Network Worm Detection using Markov's and Cantelli's Inequalities

Miranda Mowbray

HP Laboratories Bristol – Filton Road, Stoke Gifford, Bristol BS34 8QZ – U.K

miranda.mowbray@hp.com

Abstract. *This paper presents a method of detecting network worms, which makes use of Markov's and Cantelli's statistical inequalities. This method is compared with a detection method based on one used in a commercial security product, using a data set consisting of over 3 million packets sampled from an enterprise network. The Markov-Cantelli detection method produces considerably fewer false alarms than the comparison method.*

1. Introduction

This paper presents a low-cost method of detecting network worms, so that they can be mitigated. The aim of this research was to meet a set of requirements stated by major customers for a potential commercial product in this area.

The customers' requirements were challenging. It must be possible to detect and report a worm within 10 minutes. The solution must work reasonably well 12 hours after installation. It is not enough for the detection method to detect whether or not the network is currently suffering a worm attack; it is also necessary to identify the particular machines or flows involved in the attack, so that the attack can be mitigated. The method must not produce too many false alarms. Most importantly, the solution must be low cost: some of the products in this area are priced out of the reach of small or medium sized businesses.

These customer requirements have other implications for the detection method. Normal network traffic varies, both between networks and over time, and normal traffic for a given network may change permanently when new machines or applications are added, so it is not possible to know upfront what behavior is normal for a network. Since the solution must work reasonably well 12 hours after installation, and it is not in general possible to obtain data from the network before installation, there cannot be a training period of more than 12 hours for the method to learn normal behaviour for the system. Similarly, the method must adjust to permanent changes in network behaviour without a long training period. The method has to work with sampled data, because collecting data from all traffic packets is too costly. Finally, the method cannot rely on an updating service that provides signatures or behavioral descriptions for new attacks, because such services are costly to operate.

The rest of this paper describes and evaluates a detection method that does manage to meet the customers' requirements. In particular, in simulations comparing it with the detection method used in a commercial network security product, the new method gave less than a quarter of the number of false alarms, even when the parameter governing sensitivity was set four orders of magnitude higher than for the comparison method.

Section 2 describes related work and other approaches to network attack detection. Section 3 presents the detection method. The method makes use of two well-known statistical inequalities, Markov's inequality and Cantelli's inequality. Markov's and Cantelli's inequalities are given in Subsection 3.1, and the detection method using these inequalities by the detection method is presented in Subsection 3.2. Section 4 gives the simulation results. The final two sections discuss possible refinements of the method and possible applications to the detection of some other types of network problems.

2. Related Work

There are essentially two approaches to detecting network attacks, including worms. The first approach relies on knowing signatures or behavioural descriptions either of normal network behaviour [Feather and Maxion 1993] or of known attacks (see eg. [Lewis 1993]), or of both. For example, 376 Bytes sent to UDP port 1434 is a signature of the SQL Slammer, and a possible behavioural description for identifying some malware spread by IRC-controlled zombies is that a machine that performs a peer-to-peer download, then makes a remote IRC connection, and then uses over 90% of local bandwidth. The BLINC classification method [Karagiannis et al. 2005] identifies signatures of communication patterns at the social, functional and application levels. One possibility is to use data mining to generate signatures [Zurutuza et al. 2008].

This first approach is used by Checkpoint and Arbor Networks, the two companies who dominate the market for network attack detection [Kolodgy 2006]. Since normal network traffic varies and the detection method cannot have a long training period to learn its characteristics, a detection method relying on knowledge of long-term signatures or behavioural descriptions of normal traffic is unlikely to be useful for addressing the problem outlined in Section 1. Attackers may adapt their methods to avoid producing the signatures of previous attacks (for example, varying the ports they use, or having self-modifying code, or varying/disguising the types of site used to infect machines and turn them into zombies). An approach based on knowledge of attacks cannot be effective unless it has access to an ongoing service that continuously looks for new attacks or mutations of old attacks, and produces signatures or behavioural descriptions for them. The service ensures that the detection mechanism does not have to reinvent the wheel in the sense of working out for itself how to identify new attacks. However the provision of such a service does not scale well with current high speed networks, and is costly. Checkpoint and Arbor Networks' solutions rely on such services, but a low-cost solution cannot.

This paper does however retain one idea from this approach, which is to choose measures of network traffic likely to reveal the presence of worms. The measures that will be used are ones for which it is hard to design an efficient worm attacks that does not have a large value of the measure. This general principle was introduced in [Williamson 2003].

The second approach to detecting network attacks is to assume that they will produce unusual traffic, and use statistical analysis techniques on network traffic measurements to detect unusual traffic patterns in the network. The most popular statistical analysis technique to use for this purpose is probably Principal Component Analysis [Lakhina et al. 2004]. However, it has been shown that this technique has some challenges, in particular when using this technique it is inherently difficult to pinpoint the flow causing the problem [Ringberg et al. 2007].

An important category of statistical analysis methods is change detection, which detects sudden large changes from forecast behaviour. A comparative study of several change detection techniques [Soule et al. 2005] (including wavelets [Barford et al. 2002]) found the best performer across all validation tests to be the simplest one studied, which gives an alarm if the variance of the difference from the forecast is large. Forecasting methods include experimental weighted moving average (EWMA), Kalman Filtering, Holt-Winters, and Box-Jenkins autoregressive models (see eg. [Krishnamurthy et al. 2003]). The Kalman Filtering model assumes that the measures used are linear combinations of underlying variables with Gaussian distributions. Although it still can give good results for measures that are not too far from this model, the measures used in this paper (and elsewhere) to detect network worms are a very bad fit, as they are discrete-valued with the lower values being most common. Holt-Winters requires the periodicities in the network traffic to be known in advance; section 7 of [Lakhina et al. 2004] demonstrates that this is more difficult than might be imagined. Autoregressive processes can be used to detect abrupt changes correlated between several measures [Thottan and Ji 2003]; however, network worms may only affect one of the measures chosen to reveal the likely presence of worms.

Another statistical technique is to look for abrupt changes in the entropy of network measurements. Although this is intellectually appealing, experiments using sampled traffic collected during worm outbreaks show that very simple measures such as a count of unique destination ports—which is similar to one of the measures that will be used in this paper—can outperform entropy-based techniques [Tellenbach et al. 2007].

The requirement that the solution should work reasonably well after a training period of 12 hours excludes the use of statistical techniques that have a long training period (such as those used by the Mazu Profiler product, which does profiling according to the day of the week—or even according to the week in the quarter—and so requires at least a week’s data for training). In fact, any solutions that rely on knowledge of a detailed traffic profile for the network may have difficulty meeting this requirement. The requirements also exclude the use of statistical techniques that have high computational complexity and thus would be expensive to use to perform real-time traffic analysis in a large network (such as analysis using neural networks [Manikopoulos and Papavassiliou 2002]).

3. A Markov-Cantelli Worm Detection Method

The detection method presented in this section shares with the statistical analysis approach the aim of identifying unusual traffic, based on forecasting from recent behaviour.

The first step of the detection method is to identify measures of sampled network traffic such that any effective worm of a particular class will produce traffic with a high value of the measure. The “IP address sweep”, which is the number of different destination IP addresses connected to by a given source IP address in a minute’s worth of sampled traffic, is one such measure. This is basically the same as the measure introduced in [Williamson 2003], except that Williamson’s measure uses a second’s worth of all connections from the IP address. If a worm propagates rapidly, the IP addresses that it propagates from will have high IP address sweep values. The other two measures used count the number of switch ports used by a flow using a particular protocol, UDP for the second measure and TCP for the third measure, to detect TCP and UDP worms that use port scanning.

For all three measures, efficient worm attacks of the appropriate type will produce unusually high values of the measure, not unusually low values. Moreover, the measures only take non-negative values. The statistical analysis techniques described in Section 2 can be applied to measures that take negative values, so they do not use this fact. In contrast, the detection method described in this section does use this information, via two well-known statistical inequalities, Markov’s and Cantelli’s inequalities.

3.1. Markov’s and Cantelli’s Inequalities

Markov’s inequality states that for any variable X taking only non-negative values, for which the mean m is finite, and any positive real number a,

(1) $\Pr(X >= a) <= m/a$

Cantelli’s inequality (which is also known as the one-sided Chebyshev inequality) states that for any variable X for which the mean m and the standard deviation s are finite, and any positive real number a,

(2) $\Pr(X – m >= a) <= s^2/(s^2+a^2)$

Combining these two inequalities, it follows that for any variable X taking only non-negative values, for which the mean m and the standard deviation s are finite, and for any real number $a> m$,

(3) $\Pr(X >= a) <= \min\{ m/a , s^2/(s^2+(a-m)^2) \}$

Both parts of the right-hand side of (3) are necessary, because neither of the inequalities (1), (2) is stronger than the other. There are some distributions and real numbers a satisfying the conditions of (3) for which the upper bound on $\Pr(X>=a)$ given by Markov’s inequality is tighter than the one given by Cantelli’s inequality, and others for which Cantelli’s equality gives the tighter bound.

Suppose that $0<p<1$ and the conditions for (3) hold. Let $r>0$ be a positive real number. Define T (which depends on m , s , p, r) as follows:

(4) $T = \min\{ m/p ,\ s.(1/p-1)^{1/2} + m \}$ if $s>0$, otherwise $T = m+ r$

then (3) implies that

(5) $\Pr(X >= T) <= p$

3.2. Using the Inequalities

The detection method has as parameters a threshold probability p with $0<p<1$, a window size w, and a small real number $r > 0$. (Subsection 4.1 will discuss how to choose these parameters).

When a sampled packet passes through the switch that is operating the detection method, the switch records the information necessary to later calculate the measures. Once a minute, the first measure is calculated for each source IP address that had sent at least one packet sampled by the switch during the previous minute, and the other two measures are calculated for each flow containing a packet sampled by the switch during the previous minute. For each measure, the value calculated for each relevant source IP address or flow is appended to a list of values found for that measure.

When a multiple of w packets has been sampled, an estimated mean m_1 and standard deviation s_1 for the first measure are calculated as the sample mean and standard deviation of the values in the list of values found for the first measure, and the list is emptied. (Note that the calculation of m_1 and s_1 uses data from sampled packets from all source IP addresses that sent at least one of the preceding w packets. It is not the case that different means are estimated for measures from different source IP addresses, rather one overall mean is calculated for the first measure using data from all the previous w sampled packets. This point will be returned to later, in Section 5.) The current threshold value for the first measure is then set to T_1, where T_1 is given by setting $m = m_1$, $s = s_1$ in equation (4). An analogous process is carried out for the second and third measures, setting the current thresholds T_2, T_3 for these measures.

The current thresholds are then used to detect possible worms. Whenever the value of a measure for some IP address or flow is calculated at the end of a minute and is found to be greater than or equal to the current threshold for that measure, an alarm is raised for this IP address or flow.

The justification for raising an alarm in this circumstance is that by (5), there is probability at most p that the data point in question comes from *any* distribution with non-negative values whose mean and standard deviation are those estimated from the data from the preceding w sampled packets. This justification uses the assumptions that the underlying distributions of the measures in normal traffic take only non-negative values and have finite mean and variance. It does not make any other assumptions about these distributions. In particular, it does not assume that they can be well approximated by Gaussians.

Since all the measures are used in the same way, the detection method can be very easily extended to make use of new measures. If a new measure of a minute's sampled traffic is discovered which takes non-negative values, and takes unusually high values if an effective worm of a particular type is present, it would be straightforward to use the new measure in addition to (or instead of) the three described above. For example a measure combining the IP address sweep with the failure ratio, inspired by RBS+TRW [Jung et al. 2008], might perform well. Measures based on data collected over other time intervals other than a minute could be used in the same way.

3.3. After an Alarm is raised

The detection method is intended to be used as an initial filter within a more complex network security system. The effect of the alarm is that information about the possible attack is passed on to another part of the software for further processing to decide whether or not the alarm will be passed on to the network administrator.

For example, machines indicated as suspicious for fewer than three consecutive minutes might not be reported to the administrator, and/or machines that have already been recently reported to the administrator as suspicious might not be reported again if they continue to be suspicious. There are network security products that correlate and prioritize security alarms received from different sources in different parts of the network, suppressing those with low priority. For instance QRadar by Q1Labs can suppress alerts of attacks by less virulent attackers on less vulnerable and less financially valuable network objects reported by less reliable alert sources. Other network security solutions allow administrators to set the system to suppress alarms concerning a particular machine that is a frequent cause of false alarms, or future alarms arising from a particular incident that is already under investigation.

If the alarm is not suppressed, and the administrator decides to take action, there need to be effective mechanisms available for mitigating the attack - for example, the administrator might be able to temporarily or permanently block the relevant machine or flow from the network. If there are no effective mitigation mechanisms then there is not much use in detecting the worm.

4. Simulation Results

This section reports the results of simulations of the Markov-Cantelli detection method described in the previous section. The input data set for all the simulations consists of data from over 3 million packets, which were sampled (using sFlow-type sampling [Phaal et al. 2001]) from a switch in an enterprise network. A 12-day sampling period was used, long enough to check for differences in network use patterns on weekends.

As a comparison, simulations using the same input data set are also reported for a method that sets the alarm threshold for a measure to be such that there is probability q of a data point at the threshold occurring in a Gaussian distribution whose mean and variance are equal to the estimates calculated as described in Subsection 3.2. The value q is a parameter for this method. This method of using Gaussians to model the distributions of these measures in normal traffic is used as an initial filter for worm detection by a commercial network security product. (This paper does not name the product or its manufacturer, for reasons of commercial confidentiality.) It is possible that other commercial products operate in a similar fashion, but this is hard to determine: understandably, network security companies can be reluctant to make public the exact algorithms that they use for worm detection. The unavailability of this information meant that it was not possible to make a comparison with a different commercial product.

4.1. Choice of Parameters

The easiest of the three parameters p, w, r to choose is r. The smaller r is, the tighter the bound given by (5) is for the case $s = 0$. It does no harm to choose r as small as possible, commensurate with the arithmetic precision of the calculations used by the detection method. Throughout this section *r* is set to 10^{-6}.

The window size *w* has more complicated effects. On the one hand, a larger window means that the estimates of the mean and standard deviation of measures are based on more data, and so are more likely to be accurate, provided that the short-term distribution for normal traffic does not change much during the time of the sampling of the *w* packets from which the mean and deviation are estimated and the subsequent *w* packets for which this estimate is used. More accurate estimates of the mean and standard deviation should mean more accurate identification of worms. Also, if the window size is small, there is a danger that a sufficiently large proportion of the *w* sampled packets will be from a worm that the detection method will estimate the worm's behaviour to be normal. On the other hand, in time periods during which the distribution does change significantly, a larger window will have the effect that the detection method will take a longer time to adjust to the new distribution.

An additional consideration is that no alarms are raised while the initial *w* packets are sampled. Since the initial training period is required to be at most 12 hours, the value *w* must be small enough for the first *w* packets to be sampled within this time. An alternative design for the detection method would have been to recalculate thresholds at fixed time intervals (of at most 12 hours), instead of every time a multiple of a fixed number of packets had been sampled. One reason for not choosing this alternative design is that an attacker might be able to determine the times when the thresholds would be recalculated, and take advantage of this knowledge. Another reason is that it seems sensible to recalculate thresholds more frequently during periods of heavy traffic. In practice it may be difficult to be certain that at least *w* packets will be sampled during the first 12 hours, but there is a work-around, which is to do the first threshold calculations a fixed time interval after the start, and subsequent threshold calculations after a multiple of *w* sampled packets.

For the simulations, however, the input data set is known in advance, so this problem does not arise. The number of packets sampled in the first 12 hours of the data set is slightly over 150,000. In the simulations *w* takes values between 50,000 and 150,000 in steps of 6,250, with a default value of 100,000.

The parameter *p* (or *q*) governs the sensitivity of the detection method. Too high a value will have the effect of there being unacceptable numbers of false alarms. Too low a value will result in the detection method failing to detect some worms. In a test simulation in which *p* (the threshold probability for the Markov-Cantelli method) was set equal to *q* (the threshold probability for the comparison method), both the methods clearly identified a possible worm; however, there was a huge difference in the number of alarms generated by the two methods, with the comparison method producing many more alarms. Because of this, in all the simulations reported below *q* is set to a small value, 10^{-6}, and *p* is set to values that are orders of magnitude larger than *q* – between 0.001 and 0.01 in steps of 0.0005, with a default value of 0.01.

The simulations whose results are reported below consist of 19 simulations of the Markov-Cantelli method with default window size and different threshold probabilities; 16 of the Markov-Cantelli method with default threshold probability and different non-default window sizes; and 17 of the comparison method with default threshold probability and different window sizes.

4.2. Illustration: Threshold for the First Measure

Figure 1 illustrates the operation of the Markov-Cantelli detection method by showing how the threshold for the first measure changes over time, based on the 12 days' input data set. Each point represents a source IP address during a minute's time interval. The y-coordinate indicates the IP address sweep measure for this source address. The line shows the alarm threshold for this measure; source IP addresses corresponding to points above the line are regarded as potential worm sources.

It can be seen that the threshold adjusts flexibly to changes in normal traffic. The peak after 12000 minutes is identified as a possible worm attack, but the values of the measure for the normal traffic shortly after this peak lie beneath the threshold, even though they are higher than those typical in the normal traffic before the peak, and also higher than previous threshold values. It can also be seen that the detection algorithm does not become desensitized as a result of these higher values. As the values return towards previous levels towards the end of the 12 days, the threshold decreases with them.

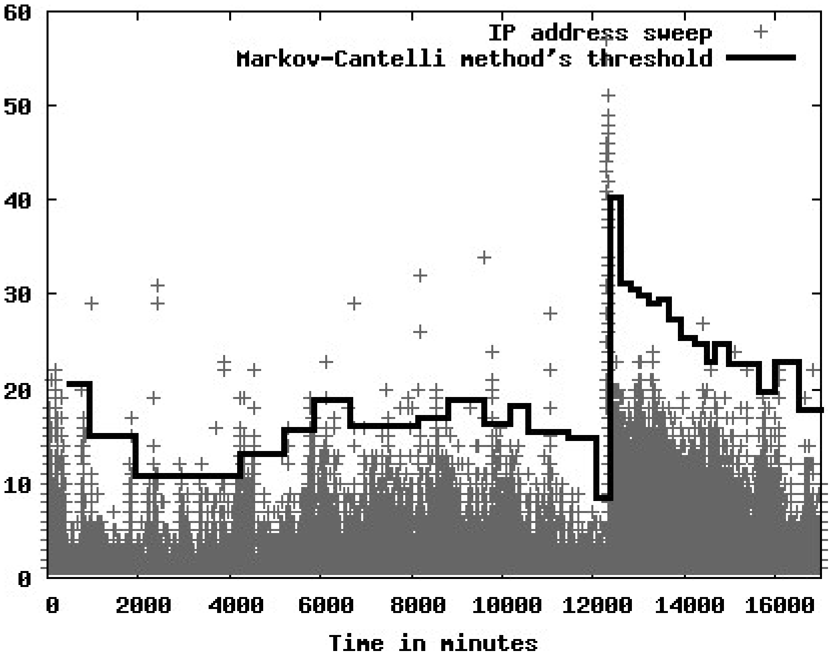

Figure 1. IP address sweep measure for the input data set. The line indicates the threshold value generated by the Markov-Cantelli detection method with *w* = 100,000, *p* = 0.01.

Figure 2 shows the equivalent graph for the comparison detection method with the same window size and with threshold probability 10^{-6}. It can be seen that this detection method also adjusts flexibly to the changes in traffic, but that it produces many more alarms than the Markov-Cantelli detection method illustrated in Figure 1, even though the threshold probability is four orders of magnitude lower than that for Figure 1.

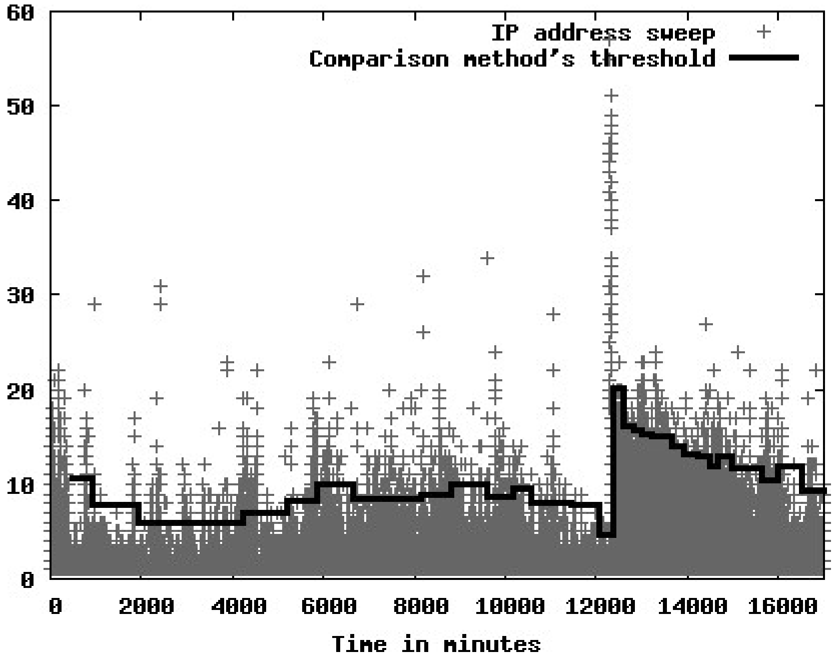

Figure 2. IP address sweep measure for 12 days' data. The line indicates the threshold value generated by the comparison detection method with w = 100,000, $q = 10^{-6}$.

For all the parameter values used in the simulations, both detection methods identified the peak in the IP sweep measure shortly after 12000 minutes as possibly indicating a worm; each simulation resulted in at least 45 alarms in connection with this peak. It is unlikely that all these alarms would be suppressed as the result of the further processing by the network security system. On the other hand, it is quite possible that most of the alarms connected with the peak would be suppressed. They are clustered in time and involve only two source IP addresses, so it would be straightforward for the system to identify that they may be all the result of just one or possibly two incidents, and to suppress later ones so as not to annoy the network administrator.

The values taken by the other two measures did not exhibit suspicious peaks of this type.

4.3. Effect of changing the threshold probability

Figures 3 and 4 show the effect of changing the threshold probability p on the number of alarms given by the Markov-Cantelli detection method. The larger p is, the more sensitive the detection method is, and so the more alarms there are.

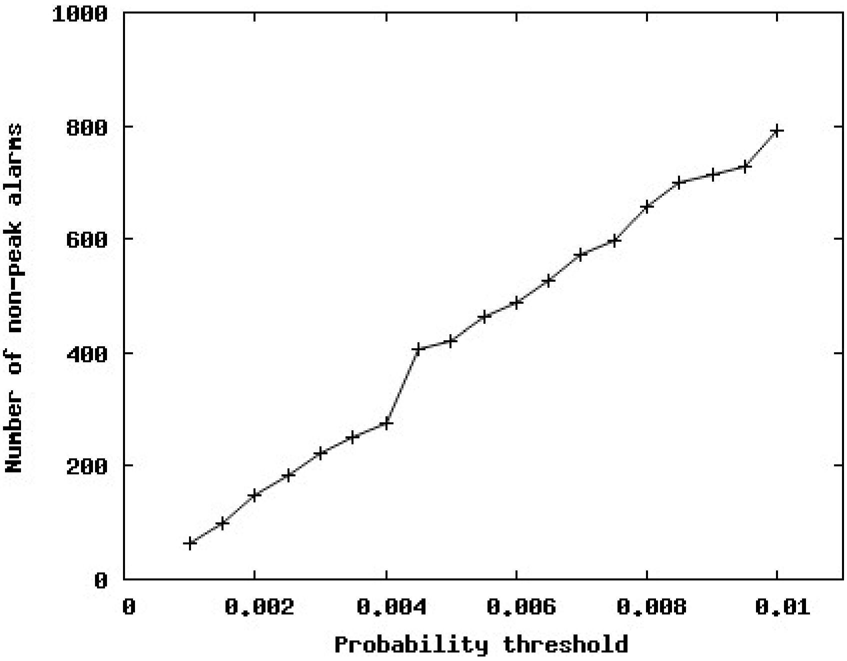

Figure 3. Effect of *p* on the number of alarms for the Markov-Cantelli detection method (*w* = 100,000). The y-coordinate gives the number of alarms given over the 12 days, resulting from any of the 3 measures, but excluding those connected with the peak of the first measure.

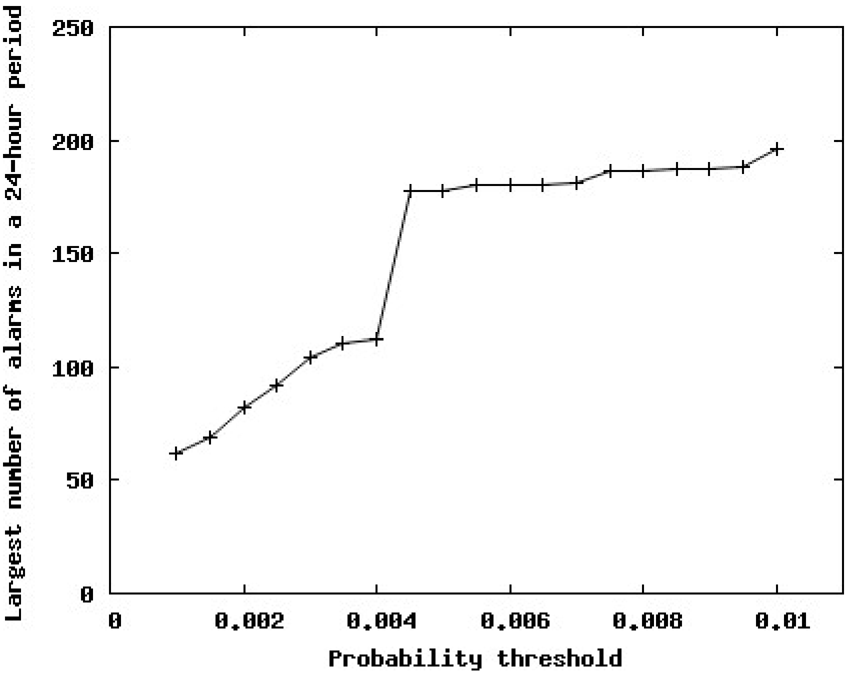

Figure 4. Effect of *p* on the number of alarms in a day for the Markov-Cantelli detection method (*w* = 100,000). The y-coordinate gives the maximum number of alarms in any 24-hour period over the 12 days.

Figure 3 shows that the number of alarms that are not connected with the peak in the first measure increases approximately linearly with *p*. Even for the highest value of *p* simulated, the number of packets that raise a non-peak alarm is less than 800, out of a total of over 3 million sampled packets. The number of alarms that are reported to the network administrator will be smaller again. For example, if alarms are not reported unless the relevant source IP addresses or flow has also been indicated as suspicious in the preceding two minutes, this reduces the number of reported non-peak alarms to 290. For $p = 0.001$, this reduces the total number of reported non-peak alarms from 62 to 34.

The average alarm rate is not the only statistic of interest connected with alarms. It is easier for network administrators if the alarms are not too closely clustered, so that as an extreme example the nearly 800 non-peak alarms for p=0.01 indicated in Figure 3 do not all occur in the same half-hour. Figure 4 shows how the value of *p* affects the total number of alarms (including alarms connected with the peak of the first measure) in the worst 24-hour period, and demonstrates that the effect of *p* in this case is nonlinear.

Given that the simulations with p=0.001 clearly identified the peak in the first measure as a possible worm and gave the fewest non-peak alarms, it might be sensible to use this value rather than p=0.01 when using the Markov-Cantelli detection method in practice. The reason for using p=0.01 as the default value in these simulations was to make clear the very sizeable difference in performance between this method and the comparison method.

4.4. Effect of changing the window size

Figures 5 and 6 illustrate the effect on alarm numbers of changing the window size *w*. The y-coordinates give values of the same statistics as in Figures 3 and 4 respectively. The graphs show the simulation results both for the Markov-Cantelli detection method with threshold probability 0.01, and for the comparison method with threshold probability 10^{-6}. As in the comparison between Figures 1 and 2, the most striking feature of these two graphs is that the comparison method gives many more alarms (more than four times as many, for every window size simulated), despite using a threshold probability that is orders of magnitude lower.

Figure 5 shows that the number of non-peak alarms decreases slightly as the window size *w* increases. One reason for this is that none of the first *w* packets can give rise to an alarm, but the gradients in Figure 5 are steeper than would be the case if this were the only reason. There is also a contribution from the fact that the estimates of mean and variance are more accurate for larger window sizes. Figure 6 shows that the slight decrease in overall numbers of non-peak alarms does not translate neatly into a slight decrease in the numbers of alarms in the worst 24-hour period, as this statistic is more volatile.

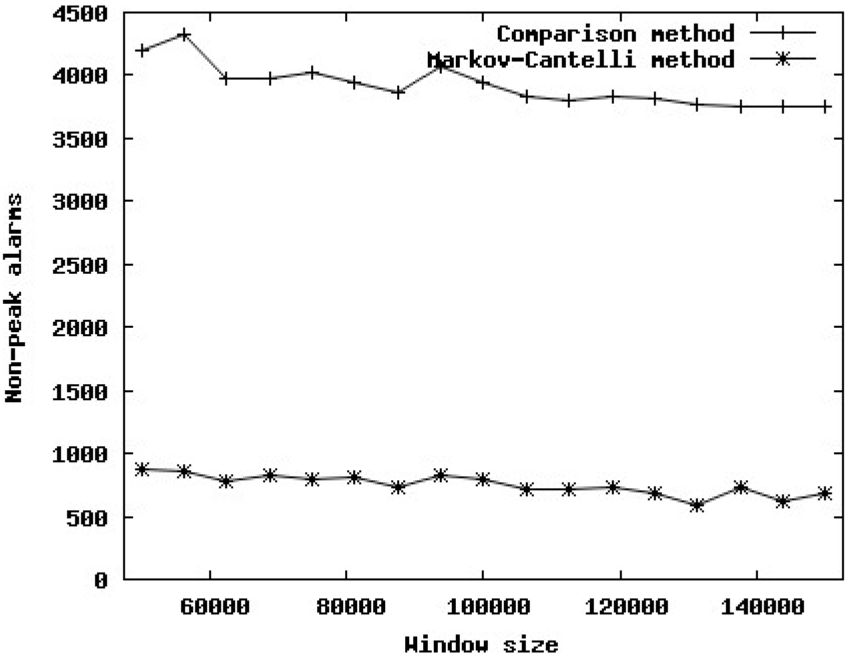

Figure 5. Effect of w on the number of alarms for the Markov-Cantelli detection method with p = 0.01 and the comparison method with q =10^{-6}. The y-coordinate gives the number of alarms given over the 12 days, resulting from any of the 3 measures, but excluding those connected with the peak of the first measure.

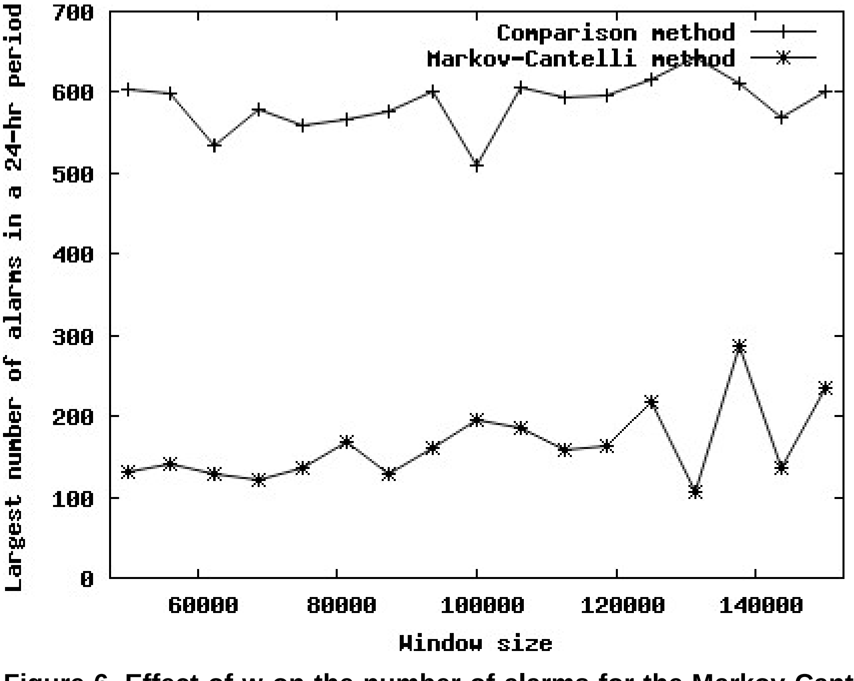

Figure 6. Effect of w on the number of alarms for the Markov-Cantelli detection method with p = 0.01 and the comparison method with q =10^{-6}. The y-coordinate gives the maximum number of alarms in any 24-hour period over the 12 days.

5. Refinements

It was mentioned in Subsection 3.2 that the Markov-Cantelli detection method calculates one threshold for use with each measure; it does not calculate different thresholds for different source IP addresses or different network flows. This may be a cause of inaccuracy if, as is often the case, different flows have very different typical traffic characteristics. A possible refinement to the detection method is to set different thresholds for the same measure applied to different flows. Instead of the single list for each measure described in Subsection 3.2, there will be a separate list for each pair consisting of a measure and a flow, or a measure and a source IP address.

One potential problem with this refinement is that if a new flow caused by a worm attack enters the network, its behaviour will be assumed to be normal. Another potential problem is scalability: it may be expensive in terms of computation and storage to keep track of every flow in a large network. Moreover, flows with very low traffic may not be sampled often enough recent data to allow accurate estimates of the means or variances of the measure applied to the flow. A compromise is to calculate separate current thresholds for the top few flows from which the largest number of packets have been sampled in the last window, and to lump together all the other flows, calculating one overall current threshold for them based on the sampled packets from flows other than the top few. If a threshold still needs to be calculated without there being enough sampled packets to give accurate estimates, previous estimates can be reused.

Another possible refinement is that if an alarm is raised in connection with a sampled packet, the values of the measures for that packet are not added to the lists used for the calculation of thresholds. The estimated mean and variance are intended to model normal traffic, so if a packet is suspected not to be normal traffic it is reasonable to exclude it when building the model.

6. Other Applications

The detection method described in Section 2 may also detect - and hence assist in mitigating - some network intrusions, denial-of-service attacks, misconfigurations, and hyperactive rogue servers, which may cause unusually high values of the measures used for worm detection.

By using other measures with appropriate properties, it should be possible to use the Markov-Cantelli detection method to detect other types of network problems. For example, it could be used to detect network attacks that use DNS tunnelling, by using as a measure the number of DNS packets sent by a particular source IP address in a minute.

Acknowledgement

This paper is a result of a collaboration between the author and Matt Hall. Matt contributed several of the best ideas in this paper, and also provided the input data set for the simulations. The paper would not exist without him.

References

Barford, P., Kline, J., Plonka, D. and Ron, A. (2002). A Signal Analysis of Network Traffic Anomalies. *Proceedings of the ACM SIGCOMM conference on Internet*

Meaurement (IM'02), pages 71-82.

Feather, F and Maxion, R. (1993). Fault Detection in an Ethernet Network using Anomaly Signature Matching. *ACM SIGCOMM Computer Communication Review* 23(4):279-288.

Jung, J., Milito, R. and Paxson, V. (2008). On the adaptive real-time detection of fast-propagating network worms. *Journal in Computer Virology*, 4(3):197-210.

Karagiannis, T., Papagiannak, K. and Faloutsos, M. (2005). BLINC: multilevel traffic classification in the dark. *ACM SIGCOMM Computer Communication Review,* 35(4):229-240.

Kolodgy, C. (2006). Worldwide Threat Management Software 2006-2010. IDC study.

Krishnamurthy, B., Sen, S., Zhang, Y. and Chen, Y. (2003). Sketch-based Change Detection: Methods, Evaluation, and Applications. *Proceedings of the ACM SIGCOMM conference on Internet Management (IM'03)*, pages 234-247.

Lakhina, A., Crovella, M. and Diot, C. (2004). Diagnosing Network-Wide Traffic Anomalies. *ACM SIGCOMM Computer Communication Review* 34(4):219-230.

Lewis, L. (1993). A Case-Based Reasoning Approach to the Management of Faults in Communication Networks. *Proceedings of IEEE INFOCOM'93*, volume 3 pages 1422-1429.

Manikopoulos, C. and Papavassiliou, S. (2002). Network Intrusion and Fault Detection: A Statistical Anomaly Approach. In *IEEE Communications Magazine*, Oct 2002 edition, pages 76-82. IEEE Press.

Phaal, P., Panchen, S. and McKee, N. (2001). InMon Corporation's sFlow: A Method of Monitoring Traffic in Switched and Routed Networks. IETF RFC 3176.

Ringberg, H., Rexford, J., Soule, A. and Diot, C. (2007). Sensitivity of PCA for Traffic Anomaly Detection. *Proceedings of the ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 109-120.

Sorenson, H. (1985), Kalman Filtering: Theory and Application, IEEE Press.

Soule, A., Salamatian, K. and Taft, N. (2005). Combining filtering and statistical methods for anomaly detection. *Proceedings of the ACM SIGCOMM conference on Internet Measurement (IMC'05)*, pages 331-344.

Tellenbach, B., Brauckhoff, D. and May, M. (2007) "TIK Report 275", ftp://ftp.tik.ee.ethz.ch/pub/publications/TIK-Report-275.pdf, June.

Thottan, M. and Ji, C. (2003). Anomaly Detection in IP Networks. *IEEE Transactions on Signal Processing* 51(8): 2191-2204.

Williamson, M. (2002). Throttling viruses: Restricting propagation to defeat malicious mobile code. *Proceedings of the Annual Computer Security Applications conference (ACSAC'02)*, pages 61-68.

Zurutuza, U., Uribeetxeberria., R. and Zamboni, R. A Data Mining Approach for Analysis of Worm Activity Through Automatic Signature Generation. *Proceedings of the 1st ACM workshop on AISec*, pages 61-70.