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 self-learn by analysing its own traffic patterns. 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. Rejected traffic falling outside the projected level for the network under study 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 external access not allowed by the network policies and on the other hand keeping the network up and running to provide these resources on a 24x7 basis without interruption. While this combined goal seems to be reasonable, it is difficult to achieve, requiring substantial amount of design work to produce a secure model for any network. Achieving such a goal 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 caused by 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 of these firewalls do not know how to handle any data packets 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 a firewall is necessary to base-line firewall activities as it has become difficult to predict attacks. Hardening is effective in blocking illegal access to the network resources but it cannot stop other sorts of network attacks by external and internal entities to the open ports, as 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.

The reduction of DoS from external sources will give the network more stability and reliability to deal with other problems caused by internal network activities. Firewalls are good, 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, and researching this area can enhance our understanding of how to provide more protection to large networks that are connected directly to the Internet.

While a firewall needs to be maintained by humans it also records its own activities in the form of log files. These logs can potentially collect, store and analyse firewall activity data to subsequently provide a proactive mechanism to defend the network from future attacks. The idea might look straightforward, however, the implementation is complex where traffic levels on large networks are high, with some networks logging millions of packets every hour.

The aim of this paper is to describe research carried out using TCP/IP traffic data from a live network to produce a self-learning procedure for protection against DoS attacks. The general approach is to statistically develop a forecast of expected 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. The paper commences with a discussion on the shortcomings of current intrusion detection approaches to effectively handle DoS, and describes the development of the firewall baseline and the traffic prediction model.

FIREWALLS AND NETWORK SECURITY

The primary rule 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).

A packet travelling across the network firewall using TCP/IP represents a network communication activity that can be an attack against any network host sitting behind this network firewall (Oppliger, 1997). Firewalls use a screening process on every packet based on pre-defined rules to detect attacks, and this process is very important and sensitive to the whole network infrastructure security. These activities (normally logged by the firewall) are very rich in data and can be analysed later to obtain further information about existing and future attacks. It can also be used to evaluate firewall security, performance and management levels (Noureldien & Osman, 2000).

Firewalls can provide network communication security by applying encryption; however, this normally leads to an overhead on the firewall's ability to process requests unless high-speed links are available to the network. A good example of encryption protocols is the Secure Internet Protocol (IPSec) which provide a secure communication channel that can be used to implement a Virtual Private Network (VPN) using the firewall when accessing network resources via an external host from the Internet (Stallings, 2006). The amount of security that can be provided by firewalls is very 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 which require special tools to be incorporated with the firewall to scan every packet.

With the introduction of some new protocols to the Internet such as Internet Protocol Version 6 (IPv6) to replace existing IPv4, IPSec and secure certificate systems used with Secure Shell Layer (SSL), it becomes more important to revise the way current data pattern matching is used by firewalls. These need to be up-to-date in order to be able to screen and filter datagram packets that use these new protocols (Smith & Bhattacharya, 1999). This adds complexity to how firewalls are managed, however, attack patterns in principal remain the same but the method of carrying out the attack over TCP/IP has changed.

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 (Noureldien & 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.

NETWORK SECURITY USING IDS

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 suitable 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 role-based system. The problem has to occur before it can be identified and configured as a static role in the firewall in order to be used for future filtration and protection.

In DoS attacks, the IDS requires pattern updates to recognise new attacking agents sent by foreign networks it needs to stop in order to be reconfigured. Current IDS models look for a matched pattern of information and activities that can be analysed and categorised as malicious behaviour by the IDS to protect the network from intruders (Vigna, Valeur & Kemmerer, 2003). Although this has been the basic implementation for an IDS in a networked environment it is also one of the main limitations in its ability to detect new attacking techniques without being attacked first.

Looking for patterns might seem straightforward for an IDS which compares network traffic with attack data to pick intrusion patterns (Stolfo & Lee, 2000). However this is more of a static process that the system cannot perform dynamically. An IDS can also make use of decision roles 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 the IDS (Lee & Heinbuch, 2001). Recent implementations of IDS models are classified as either Host-based Intrusion Detection System (HIDS) or Network-based Intrusion Detection System (NIDS).

In the case of HIDS the IDS system is installed directly on the host and every monitored host within the network can itself be considered as an individual IDS. This can be helpful as the number of hosts is limited and manageable, however, most of the time networks have multiple hosts using multiple Network Operating Systems (NOS) environments with multiple applications which make it difficult to manage and use the same set of pattern classifiers and role based detection mechanisms.

On the other hand NIDS uses a more broad approach of IDS toward networked systems by dedicating 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, such as Cisco, UNIX and Windows (Iheagwara & Blyth, 2002; Ollmann, 2003). In addition, as no pattern exists for the attacking agent to be classified as an attack the new attack cannot be defined by the IDS system which will then pass the attacking agent to the network. This causes the attacking agent that is already within the network to start flooding the network with traffic making it unusable. The quicker the IDS can detect the attack the quicker an intrusion can be stopped before causing damage to the network (Chang, 2002).

EXPERIMENT IN SELF-LEARNING FIREWALL

A common problem with an 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, Longstaff and Carley (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 SYNC 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 if the source attacker utilises 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 IDSs. Such an approach is possible by 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.

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, a second round of data was collected from logs from the same firewall network 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 protocol 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.

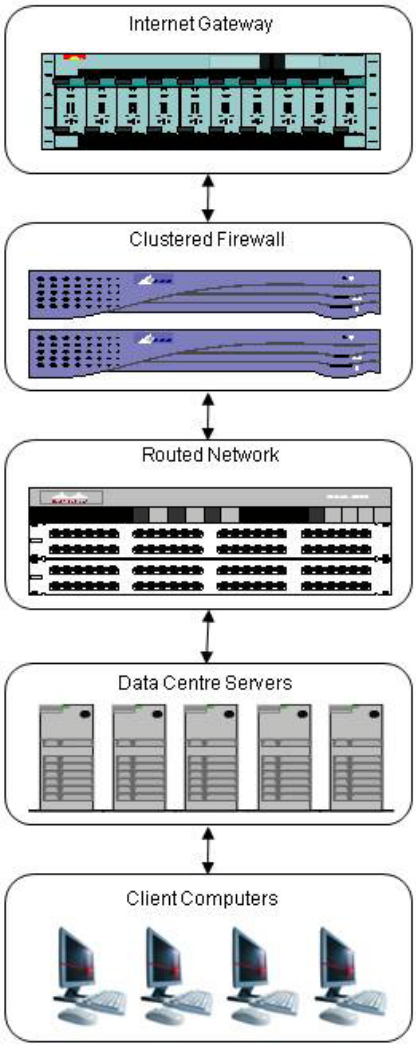

Figure 1 – Infrastructure of large network used in this project

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.

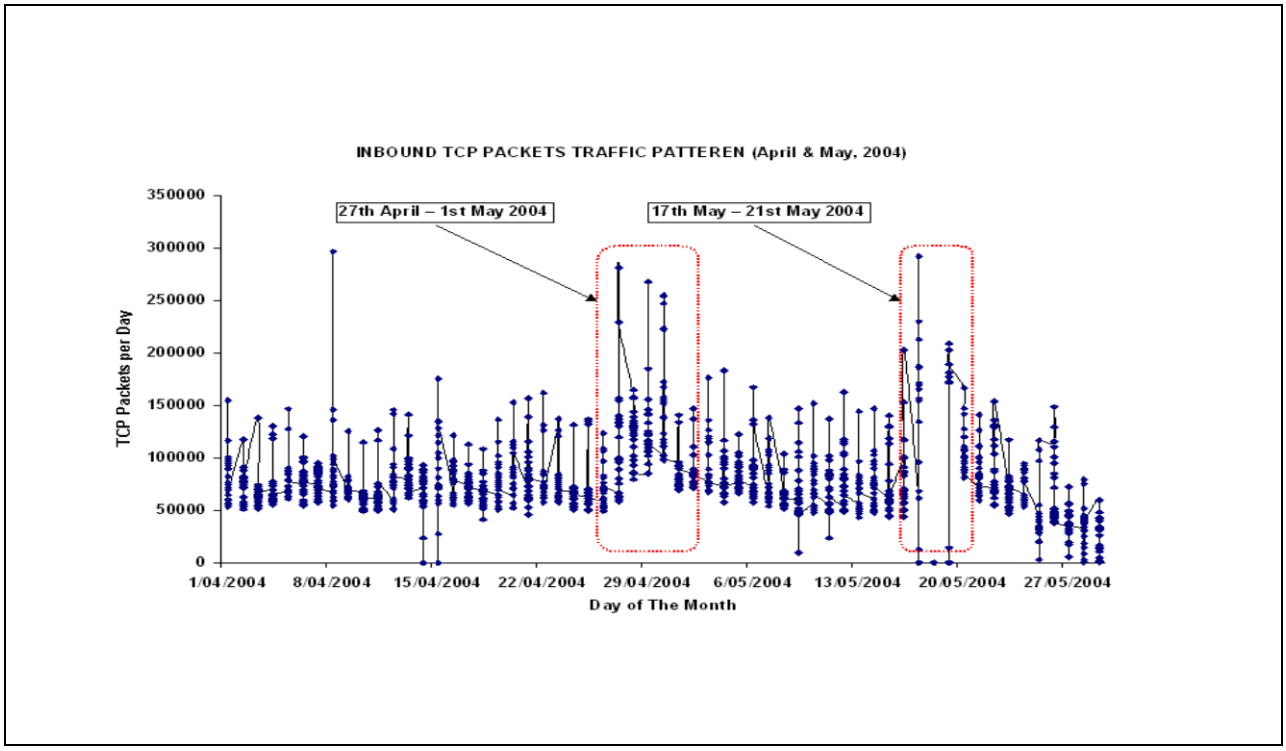

Figure 2A – Rejected Packets Residuals Using Regression and Holt-Winter Forecasting Models.

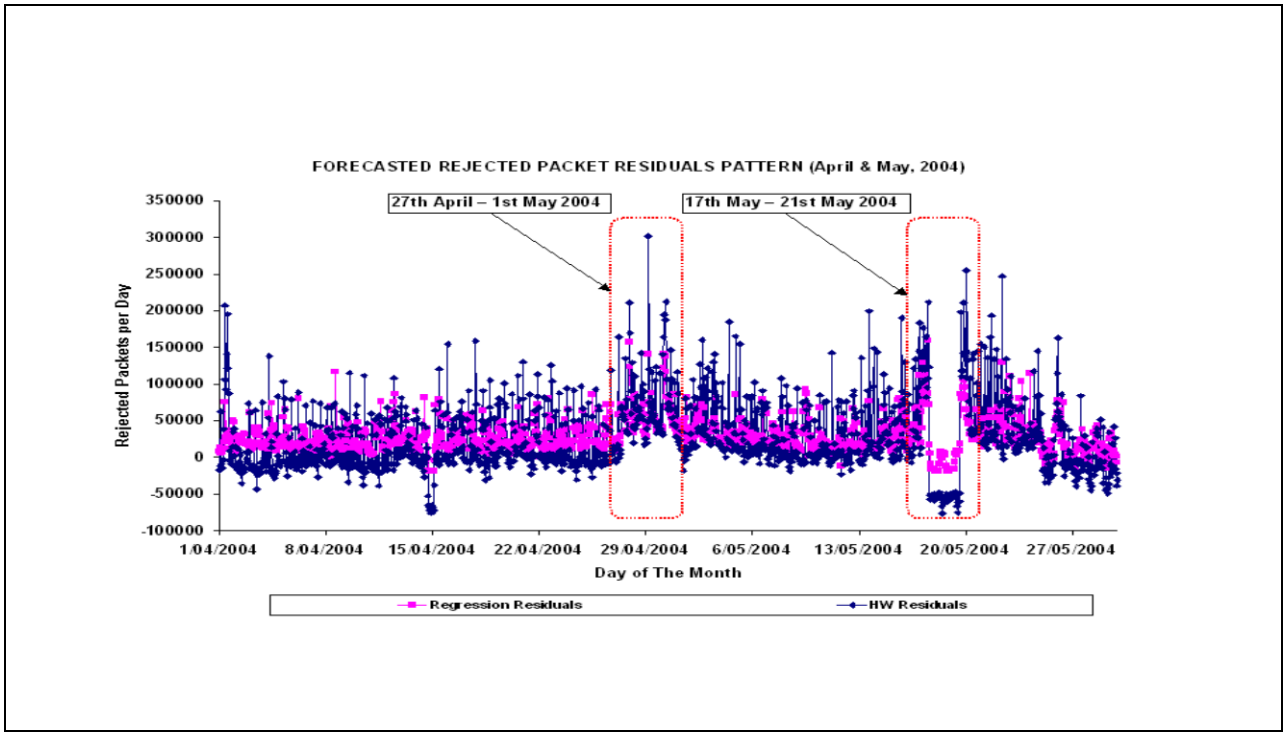

Figure 2B – Inbound TCP Packets Traffic Pattern within the Baseline Data Set.

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 ware 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 ware uses 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 packet 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 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 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.

These tables also 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% 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 re-calculated 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.

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 displays 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 research model involving analysis of past data could 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. Further research is needed to confirm these findings in different networks and enhance self learning techniques by applying neural network modelling 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.

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.

Easley, M. & Stiennon, R. (2002), Intrusion Prevention will Replace Intrusion Detection, *Gartner Research Note* No. T-17-0115, pp. 1-5.

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.

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.

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.

Oppliger, R. (1997), Internet Security: Firewalls and Beyond, *Communications of the ACM Journal,* vol. 40, no. 5, pp. 92 - 102.

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.

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.

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.

COPYRIGHT