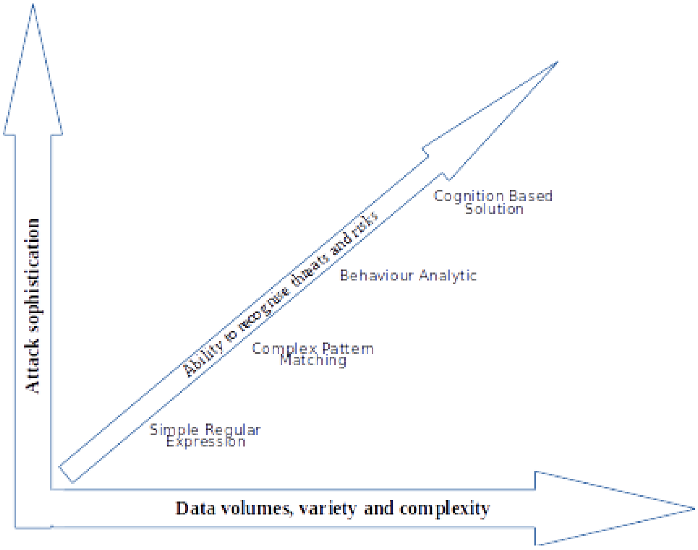
Attack sophistication
Ability to recognise threats and risks
Cognition Based Solution
Behaviour Analytic
Complex Pattern Matching
Simple Regular Expression
Data volumes, variety and complexity

Fig. 1.: Automated threat detection: attack sophistication vs detection technique to employ

3. Methodology (Resource efficient monitoring)

An incident occurs when an attack is carried out. Each incident area (e.g. scanning, compromising, malicious code) produces different indicators which spread out over time and space. Given the mean time to detect (MTTD) of a modern attack is about 200 days, it is necessary to maintain a long history of what is happening in the environment. Most systems cannot keep enough event data to track across extended time intervals due to the storage overheads. As a result, the scarcity of attack data within a short period allows the attacker to go undetected. On the other hand, as contemporary enterprise networks scale up in size and speed, a huge volume of traffic has a cost ramification for collection and processing. Resources of network devices are comparatively expensive and scarce. Such resources need to be utilised on their regular activities than utilising on monitoring activities. Therefore, to be a practical automated solution, the proposed system should be computationally inexpensive. We propose continuous monitoring via node profiling and analysis as described in our previous works [2–5]. It uses information fusion and evidence accumulation in computing node profiles.

3.1. *Computing node profiles*

Node profiling is the method of evidence fusion across space and time by updating node score dynamically based on changes in evidence. Profiling computes a suspicion score s_w for each node in the system during a smaller time window w. That score is updated as time progresses to compute a node score N_W such that,

$$N_W = \sum s_w \tag{1}$$

, for a larger observation window $W = \sum w$.

3.1.1. *Computing s_w*

Evidence to compute s_w can be collected from any relevant source of information[a] and convert them to descriptors (a.k.a features), say $D = \{d_1, d_2, d_3, ..., d_n\}$, subject to the condition that chosen descriptors have some sort of predictive power of the monitoring behaviour defined in the hypothesis H.

[a] e.g. packet/flow information from L3 switches or outputs of signature based IDSs, anomaly detection components, antivirals, file integrity checkers, SNMP-based network monitoring systems or any other source

H - the hypothesis that given node[b] is under attack, i.e. an attacker has initiated at least the first step of the attack lifecycle (see Section 2).

Then s_w is defined as:

$$s_w = \begin{cases} 1 & \text{if } \Lambda = ln\frac{L(H|D)}{L(\neg H|D)} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $L(.)$ denotes the likelihood. Λ expresses how many times more likely the observed evidence in D are under one model (say H) than the other ($\neg H$). Various machine learning algorithms, either supervised or unsupervised, can be employed to estimate Λ depending on the context and data availability.

3.1.2. *Computing Λ using Naive Bayes model*

A key application of predictive analytics is to classify entities and events based on a knowledge of their attributes. In this chapter, for the purpose demonstration, we employ a Naive Bayes classifier with communication flow attributes to compute Λ as follows.

Let $D = [d_1, \ldots, d_n]$ be a n-dimensional space, where n is the total number of attributes (descriptors) chosen to describe a communication flow in a computer network. Note that we assume remote attacks here and hence victims node receiving malicious communication flows during the attack lifecycle. Using the notation of conditional probability and well known Bayes theorem,

$$P(H/D) = \frac{\prod_{k=1}^{n} P(d_k/H) \cdot P(H)}{P(D)} \tag{3}$$

$$P(\neg H/D) = \frac{\prod_{k=1}^{n} P(d_k/\neg H) \cdot P(\neg H)}{P(D)} \tag{4}$$

Dividing equation (3) by (4) and taking logarithm,

$$ln\frac{P(H/D)}{P(\neg H/D)} = ln\frac{P(H)}{P(\neg H)} + \sum_{k=1}^{n} ln\frac{P(d_k/H)}{P(d_k/\neg H)} \tag{5}$$

[b] denoted by an IP address in this work

Equation 5 is the well known "Log likelihood ratio". $P(H), P(d_k/H)$ are prior and likelihoods terms while $P(H/D)$ is the posterior probability. Taking $ln\frac{L(H|D)}{L(\neg H|D)} \cong ln\frac{P(H/D)}{P(\neg H/D)}$,

$$\Lambda = ln\frac{P(H/D)}{P(\neg H/D)} \tag{6}$$

A key assumption in equations 3 and 4 is that the conditional probability of each feature given the class is independent of all other features. This is not the case in many practical problems, including ours. For example, the conditional probability of network services (e.g. HTTP, FTP, SMTP, SSH, DNS) and the port number in a communication flow are not independent of each other. But in practice, even when the independence assumption is violated and there are clear known relationships between attributes, it works anyway. A good example of this is spam filtering in which features are individual words in an email. In this case, certain word combinations tend to show up consistently in spam - for example, "online", "meds", "viagra" and "pharmacy". So, their occurrences are not independent of each other. But Naive Bayes based spam filters which assume mutual independence of features works very well in filtering spams. This can be due to two reasons. First, prediction in equation 5 depends only on the maximum, not the value of the maximum. Hence Naive Bayes classifier gets it right even if there are dependencies between features given that such dependencies do not change which class has the maximum probability value. Though there is no guarantee it can always happen, the second reason might be such dependencies often cancel out across a large set of features. However, the performance of Naive Bayes can degrade if the data contains highly correlated features as they over-inflating their importance (i.e. voting for twice in the model). Therefore it is better to evaluate the correlation of attributes using a correlation matrix and remove those features that are the most highly correlated, and test the performance before and after such a change and stick with better results.

3.1.3. *Pros and cons using Naive Bayes*

There are pros and cons of using Naive Bayes in this problem. As mentioned above, due to the nature of the problem, the proposed system should be computationally inexpensive and lightweight. Using Naive Bayes, calculating the probabilities for each attribute is very fast, hence the system can retrain quickly as the data changes because temporal drift is a major issue in Cybersecurity problems. Another advantage of Naive Bayes is its Naive (independent) assumption can be exploited to speed up the execution of the algorithm. Attribute probabilities can be calculated in parallel using different CPUs, machines or clusters in real-world applications. This is useful as global networks scale-up in traffic, volume and speed. Naive Bayes does not need a lot of training data to perform well. Because interactions between attributes are not considered in model training, hence less training data needed than some other popular algorithms (e.g. logistic regression). However, Naive Bayes will not be reliable if there are significant differences in the attribute distributions between training and test cases. Zero observations problem is a special case of this. After such cases have been identified the model should be updated.

3.1.4. *Compute s_w using unsupervised learning*

This section discusses how to employ unsupervised learning, in particular an autoencoder neural network model, in estimating s_w. An autoencoder neural network is an unsupervised learning algorithm. It applies back propagation, setting the output values to be equal to inputs (i.e. $\hat{D} \approx D$), by learning an approximation to the identity function in the model. Figure 2 depicts an autoencoder model.

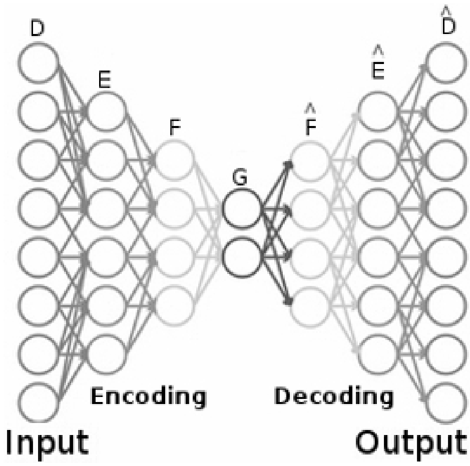

Fig. 2.: An autoencoder model.

To compute suspicion score, s_w is defined as:

$$s_w = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_i - \hat{d}_i)^2} \tag{7}$$

In other words, s_w is the reconstructed mean squared error (MSE) of the autoencoder model during a smaller time window w. As far as anyone knows, the total number of malicious data is far less than the total number of benign data in many real-world security problems. Hence, we can expect that the model learns patterns in benign data than malicious data that it can't easily see. As a result, high reconstructed MSE (s_w) values can be expected for malicious observations, which can be used to distinguish victim nodes from normal nodes.

Nevertheless, from the machine learning perspective, it is still important to evaluate different algorithms to see which algorithm performs best in terms of false alarms and computational cost in estimating Λ. However, it is out of the scope of this work.

3.2. *Analysis*

We compute a node score for each node in the system as described above. Aggregating suspicious scores over time helps to accumulate relatively weak evidence for long periods. These accumulated terms can be used as a measurement of the level of suspicion of a given node at any given time. If an attacker activity pattern is sufficiently reflected by profiles then detecting anomalous profiles would be sufficient to identify attackers. Hence, our task is detecting anomalous profiles in a given set of node profiles. We use a statistical method to detect anomalies subject to the assumption that normal node profiles in a given set follow an unknown Gaussian distribution. Testing of our hypothesis for any given time is a Bernoulli trial. Accumulated Bernoulli trials make a Binomial distribution which can be approximated by a Normal distribution. In practice, the setup where we have the distribution would be very well a mixture of Gaussian.

For each profile score N_W, its Z score is computed as:

$$z = \frac{N_W - \mu_W}{\sigma_W} \tag{8}$$

Where μ_W and σ_W are mean and standard deviation of the dataset at time window W. A test instance is declared to be anomalous if $z \geq T = k$. Note that the threshold T adjusts itself according to current state of a network as μ_W and σ_W change. k can be setup to 1,2 or 3 using the

68%-95%-99.7% rule of detecting outliers with Z-scores. For example, by setting k equals to 2, Z-scores in 2-3 range can be considered as borderline outliers. If the data (N_W values) is skewed, it is recommended to apply a transformation technique (e.g. power transformation or logarithms) first to move the dataset back to the normal bell shape and then apply the outlier detection technique.

Above simple outlier detection method subject to the masking and swamping effects. Multiple outliers may influence the value of the test statistic enough so that no points declared as outliers. On the other hand, swamping can occur when there is no outlier (or an outlier with very slight deviation). Therefore we complement the outcome of the above outlier detection method with graphical methods. Graphics can often help identify cases where masking or swamping may be an issue. Our analysis compares each node's activity changes to activity changes in its peer group. Looking at one's aberrant behaviour within similar peer groups (e.g. same user types, subnet, departments, job roles) would give better results in terms of false alarms than setting a universal baseline. Hence first classifying similar nodes into peer groups (e.g. web servers, file servers, clients), based on behaviour related attributes/features, and then applying the monitoring algorithm is recommended. Finding suitable classification algorithms for this task is left as future work.

4. Experimental setup

4.1. *Dataset description*

A third party dataset consists of malicious and normal traffic is used in this work. According to the authors [6], IXIA PerfectStormOne Tool[c] has been used to generate synthetic contemporary attacks (see section 4.1.2) within realistic modern normal activities. A 100GB of raw network traffic has captured.

4.1.1. *Initial features*

Forty nine features are extracted and categorised into five groups [6]:

- Flow related features: these features include the flow-related attributes between two nodes in a computer network such as source

[c] https://www.ixiacom.com/products/ixnetwork

and destination IP addresses, port numbers and the protocol type (e.g. TCP, UDP)

- Basic features: attributes that represent protocols connections (e.g. duration, source to destination bytes, time to live, packets loss, service, bits per second, packet count)
- Content related features: encapsulates the TCP/IP layer related attributes (e.g. TCP window advertisement value, TCP base sequence number, packet size)
- Time related features: contains the time related attributes (e.g. jitter, start time, inter packet arrival time, round-trip time, time between SYN and SYN_ACK, time between SYN_ACK and the ACK)
- Additional generated features divided into two groups: general purpose features (each feature has its own purpose) and connection features (built from the flow of 100 record connections based on the sequential order of the last time feature)

None of the above features is attack dependent and can be extracted from any given traffic flow, and hence can be considered as a general-purpose feature set for monitoring. Readers are invited to refer to [6] for more details about data generation, features and attack types.

4.1.2. *Attack types*

In addition to the day to day normal activities, following attack types have been produced in the dataset. Note that one or more activity types described below can be presented at different stages of the attack lifecycle. For example, Fuzzers and Exploit can be presented at either stage 1, 2, or 5 of the attack lifecycle. Ability to detect them will stop the attacker reaching the final stage of the attack where exfiltration, corruption and disruption happening, and hence can be considered as an early detection. However, we will mostly focus on reconnaissance activities in this work as obviously, it would be the first step of many network-based computer attacks.

- Fuzzers: an attack in which the attacker attempts to discover security loopholes in a program, operating system, or network by feeding it with the massive inputting of random data to make it crash.
- Analysis: a type of variety intrusions that penetrate the web applications via ports (e.g., port scans), emails (e.g., spam), and web

scripts (e.g., HTML files).

- Backdoor: a technique of bypassing a normal authentication, securing unauthorised remote access to a device, and locating the entrance to plain text as it is struggling to continue unobserved.
- DoS: an intrusion which disrupts the computer resources via memory, to be extremely busy to prevent the authorised requests from accessing a device.
- Exploit: a sequence of instructions that takes advantage of a glitch, bug, or vulnerability to be caused by an unintentional or unsuspected behaviour on a host or network.
- Generic: a technique that establishes against every block cypher using a hash function to collision without respect to the configuration of the block-cypher.
- Reconnaissance: can be defined as a probe; an attack that gathers information about a computer network to evade its security controls.
- Shellcode: attacker penetrates slight piece of code starting from a shell to control the compromised machine.
- Worm: an attack whereby the attacker replicates itself to spread on other computers. Often, it uses a computer network to spread itself, depending on the security failures on the target computer to access it.

4.1.3. *Exploring the dataset*

To investigate if all the descriptors in the initial feature set have some sort of predictive power of the monitoring behaviour, we plot boxplots for each descriptor separately in malicious class as well as in benign class. Due to the space constraints, Figure 3presents only twenty boxplots of them. As shown in Figure 3, there is a certain group of features which have different distributions in malicious class than benign class. These features have the discriminating power and hence inform more in classification/anomaly detection model. To reduce the computational cost features having the same distributions in both classes can be removed from unless they have any other form of information such as semantic information which is not encoded in the data. Therefore feature selection and reduction is necessary to remove unnecessary features (e.g. features not informed in the model and highly correlated features) from the feature set.

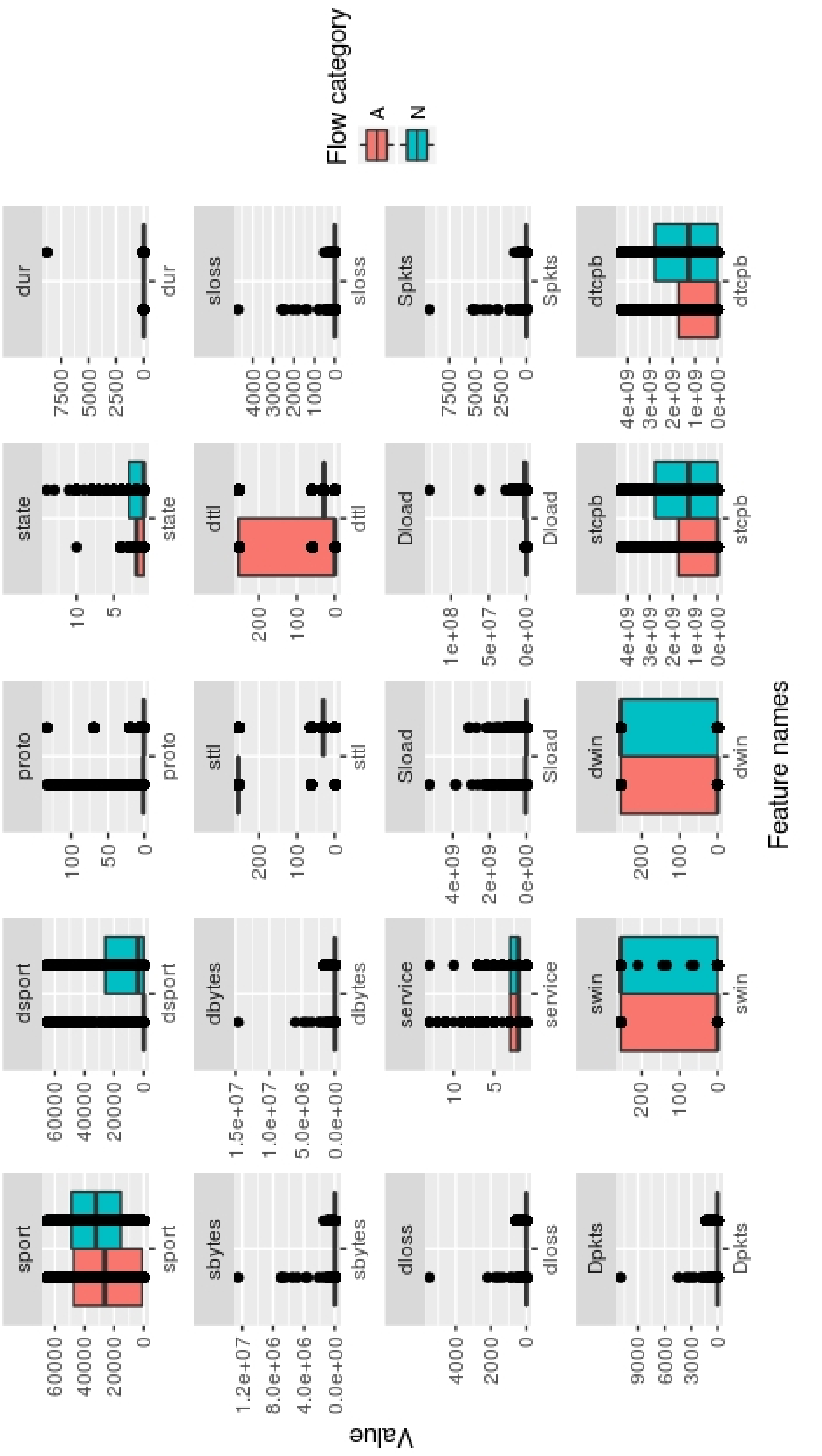

Fig. 3.: Distributions of each feature in two classes (A-attack, N-normal).

4.2. *Feature selection using empirical analysis*

As mentioned above, due to the scarcity of resources on monitoring devices, reduction to the computational cost is vital in our problem. Hence feature selection and reduction play an important role. Features described in section 4.1.1 were chosen as the initial feature set, and then a random forest model [7] was built and tuned using R's random forest package (version 4.6-12) [8]. Then, for the feature trade-off analysis in section 5.2, top important features are selected using the mean decrease Gini (see Figure 4).

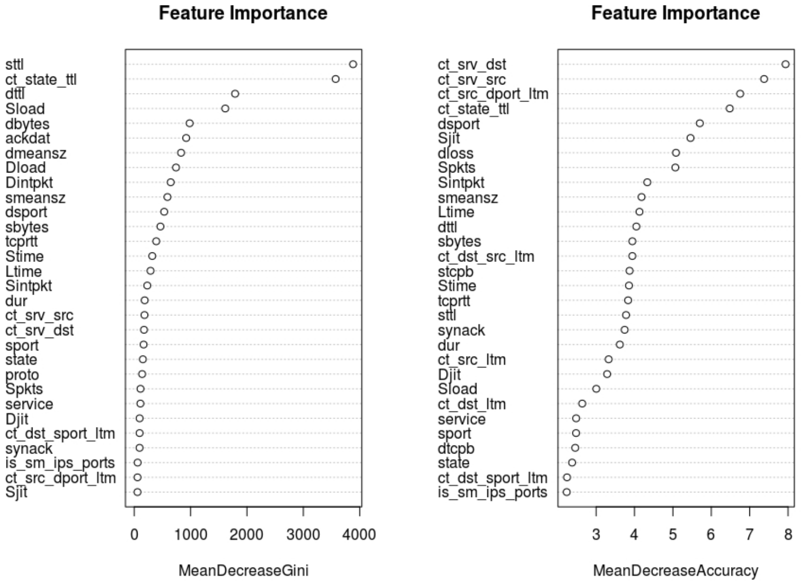

Fig. 4.: Feature selection: using empirical analysis.

5. Experimental results

In this section, experimental results are presented. We use graphical forms (e.g. Z-Score graphs) to present information. Visualisation helps to quickly recognise patterns in data as well as spotting masking and swamping effects mentioned above.

5.1. *Monitoring for reconnaissance*

Reconnaissance is a type of intrusion activities that can occur at the very early stages of the attack lifecycle. In an active reconnaissance, an intruder engages with the targeted system to gather information about vulnerabilities. So, our proposed technique was investigated against reconnaissance activities in the dataset. As shown in figure 5, proposed approach can detect all victims of reconnaissance activities in the dataset. It includes ten IP addresses, namely, 149.171.126.18, 149.171.126.17, 149.171.126.11, 149.171.126.12, 149.171.126.13, 149.171.126.15, 149.171.126.16, 149.171.126.19, 149.171.126.14 and 149.171.126.10 in the dataset. These IP addresses denoted by red dotted lines in figure 5. All other nodes (30 IP addresses) are normal and denoted by black lines. As expected, they keep suspicious score near zero as they are not targeted by any reconnaissance attempt during the monitoring period.

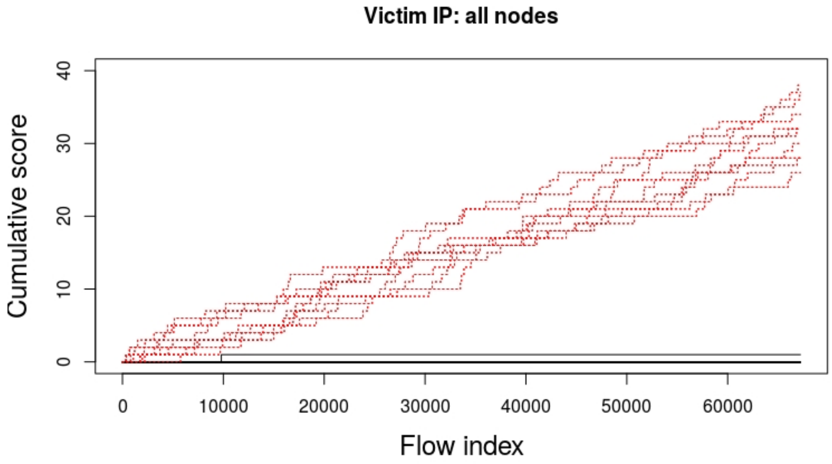

Fig. 5.: Monitoring for all nodes (forty nodes in the subnet). Victims of reconnaissance attempts are denoted by red dotted lines. All other nodes denoted by black lines in the above graph keep suspicious score near zero.

In order to closely investigate how our algorithm increment profile scores over the time, figure 6 visualises the profile scores of node 149.171.126.18 against time and event arrivals. In figure 6, data points with + sign denotes a reconnaissance attempt as it happens over the time from multiple source IPs while all other points denote a normal activity. As obvious, reconnaissance always increments the suspicious score of a node in the graph.

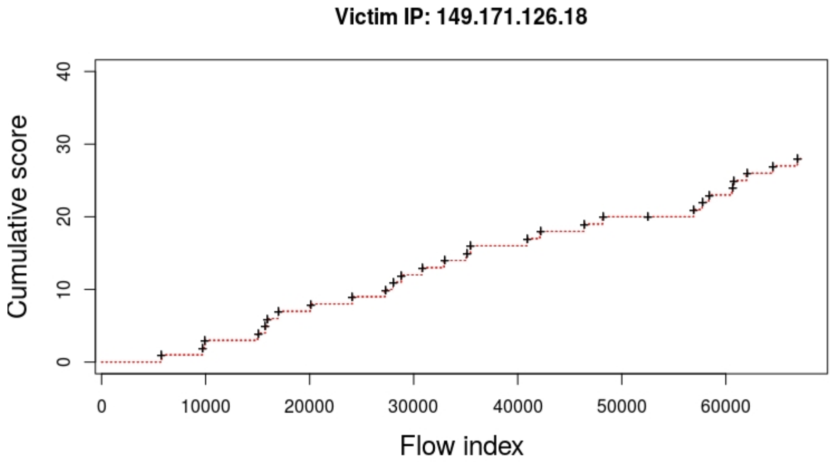

Fig. 6.: Increasing the suspicious score: + sign denotes a reconnaissance attempt as it happens over the time from multiple source IPs.

It should be noted that computing cumulative scores N_W itself and then presenting them as in figure 5 are not enough in detecting on going malicious activities. Because there is no sense of a threshold in that representation. Converting to Z-scores is required. In Z-score graphs in figure 7, nodes corresponding to red dotted lines denote victims. We set threshold as $T = 2$ in this work and then victims nodes are near the T, and importantly there is a clear visual separation between the set of normal nodes and anomalous nodes. Hence it is possible to recognise victims using the proposed method.

5.2. *Feature trade-off analysis*

As contemporary enterprise networks scale up in size and speed, huge volume of traffic has a cost ramification for security monitoring. Therefore, as mentioned in section 3, in order to be a practical automated CEWS solution, proposed system should be computationally inexpensive. Feature reduction play an important role in developing light weighted solutions, which could be motivated as long as it preserves the required level of preci-

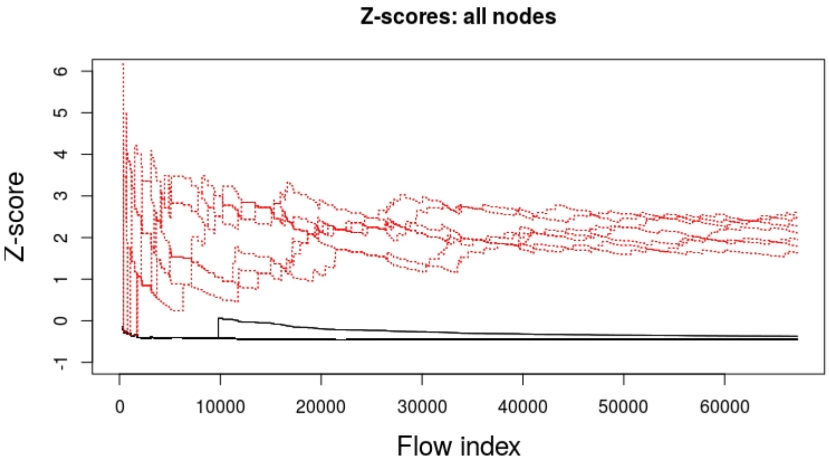

Fig. 7.: Z-scores: compassion within the peer group. Red dotted lines denote victim nodes.

sion. We will investigate how number of features affects on the performance as follows.

First, the most 5 importance features (top 5) were selected as input features (see section 4.2) to the proposed algorithm, and then training time, testing time (per test case) and false alarm rates were recorded. With regards to the false alarms, misclassification of flows were counted. The same experiment was repeated nine times by keeping all parameters unchanged, except number of features which were varied as top 5, top 10, top 15 and so on until top 45 as described above. Training and testing time were calculated on a laptop with an Intel(R) Core(TM) i7-6820HQ CPU @ 2.70GHz and 8GB of RAM. Ubuntu 16.04 operating system was running on the laptop.

Figure 8 presents the number of features vs training time. As shown in figure 8, number of features is proportional to the model training time. Lower the number of features selected the better for training time. Figure 9 presents the number of features vs testing time. Though there are slight drops at points 15 and 40, the graph has a increasing trend in general for test time. Figures 10 and 11 present false alarms rates against number of features used in the model. As obvious from figures, increasing number of features does not always reduce the false alarms. In fact, in this particular case, it will start to increase false negative rates if we increased the number of features beyond 20. From the security point of view false negatives are more critical than false positives though later would affect on user convenience. As per this analysis, using top 10 importance features would be the best combination in terms of computational cost and false alarms rates.

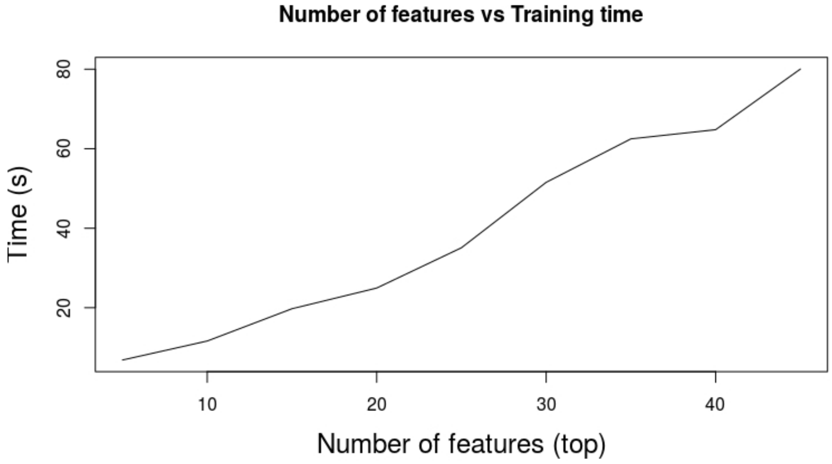

Fig. 8.: Number of features vs training time.

5.3. *Monitoring for other activities*

As explained in section 4.1.2, ability to detect any intruded activities simulated in the dataset will stop the attacker reaching final stage of the attack where exfiltration, corruption and disruption happening, and hence can be considered as an early detection. Therefore this section briefly investigates ability to employ proposed method to detect other intruded activities included in the dataset.

We use the entire dataset, without excluding any specific type of malicious activities, to produce the cumulative and Z-sore graphs as mentioned above. Figures 12 and 13 presents the cumulative and Z-score graphs respectively. Red dotted lines denote the victim nodes of different attack activities simulated in the dataset. Note that it includes the same victim IPs mentioned in section 5.1. Readers should notice the changes to the

September 5, 2018 17:54 ws-rv9x6 A pre-print of an article published in Data science for cyber security. Security science and technology, 3. London: World Scientific, chapter 5, pages 95-126. The final authenticated version is available online at: https://doi.org/10.1142/9781786345646$_0$05 output page 19

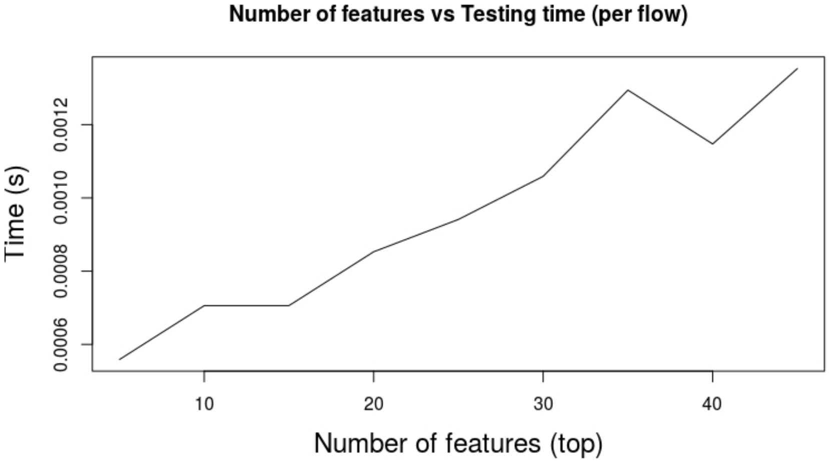

Fig. 9.: Number of features vs testing time (per test case).

limits of cumulative scores in figure 12. Its value increases from 40 (in figure 5) to 1500 for one node. This has happened due to the contribution of other activities to increment node score in addition to the reconnaissance. While this helps to spot that node quickly, in the same time it influences the values of test statistic of other nodes to suppress (see figure 13). However, in practice, this wont be a problem as soon as highly deviated victim is spotted and stop, rest of nodes start to stand out as μ_W, σ_W and T adjusts itself according to the current state of the network. Therefore the higher the number of suspicious activities is the better for early detection using proposed method.

6. Related literature

An extensive survey of collaborative intrusion detection proposals can be found in [9]. Collaborative intrusion detection works aim at sharing, correlating and cooperatively analysing sensor data collected from many organisations located in different geographical locations and hence producing early warnings on ongoing malicious activities. An infrastructure and organisational framework for a situation awareness and early warning system presented in [10]. eDare (Early Detection, Alert and Response system) [11] and the Agent-based CEWS [12] are similar efforts. However information

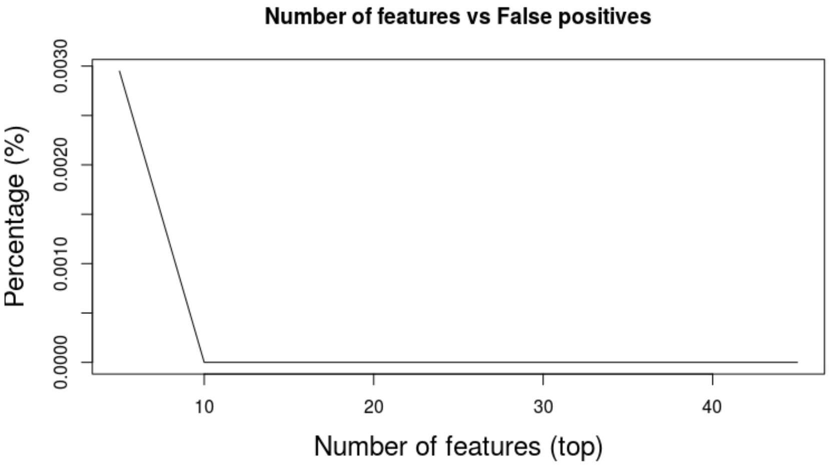

Fig. 10.: Number of features vs false positive rates.

exchange can be seen as a major barrier for CEWS' advances. The Internet motion sensor, a globally scoped Internet monitoring system, statistically analyses dark net traffic that needs to be interpreted by humans [13]. DShield internet storm centre collects firewall and IDS logs world wide and incorporates human interpretation and action in order to generate predictions and advice [9] while eCSIRT.net [14] comprises of a sensor network which collects and correlates alerts for human inspection. DeepSight intelligence collects, analyses and delivers cyber-threat information through a editable portal and datafeeds, enabling proactive defensive actions and improved incident response [15]. Human analysis and data mining is incorporated in order to provide statistics. In the context of security, data and information sharing is difficult between different organisations and nations due to various reasons [16, 17].

Situational awareness is an essential part of an CEWS which includes awareness of suspicious network related activities that can take place at all levels in the TCP/IP stack [18]. Such activity can range from low-level network sniffing to suspicious linguistic contents on social media. Various network measurements and techniques (e.g. packet inter arrival times [19], deep packet inspection [20], game theory [21]) have been employed. The idea for a common operational picture (big picture) is presented [22, 23]. A systematic review of cyber situational awareness can be found in [18].

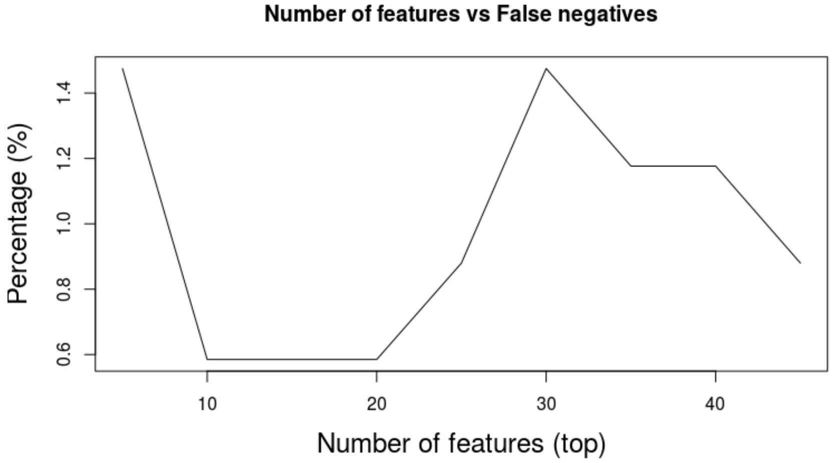

Fig. 11.: Number of features vs false negative rates.

However instead of addressing the full complexity, above solutions concentrated on a particular issue of the problem and some solutions (e.g. deep packet inspection) are neither feasible in practice nor suitable for real time analysis yet.

Sensing in-progress attacks requires strategically placed sensors throughout the cyberspace. Current sensor networks for CEWS have a simple monolithic structure [24], where data is acquired at the network edges and then transmitted over a dumb infrastructure to a central location for analysis. This can cause various issues to the analysis due to many reasons such as nonidentical measurements, nonidentical local detectors and noisy channels [25]. High computational cost is another significant issue. Hence computationally fast and accurate methodology to evaluate the error, detection, and false alarm probabilities in such networks is essential. Optimal sensor placement strategies for CEWS is discussed in [26]. Authors study correlation between attack patterns of different locations (national and international) and explore how sensors should be located accordingly. The design and analysis of sensor networks for detection applications has received considerable attention during past decades [27].

In order to early warn, fusion of different network measurements from different sources is essential. Fusion of cyber related information from a variety of resources including commercial news, blogs, wikis, and so-

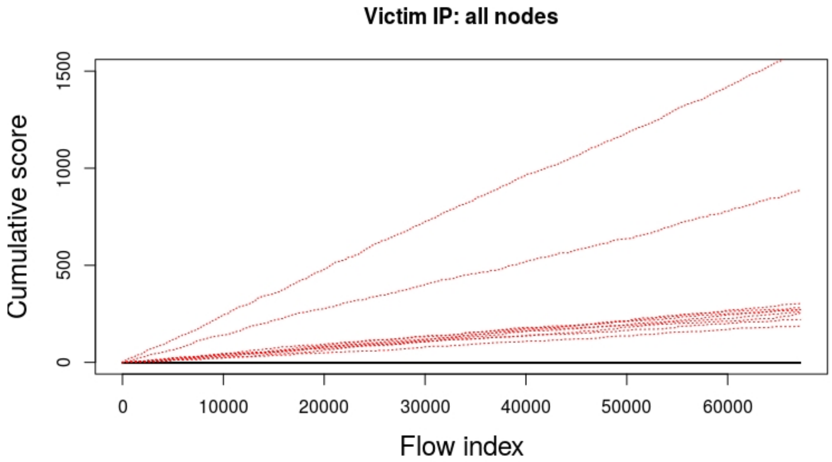

Fig. 12.: Profile Scores: cumulative scores of all nodes in the network while being a target of all possible malicious activities including reconnaissance. Victims of malicious activities are denoted by red dotted lines.

cial media sources is proposed in [28]. Bayesian fusion for slow activity monitoring [3, 29], high speed information fusion for real time situational awareness [30], JDL data fusion model to computer networks [31], detecting network data patterns [32], combining data from sensors using ontology methods [33] and fuse security audit data with data from a psychological model [34] are few of them to mention. Using web-based text as a source for identifying emerging and ongoing attacks can be found in [35].

An open, adaptable, and extensible visual analytic framework is provided in [36]. All data is treated as streaming and visualises them using machine learning techniques [37], live network situational awareness system that relies upon streaming algorithms included [38], fast calculations of important statistical properties of high speed and high volume data [38], sophisticated visualization of attack paths and automatic recommendations for mitigation [39] are some interesting works.

Threat scenario provides an important aspect to the early warning discussion. For example, early warning on malware propagation can be easier than warning on DOS attack. Focus to early warning on particular threat type is common (e.g. [40–44]). A malware warning centre is proposed in [40] while [42] aims for distributed and large-scale malware on the Internet. A

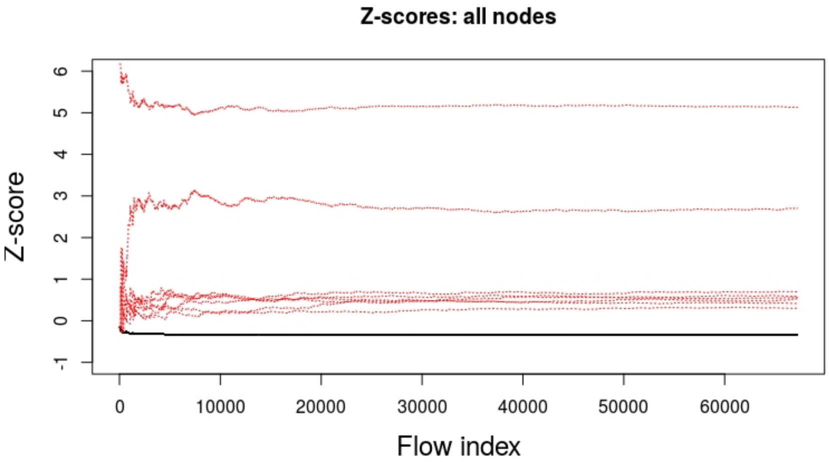

Fig. 13.: Profile Scores: Z Scores of all nodes in the network while being a target of all possible malicious activities including reconnaissance. Victims of malicious activities are denoted by red dotted lines.

worm propagation stochastic model is built [43]. Authors propose a logical framework for a distributed early warning system against unknown and fast-spreading worms. An open-source early warning system to estimate the threat level and the malicious activities across the Internet is provided [44]. Limiting to a certain threat type is a major drawback of these proposals. They cannot simply extend for newly emerging threats.

7. Research challenges

Traditional defences are simply not matching for today's adversaries as less than 1% of successful advanced threat attacks are spotted by SIEM-systems [45]. Once pass through the perimeter defences, the attacker can persist for long periods by moving laterally across the network and compromising as many systems as possible. Using seemingly legitimate actions, the attacker can then exfiltrate sensitive data or intellectual property. Hence continues monitoring of the behaviour of systems/users is required. Unlike most traditional solutions which focus on one or two steps in the attack lifecycle, as shown in this work, our proposed method can counter adversary's activities at early stages of the attack lifecycle. As a result, by analysing the data and following the digital footprints of the attacker, a security professional can focus on disrupting the adversary's attack before she can achieve her goal.

Ability to early-warn depends on three factors: attack progression rate (e.g. a malware propagation vs denial of service (DOS) attack), amount of evidence produced at each stage, and the ability to acquire such evidence by sensors. Rest of the section highlights a few challenges associated with these factors.

7.1. *Generic set of indicators*

In other domains, such as natural disasters (e.g. tsunami), early warnings are well established, and arguably simple when compared to the early warnings on the cyberspace. For example, in kinetic warfare, intelligence officers study different sources of intelligence (e.g. listen to communications, satellite imagery) to looking for known preliminary indicators of military mobilisation. In medical diagnosis, preliminary indicators such as feeling thirsty, tired, losing weight and blurred vision can early warn an individual about diabetes. But on the cyberspace, it is not clear what these indicators are or how they can be observed [46]. This presents a huge problem when trying to develop CEWS. As some scholars argue [46, 47], CEWS cannot be developed from a purely technical perspective. They must consider more than just technical indicators and require significant input from other disciplines such as international relations and sociology since the focus of CEWS should be to warn of an impending attack rather than detecting when it in progress. However the biggest challenge, a generic set of indicators (signs) of preparation for an attack on the cyberspace is not well established (understood) yet [48].

7.2. *Gathering evidence*

The cyberspace has a diversity. For example, it consists of different topological structures (e.g. PAN, LAN, MAN, WAN), different kind of networks (e.g. open Internet, darknet, honeynet, demilitarized zone) and different types of users (e.g. universities, health care system, the traffic system, power supply, trade, military networks). These entities produce events in different types and rates and have different analysis objectives and privacy requirements. To provide a representative image of the cyberspace at any given time, CEWS have to collect and process data from a range of these different entities. Employing a large monolithic sensor network for intelligence gathering on the cyberspace would not be possible due to these variations.

7.3. *Uncertainty reasoning*

The cyberspace is an uncertain place. Hence cyber defenders have to deal with a great deal of uncertainty [3, 49] which is compounded by the nature of computing. Any future CEWS that seeks to model and reasoning on the cyberspace has to accept this ground truth and must deal with incompleteness (compensate for lack of knowledge), inconsistencies (resolve ambiguities and contradictions) and change (update the knowledge base over time). For example, entering misspelled password can be a simple mistake by an innocent user or a password guessing attempt by an attacker. Cyber defenders do not know who the attackers nor their location. Some suspicious events, e.g. a major router failure could generate many ICMP unreachable messages while some computer worms (e.g. CodeRed and Nimda) generate the same in active probing process, can appear as part of an attack as well as can originate from normal network activities. Other contextual information should be utilised to narrow down the meaning of such data [3].

7.4. *Scalability*

In principle, it is possible to log every activity on every device on the cyberspace, but in practice, security analysts cannot process these logs due to their vagueness as attack indicators as well as the sheer volume of data. The biggest challenge is how to start from imprecise and limited knowledge about attack possibilities, and quickly sift through a huge volume of data to spot a small set of data that altogether makes the picture of attacks clear. As volume and rate of traffic are rising, an inspection of every individual event is not feasible. A data reduction is needed [3].

7.5. *Information fusion*

As mentioned earlier, CEWS cannot be developed from a purely technical perspective. Given the huge number of possible data sources and an overwhelming amount of data they generate, a data reduction method is essential to enable continuous security monitoring [50]. Future CEWS require fusing as many data sources as possible. Though it is not an exhaustive list, potential data sources for this task would be: network data traffic, log files, social media, mobile location traces, mobile call traffic, web browsing traces, content popularity, user preferences, spatial/geographic distribution of network elements, network topology (router and AS level), network paths, protocol traces, social network structure and other security intelligence either system or social level.

7.6. *Evaluation*

Getting validity for a novel method is only possible through a proper evaluation. But in this research area, evaluation of novel algorithms against real-time network data is a challenge. Real network traffic datasets with ground truth data on attack activity are difficult to obtain. Any such effort faces the uncertainty of success in investigating relevant patterns of activities. One solution to this problem would be to develop monitoring algorithms based on unary classification as it is relatively easier to find clean datasets than malicious ones, or providing mathematical proof for novel methods.

Machine learning algorithms need to be verified to find out their precise performance in real data. Specially in network computer security it is really important to have good datasets, because the data in the networks is infinite, changing, varied and with a high concept drift. These issues force us to obtain good datasets to train, verify and test the algorithms.

7.7. *Incident data integrity and retention*

No matter how persuasive evidence may be, it can be thrown out of court if you somehow alter it during the evidence collection process. Make sure you can prove that you maintained the integrity of all evidence. You may not detect all incidents as they are happening. Sometimes an investigation reveals that there were previous incidents that went undetected. It is discouraging to follow a trail of evidence and find that a key log file that could point back to an attacker has been purged. Carefully consider the fate of log files or other possible evidence locations. A simple archiving policy can help ensure that key evidence is available upon demand no matter how long ago the incident occurred.

References

[1] J. Biskup, B. Hämmerli, M. Meier, S. Schmerl, J. Tölle, and M. Vogel, 2. 08102 working group-early warning systems, *InProceedings biskup_et_al:*

DSP. p. 1493 (2008).

[2] S. A. Shaikh and H. K. Kalutarage, Effective network security monitoring: from attribution to target-centric monitoring, *Telecommunication Systems*. **62**(1), 167–178 (May, 2016). ISSN 1572-9451. doi: 10.1007/s11235-015-0071-0. URL `https://doi.org/10.1007/s11235-015-0071-0`.

[3] H. K. Kalutarage, S. A. Shaikh, I. P. Wickramasinghe, Q. Zhou, and A. E. James, Detecting stealthy attacks: Efficient monitoring of suspicious activities on computer networks, *Computers & Electrical Engineering*. **47**, 327–344 (2015). ISSN 0045-7906. doi: http://dx.doi.org/10.1016/j.compeleceng.2015.07.007. URL `http://www.sciencedirect.com/science/article/pii/S0045790615002384`.

[4] H. K. Kalutarage, S. A. Shaikh, Q. Zhou, and A. E. James. Monitoring for slow suspicious activities using a target centric approach. In *Information Systems Security*, pp. 163–168. Springer (2013).

[5] H. K. Kalutarage, S. A. Shaikh, Q. Zhou, and A. E. James. Sensing for suspicion at scale: A bayesian approach for cyber conflict attribution and reasoning. In *Cyber Conflict (CYCON), 2012 4th International Conference on*, pp. 1–19 (2012).

[6] N. Moustafa and J. Slay, The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set, *Information Security Journal: A Global Perspective*. **25** (1-3), 18–31 (2016).

[7] L. Breiman, Random forests, *Machine Learning*. **45**(1), 5–32 (Oct, 2001). ISSN 1573-0565. doi: 10.1023/A:1010933404324. URL `https://doi.org/10.1023/A:1010933404324`.

[8] A. Liaw and M. Wiener, Classification and regression by randomforest, *R News*. **2**(3), 18–22 (2002). URL `http://CRAN.R-project.org/doc/Rnews/`.

[9] C. V. Zhou, C. Leckie, and S. Karunasekera, A survey of coordinated attacks and collaborative intrusion detection, *Computers & Security*. **29**(1), 124–140 (2010).

[10] B. Grobauer, J. I. Mehlau, and J. Sander. Carmentis: A co-operative approach towards situation awareness and early warning for the internet. In *IMF*, pp. 55–66 (2006).

[11] Y. Elovici, A. Shabtai, R. Moskovitch, G. Tahan, and C. Glezer. Applying machine learning techniques for detection of malicious code in network traffic. In *KI 2007: Advances in Artificial Intelligence*, pp. 44–50. Springer (2007).

[12] K. Bsufka, O. Kroll-Peters, and S. Albayrak. Intelligent network-based early warning systems. In *Critical Information Infrastructures Security*, pp. 103–111. Springer (2006).

[13] M. Bailey, E. Cooke, F. Jahanian, J. Nazario, D. Watson, et al. The internet motion sensor-a distributed blackhole monitoring system. In *NDSS* (2005).

[14] CSIRT_Network. The european computer security incident response team network. `http://www.ecsirt.net/` (June, 2015).

[15] Symantec. Cyber security: Deepsight intelligence. `http://www.symantec.com/deepsight-products/` (June, 2015).

[16] M. Brunner, H. Hofinger, C. Roblee, P. Schoo, and S. Todt. Anonymity and privacy in distributed early warning systems. In *Critical Information Infrastructures Security*, pp. 81–92. Springer (2011).

[17] R. Koch, M. Golling, and G. D. Rodosek. Evaluation of state of the art ids message exchange protocols. In *International Conference on Communication and Network Security (ICCNS)* (2013).

[18] U. Franke and J. Brynielsson, Cyber situational awareness–a systematic review of the literature, *Computers & Security*. **46**, 18–31 (2014).

[19] P. Harmer, R. Thomas, B. Christel, R. Martin, and C. Watson. Wireless security situation awareness with attack identification decision support. In *Computational Intelligence in Cyber Security (CICS), 2011 IEEE Symposium on*, pp. 144–151 (April, 2011). doi: 10.1109/CICYBS.2011.5949399.

[20] D. King, G. Orlando, and J. Kohler. A case for trusted sensors: encryptors with deep packet inspection capabilities. In *Military Communication Conference, MILCOM 2012*, pp. 1–6 (2012).

[21] H. He, W. Xiaojing, and Y. Xin. A decision-support model for information systems based on situational awareness. In *Multimedia Information Networking and Security, 2009. MINES '09. International Conference on*, vol. 2, pp. 405–408 (Nov, 2009). doi: 10.1109/MINES.2009.130.

[22] Y. Cheng, Y. Sagduyu, J. Deng, J. Li, and P. Liu. Integrated situational awareness for cyber attack detection, analysis, and mitigation. In *SPIE Defense, Security, and Sensing*, pp. 83850N–83850N (2012).

[23] J. Preden, L. Motus, M. Meriste, and A. Riid. Situation awareness for networked systems. In *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2011 IEEE First International Multi-Disciplinary Conference on*, pp. 123–130 (Feb, 2011). doi: 10.1109/COGSIMA.2011.5753430.

[24] A. Theilmann. Beyond centralism: The herold approach to sensor networks and early warning systems. In *Proceedings of First European Workshop of Internet Early Warning and Network Intelligence (EWNI 2010)* (2010).

[25] S. Aldosari, J. M. Moura, et al., Detection in sensor networks: The saddlepoint approximation, *Signal Processing, IEEE Transactions on*. **55**(1), 327–340 (2007).

[26] J. Göbel and P. Trinius. Towards optimal sensor placement strategies for early warning systems. In *Sicherheit*, pp. 191–204 (2010).

[27] P. K. Varshney, *Distributed detection and data fusion*. Springer Science & Business Media (1997).

[28] T. Morris, L. Mayron, W. Smith, M. Knepper, R. Ita, and K. Fox. A perceptually-relevant model-based cyber threat prediction method for enterprise mission assurance. In *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2011 IEEE First International Multi-Disciplinary Conference on*, pp. 60–65 (Feb, 2011). doi: 10.1109/COGSIMA.2011.5753755.

[29] H. Chivers, J. A. Clark, P. Nobles, S. A. Shaikh, and H. Chen, Knowing who to watch: Identifying attackers whose actions are hidden within false alarms and background noise, *Information Systems Frontiers*. **15**(1), 17–34 (2013).

[30] M. Sudit, A. Stotz, and M. Holender. Situational awareness of a coordinated cyber attack. In *Defense and Security*, pp. 114–129 (2005).

[31] S. Schreiber-Ehle and W. Koch. The jdl model of data fusion applied x2014; a review paper. In *Sensor Data Fusion: Trends, Solutions, Applications (SDF), 2012 Workshop on*, pp. 116–119 (Sept, 2012). doi: 10.1109/SDF.2012.6327919.

[32] R. Paffenroth, P. Du Toit, R. Nong, L. Scharf, A. P. Jayasumana, and V. Bandara, Space-time signal processing for distributed pattern detection in sensor networks, *Selected Topics in Signal Processing, IEEE Journal of.* **7**(1), 38–49 (2013).

[33] M. L. Mathews, P. Halvorsen, A. Joshi, and T. Finin. A collaborative approach to situational awareness for cybersecurity. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2012 8th International Conference on*, pp. 216–222 (2012).

[34] F. L. Greitzer and D. A. Frincke. Combining traditional cyber security audit data with psychosocial data: towards predictive modeling for insider threat mitigation. In *Insider Threats in Cyber Security*, pp. 85–113. Springer (2010).

[35] K. Grothoff, M. Brunner, H. Hofinger, C. Roblee, and C. Eckert, " problems in web-based open source information processing for it early warning (2011).

[36] D. Jonker, S. Langevin, P. Schretlen, and C. Canfield. Agile visual analytics for banking cyber x201c;big data x201d;. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pp. 299–300 (Oct, 2012). doi: 10.1109/VAST.2012.6400507.

[37] L. Harrison, J. Laska, R. Spahn, M. Iannacone, E. Downing, E. M. Ferragut, and J. R. Goodall. situ: Situational understanding and discovery for cyber attacks. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pp. 307–308 (Oct, 2012). doi: 10.1109/VAST.2012.6400503.

[38] W. W. Streilein, J. Truelove, C. R. Meiners, and G. Eakman. Cyber situational awareness through operational streaming analysis. In *MILITARY COMMUNICATIONS CONFERENCE, 2011-MILCOM 2011*, pp. 1152–1157 (2011).

[39] S. Jajodia, S. Noel, P. Kalapa, M. Albanese, and J. Williams. Cauldron mission-centric cyber situational awareness with defense in depth. In *Military Communications Conference, 2011-MILCOM 2011*, pp. 1339–1344 (2011).

[40] C. C. Zou, L. Gao, W. Gong, and D. Towsley. Monitoring and early warning for internet worms. In *Proceedings of the 10th ACM conference on Computer and communications security*, pp. 190–199 (2003).

[41] M. Apel, J. Biskup, U. Flegel, and M. Meier. Towards early warning systems–challenges, technologies and architecture. In *Critical Information Infrastructures Security*, pp. 151–164. Springer (2010).

[42] M. Engelberth, F. C. Freiling, J. Göbel, C. Gorecki, T. Holz, R. Hund, P. Trinius, and C. Willems, The inmas approach (2010).

[43] E. Magkos, M. Avlonitis, P. Kotzanikolaou, and M. Stefanidakis, Toward early warning against internet worms based on critical-sized networks, *Security and Communication Networks.* **6**(1), 78–88 (2013).

[44] S. Kollias, V. Vlachos, A. Papanikolaou, P. Chatzimisios, C. Ilioudis, and

K. Metaxiotis. Measuring the internet's threat level: A global-local approach. In *Computers and Communication (ISCC), 2014 IEEE Symposium on*, pp. 1–6 (2014).

[45] V. R. Team, 2014 data breach investigations report (2014).

[46] M. Robinson, K. Jones, and H. Janicke, Cyber warfare: Issues and challenges, *Computers & Security*. **49**, 70–94 (2015).

[47] A. Sharma, R. Gandhi, W. Mahoney, W. Sousan, Q. Zhu, et al. Building a social dimensional threat model from current and historic events of cyber attacks. In *Social computing (SocialCom), 2010 IEEE second international conference on*, pp. 981–986 (2010).

[48] H. Kalutarage, S. Shaikh, B.-S. Lee, C. Lee, and Y. C. Kiat, *Early Warning Systems for Cyber Defence*, In eds. J. Camenisch and D. Kesdoğan, *Open Problems in Network Security: IFIP WG 11.4 International Workshop, iNetSec 2015, Zurich, Switzerland, October 29, 2015, Revised Selected Papers*, pp. 29–42. Springer International Publishing, Cham (2016). ISBN 978-3-319-39028-4. doi: 10.1007/978-3-319-39028-4_3. URL https://doi.org/10.1007/978-3-319-39028-4_3.

[49] H. Kalutarage. *Effective monitoring of slow suspicious activites on computer networks*. PhD thesis, Coventry University (2013).

[50] K. Dempsey, *Information security continuous monitoring (ISCM) for federal information systems and organizations*. US Department of Commerce, National Institute of Standards and Technology (2011).