

2008

Identifying DOS Attacks Using Data Pattern Analysis

Mohammed Salem

Curtin University of Technology

Helen Armstrong

Curtin University of Technology

DOI: [10.4225/75/57b56484b8771](https://doi.org/10.4225/75/57b56484b8771)

Originally published in the Proceedings of the 6th Australian Information Security Management Conference, Edith Cowan University, Perth, Western Australia, 1st to 3rd December 2006.

This Conference Proceeding is posted at Research Online.

<http://ro.ecu.edu.au/ism/55>

Identifying DOS Attacks Using Data Pattern Analysis

Mohammed Salem, Helen Armstrong
School of Information Systems
Curtin University of Technology
GPO Box U1987, Perth, Western Australia 6845
mohammed.salem@student.curtin.edu.au
h.armstrong@curtin.edu.au

Abstract

During a denial of service attack, it is difficult for a firewall to differentiate legitimate packets from rogue packets, particularly in large networks carrying substantial levels of traffic. Large networks commonly use network intrusion detection systems to identify such attacks, however new viruses and worms can escape detection until their signatures are known and classified as an attack. Commonly used IDS are rule based and static, and produce a high number of false positive alerts. The aim of this research was to determine if it is possible for a firewall to analyse its own traffic patterns to identify attempted denial of service. Statistical analyses of firewall logs for a large network were carried out and a baseline determined. Estimated traffic levels were projected using linear regression and Holt-Winter methods for comparison with the baseline. The research proposes a Neural Network model for forecasting rejected traffic falling outside the projected level for the network under study that could indicate an attack. The results of the research were positive with variance from the projected rejected packet levels successfully indicating an attack in the test network.

Keywords

Firewall, denial of service, baseline, intrusion detection, network security

INTRODUCTION

The expansion of networked systems in the past decade has led these networks into a security dilemma. On the one hand these networks try to protect networked resources from unauthorised external access and on the other keeping the network available on a 24x7 basis. While these goals seem to be reasonable, this is difficult to achieve, requiring substantial amount of design to produce a secure model for any network. Achieving such goals utilising the Internet is an even more complex and tedious process. Network administrators spend substantial time and effort trying to secure their networks from known and unknown threats resulting from the open nature of the Internet.

The common idea behind any firewall is to allow legitimate entities to access shared networked resources based on predefined policies. The problem is that most firewalls do not know how to handle traffic unless it is predefined within the policy. In other words they lack the ability to learn from past experience and thus rely on human intervention. In most cases firewall administrators harden their firewalls by closing every port then setting rules to open certain ports as needed by the network users and applications.

Hardening firewalls is effective in blocking illegal access to the network but cannot stop other sorts of network attacks by external and internal entities to the open ports. This is because they are considered legitimate activities by the firewall. In some cases even the most hardened firewalls can fall into an attacker's trap by responding to the attack packets rather than dropping them and continuing to process normal network activities. In the context of this paper this is a Denial of Service attack (DoS).

In a practical networking environment, DoS can be defined in different ways depending upon the target (i.e. specific application or service):

1. Attacks against application servers: for example web servers, causing servers to be unavailable for public use.
2. Flooding network gateways and firewalls: for example flooding with thousands or millions of Transmission Control Protocol / Internet Protocol Suite (TCP/IP) packets causing either slowness in network activities or the network to become completely unusable till all packets are dropped from the network.
3. Attacking mail gateways: for example sending mail ware Simple Mail Transfer Protocol (SMTP) packets that can block email systems for a long period of time before they can be cleared.

This paper deals with DoS attacks of the second type, flooding the firewall with thousands of packets making it either unstable or unusable. Such a DoS attack will flood the network with randomly generated packets, where the firewall will respond by rejecting these packets if it has been configured properly, however, it will not serve the legitimate network users as it is busy rejecting these randomly arriving packets.

Firewalls are good for filtering traffic, however, they need to be continually monitored and analysed in order to be more efficient against attacks (especially DoS attacks). Early detection of an attack against the network aids fast elimination and more effective network protection. Firewall logs can collect, store and analyse firewall activity data to subsequently provide a proactive mechanism to defend the network from future attacks. The implementation of this can be complex where traffic levels on large networks are high, with some networks logging millions of packets every hour. It is not possible to eliminate DoS attacks completely, and protection against these attacks is expensive and time consuming. The most effective DoS defence approaches detect and block attack traffic close to the source, and this task is complex as it is difficult to discriminate between legitimate and malicious traffic (Peng et al. 2007).

The aim of this paper is to describe research carried out using TCP/IP traffic data from a live network to produce an anomaly-based self-learning procedure for identification of DoS attacks. The general approach is to statistically develop a forecast of expected network traffic levels based upon a baseline derived from normal traffic on that network. This forecast is then compared to real-time activities to indicate possible DoS attacks.

FIREWALLS AND IDS

The primary role of any firewall is to protect the network based on pre-defined rules designed as per the computer network local security best practice policy (Smith & Bhattacharya 1999). The configuration of firewall rules is very important to differentiate between normal network activities and attacks. Some networks include additional protection systems on top of the firewall such as Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS).

Activities logged by the firewall are very rich in data and can be analysed later to obtain further information about existing and future attacks. They can also be used to evaluate firewall security, performance and management levels (Noureddien & Osman 2000). The amount of security that can be provided by firewalls is limited when an attacker initiates an internal attack as this falls within the Local Area Network (LAN) boundaries. It is also limited in protecting against virus attacks due to the large number of software applications used within the network environment requiring special tools to be incorporated with the firewall to scan every packet.

Even though current firewall systems can be implemented on a variety of software and hardware systems, the basic idea behind firewalls remains the same. Different types of firewalls can be found on routers or even on dedicated network servers. The first type of common firewall is the packet filter firewall which works by reading the source IP address from each packet header (Stallings 2006). Packet filtering firewalls filter incoming and outgoing packets and either allow or block the traffic, and if the packet is allowed to or from the network it will be forwarded to the next hub as per the routing table information for this packet's destination. On the other hand, if the packet is blocked it will be discarded. Packet filtering firewalls are useless if an external attacker has spoofed the IP header with an internal IP address that the firewall will allow to get inside the network. In addition, packet-filtering firewalls fail to meet most of network application requirements.

The second type of firewall is the application level firewall that allows network services (e.g. Telnet, FTP, etc.) to be established and used within predefined criteria controlled by firewall policies. However, even market leading firewalls such as Check Point Firewall-1 still lack a self-learning mechanism, in other words they cannot learn from past attacks dynamically (Noureddien & Osman 2000). Firewalls are designed by humans and need human intervention in order to be kept up-to-date with latest security patches and rules configuration.

Firewalls can provide an authorized flow of traffic to inside the network and all traffic to pass to outside the network with a convenient service, direction, user and behaviour control, however, this type of protection is no longer a suitable type of protection when it comes to internal or external attacks, worms and viruses (Stallings 2006). No complete immunity can be obtained using a firewall because it is a rule-based system. The problem has to occur before it can be identified and configured as a static rule in the firewall in order to be used for future filtration and protection.

IDS are commonly used to supplement the firewall in identifying network attacks and the quicker the IDS can detect the attack the quicker an intrusion can be stopped before causing more potential damage to the network (Chang 2002). The most utilised type of IDS is misuse detection where firewall logs are compared with signatures of known attacks. Current misuse IDS models look for a matched pattern of information and activities that can be analysed and categorised as malicious behaviour by IDS to protect the network from intruders (Vigna et al. 2003). Misuse detection can efficiently detect known attacks if the signatures are known, resulting

in a fast match with attack patterns stored in the IDS. Looking for patterns might be straightforward for an IDS which compares network traffic with attack data to pick intrusion patterns (Stolfo & Lee 2000). Misuse detection systems commonly exhibit low false positive rates, as an alarm is only raised if the log data possesses an attack signature. The main disadvantage of misuse detection IDS is that they cannot detect novel attacks that leave unknown signatures (Ghosh & Schwartzbard 1999). This means that a new type of DoS attack will go undetected by the misuse detection IDS until its signature is known and recorded. There is, therefore, a gap between the time the new attack appears and the time it is recognised as an attack by the coding of its signature in the IDS. This is one of the main limitations in the ability of misuse detection IDS to identify new attacking techniques without being attacked first.

Ghosh and Schwartzbard (1999) categorise misuse detection approaches into expert systems, model-based reasoning, state transition analysis and keystroke dynamics monitoring. Although the misuse approach is a static process not able to identify and recognise novel attack mechanisms the majority of research into intrusion detection (and the commercial products available) are misuse based, requiring signature matching.

The alternate approach to intrusion detection is anomaly based, which do not scan for identifiable patterns of attack, but performs analyses on past behaviour and build models of acceptable behaviour. This approach requires statistical analyses of previous network activity to determine a 'normal' range for comparison against current traffic. IDS can also make use of decision rules to decide, and neural networks (NN) and data mining theories have been used to enhance the performance of IDS systems against network attacks by providing an intelligent mechanism to learn more from past audit data collected from IDS (Lee & Heinbuch 2001). Although extensive research has been carried out in misuse detection, research into pre-emptive approaches is less prolific. Categories of anomaly based research identified by Ghosh and Schwartzbard (1999) include rules for normal behaviour, statistical models of user or program profiles, and machine learning to recognise anomalous user or program behaviour. More recent research includes areas such as privileges flows and system calls (Cho & Park 2003, Kruegel et al. 2003), specifically designed architectures (Bolzoni et al. 2006, Zanero 2008), web application profiles and protocols (Estevez-Tapiador et al. 2003, Kruegel & Vigna 2003, Robertson et al. 2006), data mining (Sequeira & Zaki 2002) and neural networks (Mukkamala et al. 2002, Ordóñez-Cardenas & Romero-Troncoso 2008, Zhang, Jiang & Kamel 2005). Anomaly detection approaches require the establishment of a baseline derived from the normal behaviour of the network being monitored, which is then used as the basis for identifying deviations in order to detect potential DoS attacks.

This paper focuses on network based IDS (NIDS) which dedicate a specific network device to act as a network activity monitor and sensor, and when a malicious activity is detected it alerts the network and blocks the intruders from doing further damage to the network participating hosts (Allan 2003). In spite of enhancements to expedite their ability to detect, alert and block intruders, traditional NIDS still suffer from re-configuration problems that make generalization and application to all network configurations difficult. This is due to protocol and configuration language dependencies while re-configuring classification methods in different NOS software and hardware environments (e.g. Cisco, UNIX and Windows) (Iheagwara & Blyth 2002, Ollmann 2003).

EXPERIMENT IN SELF-LEARNING FIREWALL

A common problem with IDS is the rate of inaccuracy and the high number of false positive alerts leading to management overheads without a valid source of an attack toward the network. This is due to the IDS using multiple algorithms when classifying an attack (Easley & Stiennon 2002). Chen et al. (2004) explain the limitations of three types of IDS in the handling DoS/DDoS attacks:

1. Congestion based detection algorithm: can only be applied while the network is congested and can cause a lot of false positive alerts raised by normal network traffic at the time of the attack.
2. Anomaly based detection algorithm: can only be applied to TCP SYN packets and need to have a TCP protocol and sub-protocols (e.g. ICMP and UDP) thresholds to be known before it can declare a DoS attacks. This method can also generate false positives if the normal network traffic reaches the pre-defined thresholds.
3. Source based detection algorithm: can be used surely if the source attacker utilise a spoofed IP address, however, if the system cannot distinguish between original trusted IP address and a spoofed IP it will generate a false alert.

If a firewall were to analyse its own traffic logs and forecast traffic patterns for 24-48 hours it may be able to detect DoS as they occur, thus limiting the above disadvantages of current IDS. Such an approach is possible using data statistical modelling to produce a firewall pattern baseline from firewall logs without IDS coexistence in the network. Expected firewall traffic is then forecast based upon this baseline and actual traffic compared to the estimate. The expectation is that changes within the firewall traffic patterns can be interpreted as attacks as long as a minimum margin can be defined to eliminate false alerts.

This research analysed a set of cross-sectional past data logs for a large networked environment comprised of 15000+ systems incorporating hardware and software from different software vendors (see Figure 1 for the high level network configuration). This network was protected with a Check-Point Firewall-1, running on clustered UNIX platforms that serve multiple sites within the organisation. Both servers and hosts sitting behind the firewall represent a wide range of networked platforms (Microsoft Windows, Novell, UNIX and Linux). As with any large network the firewall configuration had been hardened to eliminate any possibility of network attacks to gain access behind the firewall, and hence the Check-Point Firewall-1 only logged activities of open ports as required by the applications used within the environment. In addition logging included the recording of dropped packets between LAN and WAN, these being classified as rejected packets.

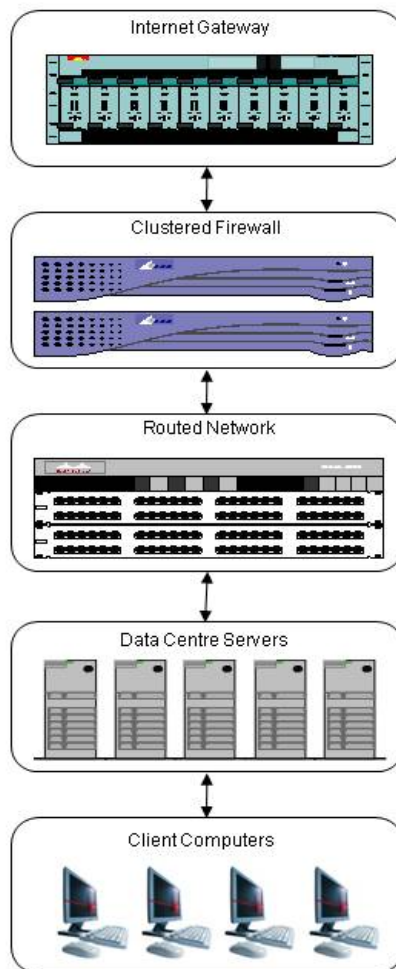


Figure 1 – Infrastructure of large computer network used in this research

DATA COLLECTION AND ANALYSIS

The data was collected in two stages. In stage one a number of cross-sectional statistical data samples were collected from existing network firewall logs representing live production network protocol activities on a daily basis for a period of three months. This data actually consisted of millions of readings per day and a Java routine was used to pre-prepare the data in 24 one-hour formats. This three month data set was used to develop the baseline model consisting of a total of 2184 observations (91 days x 24 observations per day) collected in the first stage. After developing the baseline model, another round of data was collected from logs from the same firewall network logs within a change controlled environment (i.e. no changes to the network firewall logging process occurred) over a further two months. This data was also prepared in 24 hour observations providing a total of 1416 observations (59 days x 24 observations per day). This second set of data was then compared with the baseline model forecasts for quantitative analysis.

The collected data contained a variety of protocol activities such as TCP, IP, ICMP and UDP. These protocols represent the major parts of the standard TCP/IP suite of protocols required for Internet connections and communications to occur, therefore, the sampling of the data was designed based on a group of these protocols activities. The protocols were classified based on traffic direction (i.e. inbound traffic or outbound traffic) and packet status (i.e. packet been accepted or rejected).

As the type of collected data was numeric, a quantitative approach for data analysis was deemed to be the most suitable to study the casual relationship between dependent factors within the designed baseline model. Two quantitative statistical forecasting methods were used to produce the baseline model: Holt-Winter Multiplicative Smoothing Method and Linear Multiple Regression Method. The main reason for selecting these two methods was that both methods cover for trend and seasonality components at the same time. In addition Microsoft Excel was used to triangulate with MINITAB statistical software results to assure integrity for both forecasting methods.

As the research explored the relationship between log patterns and DoS attack against network availability, the measurement unit selected was the total number of rejected packets per hour. The Holt-Winter method of forecasting uses an exponential smoothing forecasting technique that facilitates discovery of the underlying pattern within the time series data while eliminating the effects of any trend and seasonal components within the time series data.

The linear regression model was selected as a forecasting tool to estimate the total number of rejected packets formulated as an equation of selected protocol packet activities and status (inbound or outbound through the firewall). The aim of this statistical model was to see how closely the rejected packets matched the results found by the Holt-Winter forecasts in order to determine the reliability of results from both methods.

RESEARCH FINDINGS

Due to the need to classify the packet status as either accepted or rejected, only inbound and outbound statistics were suitable for such a prediction of rejected packets over time. The regression model supports the above premise as it only used the inbound traffic protocols packets to predict the total number of rejected packets. Figures 2A and 2B illustrate a common pattern appearing in the forecast residuals graphs for the Holt-Winter and the Linear Multiple Regression approaches. Figure 2A displays the actual inbound TCP traffic on the network under study over a two month period. Figure 2B plots the estimated rejected packet residuals using the linear regression and Holt-Winter forecasting methods. The light colour plots are linear regression residuals and the dark plots show the Holt-Winter residuals. The time periods in these figures are particularly significant, as discussed below.

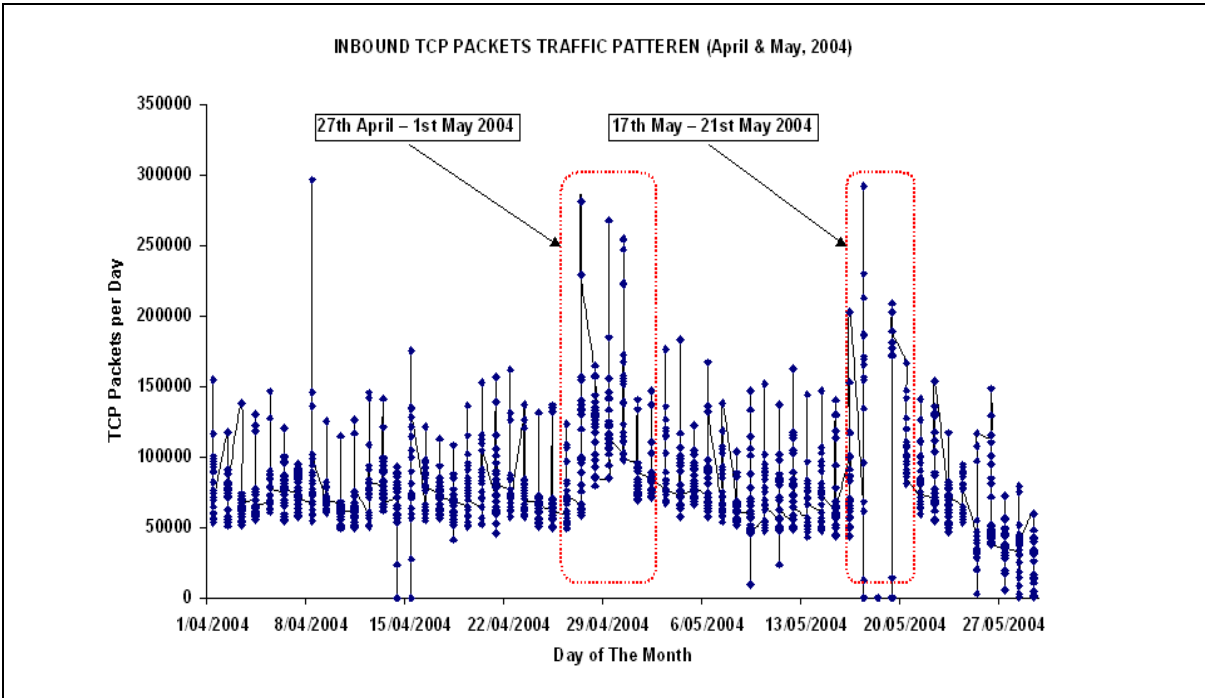


Figure 2A — Rejected Packets Residuals Using Regression and Holt-Winter Forecasting Models.

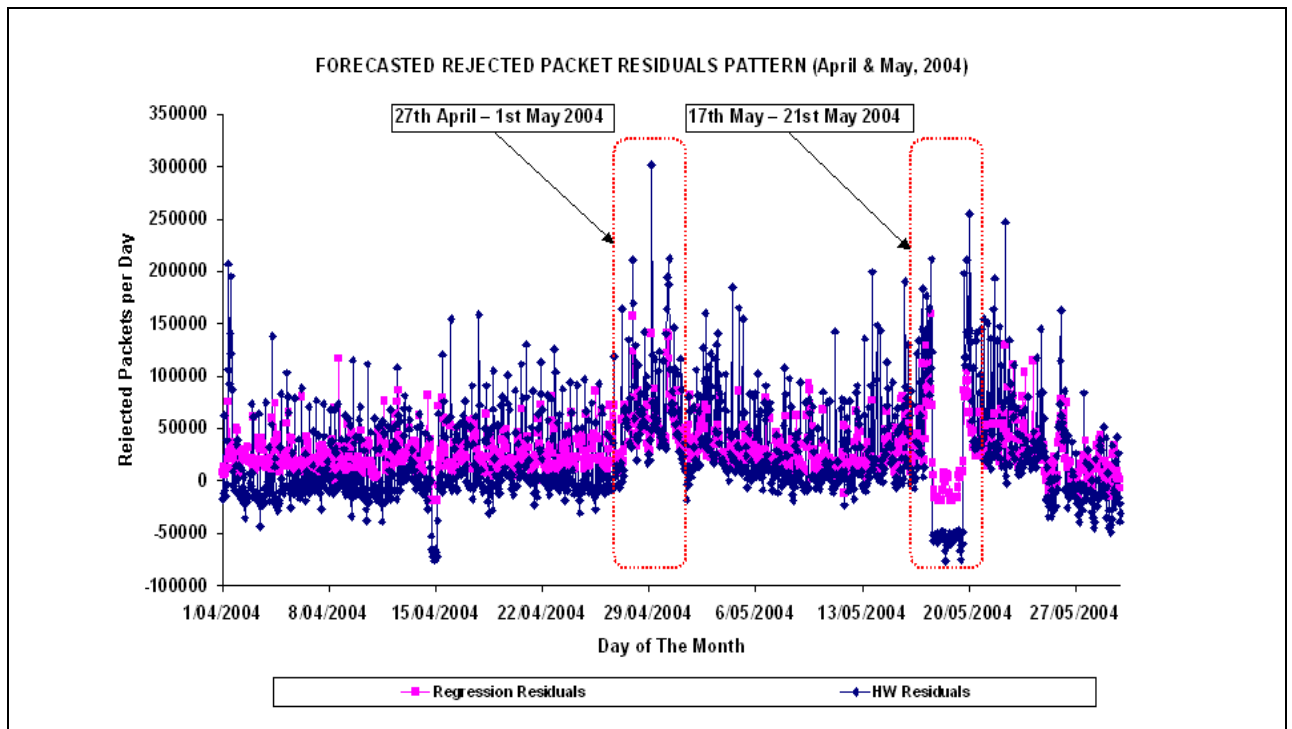


Figure 2B — Inbound TCP Packets Traffic Pattern within the Baseline Data Set.

Residuals Anomalies Analysis

Figure 2B shows two spikes in the rejected packets residual pattern (both marked by a dashed line box). The large spike in the amount of packets occurred during Period 1 (i.e. 27th April to 1st May 2004) and investigating the firewall audit logs showed no explanation for this large spike in the residuals. A mix of an increase and a decrease in the packet levels appears within Period 2 (i.e. 17th May to 21st May 2004). When investigating firewall audit logs, it was found that the loss of data was due to the introduction of a new firewall cluster node that affected the logging process causing the loss of some logging data, and this was rectified on the final day of Period 2.

After further investigating activities on the Internet over Period “1” it was found that an Internet mail-worm named “SASSER” had spread during this period and only reported to the Internet community as early as the 1st May 2004 at 14:24 GMT time (SANS 2004). However, the SASSER had overwhelmed the network firewall for nearly four days before it was discovered. The SASSER mail-worm employs a vulnerability within the Microsoft Windows 2000 operating system called the Local Security Authority Subsystem Service (LSASS) that was discovered first in 26th April, 2004. However, the link between the LSASS vulnerability and SASSER was not made until the 30th April 2004. “SASSER” uses certain ports within TCP to spread itself (ports 445, 5554 and 9996), and this corresponds to the inbound TCP packets received from the Internet over the WAN gateway (see Figure 2B).

In this research, the case study network firewall was not based on a Microsoft Windows operating system and it blocked the SASSER packets requests from passing to the LAN network as these ports are not usually open. Therefore, the only effect of SASSER was limited to flooding the network with a large number of TCP packets passing through the gateway to the firewall interfaces and subsequently slowing down the whole network. This is a typical network DoS attack launched randomly from a SASSER infected Windows system targeting any network connected to the Internet.

Identifying DoS Attack

In order to declare a DoS attack based on the information provided by the residuals obtained from both forecasting baseline models (i.e. Holt-Winter and linear regression models), a maximum accepted margin of rejected packets per hour needs to be defined. By using the real data (not forecasted data) the number of

rejected packets from both inbound and outbound packets travelling across the firewall between January 1st and March 31st, 2004 could be calculated. This helped in obtaining the average number of rejected packets from the network (without having an attack) which was then compared with the average in the period from April 1st to May 29th, 2004 when the network is under attack. The difference is the maximum number of rejected packets that can be allowed before declaring a DoS attack. In order to obtain a percentage representation of the accuracy margin the total rejected packets per hour is divided by the total number of packets travelling in and out of the firewall per hour. Tables 1 and 2 show these calculations.

	Inbound Traffic	Outbound Traffic	Rejected Traffic
Average Number of Packets per Hour	93748.41	342.12	51658.11
Total	94090.53		51658.11
Rejected to Total Traffic Percentage	54.9 %		

Table 1 – Rejected Statistics (January-March 2004)

	Inbound Traffic	Outbound Traffic	Rejected Traffic
Average Number of Packets per Hour	108924.92	366.91	89888.85
Total	109291.83		89888.85
Rejected to Total Traffic Percentage	82.2 %		

Table 2 – Rejected Statistics (April-May 2004)

Tables 1 and 2 show the following:

1. The amount of outbound traffic is very low, even negligible compared to the inbound traffic passing through the network at the same period. It appears that the inbound traffic influences the number of rejected packets more significantly than the outbound traffic.
2. There is an increase of approximately 38,231 packets in the rejected packets per hour on average (89,888.85 – 51,658.11). If the number of rejected packets per hour increases by 38,231 packets on this particular network it is highly likely that this network is under attack.

Tables 1 and 2 indicate that the average percentage of rejected packets without attack is 54.9% and the average percentage of rejected packets with DoS attack is 82.2%. The percentage difference is 27.3% (82.2% - 54.9%). As an increase of 27.3% represent a threshold above the normal level of rejected packets could represent a DoS attack launched against the network, the trigger point for this network is 27.3% and any reading above this level should indicate a possible DoS attack. Hence the anomalies within the model residuals emphasise the relationship between DoS and daily network activities. In addition the regression baseline model can be recalculated every day and then a new forecast can be generated to compare 24-hour forecasts with real time data. However as each network environment is different, the trigger line for the network under study will not necessarily be the same for other networks, and a new trigger line needs to be established for each new network.

A SIMPLE DETECTION MODEL BASED ON HISTORICAL DATA MINING

The data collected earlier from 2004 as an example of a large attack shows that a statically calculated increase of 27.3% was required to raise an alert. Therefore, finding the accurate threshold value before declaring an attack by comparing packets anomaly to normal network activity enhances the detection process. IDS and IPS can do this and provide an active response, however, it is very expensive and the majority of these products are proprietary (Thames et al. 2008). This research paper approached the problem with a solution from a network management point of view where the detection system can integrate with the firewall. This system can be tuned to use the firewall logs historical data at the beginning of the detection process and then recalculate the threshold dynamically without a predefined set of rules. The detection process is based on the change of network activities using a neural network model to forecast the level of rejected packets within the network as compared with the threshold.

Threshold Calculations

Let's assume the following:

Firewall log history period = T (this could be day, week or month without an attack presented in the data)

The average number of rejected packets per day using historical period T = Y

The average number of rejected packets per day at period T + (1 day) = D

Therefore, the baseline detection threshold $H = (D/Y) - 1$ and this must be > 0 to raise an alert

The following example shows how to calculate the threshold:

The historical data range $T = 1$ month

The average number of rejected packets during the month of January: $Y = 100000$ Rejected Packets/Day

The average number of rejected packets in the first day of February: $D = 120000$ Rejected Packets/Day

Therefore, the baseline detection threshold: $H = (120000/100000) - 1 = 0.2 = 20\%$

The baseline detection threshold represents the network's natural growth in rejected packets and anything higher than this threshold can be considered as an attack with the intention to flood the network with traffic that can lead to DoS. Therefore, comparing the rejected packets forecast obtained from Multilayer Perceptron Backpropagation (MLP) with real rejected data for a specific day can show if the difference is lower or higher than the dynamically calculated threshold. In the above example if the difference between the two is higher than 0.2 then an alert can be generated to protect the system either manually by the firewall administrator or automatically using the firewall rules itself.

Multilayer Perceptron Backpropagation (MLP) Neural Network

MLP is a type of neural network that can be constructed from an interconnected set of nodes with multiple layers as shown in Figure 3. The first layer is the input layer, the hidden layer where all inputs are transformed to the next layer based on their weights and the final layer is the output layer.

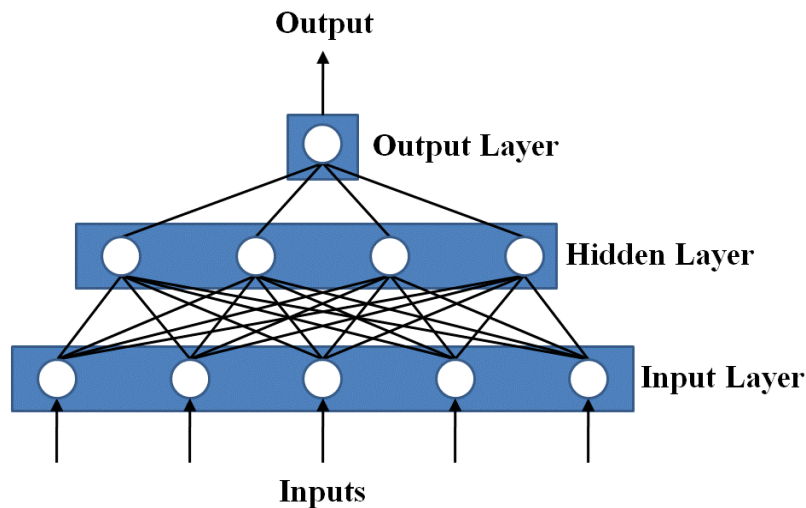


Figure 3 – Neural Network Architecture Example with One Hidden Layer (Adapted from Kennedy et al. 1997)

The MLP algorithm utilises the neural network inputs to calculate the output using a hyperbolic tangent transformation function for all hidden neurons (nodes). Therefore, the output neuron “y” is the sum of all the connection weights between the hidden neurons and the output neuron inside the network (Kennedy et al. 1997). Kennedy et al. (1997) defined the hyperbolic tangent transformation function to obtain the hidden neurons (nodes) output using the equation below:

$$\text{Output } y = \frac{(1 - e^{-net})}{(1 + e^{-net})} \quad (\text{Source: After Kennedy et al. 1997})$$

Where the (net) in the previous equation is the output node (y) as a function of input nodes (x) that can be obtained from the summation of the weights multiplied by the inputs as per the following formula:

$$y(x) = w_0 + \sum_i w_i x_i \quad (\text{Source: After Kennedy et al. 1997})$$

Where:

- w_0 is the bias of the node
- w_i is the connection weight from hidden neuron node i^{th} with input value x_i

For this research, MLP network inputs are the packets statics per a time unit (hour/day/week) for every type of packet that enters or exits the computer network firewall (e.g. TCP, IP, ICMP, UDP...etc) regardless as to whether it has an accepted or rejected status. The MLP neural network output represents the total number of rejected packets forecast for the next time period (hour/day/week).

Figure 4 shows the flow of data within MLP with recursive backpropagation processing of the weights that is used to find the optimal value of the connection weights between MLP input and output nodes (Ordonez-Cardenas & Romero-Troncoso 2008). The scaling process is a pre-processing step to prepare the inputs in order to match the hyperbolic tangent range.

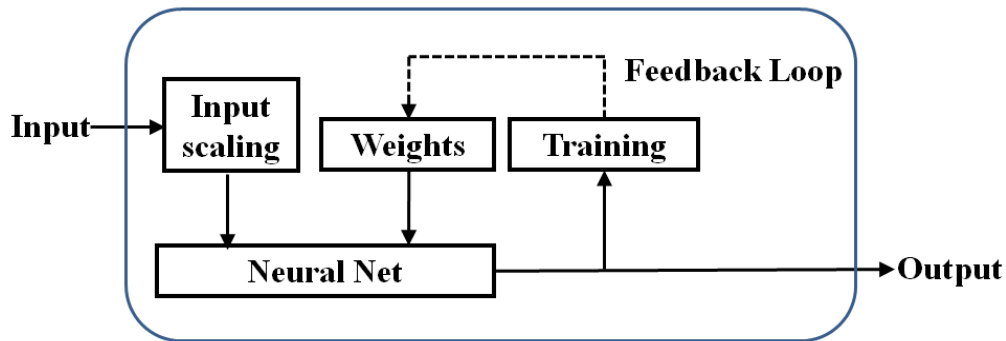


Figure 4 – MLP System Floorplan (Adapted from Ordoez-Cardenas & Romero-Troncoso 2008)

The error minimisation can be obtained by backpropagating the difference between real training output values and predicted output values of the MLP using Mean Square Error (MSE) of the overall training set with the following MSE error equation:

$$E = \sum_{i=1}^{N_{train}} \left\{ \frac{1}{2} \sum_{j=1}^J (d_{ij} - y_{ij})^2 \right\} \quad (\text{Source: After Kennedy et al. 1997})$$

Where:

- E is the total root mean square error of the network at training pattern “ i ”
- i is the index of the training pattern input value
- N_{train} is the total number of training patterns with total number of outputs “ J ”
- j is the index of the training pattern output value
- J is the training patterns outputs
- d_{ij} is the desired value of output “ j ” at training pattern “ i ”
- y_{ij} is the actual output of the training pattern

The Attack Detection Process

This research approaches the solution of the problem from a network management point of view. Instead of setting the threshold to raise an alert by configuration rules, the model will re-calculate the threshold based on the history of average rejected packets as recorded by the firewall logs. The proposed solution model is different from existing IDS/IPS in that it can change the detection rule dynamically using the following process:

- Analyse firewall logs and obtain protocol statistics
- Calculate the baseline natural growth threshold of rejected packets over the past available log history
- Forecast future levels of rejected packets
- Compare forecasted data with real data within units of hours or days.

- If the difference between real data and forecasted data exceed the threshold then raise appropriate alerts.
- Recalculate the detection threshold and compare with baseline threshold.
- Repeat previous steps if a major re-configuration occur to the firewall network to obtain the new baseline threshold.

CONCLUSION

This research paper has shown that we can predict the network firewall activities ahead of time using forecasting techniques such as Holt-Winter or Linear Regression. The anomalies described in the previous sections show a solid relationship between the patterns obtained from past logs data and future forecasted data by investigating the difference (i.e. residuals), and the DoS pattern represents as an anomaly. However, as different networks operate under different levels of inbound data traffic per hour, the maximum allowed average of rejected packet counts needs to be determined in a percentage format in order to apply to other network environments.

The proposed model involving analysis of past data with MLP can be considered a simple IDS or IPS system based on quantitative analysis rather than the IDS / IPS rule based systems. The ability to forecast even one hour ahead would give the system administrator the capability of deciding if they should stop certain activities on the network before it is too late. Alternatively more automated responses could be implemented. Further research is needed to confirm these findings in different networks and enhance self learning techniques by applying neural network modelling in order to produce more practical, self manageable and reliable results when compared with other IDS and IPS network security protection techniques.

REFERENCES

- Allan, A. (2003), Intrusion Detection Systems: Perspective, Gartner Research Technology Overview No. DPRO-95367, pp. 1-20.
- Bolzoni, D., Etalle, S. & Hartel, P. (2006). POSEIDON: a 2-tier anomaly-based network intrusion detection system, Proceedings of IWIA 2006 4th IEEE International workshop on Information Assurance, 13-14 April.
- Chang, R. K. C. (2002), Defending Against Flooding-Based Distributed Denial-Of-Service Attacks, Communications Magazine, IEEE, vol. 40, no. 10, pp. 42- 51.
- Chen, L., Longstaff, T. A. & Carley K. M. (2004), Characterization of defense mechanisms against distributed denial of service attacks, Computer & Security Journal, vol. 23, no. 8, pp. 665 - 678.
- Cho, S.B. & Park, H.J. (2003) Efficient anomaly detection by modelling privilege flows using hidden Markov model, Computers & Security, vol 22, no. 1, pp 45-55
- Easley, M. & Stiennon, R. (2002), Intrusion Prevention will Replace Intrusion Detection, Gartner Research Note No. T-17-0115, pp. 1-5.
- Estevez-Tapiador, J., Garcia-Teodoro, P. & Diaz-Verdejo, J. (2005) Detection of web-based attacks through Markovian protocol parsing, ISCC 2005, Proceedings of 10th IEEE Symposium on Computers and Communications, pp27-30 June, pp 457-462
- Ghosh, A. & Schwartzbard, A. (1999), A Study in Using Neural Networks for Anomaly and Misuse Detection, in Proceedings of the 3rd USENIX Windows NY Symposium, July 12-15, Seattle Washington
- Iheagwara, C. & Blyth, A. (2002), Evaluation of the Performance of ID Systems in a Switched and Distributed Environment: The Realsecure Case Study, Computer Networks Journal, vol. 39, no. 2, pp. 93-112.
- Kennedy, R. L., Lee, Y., Roy, B. V., Reed, C. D. & Lippmann, R. (1997). Solving Data Mining Problems Using Pattern Recognition Unica Technologies, Prentice-Hall, PTR, New Jersey.
- Kruegel, C., Mutz, D., Valeur, F. & Vigna, G. (2003), On detection of anomalous system call arguments, Lecture Notes in Computer Science, Springer,
- Kruegel, C. & Vigna, G. (2003) Anomaly detection of web-based attacks, Proceedings of CCS'03, October 27-31, Washington DC, USA
- Lee, S. C. & Heinbuch, D. V. (2001), Training a Neural-Network Based Intrusion Detector to Recognize Novel Attacks, IEEE Transactions on Systems, Man and Cybernetics, Part A, vol. 31, no. 4, pp. 294-299.

- Mukkamala, S., Janoski, G. & Sung, A. (2002) Intrusion detection using neural network and support vector machines, Proceedings of IEEE International Joint conference on neural Networks, Available at <http://serc.iisc.ernet.in>
- Noureldien, N. A. & Osman, I. M. (2000), On Firewalls Evaluation Criteria, in Proceedings of the TENCON 2000 Conference, Kuala Lumpur Malaysia, pp. 104 - 110.
- Ollmann, G. (2003), Intrusion Prevention Systems (IPS) Destined to Replace Legacy Routers, Network Security Journal, vol. 2003, no. 11, pp. 18-19.
- Ordonez-Cardenas, R. & Romero-Troncoso, R. (2008) MLP neural network and on-line backpropagation learning implementation in a low-cost fpga, Proceedings of the 18th ACM Great Lakes symposium on VLSI, GLSVLSI'08, May 4-6, 2008, Orlando, Florida, USA
- Peng, T., Leckie, C. & Ramamohanarao, K. (2007) Survey of network-based defence mechanisms countering the DoS and DDoS problems, ACM Computing Surveys, vol. 39, no. 1.
- Robertson, W., Vigna, G., Kruegel, C. & Kemmerer, R.A. (2006) Using generalisation and characterisation techniques in the anomaly based detection of web attacks, Proceedings of the Network and Distributed System Security Symposium (NDSS), Internet Society, February, USA. available: <http://seclab.tuwien.ac.at/publications.html>
- SANS (2004), Sasser Worm - Week in Review LSASS Exploit Analysis - SANSFIRE 2004, URL <http://isc.sans.org/diary.php?date=2004-04-30>, Accessed 14 Jan 2005.
- Sequeira, K., and M. Zaki. 2002. ADMIT: anomaly-based data mining for intrusions. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. Edmonton, Alberta, Canada. ACM.
- Smith, R. N. & Bhattacharya, S. (1999), Operating Firewalls Outside the LAN Perimeter, in Proceedings of the Performance, Computing and Communications Conference, 1999. IPCCC '99. IEEE International, Scottsdale, AZ USA, pp. 493 - 498.
- Stallings, W. (2006), Cryptography and Network Security, Prentice-Hall Inc., New Jersey.
- Stolfo, S. J. & Lee, W. (2000), A Framework for Constructing Features and Models for Intrusion Detection Systems, ACM Transactions on Information and System Security (TISSEC) Journal, vol. 3, no. 4, pp. 227 - 261.
- Thames, J. L., Abler, R. and Keeling, D. (2008), A Distributed Firewall and Active Response Architecture Providing Preemptive Protection, ACMSE 2008, Auburn AI, March 2008
- Vigna, G., Valeur, F. & Kemmerer, R. A. (2003), Designing and Implementing a Family of Intrusion Detection Systems, in Proceedings of the 9th European software engineering conference held jointly with 10th ACM SIGSOFT international symposium on Foundations of software engineering, ACM Press, Helsinki, Finland, pp. 88 - 97.
- Zanero, S. (2008), ULISSE, a Network Intrusion Detection System, Proceedings of CSIIRW 2008, May 12-14, Tennessee, USA
- Zhang, C., Jiang, J. & Kamel, M. (2005) Intrusion detection using hierarchical neural networks, Pattern Recognition Letters, vol 26, issue 6, pp. 779-791

COPYRIGHT

Mohammed Salem and Helen Armstrong ©2008. The authors assign Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. The authors also grant a non-exclusive license to ECU to publish this document in full in the Conference Proceedings. Any other usage is prohibited without the express permission of the authors.