Data Classification for Artificial Intelligence Construct Training to Aid in Network Incident Identification using Network Telescope Data

ABSTRACT

This paper considers ways to collect appropriate data for the training of Artificial Intelligence constructs, such as Genetic Algorithms and Neural Networks, for the identification of potential network incidents using passive network telescope data. Passive network telescopes provide a useful measure of anomalous traffic on the Internet. This is due to the fact that all network traffic received by a network telescope is by nature unsolicited. It thus follows that this traffic is either malicious or due to possible misconfiguration. This paper considers the need for automated means of analysing Network Telescope due to factors such as the vast amount of data available and the difficulty of classification. A proposed solution to this need for automation is to use make use of Artificial Intelligence techniques such as neural networks that can be trained to detect sudden changes in the composition of network traffic. A core requirement of Artificial Intelligence techniques is the need for reliable and definitive training data. While a large amount of data obtained from network telescopes exists, currently this data is not marked for known incidents. A concern related to marking data is the time involved and the accuracy of the markings. To solve these issues two methods of data generation are considered: manual and automated generation. The manual technique considers heuristics for finding network incidents while the automated technique considers building simulated data sets using existing models of virus propagation and malicious activity. Using these marked data sets some example Artificial Intelligence systems are designed for network incident discovery.

Categories and Subject Descriptors

C.2.3 [**Network Operation**]: Network monitoring; I.2.6 [**Learning**]: Knowledge acquisition

Keywords

Network Telescope, Trend Analysis, AI

1. INTRODUCTION

This paper concerns itself with three major concepts: Network Telescopes, Mathematical Modeling and Artificial Intelligence. These concepts are now discussed together with a motivation for such research and possible fields of application.

1.1 Network Telescopes

Network telescopes provide a looking glass into the world of malicious activity on the Internet [17]. This is accomplished through passively listening to traffic bound to an address space to which there are no hosts present as illustrated in Figure 1. We can assume from the fact that all traffic monitored from a network telescope is anomalous due to the fact that the traffic is completely unsolicited [15]. Further, in the case of totally passive telescopes it is impossible to complete any sort of handshake as no acknowledgements are sent back [12]. It is noted that these telescopes receive a considerable amount of traffic, though this is dependent on the size the network telescope's IP Space. For example, CAIDA (The Cooperative Association for Internet Data Analysis) [3] received, at peak times during May 2010, 1.25 Gb/s worth of data [10]. This equates to roughly 562.5 GB worth of data an hour, assuming the worst case of a consistent 1.25 Gb/s worth of data. Although, for most trend based analysis the actual packet payload is unnecessary. This reduces the amount of data considerably. Considering that mostly Ethernet Packets stacked with IPv4 and TCP for control will be observed, the packet header of each packet will typically be 56 bytes in size. This of course is an assumption of average, discounting mangled packet headers and potentially larger packet headers in the case of IPv6. CAIDA received a maximum of 190000 packets/s in the Month of May 2010 [11]. Using this together with the average packet header a value of approximately 35.67 GB/hour. While this represents a considerable decrease in the data to be analysed it should be considered that network analysis is a complex process with many factors to consider. Considering even a simple port analysis for active ports is something that would require considerable man power to accurately analyse. Thus it is necessary to consider adaptive automated techniques for the analysis of network telescope data.

1.2 Artificial Intelligence

Artificial Intelligence is a field within Computer Science that attempts to mimic the learning capabilities found in nature [21]. This varies from modeling the neural networks present in the human brain to adopting evolutionary techniques present in organisms. It however does not constraint itself to the limitations found in biology. Popular techniques in this field include Artificial Neural Networks (ANN), Genetic Algorithms and Bayesian Net-

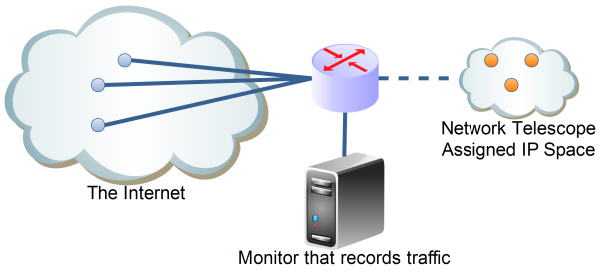

Figure 1: Simple network diagram depicting the nature of a network telescope

works. These techniques require either some sort of mechanism to determine whether they are being trained correctly. In ANN's this is provided by a set of marked training data. Genetic Algorithms solve this issue by defining a fitness function which allows the algorithm to know when an optimal solution has been found. It is this learning capability which is a attractive to incident identification as the nature of vulnerability, exploits and attacks used by cybercriminals is constantly changing.

1.3 Mathematical Modeling

It is useful when considering a complex system to produce a simplified model. That is a number of variables are defined related to the system. It is the relationships between these variables which simulate how the real systems acts. Mathematical modeling is a the field of mathematics which concerns itself with this sort of work and involves modeling population growth, physical systems and other real world phenomena. Useful functions in this realm includes: Ordinary Differential Equations, Systems of Differential Equations, Markov Systems and Non-linear systems.

1.4 Motivation and Application

There is a clear need to be able detect network incidents as they occur as this allows for measures to limit the spread of damage caused by such an incident. From a organizational and South African context understanding the nature of anomalous traffic bound for an organization or country allows the analyst to compare relative to other countries and organizations. Particularly, pertinent to South Africa, considering the planned increase in South African International bandwidth capacity, considering plans such as EASSy [2] and Main One [14].

1.5 Paper Organization

2. RELATED WORK

This section considers the work done by research bodies considering the three previously mentioned focuses of this paper.

2.1 Network telescope analysis

A considerable amount of work has been conducted by the Information Security research community at large with regards to work in the field of Network Telescope analysis. In particular the researchers at CAIDA [3] work defining the fundamentals of Network Telescopes [25]. Other work conducted by CAIDA includes observing large network incidents as they occur in particular Code Red Worm [23], SQLSlammer [22] and Witty Worm [28]. They have also developed frameworks for creating Distributed Network Telescope Nodes for the monitoring and analysis of network traffic on a global scale [13]. Work within Rhodes University has considered the relation between logical distances and packets collected by Network Telescopes [7], the graphical representations of network incidents through the use of InetViz a tool developed by van Riel and Irwin [31, 33] and mapping the Internet through space filling curves such as the Hilbert Curve [18]. While there has been worked conducted considering the statistical analysis of network traffic by clustering [20] and other means. Little work considers analysis by "simple" security metrics. This analysis acts as a stepping stone, allowing for future work into automated techniques for incident discovery. This is done by building upon the heuristics and observations obtained through manual analysis.

2.2 Use of Artificial Intelligence Techniques

Artificial Intelligence is a very topical field within computer science and a large amount of work has been done using Artificial Intelligence constructs for pattern matching. Neural networks have been used to aid in rule classification for aviation accidents [16], routing in wireless networks [6] and Data mining [9]. Artificial Intelligence is often applied in the fields of robotics, optimization and pattern identification.

2.3 Mathematical Modeling of Network Incidents

A number of researchers have considered previous network incidents and modeled them using existing models for population growth. Examples of these include: Modeling Code Red Worm as Logistic Growth [26], Virus modeling using the SIRS model [19] and Modeling the spread of Conficker C [34]. The rationale behind this type of research is that if it is understood how previous incidents occurred, it may be possible to predict future behaviour and thus understand how this influences the functioning of network as whole during these incidents. This allows the people who protect and monitor these networks to make informed decisions.

3. DATA SUMMARIZATION

Data summarization is an important process in the analysis of network traffic as it reduces the data set into more manageable components from which more meaningful analysis can be made. The original data set consists of over 33 million packets captured at Rhodes University during the time period August 2005 and September 2009 [7, 8]. This data was processed and imported into a SQL database consisting of entries containing the relevant components of each packet header. The packet payload was not included due to space and processing constraints. This data was then reduced to a smaller subset of numeric measures which provided a description of the data considering averages, medians, deviations and extrema. These statistics included Packet Counts, Summed Packet Size, Average Packet Size, Standard Deviations in Packet Size, Average TTL, Standard Deviation in TTL and Count per Hosts /32, /16 and /8. This data was grouped according to date at the hourly, daily, quarterly and yearly interval. The data was further subdivided according to type TCP, UDP and ICMP with subgroups by port for UDP and TCP and by ICMP type for ICMP.

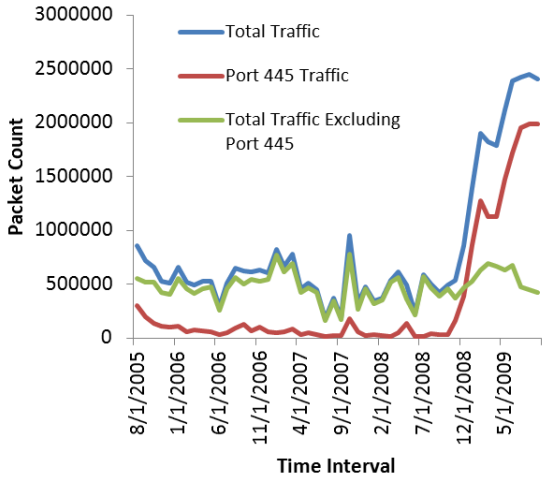

Figure 2: Plot of Packet Counts per month between August 2005 to September 2009, Highlighting the significant effect Conficker had on the distribution of anomalous activity collected from the telescope.

4. NORMALITY AND MEASUREMENT

It is important when attempting to identify incidents that are anomalous, to have some sort of base "normality" to compare relative to. To solve this complex issue, the researchers consider normality of network traffic to be the normality of a number of different measures. Some of these measures include mean, minimum, maximum and standard deviation of a quantity. Median and expected deviation are examples of what are known as "Robust Measures". Robust measures are useful as they are resistant to being influenced by large outliers in data sets. However they are more computationally expensive. This computational cost becomes an issue when dealing with large data sets, such as the RU Telescope which at current has in excess of 4 million packets stored. Noting variations in these measures or a combinations of variations will usually be indicative of a network event. This is further discussed with examples provided in Section 6. An important question is the effect that incidents have on the nature of normality. Considering Figure 2 it can be seen that the total packet counts remain relatively stable, taking into account other events have occurred and the growth in available bandwidth world-wide, until about approximately October 2008. At this point it is noted that there is a clear relationship between variations in the Total Traffic and Traffic bound for port 445. This is clearly due to the Conficker worm [32]. This is problematic for measuring future incidents due to a skewing of numerical measures. This problem is solved by removing the anomalous activity from the dataset and recalculating the measures. This is shown by the green line in Figure 2.

5. INCIDENT DISCOVERY

This paper considers two classes of incident identification. Manual identification concerns itself with human analysis and observation which then leads to incident identification. This is a broad field to work in and it is possible consider things like analysing firewall appliance logs, SNORT logs and observing live packet dumps using tools like TCP Dump. However the focus of this paper is limited to network telescope data alone. The majority of this manual identification is achieved by considering summaries devised, in section 6 . Some of these rules can also be derived from theoretical knowledge such as DDoS Backscatter and application of the Central Limit Theorem.

6. MANUAL IDENTIFICATION

Through data exploration a number of useful measures for abnormal activity were discovered. Some of these measures are discussed here with the relevant event they detected or could potentially detect. How this data could be potentially marked as training data for Artificial Intelligence constructs is discussed. Finally some of the weaknesses of this type of manual analysis are considered.

6.1 Ratio Analysis

A ratio of the total observed count and a particular quantity provides a useful and scalable measure of the traffic distribution. Whereas a plain count is not sensitive to changes that occur should there be a sudden increase in traffic flow, a ratio adapts to these changes adequately. In particular, a ratio of top quantities are of interest as these show a clear variation in traffic composition

6.1.1 Variation in Packet Count Ratios

It is expected that "top ports" will constitute the majority of packet counts. However some degree of normality is expected within these top ports assumed from the Central Limit Theorem. Thus it can be concluded that major shifts in these ratios are indicative of anomalous behavior. Mathematically this rule could be described as $P/n > c \times P_{average}/n$ where n is the sample size considered at a time interval, c an empirically derived constant which is context sensitive, P is the observed count and $P_{average}$ is the average count for previously observed intervals. From Figure 3 it is observed that in 2009 approximately 73% of all traffic received was destined to port 445. Clearly this exceeds the previous average for port 445 and considering the time period this anomaly is clearly caused by Conficker. Again the issue of time resolution is present as a yearly count identifies the incident in 2009 and not late 2008.

6.1.2 Distribution of top port counts

It would be expected that in a normal traffic distribution that the top ports would have packet counts that are well distributed over the entire period. If this were not the case it is possible that some anomalous activity has taken place. Table 1 provides measures of port 5678, which had been identified as one of the top 40 ports in the dataset. It is clear from this table that most of this data lies within one specific year and may be indicative of anomalous behavior. Researching relevant trusted security websites yields exploits that existed in Symantec Netware on port 5678 [4] during early 2008.

6.2 Analysis by Deviation in Packet Size and ICMP Type

The standard deviation, σ, of a measured quantity provides a sense of the spread of the measured quantity. While it is computationally easy to calculate, relatively speaking, it is not a robust measure, that is, it is subject to be influenced by large outliers. While this may be inappropriate when dealing with identifying norms, large outliers are exactly what may tip an analyst off to possi-

Date	Packet Count	Percentage of Total Count
2005	2	0.0037
2006	1	0.0018
2007	0	0
2008	4390	99.9945

Table 1: Table of Packet Counts and Percentage of Total Count for port 5678 grouped at a yearly level

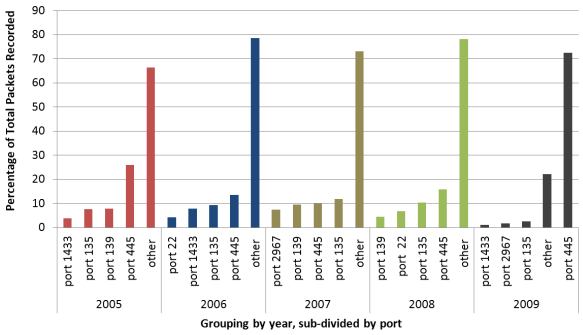

Figure 3: Percentage of total traffic for a given port grouped on a yearly scale. This graph shows a major shift in the distribution between 2008-2009, that is the majority of traffic recorded is bound for port 445.

ble incidents. As previously mentioned, from the Central Limit Theorem, it is expected that approximately 95% of all data to lie within two standard deviations of the mean. The above considerations allow us to construct a useful rule in terms of variation. That is, if there is a sudden change in the deviation of a quantity for a sufficiently large number of samples, then an incident has most likely occurred. Mathematically this can be expressed as $\sigma > 2 \times c \times \sigma_{old}$ where σ is the measured standard deviation, c is an empirically derived constant and σ_{old} is the previous measurement for the standard deviation. This measure is useful in incident discovery if it is considered that some viruses randomly pad the payload of their packets in an attempt to prevent their packets from being filtered out by a firewall and making it more difficult to design IDS (Intrusion Detection System) signatures to match the viral traffic. A sharp change in the amount of traffic on a specific port or ICMP type coupled with a change in standard deviation in packet size is a good numeric description of this sort of anomalous behavior

6.2.1 *Variation in Packet Size*

For this section port 2967 will be considered as a port of interest. The minimum size of a packet assuming an Ethernet frame is 64 bytes. In general it is expected that most packets to just above this minimum size as illustrated in Table 2. However averages in general make poor measures of variability unless the time resolution is sufficiently fine-grained. Considering the ports with the highest variation in 2007 a number of potential incidents are considered. If the deviations at monthly level are considered for port 2967 there does not appear to be any significant difference in the standard deviation as shown in Table 3. However, considering a daily standard deviation reveals 20 February 2007 to be a day of interest as shown in Table 4.

Date	Average Packet Size
February 19, 2007	61.57
February 20, 2007	69.12
February 21, 2007	60.88
February 22, 2007	60.93

Table 2: Packet Size Averages grouped at a daily level

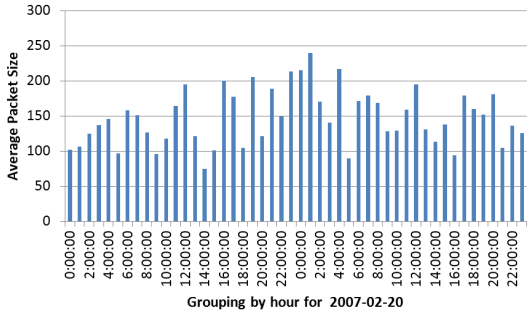

Figure 4: Plot depicting the average packet size for port 2967 for 20 February 2007 grouped at an hourly level. It is clear to see there is a large amount of variation in packet size which was atypical of previous readings.

Plotting the deviation at an hourly level shows the considerable deviation in packet sizes during this time period as shown in Figure 4. This illustrates the point that the time interval considered is highly significant. An investigation from reputable Internet security sources yields that W32.Rinbot [30] started to emerge in the wild as early as February 2007. It exploited vulnerabilities in applications running on port 2967 through specially crafted RPCs (Remote Procedure Calls) with randomized packet size padding.

6.3 Analysis by counts

Counts of data grouped according to some criteria provides a simple and fast measure of normality by comparison with previously measured means. It should be noted that this sort of measure is prone to be ineffective should there be a significant change in the logical or physical network topology of the Network Telescope.

6.3.1 *Cases of denial of service through spoofing*

Month	Packet Count	Std. Dev of Packet Size
December 2006	159675	0.66
January 2007	144334	2.93
February 2007	104435	4.19
March 2007	96008	2.78
April 2007	20062	2.83
May 2007	16666	1.96
June 2007	21828	2.98
July 2007	25812	1.69
August 2007	7323	4.07

Table 3: Packet Count and Standard Deviation of Packet Size measured at monthly intervals

Day	Packet Count	Std. Dev of Packet Size
February 16, 2007	3142	0.23
February 17, 2007	14029	0.85
February 18, 2007	3423	0.88
February 19, 2007	5802	1.48
February 20, 2007	21563	57.37
February 21, 2007	3006	1.88
February 22, 2007	3507	1.61
February 23, 2007	5049	1.00

Table 4: Table of Packet Count and Standard Deviation of Packet Size measured at daily intervals

Date	Packet Count
November 2008	1195
December 2008	1722
January 2009	2962
February 2009	163213
March 2009	2142

Table 5: Table of ICMP Type 11 Packet Count by month

A common technique used in DDoS is to spoof an IP range and then use this range to attack a server. This of course, assuming sufficient load, causes the server to incidentally stop responding resulting in times outs. These time outs result in the generation of ICMP Type 11 packets which are sent back to the spoofed address space. Occasionally it occurs that this address space actually belongs to a Network Telescope. Table 5 alludes to the possibility of this sort of activity occurring February of 2009. After considering tables representing packet counts for ICMP type 11 at a daily resolution it was determined that some form of DDoS could potentially have been perpetrated during 17 - 18 February 2009 as shown in Figure 4.

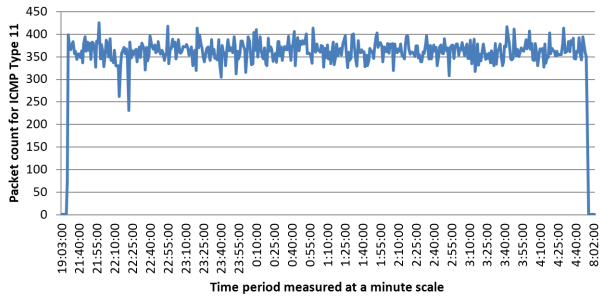

Figure 5: Plot depicting the sudden appearance of ICMP Type 11 Traffic during the 17th and 18th of February 2008.

6.4 Marking data for training purposes

Having found network incidents it is a relatively trivial process of marking the relevant details pertaining to said incident and then constructing a database of these entires. Table 6 shows a potential entry in such a table. This data can then be used to train neural classifiers. The rules used to find these incidents themselves are also of great value.

Name	Conficker
Start Time	14 October 2008
End Time	Unknown
Port	445
Severity	High
IP Range	No specific range
Comment	Exploits a 445 vulnerability on Windows based system. Is thought to be used to construct a very large commercial botnet

Table 6: Potential entry in database describing the Conficker Incident

Optimizing the condition values and rule combinations to maximize the number of incidents discovered would also define a useful Artificial Intelligence construct and is consider in section.

6.5 Issues with Manual Analysis

The process of identifying events and devising heuristics is a time consuming process and lacks completeness in the analysis. To carefully consider 65536 ports is no small feat and this is compounded by the fact that an analysis of network traffic that relies on ports alone is very limited in what can be concluded. As has been identified in this section, the time interval at which events are considered greatly affects the outcome of the analysis, considering very fine time scales is particularly costly in terms of processing and time required for analysis. Further, this sort of analysis relies on the fact that incidents are well documented and this information is available, which is often not the case.

7. MODELING

Mathematical modeling of Network Incidents is important in order to give researchers and analysts an understanding of the the nature of the threat that is posed by large scale incidents. From a malware perspective, by creating and understanding models of viral activity it is possible for researcher to predict the behaviour of new viruses. Further these models can be combined with other theory such as queuing theory to predict network failure at choke points [27]. In this section a number of modeling techniques will be discussed with relation to their importance in a network incident detection system.

7.1 Linear Growth

Consider a population that grows at a constant rate. That is $\frac{dy}{dx} = a$. Solving this simple equation leads to a solution of the form $y = ax + b$.

Linear Growth is not a growth type that is typically observed in nature except for shortish periods of time. That is is to say a population may growth linearly for x times units until the natural resources are depleted. In a similar sense the researchers do not expect to observe classical linear behaviour in Network Incidents. However, it is possible to consider $y = a$ as a naive model of normality. This naive approach is justified by the fact that most ports receive almost no traffic at all, this is illustrated further if it is considered that most of the traffic observed by rutelescope is composed for the most part of a few ports, as was illustrated in Figure 3. From some limited observation made during the manual analysis, ports that do

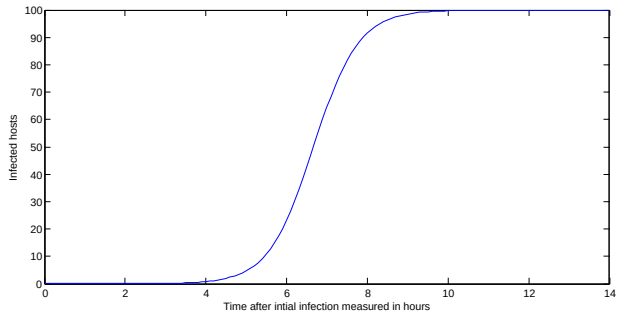

Figure 6: Plot of Logistic Growth with M = 100 and T = 12.

receive traffic tend to have counts that are randomly situated around a straight line of form $y = c$. This makes a case for linear growth with no gradient as a naive pattern for normality. Of course perturbations are expected from this sort of growth and thus systems that attempt to detect normality need to be able to adjust for this.

7.2 Geometric Growth

Geometric growth is defined as a population that grows proportionally to the current population size. Mathematically this can be expressed as $\frac{\mathrm{dy}}{\mathrm{dx}} = ay$. Solving this differential equation yields $y = Ce^{kt}$. Geometric growth could be expected in the case of some viral outbreaks. That is as, as more machine become infected with said virus, so more scans for the vulnerability are made and thus exponentially more of these packets are received by a network telescope.

7.3 Logistic Growth

Logistic Growth is a popular model for describing the growth of a population that is unable to grow indefinitely in size. This is due to the limitations imposed by the environments, such as food and shelter. Logistic Growth is a popular choice for modeling the growth of population sizes for organisms. Logistic growth has been shown to accurately model Code Red Worm growth [27].

The Logistic growth equation can be derived from considering that the rate of growth of a population is proportional to $\frac{\mathrm{dy}}{\mathrm{dx}} = ay - by^2$. Solving this equation leads to the equation $y = \frac{My_0}{y_0+(M-y_0)e^{-at}}$. After further manipulation, the usual form of Logistic growth emerges as $y = \frac{M}{1+e^{-k(t-T)}}$. Figure 6 shows normal logistic growth.

7.4 SIR

The SIR model is used for cases of infectious diseases spreading in a closed population. Individuals in these groups are broken up into three separate groups. Namely Susceptible (S), Infected (I) and Removed (R). Figure 7 shows the transitions of individuals from one group to another. Arguably in the case of computer virus modeling some of the machines that enter the removed population actually re-enter the susceptible population. To model this the SIRS model could be considered, however for simplicity sake the SIR model shall be considered. The following equations model the interactions for the SIR model.

$\frac{\mathrm{ds}}{\mathrm{dt}} = -asi.$

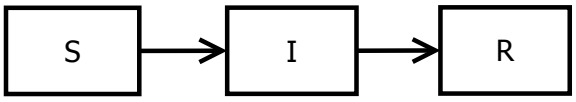

Figure 7: Diagram showing the interactions within the populations of the SIR model

$\frac{\mathrm{dy}}{\mathrm{dx}} = asi - by.$

$\frac{\mathrm{dy}}{\mathrm{dx}} = by.$

Figure 8 plots the solutions to this equation with a = 0.001 and b = 0.1. These graphs show how the various populations grow with time. The useful graph for indecent detection is that of the growth in the infected population. This because as the population grows, it should be possible to observe a change in the composition of the network traffic received by a network telescope.

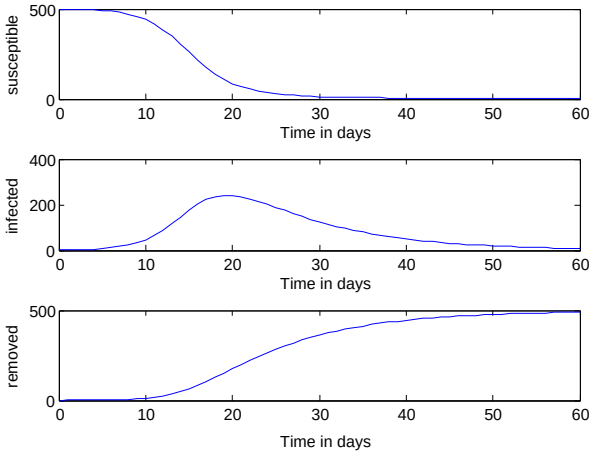

Figure 8: Plots of the solutions to the system of differential equations.

7.5 Randomly Perturbing the models

In reality it is unexpected that real growth will follow these sorts of models exactly. That is if the researchers look for growth that is exactly logistic, they are unlikely to find it. A possible solution to this problem is to randomly perturb the data sets a certain amount. Consider Figure 9 which plots both perfect logistic growth and logistic growth perturbed by up to 25%. Perturbing the models allows for the design of more robust detection systems.

7.6 Automated Generation

Using some the existing mathematical models for growth as discussed, it is a relatively trivial process of taking these models and producing simulated network captures. That is dividing a time interval and then creating packets with some properties, such as port,packet length and source IP.

8. AUTOMATED IDENTIFICATION USING ARTIFICIAL INTELLIGENCE

In the previous sections having considered techniques for the generation of training data, the paper nows con-

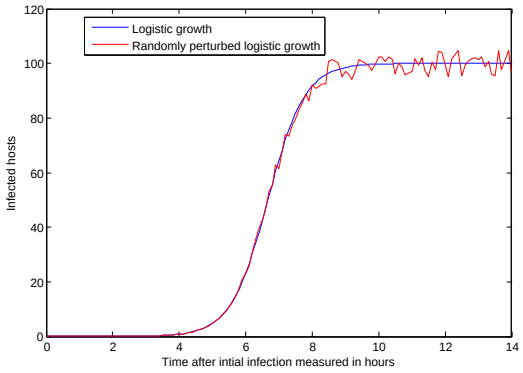

Figure 9: Plot of Logistic Growth together with a plot of the same data perturbed randomly by up to 5%.

siders ways in which these techniques can be used to develop automated systems for network incident detection. From the automated generation techniques an ANN is devised using a Feed-Forward Network architecture for growth type identification.

8.1 Classic Perceptron

A perceptron models the functioning of the neurons in the human brain. It uses a set of weights which it adjusts to match a set of inputs and bias to a set of outputs through repeatedly evaluating the input with the weights and the considering some limiting function, such as hardlim or arcsin, which then determines the output. If the correct output is obtained then the weights remain the same. Otherwise the weights are adjusted according to the error made. A more comprehensive explanation of these structures can be found considering [29, 5].

8.2 Feed Forward Neural Network

A Feed Forward Network consists of a number of neural layers. With each layer being connected to the previous layer, with potentially different weighting existing between multiple layers. During the learning phase, data is processed by one layer is then fed to the next layer as so on. Feed Forward Neural Networks make use of supervised learning. That is the neural layers that that correctly match the input and outputs patterns will produce a larger output than the other layers. This causes the weights for the neural layers that correctly identified to be increase slightly while those that failed are slightly decreased. Should the same pattern appear the neural layers that correctly identified said pattern will have even higher outputs and those that failed will have even lower outputs. Ideally the neural network will eventually learn these pattern-layer combinations [1].

8.3 Pattern Recognition

Pattern matching using Neural Networks involves two phases, namely feature extraction and classification. This is illustrated in the following example. Using the humble perceptron chained with other perceptrons to form a feed forward network is it possible to identify simple patterns through a binary representation. This is illustrated by considering a toy problem of identifying simple shapes. This idea is then extended to the identification of simple shapes to more complex patterns as those previously discussed.

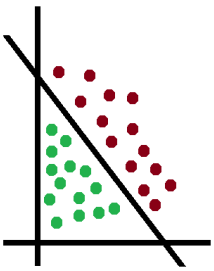

Figure 10: Figure illustrating a case of a linearly separable problem. Data belonging to different patterns is represented with a different colour.

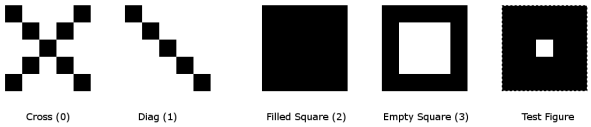

Figure 12: The images used for the simple shape identification. The number in brackets defines what each shape was assigned as an out value.

8.3.1 *Network Design and Linear separability*

The perceptron is one of the fundamental units of Artificial Intelligence. It provides a simplistic and quickly trainable means of identifying patterns which are linearly separable. This concept of linear separability can be explained by considering Figure 10. The idea is that in order to use a perceptron to classify patterns, the two patterns need to be divided into two groups by a line. This may not always be possible. The solution to this is to use multiple neural nets to define boundaries for pattern identification. The researchers decided to use three neural nets with different limiting functions as shown in Figure 11. That is the output of one perceptron with an arcsign limiter is fed as the input to another arsign limited perceptron. Finally the output from the second perceptron is fed to a perceptron with a purelin limiting function. This allows for regions for patterns to be defined, instead of requiring the patterns to be linearly separable. This describes a feed-forward neural network architecture.

8.3.2 *Binary Representation*

A number of simple shapes were chosen for binary representation as shown in Figure 12. These images were created and saved with as greyscale images. Loading these images into Matlab after some scaling yields a binary representation of the image. For example, the cross is represented in binary form as is shown in the following matrix.

$$binaryrep(cross) = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

These binary representation matrices are reduced to vectors containing all of the elements of the matrix. For example, the partial vector representation of the cross follows.

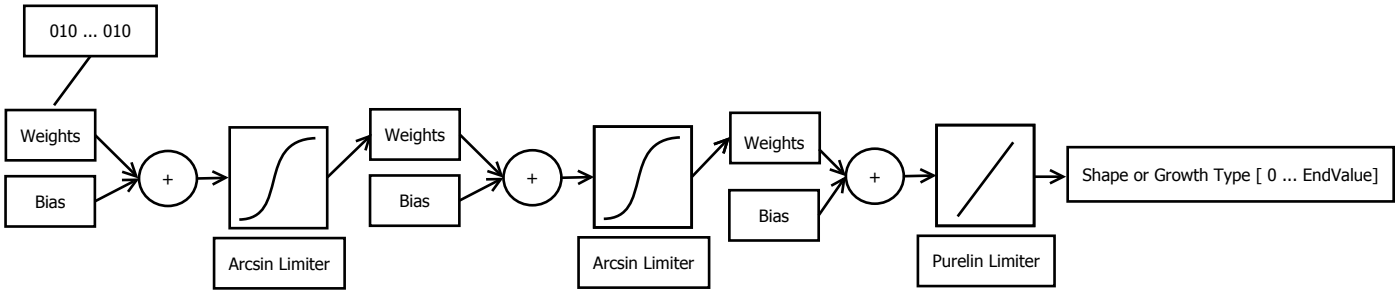

Figure 11: The Feed-Forward Network structure used for shape identification.

$$vectorrep(cross) = \begin{bmatrix} 0 & 1 & 1 & ... & 0 \end{bmatrix}$$

These vectors form the input to the feed-forward neural network. With the output being an integer assigned to each shape. After training, using the binary vector representation of the shapes as data, it is possible to detect these shapes. This was illustrated by simulation using the same binary matrix used for training that represented the cross shape. This returned a value of -0.000034, which is extremely close to the assigned value of 0. Of course as the acceptable error value was set to 0.005, a value of exactly 0 is not expected. Further it can be shown that this system allows for slight corruption of the image, a desirable property, as shown by classification of Figure. Constructing the binary vector for this image and then simulating it against the trained neural net yields a value of 2.6, that is slightly favouring the filled in square than the empty square. This provides a valuable property for growth identification as what is being classified may by slighted "corrupted". This classification is possible as long as suitable training data with suitable outputs are chosen and the data isn't too severely corrupted.

8.3.3 Extension to growth identification

Primarily, the researchers are concerned with any traffic that deviates from normality. Thus being able to define various types of growth is useful. However, perhaps slightly confusing one growth type for another, when the data shown consists of a combination of many different models, is acceptable, however it is not acceptable to confuse normality with irregularity. Data was generated using models previously discussed, with this data scaled according to the maximum value and shifted up by the minimum value in the data. This reduces misidentification due to one data set being scaled or shifted differently to the model used for training. Using the data generated an algorithm was devised to represent this as a binary matrix.

```
function Y = binaryreps(func,varargin)

    inc = 0.1;
    x = 0:inc:10;

    y = func(x,varargin{:});
    y = roundsp(10*y./max(y));

    binrep = [];
    i = 9.9 ;


    while i >= 0

        temp = zeros(1,100);
        f = find(y == i);
        if(~isempty(f))
            temp(f) = 1;
        end
        binrep = [binrep; temp];
        i = roundsp(i - 0.1);
    end
end
```

This produces binary matrices for the growth curves at a resolution 0.1 units due to the nature of the 100x100 matrix produced. Further precision is possible, but was deemed unnecessary at this point as 0.1 increments provided sufficient accuracy for identification. Figure 13 gives a visual representation of binary representation of Geometric growth.

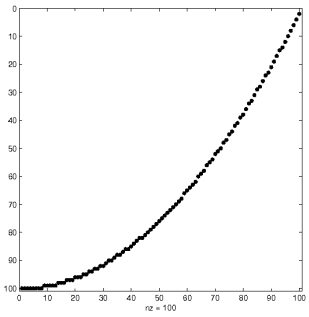

Figure 13: Plot of the sparsity matrix for the binary representation of Geometric Growth.

8.4 Application

Using this sort of technique a system could be implemented that would take in set of counts by a certain criteria. The system would then flag behaviour that is deviates significantly from the modeled normality. This is attempted by feeding this data to the trained neural network for growth identification, which results in a classification.

9. FUTURE WORK

The authors indent to extend this research into Network Incident detection by considering some of the following ideas.

- Constructing incident listings in both an automated and manual sense. One potential automated technique is to build a web-crawler that collects data from reputable sources such the US CERT, The Internet Storm Center and SANS. The incident shall

then be classified according to the anomalous activity that defines it. For example, Conficker could be defined as rapid growth in port 445 affecting Windows machines.

- Optimization of heuristic rules through Genetic Algorithms for effective incident discovery. This sort of research would consider ways to artificially breed rules for incident discovery. This is in order to maximize the number of incidents detected while simultaneously minimizing the number of false-negatives. This sort of work is reminiscent of the work done by Nottingham and Irwin in breeding rules for fast packet classification [24].

- Aggregating data collected by multiple Network Telescope nodes to allow for more distributed monitoring.

- Sliding averages, these are values that consider the data for a certain period of time and change appropriately according to the context.

- Considering more advanced statistical techniques such as ANOVA and regression analysis as good security metrics for incident identification.

- Make use ODE curve fitting and general curve fitting on traffic datasets in an attempt to recognize similar models that are significantly different from normality curves.

10. CONCLUSION

This paper has considered approaches in manual and automated identification of anomalous activity. The manual techniques involved constructing rules considering a number of measures of normality. These rules were then used to find cases where (relative) normality had been violated; these cases were then justified by considering particulars of the incidents, such as port and date, within the context of well known security sites. A number of models for Network Incident growth were considered. A simple of Feed-Forward Neural Network was considered for shape identification. This idea was extended to growth identification using the models considered previously.

11. REFERENCES

[1] [Online] Available : `http://www.let.uu.nl/uilots/lab/resources/praat/Feedforward_neural_networks.html`. [Last Accessed : 28 May 2010].

[2] Eastern Africa Submarine Cable System (EASSy). [Online]. Available: `http://www.eassy.org/`. [Accessed: Apr 20, 2010].

[3] The Cooperative Association for Internet Data Analysis. [Online]. Available: `http://www.caida.org/home/`. [Accessed: 21 April,2009].

[4] Vulnerability Summary for CVE-2008-1701. [Online]. Available : `http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2008-1701`, Accessed.

[5] An introduction to neural networks. [Online] Available : `http://www.cs.stir.ac.uk/~lss/NNIntro/InvSlides.html`. [Last Accessed : 28 May 2010], April 2003.

[6] Julio Barbancho, Carlos León, F. J. Molina, and Antonio Barbancho. Using artificial intelligence in routing schemes for wireless networks. *Comput. Commun.*, 30(14-15):2802–2811, 2007.

[7] R. Barnett and B. Irwin. An analysis of logical network distance on observed packet counts for network telescope data. In *SATNAC 2009*, 2009.

[8] Nick Pilkington Barry Irwin and Blake Friedman. A geopolitical analysis of long term internet network telescope traffic. In *SATNAC*, 2007.

[9] Dhruba K. Bhattacharyya and Syamanta M. Hazarika. *Networks, Data Mining, And Artificial Intelligence: Trends And Future Directions*. Narosa Pub House, 2006.

[10] CAIDA. Caida's chicago a passive network montior, application bits/second. [Online] : `http://www.caida.org/data/realtime/passive/?monitor=equinix-chicago-dirA&row=timescales&col=sources&sources=app&graphs_sing=ts&counters_sing=bits&timescales=24&timescales=168&timescales=672&timescales=17520`. [Accessed: Apr 20, 2010], May 2010.

[11] CAIDA. Caida's chicago a passive network montior, applicationpackets/second. [Online] : `http://www.caida.org/data/realtime/passive/?monitor=equinix-chicago-dirA&row=timescales&col=sources&sources=app&graphs_sing=ts&counters_sing=packets&timescales=24&timescales=168&timescales=672&timescales=17520`. [Accessed: Apr 20, 2010], May 2010.

[12] CAIDA. Passive Data Collection: UCSD Network Telescope. [Online]. Available: `http://www.caida.org/data/passive/network_telescope.xml`. [Accessed: April 19, 2010], January 2010.

[13] Kimberly Claffy, Young Hyun, Ken Keys, Marina Fomenkov, and Dmitri Krioukov. Internet Mapping: From Art to Science. In *CATCH '09: Proceedings of the 2009 Cybersecurity Applications & Technology Conference for Homeland Security*, pages 205–211, Washington, DC, USA, 2009. IEEE Computer Society.

[14] Main One Cable Company. Main One Cable. 2009. [Online]. Available: `http://www.mainonecable.com/`, [Accessed: Apr. 20, 2010].

[15] G. Voelker St. Savage D. Moore, C. Shannon. Network Telescopes: Technical Report. [Online]. Available: `http://www.caida.org/publications/papers/2004/tr-2004-04/tr-2004-04.pdf`. [Accessed: 10 April, 2010], 2004.

[16] Feyza Gürbüz, Lale Özbakir, and Hüseyin Yapici. Classification rule discovery for the aviation incidents resulted in fatality. *Know.-Based Syst.*, 22(8):622–632, 2009.

[17] Uli Harder, Matt W. Johnson, Jeremy T. Bradley, and William J. Knottenbelt. Observing Internet Worm and Virus Attacks with a Small Network Telescope. *Electron. Notes Theor. Comput. Sci.*, 151(3):47–59, 2006.

[18] Barry Irwin and Nick Pilkington. High Level Internet Scale Traffic Visualization Using Hilbert Curve Mapping. In *VizSEC*, pages 147–158, 2007.

[19] Qiming Liu, Rui Xu, and Shaojie Wang. Modelling and Analysis of an SIRS Model for Worm Propagation. In *CIS '09: Proceedings of the 2009*

International Conference on Computational Intelligence and Security, pages 361–365, Washington, DC, USA, 2009. IEEE Computer Society.

[20] David Marchette. A statistical method for profiling network traffic. In *Proceedings of the Workshop on Intrusion Detection and Network Monitoring*, pages 119–128, Berkeley, CA, USA, 1999. USENIX Association.

[21] John McCarthy. What is artifical intelligence ? [Online]. Available : `http://www-formal.stanford.edu/jmc/whatisai/node1.html`, November 2007.

[22] David Moore, Vern Paxson, Stefan Savage, Colleen Shannon, Stuart Staniford, and Nicholas Weaver. Inside the Slammer Worm. *IEEE Security and Privacy*, 1(4):33–39, 2003.

[23] David Moore, Colleen Shannon, and k claffy. Code-Red: a case study on the spread and victims of an internet worm. In *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 273–284, New York, NY, USA, 2002. ACM.

[24] Alastair Nottingham and Barry Irwin. GPU packet classification using OpenCL: a consideration of viable classification methods. In Barry Dwolatzky, Jason Cohen, and Scott Hazelhurst, editors, *SAICSIT Conf.*, ACM International Conference Proceeding Series, pages 160–169. ACM, 2009.

[25] Shirley Payne. A Guide to Security Metrics, SANS Institute. pages 1–2, 2002.

[26] Giuseppe Serazzi and Stefano Zanero. Computer virus propagation models. In *In Tutorials of the 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems (MASCOTS03*. Springer-Verlag, 2003.

[27] Giuseppe Serazzi and Stefano Zanero. Computer Virus Propagation Models. In *In Tutorials of the 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems*. Springer-Verlag, 2003.

[28] Colleen Shannon and David Moore. The Spread of the Witty Worm. *IEEE Security and Privacy*, 2(4):46–50, 2004.

[29] Christos Stergiou and Dimitrios Siganos. Neural networks. [Online] Available : `http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html` [Accessed : 26 May 2010], 1996.

[30] Symantec. W32.rinbot.a. [Online]. Available `http://www.symantec.com/security_response/writeup.jsp?docid=2007-021615-1555-99`. [Accessed : 17 April,2010], February 2007.

[31] Jean-Pierre van Riel and Barry Irwin. InetVis, a visual tool for network telescope traffic analysis. In *Afrigraph*, pages 85–89, 2006.

[32] US-CERT. Conficker worm targets microsoft windows systems. [Online] : `http://www.us-cert.gov/cas/techalerts/TA09-088A.html`, April 2009.

[33] Jean-Pierre van Riel and Barry Irwin. Identifying and Investigating Intrusive Scanning Patterns by Visualizing Network Telescope Traffic in a 3-D Scatter-plot. In *ISSA*, pages 1–12, 2006.

[34] Rhiannon Weaver. A probabilistic population study of the conficker-c botnet. In *PAM*, pages 181–190, 2010.