2008 IFIP International Conference on Network and Parallel Computing

A Novel Sliding Window Based Change Detection Algorithm for Asymmetric Traffic

Ejaz Ahmed, Andrew Clark, George Mohay
Queensland University of Technology, Brisbane, Australia
{e.ahmed, a.clark, g.mohay}@qut.edu.au

Abstract

The effects of network attacks may result in abrupt changes in network traffic parameters. The speedy identification of these changes is critical for smooth network operation. This paper illustrates a sequential analysis technique for detecting these unknown abrupt changes in asymmetric network traffic. A novel sliding window based adaptive cumulative sum (CUSUM) algorithm is used to detect the cause of such variations in network traffic. The significance of the proposed algorithm is two-fold: (1) automatic adjustment of the change detection threshold while minimising the false alarm rate, and (2) timely detection of an end to the anomalous traffic. The validity of the proposed technique is investigated by experimentation on simulated data and on 18 months of real network traces collected from a class C darknet. Comparative analysis of the proposed technique with a traditional CUSUM method demonstrates its superior performance with high detection accuracy and low false alarm rate.

1. Introduction

In recent years different threat monitoring techniques have been developed to detect malicious activity including intrusion detection systems, honeypots and black hole monitoring. These techniques detect attacks by monitoring either used [17] or unused Internet address spaces [1, 2, 8–10, 12, 14, 15]. In contrast to used address spaces where there are live hosts connected to the Internet, an unused address is a routable Internet address which is not a live production host, as a result the traffic observed on such an IP address is by definition unsolicited and likely to be either opportunistic or malicious, including traffic from nodes infected by worms, traffic generated by random network probing tools or viruses, backscatter traffic from distributed denial of service attacks or unintentional traffic from misconfigured nodes. Blocks of such unused Internet addresses are commonly referred to as network telescopes, blackholes or darknets by the research community.

Statistical analysis of network traffic parameters has been successfully used in identifying new and ongoing attacks. It is observed that during normal operations the properties of parameters describing the network traffic either remain constant or vary slowly over time [3, 11]. On the other hand malicious activity usually transforms these parameters in such a way that their statistical properties no longer remain constant resulting in abrupt changes. The problem of identifying malicious activities can thus be formulated as a change point detection problem: to detect changes in the traffic's statistical properties as quickly as possible with a minimal false alarm rate [3].

In this paper we use sequential analysis techniques to identify changes in the behaviour of network traffic targetting a darknet to unveil both ongoing and new attack patterns. We propose an online sliding window sequential change-point detection algorithm based on the non-parametric cumulative sum (CUSUM) method. We provide a detailed analysis of the effectiveness of the algorithm using both synthetic data and 18 months of a real network traffic collected from a dedicated unused class C address block.

2. Related Work

Many researchers have used the observation that a malicious activity usually transforms the network parameters in such a way that their statistical properties no longer remain constant resulting in abrupt changes. In [5], Chan et al. use a non parametric CUSUM algorithm to identify specific worms which use a hit list of potential target IP addresses to propagate through the network. Each incoming source address is weighted based on heuristics and the total weight in a given time window is calculated. A non-parametric CUSUM is then applied on the total (weighted) source address count. If the current weighted value is greater than a predefined percentage of the mean then the calculated CUSUM score is subjected to a threshold test in order to

978-0-7695-3354-4/08 $25.00 © 2008 IEEE
DOI 10.1109/NPC.2008.81

identify a worm outbreak. The success of the proposed algorithm largely depends upon the threshold value which is pre-selected by the authors. Moreover the specific use of the proposed algorithm in detecting only hitlist worms makes it less favourable in the analysis of network traffic containing diverse anomalous activities.

In [4], Bo et al. also applied the non-parametric CUSUM algorithm to identify worm attacks. In this paper the number of unique destination hosts that a given source attempts to connect to in a given time interval is first calculated. The algorithm is then applied on the calculated sequence to estimate the CUSUM score. The calculated score is then subject to a threshold test to identify the worm activity. The authors have used a constant threshold value for detection of worm attacks within a lab environment.

In [6, 16], the authors have used the non-parametric CUSUM algorithm to detect denial of service attacks based on SYN flooding. Chen et al. [7] provided a framework to detect distributed denial of service (DDoS) attack using a distributed change detection algorithm based on the non-parametric CUSUM.

Siris et al. [13] used a parametric CUSUM change detection algorithm to detect TCP SYN flooding attack. Different design parameters have been evaluated using two data sets containing both high and low intensity attacks. The attacks were generated synthetically and the real network traffic is used as background noise. The authors have compared two change point detection techniques namely the adaptive threshold and parametric CUSUM algorithms. In both these techniques the results were subject to a pre-selected threshold test to identify the change points. Although both low and high intensity attacks were used to analyse the performance of the algorithm, the effect of varying attack characteristics such as duration and frequency were not considered.

While these methods can detect anomalies that cause abrupt changes in the network traffic parameters, identification of an end of the attack is not provided. While timely detection of an attack is critical, we argue that identification of an end to the anomalous activity is equally important to successfully detect subsequent attack patterns and to reduce the false alarms. Moreover the use of a static threshold, for detection of change points, subjects them to higher false alarms in the event of different and diverse attack patterns. We aim to overcome these problems using two different approaches and our proposed technique is distinguished from previous work in the following regards:

1) We propose a variable sliding window based non-parametric CUSUM algorithm to detect abrupt changes in data series. The use of a sliding window improves the efficiency of the algorithm by removing the effect of data points related to the last detected change point thus only keeping the normal data points within the sliding window at any given point in time. It also assists in identifying when the anomalous activity has ended, quickly removing its effect and reducing the false alarm rate.

2) Our technique uses a dynamic threshold in order to detect change points rather than pre-defined thresholds. Use of a dynamic threshold allows the algorithm to perform better during diverse anomalous activities.

3) The effectiveness of the proposed technique is proved using not only synthetic data sets but also using 18 months of real network traffic collected from a dedicated unused class C address block. Although the proposed algorithm can be used in detecting anomalous activities embedded in normal network traffic, our aim is to use the proposed algorithm to detect unusual behaviours in large collection of unsolicited traffic observed on a dedicated class C Darknet.

3. Detection Mechanism

During malicious activity, it is expected that the statistical properties of the traffic parameters no longer remain constant, resulting in the abrupt change. These change points can be detected using sequential analysis methods such as the cumulative sum (CUSUM) change point detection algorithm. CUSUM is a sequential analysis technique which assumes that the mean value of the parameter under observation will transform from negative to positive in the event of a change in its statistical properties. In this section a sliding window based CUSUM algorithm will be discussed for identification of abrupt changes in the traffic behaviour.

3.1. Sliding Window

The analysis of time-varying parameters is usually performed over parameter values covered by a window of infinite length, considering all the previous values, or by analysing values within a window of finite but fixed length. However the use of a fixed length window for change detection in time-varying parameters might lead to incorrect analysis either by delaying the detection of a change point or even not detecting a change point in the first place. We aim to tackle this problem by using a sliding window mechanism where its length is adjusted after a change in the parameter value is detected. Two different variable length sliding window techniques are proposed. These techniques differ in their post change window adjustment strategy.

The basic idea in the proposed variable length sliding window is to reduce window size to a minimum value whenever a change in system parameter is detected. The window will remain in the fixed length state until a new change point is detected or the current change point is terminated. After the end of change point is detected, the window will either progressively expand, discarding all the pre-change parameter values or it can progressively expand continuing from its pre-change size. In both cases parameter values responsible for the current change point will be discarded. We name these two techniques VALS-1 and VALS-2 respectively (where VALS stands for variable length sliding window).

In VALS-1 the sliding window will start with a minimum length of W_{min} and will progressively expand to maximum length W_{max} given that there is no change in parameter properties. After reaching W_{max} the window starts sliding. Let us assume that the parameter properties have changed at time t_s and the change continues until time t_e after which the parameter returns to a steady state. Upon detection of this change the sliding window will reduce to constant size W_{con} at time t_s such that $W_{con} > W_{min}$. At this stage the window is initialized with the parameter value responsible for the change. The window will remain in this state and continue to slide until a new change is detected or the current change expires. At time t_e, the window will start with W_{min}, initialize with post changed element and continue to expand until it reaches W_{max}. The window will continue to slide until a new change point is detected.

In VALS-2 the window behaviour is same as VALS-1 until the change has detected at time t_s. At time t_s, a successful detection of a change will reduce the window size to constant size W_{con}. A second window W' will hold all the parameter elements before change such that the length of $W' <= W_{max}$. The window will proceed as VALS-1 until the end of change is detected at t_e. At t_e, window W' will expand if Length$(W') < W_{max}$ or slide if Length $(W') = W_{max}$, without considering parameter elements related to the change point. The window will continue to slide until a new change point is detected

3.2. Change Point Detection Algorithm

In sequential analysis, change detection methods can be categorized as offline or online change detection algorithms. In an offline change detection algorithm the process of data acquisition is completed before applying the algorithm. Whereas the basic idea of an online change detection algorithm is to detect change as early as possible which is critical for network operations. Suppose that a random process X is sampled at a fixed time interval t resulting in a sequential observation X_t. After each sampling period a decision is computed to decide whether or not there is a transformation in process statistical properties resulting in a change point. The test for signalling a change at time t_0 from observations y_i and y_k, is based on log likelihood ratio, S_n , shown in Equation 1

$$S_n = \sum_{i=1}^{k} s_i \tag{1}$$

where

$$s_i = \ln \frac{P_{\theta_1}(y_i)}{P_{\theta_0}(y_i)} \tag{2}$$

where θ_0 and θ_1 specify two hypotheses with probabilities P_{θ_0} and P_{θ_1} respectively. The value $i = 1$ represents the first element within a sliding window and k represents the last element in the sliding window at time t_0. For the change detection it is assumed that the log likelihood ratio shows a negative drift before change and a positive drift after change. Therefore, the relative information for change detection lies in the difference between the log likelihood ratio and its current minimum value [3]. Thus the CUSUM score can be represented as:

$$g_k = S_k - m_k \tag{3}$$

where

$$m_k = \min_{1 \le j \le k} S_j \tag{4}$$

The CUSUM score given in Equation 3 is then compared with threshold value h to identify a change. Thus the alarm for the CUSUM algorithm is given by

$$t_a = min\{k : g_k \ge h\} \tag{5}$$

According to Basseville et al. [3], the decision rule given in Equation 3 can be rewritten in a recursive manner as

$$g_k = \begin{cases} g_{k-1} + \ln \frac{P_{\theta_1}(y_k)}{P_{\theta_0}(y_k)} & \text{if } g_{k-1} + \ln \frac{P_{\theta_1}(y_k)}{P_{\theta_0}(y_k)} > 0 \\ 0 & \text{if } g_{k-1} + \ln \frac{P_{\theta_1}(y_k)}{P_{\theta_0}(y_k)} \le 0 \end{cases} \tag{6}$$

where g_0=0. From Equation 2, the above equation can be compacted into:

$$g_k = (g_{k-1} + s_k)^+ \tag{7}$$

Using the above equations the change point can be computed. But due to the lack of a complete model of $\{y_k\}$, it is difficult to compute g_k as no prior information about the underlying process distribution is available. One way to solve this problem is to use a non-parametric approach which does not make any assumptions about the underlying process probability distribution. In the case of a non-parametric CUSUM algorithm Equation 7 can be rewritten as $g_k = (g_{k-1} + s_k)^+$, $g_0 = 0$ and the corresponding decision rule can be expressed as

$$d_k(g_k) = \begin{cases} 0 & \text{if } g_k \le h \\ 1 & \text{if } g_k > h \end{cases} \tag{8}$$

where d_k is a decision at time k considering threshold value h. If the CUSUM score, g_k, is less than or equal to the given threshold the decision will be zero indicating normal operation and if g_k is greater than the threshold, the decision will be one indicating a change in parameter properties. To achieve this, it is necessary to transform sequence $\{y_k\}$ to a new sequence $\{x_k\}$ so that it has a negative mean during normal operation and a positive mean during malicious activity. Thus

$$x_k = y_k - \alpha \tag{9}$$

where α is a constant and is considered to be a upper bound on the mean of $\{y_k\}$. Thus Equation 9 becomes

$$x_k \quad = \quad y_k - \alpha * \overline{y}_k \qquad (10)$$

Where $\overline{y}_k$ is the estimated mean at time k, which can be calculated by

$$\overline{y}_k = \frac{k-1}{k}\overline{y}_{k-1} + \frac{1}{k}y_k \qquad (11)$$

Thus with the help of Equations (8 -11) a change point can be identified using the non-parametric CUSUM method.

3.3. Design Parameters

In order to identify a change the calculated CUSUM score is compared with the threshold h, see Equation 8. One way to achieve this is to pre-select a fixed value of the threshold h. But due to the diversity in different network traffic parameters it is difficult to select a single threshold value suitable for all parameters. Also in the event of different and diverse attack patterns the fixed threshold value results in an increase in detection delay or no detection at all. The better alternative is to dynamically select the threshold value to identify the start of an anomalous event. The dynamic threshold can be calculated as

$$E_n^{start} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})}{N}} \qquad (12)$$

The above equation sets the threshold value dynamically to the standard deviation of the elements within a window of size N at any given time t. It is observed that this method gives optimal results once the sliding window has reached its maximum size.

While timely detection of attacks is critical, identification of an end to the anomalous activity is equally important to successfully detect subsequent attack patterns and to reduce the false alarms. We have achieved this by comparing the CUSUM score with another threshold value given by

$$E_n^{end} \quad = \quad 0.25 * E_n^{start} \qquad (13)$$

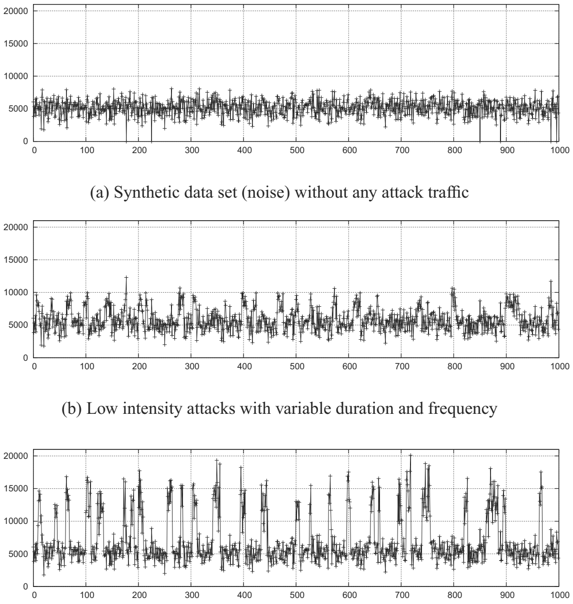
According to above equation an alarm cannot be cancelled until the CUSUM score is substantially reduced. In addition, to avoid missing alarms a separate counter τ is also used along with E_n^{end}. The alarm is not cancelled until timer τ reaches a specified value. Currently detection of the end is limited to a single change point, subsequent detection of the end of nested change points is a part of our future work. Other design parameters including sliding window size N and upper bound on mean α are selected based on experimental analysis with synthetic data and will be discussed in the following section

4. Performance Evaluation and Results

In this section we compare the performance of our two proposed sliding window based non-parametric CUSUM algorithms with the traditional non-parametric CUSUM algorithm without any window. For this we use both synthetic traffic and real traces from a class C darknet. The performance metrics considered include the detection probability (the percentage of attacks for which alarm was raised) and false alarm rate (the percentage of alarms not corresponding to an actual attack). In addition we seek to investigate the effect of window size N and upper bound on mean α on the performance of our proposed algorithm.

4.1. Synthetic Data

The synthetic data set consists of around one thousand data points distributed uniformly with mean 5200 and standard deviation 1200. This closely approximates the data collected on a class C Darknet over a period of 18 months and removes the effect of any outliers present in the data. This data set is then used as background noise and attacks were generated synthetically, which allowed us to control the characteristics of the attacks and investigate the performance of our proposed algorithms. Both high intensity attacks, whose mean amplitude is 250% higher than the mean traffic rate, and low intensity attacks, whose mean amplitude is 50% higher than the mean traffic rate were generated to test the effectiveness of the algorithm [13].



(a) Synthetic data set (noise) without any attack traffic

(b) Low intensity attacks with variable duration and frequency

(c) High intensity attacks with variable duration and frequency

Figure 1. Synthetic data set

The proposed algorithm is tested on five synthetic data sets, each containing different attack characteristics. Figure 1 shows some of the synthetic data sets used, the horizontal axis represents observation period in unit time while the vertical axis represents the number of packets. Table 1 summarises the characteristics of each data set. The attack duration and attack frequency is specified in terms of observation period. For example attack duration of 10 means that the attack will continue for 10 consecutive observation periods and attack frequency of 100 means that there will be attack after every 100 observation periods.

Table 1. Characteristics of different data sets

Data set	Attack Intensity	Attack Duration	Attack Frequency	Number of Attacks
1	high	10	100	9
2	high	variable	variable	27
3	low	10	100	9
4	low	variable	variable	24
5	variable	variable	variable	34

4.2. Experimentation

For the selection of design parameters both VALS-1 and VALS-2 along with the traditional CUSUM algorithm without any window is applied to all five synthetic data sets described above. Experiments were performed with different window sizes, N, ranging from 2 to 300, and upper bound on mean, α, with values from 0.5 to 3. The motivation behind this was to select the optimal vales for both N and α giving least false alarms with high detection accuracy.

Effect of Upper Bound on Mean: In order to analyse the effect of the upper bound on the mean, different attack scenarios including high, low and variable attack intensity were considered. In the case of high intensity attacks analysis with both fixed duration, fixed frequency attacks and variable duration, variable frequency attacks were performed. Due to space limitation, only attacks with variable duration and variable frequency will be discussed.

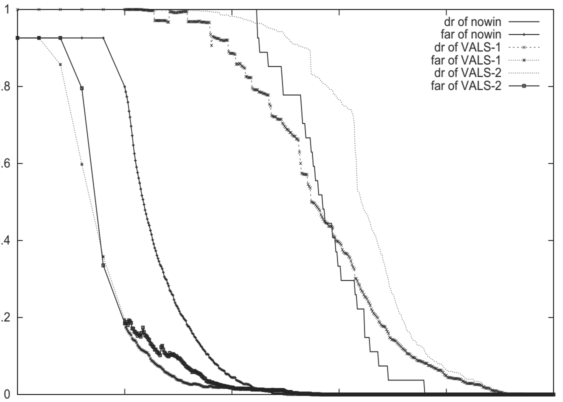


Figure 2. Effect of α: high intensity attacks

Figure 2 shows the results for data containing high intensity attacks having variable attack duration and variable attack frequency. Our intent was to investigate the effect of α on algorithm performance in terms of its detection accuracy and false alarm rate. The horizontal axis in these figures is the α value and the vertical axis is detection rate (dr) or false alarm rate (far). Each point in the graph corresponds to detection rate or false alarm rate averaged over entire range of window sizes for a specific α value.

Observe that VALS-2 gives better performance than both VALS-1 and the traditional CUSUM. This is due to the pre-change and post-change behaviour of the sliding window. The removal of data points related to the current anomalous activity and consideration of pre-anomalous data points is the major factor in its superior performance. This helps VALS-2 adapt to dynamic attack conditions, gradually learning the normal behaviour. Whereas the restart of the learning process in VALS-1 after the end of anomalous activity and the presence of data points related to past change points in the traditional CUSUM results in degradation of their performance under dynamic attack conditions.

In the case of high intensity attacks with fixed duration and intensity it is observed that the traditional CUSUM algorithm gives high detection and high false alarm rates as compared to the proposed algorithms: VALS-1 and VALS-2. On the other hand the performance of the proposed algorithms is almost identical. Both detection and false alarm rates tend to decline with the increase in α value. The high detection accuracy of the traditional CUSUM comes at the cost of a high false alarm rate. Comparatively the performance of the proposed algorithms was optimal for α between 1.35 and 1.5, giving more than 95% detection rate and less than 5% false alarm rate.

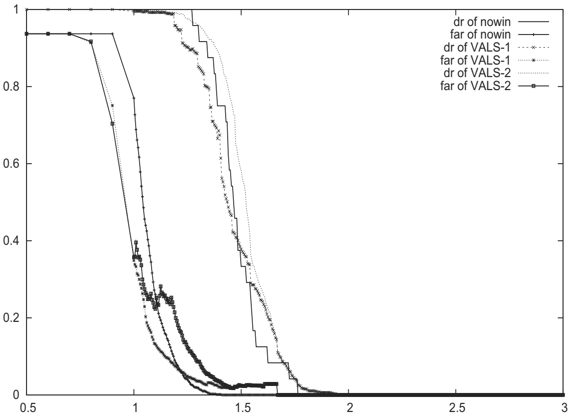


Figure 3. Effect of α: low intensity attacks

Let us now consider low intensity attacks. Both low intensity attacks with fixed duration, fixed attack frequency and variable duration, variable attack frequency were considered. Figure 3 shows the performance of different algorithms when low intensity attacks having variable duration and variable frequency were considered. The performance of all three algorithms was almost identical in this case. The optimal result in this case is for α value between 1.35 and 1.4.

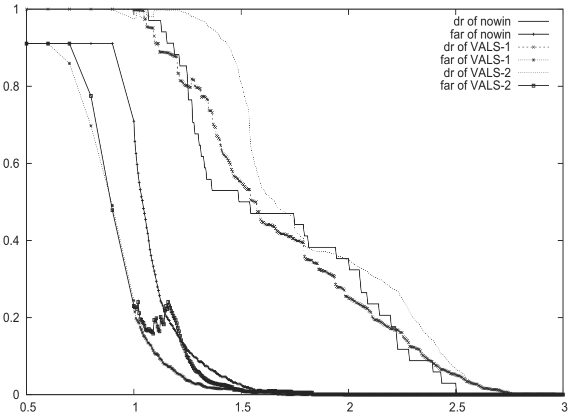


Figure 4. Effect of α: var. intensity attacks

For variable intensity attacks both high and low intensity attacks were considered with variable duration and variable frequency. The motivation was to analyse the performance of algorithms under dynamic attack conditions. Figure 4 shows the performance of algorithms under such conditions. Observe that VALS-2 gives better performance than VALS-1 and the traditional CUSUM due to the reasons described above. Moreover VALS-1 gives less false alarms than the traditional CUSUM with almost identical detection rate. VALS-2 gives optimal performance for α between 1.3 and 1.5.

Effect of Window Size: Let us now consider the effect of different sliding window sizes under different attack conditions. For this only VALS-1 and VALS-2 will be considered.

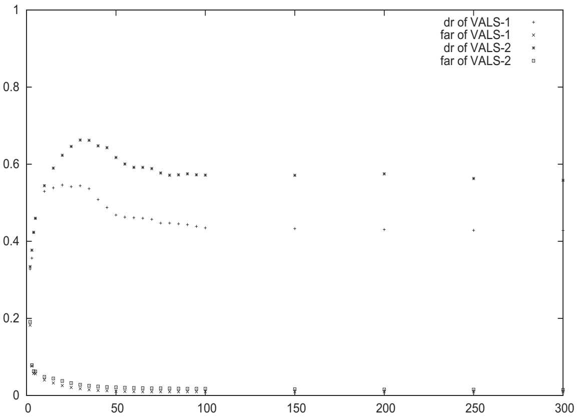


Figure 5. Effect of N: high intensity attacks

Figure 5 shows the effect of window size N on the detection rate and false alarm rate during high intensity attacks with variable duration and frequency. The horizontal axis in these figures is the N value and the vertical axis is the detection rate or false alarm rate. Each point in the graph corresponds to detection rate or false alarm rate averaged over entire range of values for a specific window size. Observe that the performance of VALS-2 is better than VALS-1. This is due to the difference in pre-change and post-change behaviour of the sliding window as discussed previously.

In the case of low intensity attacks it is observed that both the algorithms performed equally in both fixed and variable cases with $N \geq 60$ having the minimum effect on the algorithms.

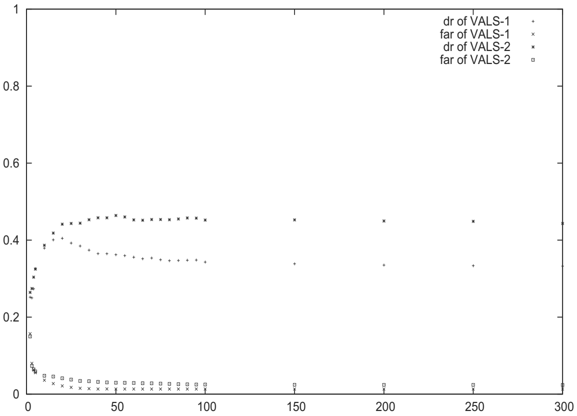


Figure 6. Effect of N: var. intensity attacks

Figure 6 shows the performance of VALS-1 and VALS-2 during variable attacks.Observe that VALS-2 gives better performance than VALS-1. It is observed that selecting $N \geq 60$ does not have any effect on the performance of these algorithms. Based on the above discussion, the design parameters used in the rest of the paper are $N = 100$ and $\alpha = 1.375$ unless otherwise noted.

4.3. Real Darknet Traffic

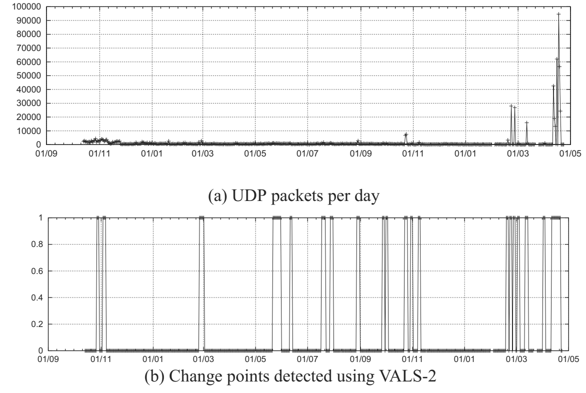
In this section, we will discuss the results of applying the VALS-2 algorithm to 18 months of real network traces collected from a class C darknet. Our intent is to use the proposed algorithm to detect unusual behaviours in a large collection of unsolicited traffic.

We will focus our discussion on the analysis of UDP traffic observed on the Darknet during 15th October, 2006 and 20th April 2008. Figure 7 shows the UDP traffic observed on the Darknet and change points detected using the proposed VALS-2 algorithm. In Figure 7(a), the horizontal axis represents observation period in days while the vertical axis represents the number of UDP packets observed on the Darknet. Figure 7(b) shows the result of applying the VALS-2 algorithm to the UDP traffic. The vertical axis in this case is the decision function, 1 indicates a change and 0 indicates no change or uniform behaviour. A total of 24 change points were detected. Due to space limitations we will limit our discussion to the 6 most significant change points.

Table 2. Summary of UDP traffic observed on the Darknet

CP	Date	Number of Packets	Sources	Destination IP	Destination Port
1^{st}	23/02/08	27947 (80.0)	866 (80.0)	x.x.x.60 (96.7)	13091 (94.6)
2^{nd}	27/02/08	26757 (56.9)	587 (70.7)	x.x.x.60 (62.5) x.x.x.221 (33.2)	13619 (60.5) 136971 (27.6)
3^{rd}	12/03/08	15827 (46.2)	509 (65.9)	x.x.x.60 (96.4)	13276 (93.4)
4^{th}	12/04/08	42544 (71.9)	537 (70.2)	x.x.x.221 (97.8)	13398 (90)
	13/04/08	18943 (52.9)	380 (61.5)	x.x.x.221 (97.3)	13048 (70.3) 13403 (24.4)
	14/04/08	13227 (43.3)	160 (31.7)	x.x.x.221 (92.5)	13408 (83.8)
	16/04/08	62025 (79.8)	991 (74.6)	x.x.x.221 (96.7)	13763 (51.3) 13929 (37.5)
	18/04/08	94607 (83.6)	1491 (80.6)	x.x.x.221 (98.5)	13493 (77.3) 13888 (7.8)
	19/04/08	56536 (74.1)	822 (68.1)	x.x.x.221 (97.5)	13240 (91)
	20/04/08	24330 (56.3)	949 (73.9)	x.x.x.221 (98.1)	13954 (43.1) 13738 (33) 13145 (14.3)
5^{th}	23/10/07	6645 (22.5)	78 (20.1)	x.x.x.91 (15.2) x.x.x.206 (15.2) x.x.x.143 (10.9)	11434 (40)
	24/10/07	7520 (23.6)	55 (16.1)	x.x.x.28 (17.1) x.x.x.221 (14.6) x.x.x.160 (7.3)	1434 (51.9)
6^{th}	28/08/07	1332 (4.4)	317 (43.48)	Various	1026:1027:1028 (80.1)

UDP traffic on port 13xxx: The most significant UDP activity was observed on destination ports 13xxx. It was observed that all UDP packets for destination port 13xxx were targeted on two destinations x.x.x.60 and x.x.x.221, with latter getting more than 93% of the total traffic. A total of 108596 unique payloads were recorded during this activity with 99.3% having a length of 27 bytes. Each source sends multiple packets using the same source port. 92% of the sources send the payload three times before modifying the payload. The first change point related to this activity was observed on 23rd February, 2008. More than 96% of the UDP traffic was destined for address x.x.x.60, with more than 94% targeted on port 13091.



(a) UDP packets per day

(b) Change points detected using VALS-2

Figure 7. Darknet traffic trace

The second change point related to the same activity was observed on 27th February, 2008. More than 95% of the traffic targeted destinations x.x.x.60 and x.x.x.221 on ports 13619 (60.5%) and 136971 (27.6%). The third change point was observed on 12th March, 2008 with 96.4% traffic destined for x.x.x.60, destination port 13276 getting 93.4% of the traffic. The fourth change point related to similar activity was observed on 12-20 April, 2008 with no activity on 15 and 17th April 2008. More than 98% of the UDP traffic was destined for x.x.x.221. It is important to note that 98% of source addresses related to these four change points were observed on the darknet only on the day of the activity.

Table 2 summarises the above mentioned activity. The value in bracket corresponds to the percentage of the total traffic observed on a particular day. Even though we do not know at this stage what really caused this activity, it is the main cause of a huge spike in the number of UDP packets collected by the darknet in the respective days. The close proximity of these change points and their distinct behaviour is indeed due to some unknown but interesting phenomenon which we aim to analyse in detail and is part of our future work.

MS-SQL Slammer: In MS-SQL slammer worm attack the attacker tries to exploit the buffer overflow vulnerability without being authenticated by the server. The increased MS-SQL slammer activity on Darknet was observed on 23rd and 24th October, 2007 (corresponding to 5^{th} change point in Table 2). During these days, port 1434 observed 40% and 51.9% of the total UDP traffic respectively.

MS Messenger NetSend spam: The MS Messenger NetSend Spam is generally related to the pop up messages on MS machines warning about registry corruption and urging the user to follow a link to “fix” the problem. By default, that “messaging” service runs on UDP/1026 for Windows 2000 and Windows XP, but it can be set to different ports. The increase in spamming activity was observed on 28th August 2007 (6^{th} change point). Observe that even the UDP traffic on that day is only 4.4% of the total traffic the proposed VALS-2 algorithm has successfully detected the change in UDP traffic behaviour.

5. Conclusions and Future Directions

We proposed and investigated two novel sliding window based change detection algorithm for asymmetric traffic, namely VALS-1 and VALS-2. The proposed algorithms use a dynamic threshold in order to detect change points and also identify the end of anomalous activity. Moreover the effectiveness of the proposed technique is investigated using both synthetic data and real network traffic collected from a dedicated unused class C address block. Comparative analysis of VALS-1, VALS-2 and the traditional CUSUM algorithm is performed under different attack conditions. It is observed that while a simple method such as the traditional CUSUM can effectively identify change in the traffic behaviour; it comes at the cost of a high false alarm rate. A sliding widow based algorithm such as VALS-2 exhibits better performance under dynamic attack conditions giving high detection and low false alarm rates. Analysis of real network traffic reveals the robustness of the proposed VALS-2 algorithm and its ability in identifying changes in traffic behaviour due to different anomalous activities. In contrast to previous work in the related area, our technique neither requires sets of attacks to be detected nor detection threshold be supplied by the user. This helps in successful detection of change points related to different malicious behaviours.

Our ongoing work focuses on extending the proposed approach to multiple traffic parameters and correlation of detected change points. Analysing the variations of change points across different parameters may help in identifying the degree of involvement of these parameters in the detected change. This will not only help in automatically identifying the primary cause of the change but also help in automatically categorizing different attacks. Currently the parameters used in the proposed algorithm are set manually based on the analysis of labelled synthetic data. A further extension is to devise a methodology for automatically adapting these parameters.

References

[1] M. Bailey, E. Cooke, F. Jahanian, A. Myrick, and S. Sinha. Practical darknet measurement. *40th Annual Conference on-Information Sciences and Systems*, pages 1496–1501, March 2006.

[2] M. Bailey, E. Cooke, F. Jahanian, J. Nazario, and D. Watson. The Internet Motion Sensor: A distributed blackhole monitoring system. In *Proceedings of Network and Distributed System Security Symposium NDSS*, San Diego, CA, February 2005.

[3] M. Basseville and I. Nikiforov. *Detection of abrupt changes: theory and application*. Englewood Cliffs, NJ: Prentice Hall, 1993.

[4] C. Bo, B.-X. Fang, and X.-C. Yun. A new approach for early detection of internet worms based on connection degree. *Proceedings of International Conference on Machine Learning and Cybernetics*, 4:2424–2430 Vol. 4, Aug. 2005.

[5] J. Chan, C. Leckie, and T. Peng. Hitlist Worm Detection using Source IP Address History. *Proceedings of Australian Telecommunication Networks and Applications Conference*, 2006.

[6] W. Chen and D. Yeung. Defending Against TCP SYN Flooding Attacks Under Different Types of IP Spoofing. *Proceedings of the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL'06)-Volume 00*, 2006.

[7] Y. Chen, K. Hwang, and W.-S. Ku. Collaborative detection of ddos attacks over multiple network domains. *IEEE Transactions on Parallel and Distributed Systems*, 18(12):1649–1662, Dec. 2007.

[8] E. Cooke, M. Bailey, Z. M. Mao, D. Watson, F. Jahanian, and D. McPherson. Toward understanding distributed blackhole placement. In *Proceedings of the ACM workshop on Rapid malcode, WORM*, pages 54–64, New York, NY, USA, 2004. ACM.

[9] E. Cooke, Z. Mao, and F. Jahanian. Hotspots: The root causes of non-uniformity in self-propagating malware. *International Conference on Dependable Systems and Networks, DSN*, pages 179–188, 2006.

[10] X. Jiang and D. Xu. Collapsar: a vm-based architecture for network attack detention center. In *Proceedings of the 13th conference on USENIX Security Symposium, SSYM*, pages 2–2, Berkeley, CA, USA, 2004. USENIX Association.

[11] J. Jung, V. Paxson, A. Berger, and H. Balakrishnan. Fast portscan detection using sequential hypothesis testing. *Proceedings of IEEE Symposium on Security and Privacy*, pages 211–225, May 2004.

[12] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the Slammer Worm. *IEEE Security & Privacy Magazine*, 1(4):33–39, 2003.

[13] V. Siris and F. Papagalou. Application of anomaly detection algorithms for detecting SYN flooding attacks. *Computer Communications*, 29(9):1433–1442, 2006.

[14] N. Vanderavero, X. Brouckaert, O. Bonaventure, and B. Le Charlier. The HoneyTank: a scalable approach to collect malicious internet traffic. *International Journal of Critical Infrastructures*, 4(1):185–205, 2008.

[15] D. Voelker and S. Savage. Inferring Internet Denial-of-Service Activity. *Proceedings of the USENIX Security Symposium*, pages 9–22, 2001.

[16] H. Wang, D. Zhang, and K. G. Shin. Change-Point Monitoring for the Detection of DoS Attacks. *IEEE Transactions on Dependable and Secure Computing*, 1(4):193–208, 2004.

[17] G. White, E. Fisch, and U. Pooch. *Computer System and Network Security*. CRC Press, 1995.