Multivariate Statistical Online Analysis for Self Protection against Network Attacks

Guangzhi Qu, Salim Hariri
Internet Technology Laboratory
ECE, The University of Arizona
http://www.ece.arizona.edu/~hpdc

Xuejun Zhu, Jionghua Jin
SIE Department
The University of Arizona

Mazin Yousif
Intel Corporation, USA
mazin.s.yousif@intel.com

Manish Parashar
The Applied Software System Laboratory,
Rutgers University

Abstract

Detection and self-protection against viruses, worms, and network attacks is urgently needed to protect network systems and their applications from catastrophic loss. Once a network component is infected by viruses, worms, or became a target of the network attacks, its operation state will shift from normal to abnormal. Online monitoring mechanism can be used to collect important aspects of network traffic and host data (CPU utilization, memory usage, etc.), that can effectively detect abnormal behaviors caused by attacks. In this paper, we develop an online multivariate analysis algorithm - MANA based on Hotelling's T^2 multivariate statistical technique [6] to analyze the behaviors of system resources and network protocols in order to proactively detect network attacks. The new algorithm builds an adaptive behavior profile of normal operation for system resources. We have validated this algorithm and showed how it can proactively detect well-known attacks such as Distributed Denial of Service, SQL Slammer Worm, and Email spam attacks.

Keyword: Abnormality Distance Metric, Multivariate Online Analysis, Attack Detection, Self Protection

1. Introduction

Network attacks have been lunched at very high rates, almost daily, and the sophistication of these attacks are also increasing exponentially. Network attacks take many different forms such as denial of service, various forms of viruses and worms. In August 2000 when Yahoo, Amazon.com, *CNN*.com and other web sites were hit by Denial of Service attacks lasting for hours [9, 10]. Worms like Code Red, Code Red II and Nimda wreaked havoc around the world during July-October 2001 [11]. *MS SQL* Slammer Worms caused 90% of vulnerable servers all over the world to be infected in 2003 within a few hours [1]. These activities have become a significant threat to the security of our information infrastructure and can lead to catastrophic results.

Attack detection, identification, and prevention within a network system is a challenging research problem due to the continuous change in Software configuration, the variety of network protocols and services being offered and deployed, and the extreme complexity of the asynchronous behaviors of attacks.

The existing attack detection strategies fall into two major categories: rule based detection and anomaly detection [8, 9]. For a rule based detection system, rules or signatures of the known attacks will be compared with the observed behavior, if there is a match, the system will raise an alarm. An obvious limitation of the rule based detection system is that it cannot detect new attacks. On the other hand, anomaly detection techniques build a baseline profile from network element's normal behavior and detect any deviation from the baseline profile in the observed data. Hence, the anomaly detection techniques will detect known and new attacks.

Network systems have multiple measurement attributes that can be used to describe its behaviors. Only one measurement attribute cannot represent accurately the system's operation states. Hence, multivariate analysis methods are needed for representing the system operation states and then use the current operational state to proactively detect abnormal behaviors.

In this paper, online multivariate analysis for attack detection was developed based on Hotelling's T^2 multivariate analysis technique. By using multivariate analysis, we can efficiently detect the viruses, worms, and network attacks at their earliest stage of propagation. The multivariate analysis statistical methods can fuse the data from multiple aspects of the network element behavior and deliver a relative distance of the current operation state from its normal state.

The paper is organized as follows. Section 2 gives the methodology of applying the multivariate statistical analysis techniques. In Section 3, we discuss how to use the multivariate statistical analysis techniques to build the vulnerability metric to detect the network attacks. Section 4 presents our experimental results that demonstrate the effectiveness of our approach to detect a wide range of network attacks. Section 5 presents concluding remarks and future work.

2. Methodology

The overall goal of our approach is to formulate a theoretical approach to use multiple measurement attributes to detect viruses, worms and network attacks efficiently and quantify the operational states of all information system resources and services.

2. 1 Measurement Attributes

In a network system, there are *P* protocols or services working together to implement the required functionality and services. In our approach, we present several measurement attributes that can be used to quantify the operational state of any system resource.

For each node of a network, each protocol can be monitored independently and the measurement attributes for each network node is assumed following a multivariate normal distribution as shown in Table l. Under these assumptions, we can model the online monitoring as a multi-sensor data fusion problem. Multivariate statistics techniques such as Hotelling T^2 control chart can be applied to build the monitoring and detection model.

Table 1: Measurement Attributes

Impacted Protocols	Measurement Attributes	Observed Behaviors
App layer	NIP/NOP: number of incoming/outgoing PDUs. IF: Invocation Frequency	IF increase 2 or 3 in order of magnitude
HTTP, DNS, SMTP, POP		NIP/NOP increases 2 to 3 in order of magnitude compared with normal scenario
Transport layer	NIP/NOP	
TCP, UDP		
Network layer	NIP/NOP AR: ARP request rate	AR increases 1 or 2 in order of magnitude
IP, ICMP, ARP		

In our approach, a network system and its components operates in one of the following three states – normal, uncertain, and abnormal. If a network component is in abnormal state, the conclusion can be drawn that the network component is infected/attacked or attacking others.

At each level of the network hierarchy (*application*, *transport*, *network*, etc.), we identify the appropriate measurement attributes that can be used to quantify the behavior of any protocol or service as being normal/abnormal (see Table 1). For example, by observing the email invocation rate, we can determine whether the email service is operating normally or not. The recent MyDoom [12] email worm caused the infected computer to generate as much as 200 emails per second. Similarly, the number of incoming *TCP* packets per second can be monitored to determine whether or not the *TCP* connection is operating normally.

When a network component experiences an attack, the measurement attributes will be severely affected by the attack and eventually increases/decrease their values such that the component starts operating in an abnormal state. As shown in Figure 1, the abnormality distance function (*ADF*) quantifies the amount of change that must occur (***Delta***) in order to move a component from normal to abnormal state.

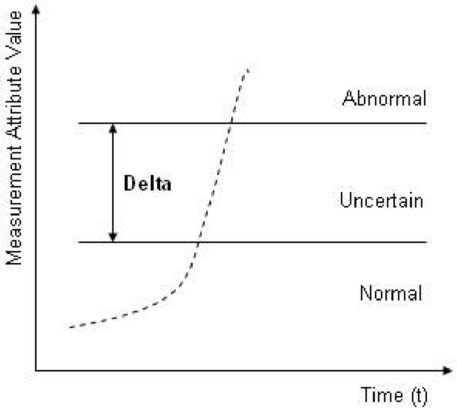

Figure 1: measurement attributes via operation states

2.2 Multivariate Statistical Analysis

Our approach aims at quantifying and characterizing the normal and abnormal operations of network centric system resources such as routers, computers, servers, and network protocols (*TCP/IP*, etc.) as well as their applications/services (*web*, *email*, etc.). For example, if the *SQL* slammer worm attack is launched on the test bed, the online monitoring traffic data will show that the *UDP* packets will be the predominant factor of all the network traffic. At the same time, the outgoing *ARP* traffic of the infected node becomes similarly active because the dependency among these two protocols. This phenomenon is caused by the huge numbers of *UDP* packets that are sent out with different *IP* destination addresses and the infected host needs the *ARP protocol* to get the *MAC* addresses of those worm packets. Actually because different protocols are monitored simultaneously, a network element acts as a multi-sensor system. Hence, multivariate analysis can be applied to study the behavior of these resources and the correlation between the measurement attributes to see whether or not they are operating in normal state or in abnormal state.

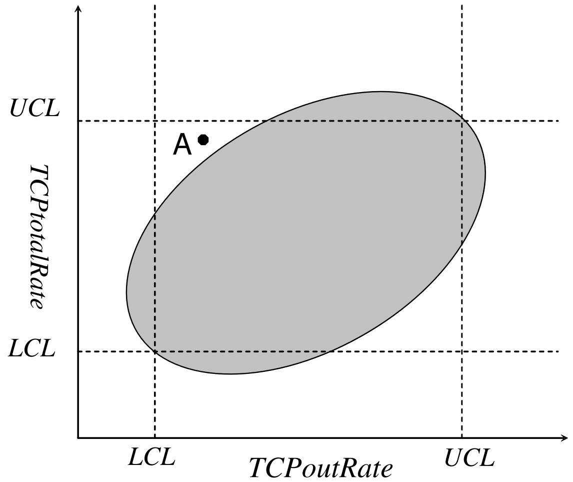

Figure 2: multivariate analysis model

The main reason for using multivariate analysis is because single measurement attribute cannot represent the operation state of the network element accurately. For example, in a spoofed *TCP SYN* attack scenario, we define a node's legitimate outgoing *TCP* packet rate as the rate of *TCP* packets that are sent with source *IP* address is the same as the node's *IP* address, denoted by *TCPoutRate*. And we use *TCPtotalRate* to denote the TCP outgoing rate of *TCP* packets including legitimate outgoing *TCP* packets and spoofed outgoing *TCP* packets. As shown in Figure 2, *UCL* and *LCL* are used to determine the upper and lower thresholds, respectively, for any node to operate normally. For example, the state of a node is quantified by point *A* as shown in Figure 2. By using only one measurement attribute – *TCPoutRate* the node is operating in normal state with respect to *TCPoutRate*. But, when we check another measurement attribute – *TCPtotalRate*, we can immediately determine that the node is operating in abnormal state and thus can detect the occurrence of the attack.

Hotelling's T^2 control chart has been used to perform anomaly detection and root cause analysis in manufacturing systems [13] and it has the capability of generating a normal region from the normal profile of multiple measurement attributes. Our approach for online multivariate analysis to proactively detect and recover from network attacks is based on Hotelling's T^2 control chart. First, we need to determine the baseline profile. According to the Hotelling's T^2 method, we need to have M ($M > 20$) observations. Suppose we have P measurement attributes. Then the normal behavior of the Measurement Attributes (**MA**) can be represented as:

$$\mathbf{MA} = \left\{ \begin{pmatrix} MA_1(t) \\ MA_2(t) \\ \vdots \\ MA_p(t) \end{pmatrix} \begin{pmatrix} MA_1(t+1) \\ MA_2(t+1) \\ \vdots \\ MA_p(t+1) \end{pmatrix} \cdots \begin{pmatrix} MA_1(t+M) \\ MA_2(t+M) \\ \vdots \\ MA_p(t+M) \end{pmatrix} \right\}$$

Based on these normal measurement attributes data set, two control limits - upper control limit (*UCL*) and Lower control limit (*LCL*) can be determined using the *M* preliminary blocks to obtain in-control data for estimation of sample mean $\overline{\mathbf{MA}}$ and covariance matrix ***S***. After that a normal region with respect to the *P* measurement attributes is determined. The sample mean $\overline{\mathbf{MA}}$ determines the normal region center (*NRC*) and the sample covariance matrix ***S*** determines the shape of the normal region as shown the shade part in Figure 2. For a pre-defined Type-I error α , the upper and lower control limits are computed :

$$UCL = \frac{(M-1)^2}{M} B_{1-\alpha/2}[P,(M-P-1)/2]$$

$$LCL = \frac{(M-1)^2}{M} B_{\alpha/2}[P,(M-P-1)/2]$$

where $B_{\alpha/2}[P,(M-P-1)/2]$ is the $1-\alpha/2$ percentile of the β distribution with P and (*M-P-1)/2* denotes the degrees of freedom.

Network attack detection requires monitoring and analysis to be carried out in real time. That means, initially we can compute the baseline profile for network element's normal behavior using offline measurements. However, the upper and lower control limits need to be updated dynamically using the new observations obtained from the online monitoring. A real time Multivariate Analysis for Network Attack detection algorithm (*MANA*) was developed based on Hotelling's T^2 control chart as show in Algorithm 1.

In the *MANA* algorithm, *L* is the number of observations. The number of observations (*L*) that we use in this paper is equal to 30. For any new observation the distance from the center of the normal state region (*NRC)* will be computed. If the distance is beyond the upper and lower bound, then the node is working in abnormal state, the self protection algorithm will be triggered to carry out the appropriate actions (e.g., filtering the attack traffic, shutting the network interface, shutting down a network node, exchanging information among agents, etc) .

```
Algorithm 1– MANA
Repeat Forever
    For (t=1; t<L) do
        Input($MA_1(t)$, $MA_2(t)$, … $MA_P(t)$ );
        D = distance($\mathbf{MA}(t)$, NRC);
        If  D < LCL or D > UCL
            Call self-protection algorithm
        If t = L
            Update(UCL, LCL, NRC)
    End for
End repeat
Algorithm 1– END
```

3. Attack Detection Approach Using Vulnerability Metric

We have used the abnormality distance metric (*AD*) to quantitatively characterize the network system state (e.g. *normal, uncertain,* and *vulnerable*) based on one measurement attribute and to accurately quantify the impact of network attacks or fault on various network components and the whole network system [3]. In this paper, we extend the Abnormality distance metric to consider multiple measurement attributes. The *AD* metric with respect to multiple measurement attributes **MA** can be defined as a function of time *t* in Equation (1).

$$AD_{\mathbf{MA}}(t) = (\mathbf{MA}-\overline{\mathbf{MA}})^t\mathbf{S}^{-1}(\mathbf{MA}-\overline{\mathbf{MA}}) \qquad (1)$$

This AD metric denotes the distance of the network node's operation state at time *t*. from the *NRC*.

4. Experimental Results and validations

We have set up an instrumented test bed environment using the resources in the Internet Technology Laboratory at The University of Arizona to validate and demonstrate our approach in achieving efficient network attack detection and quantifying the operational state of the network element as shown in Figure 3. Cisco routers are used as the core network's backbone routers. In addition, several Linux routers are used as the access routers and are programmed using Autonomia online monitoring and analysis engines. For further information about Autonomia, please refer to [2]. The test bed consists of 4 Cisco 7500 series routers and 5 10/100M switches and 40 PCs. All these computers are configured into 5 sub networks. Network services and applications such as web browsing, email service are running on the test bed. Attack library is used to inject viruses, worms, and different kinds of attacks within the test bed. The network services are monitored and analyzed using Autonomia agents. When we inject the viruses, worms or attacks into the test bed, the Autonomia agents installed on each node continuously collect the appropriate measurement attributes every second and compute the *AD* metrics associated with each protocol/service. The *MANA* algorithm is applied on these measurement attributes. The *AD* metric from the algorithm will show the operation state of the network element.

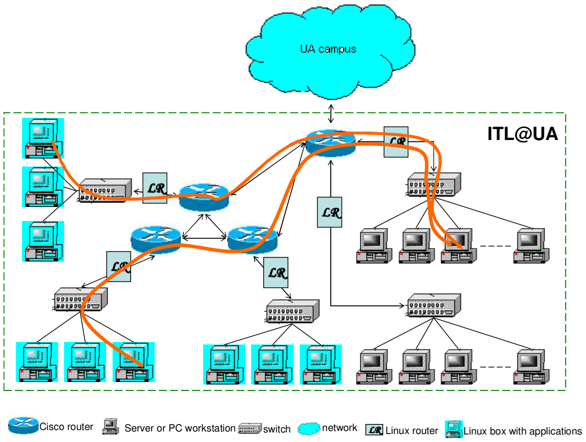

Figure 3: Test bed at the University of Arizona

In what follows, we use three attack scenarios to demonstrate our approach. The first scenario is email worm. The second scenario is *SQL* slammer worm attack and its propagation. The third one is Denial of Service attack.

4.1 Email worm

Email worm is a disruptive network attack. Worm program collects all email addresses from client email program (e.g. *MS Outlook*) and sends hundreds of emails to those email addresses with the worm program itself as attachment.

We monitored client computer email behaviors on our test bed. The email invocations frequency (*AIF*:

the number of email invocations per minute) on the client machine and *DNS* request rate are the measurement attributes used to describe the email service behavior. We apply the *MANA* algorithm to get the *AD* metric $AD_{,smtpout,\ dnsout}(t)$ where *smtpout* is the measurement attribute to denote the *smtp* outgoing rate. Similarly, *dnsout* is the outgoing rate of *dns* packets. The result $AD_{smtpout,\ dnsout}(t)$ are shown in Figure 4. The upper and lower control limits are 6.00 and 0.05. With these two thresholds, it is easy to detect the email worm occurrence when the *AD* metric is greater than 6.00 (beyond the *UCL*).

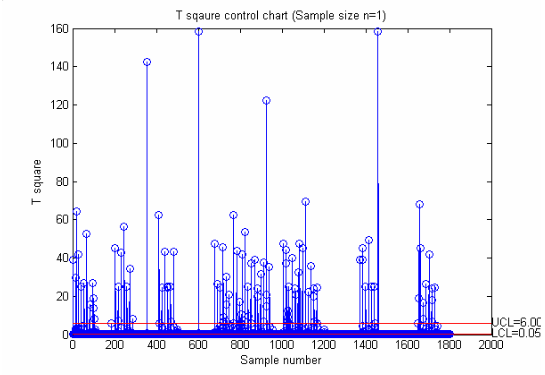

Figure 4: Abnormality distance metric for SMTP and DNS protocols/services

4.2 *SQL* slammer worm attack

In our network attack library, we have built the vulnerable *SQL* server program. The worm is a piece of code which will exploit the vulnerability within the vulnerable *SQL* server, the vulnerable *SQL* server will send 376 bytes packets via *UDP* port 1434 to the original attack launcher and other random destinations to propagate the worms.

By monitoring the network activities of the network nodes in the test bed, we use measurement attributes outgoing *UDP* packet rate and outgoing *ARP* packet rate to demonstrate our approach. During the attack, the *SQL* slammer worm generates a lot of worm packets that are sent towards random destinations which will then cause *ARP* protocol to be extremely active. Applying the *MANA* algorithm on measurement attributes *udpout* (the *udp* packet outgoing rate) and *arpout* (the outgoing rate of *arp* packets) to get the *AD* metric $AD_{udpout,\ arpout}(t)$ as shown in Figure 5. The upper and lower control limits are 7.42 and 0.05. With these two thresholds, the *AD* metric can easily detect the *SQL* slammer worm occurrence when the *AD* metric is greater than 7.42 (beyond the *UCL*). It is very clear that from 260th sample, the *AD* metric value is out of the control limit. After detecting the abnormal state, we also found that the *ARP* outgoing traffic is out of control. There is only 8 seconds delay from the beginning of the attack its detection. In this case the *AD* metric with respect to multiple measurement attributes can rapidly detect the anomaly occurrence.

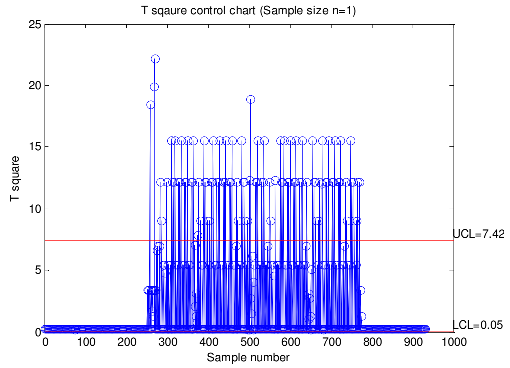

Figure 5: Abnormality distance metric for ARP and UDP protocols

4.2.1 Computation overhead consideration

We can reduce the *AD* metric computation overhead by increasing the sample size. For instance, Figures 6 shows the case when $N = 4$. As long as we choose a sample size for a subgroup properly, we can reduce the computation overhead of the online monitoring and attack detection.

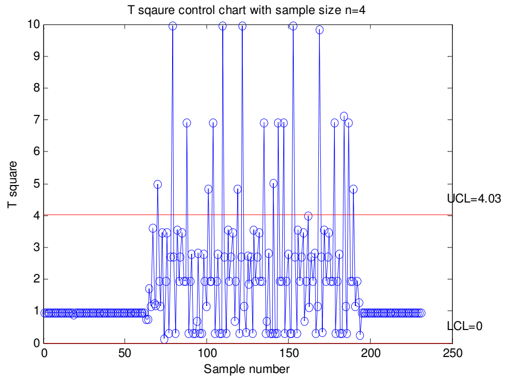

Figure 6: Abnormality distance metric for ARP and UDP protocols with sample size 4

4.3 Denial of Service (*DoS*)

In this scenario, we emulate one of the very popular *DoS* attack methods – TCP SYN attack. We launch the *DoS* attack with spoofing IP address. In our test bed, we set up a Web Server with Apache 2.0.47. During the attack, one of the network nodes will start the TCP SYN attack towards the HTTP Server. The SYN packets rate can be adjusted as constant or random. In our experiment, we set the TCP SYN packet rate as a random one. As shown in Figures 7, 8, the online monitor collects data for outgoing legitimate *TCP* traffic and total *TCP* out activity including legitimate outgoing *TCP* traffic and those spoofed. From the outgoing and incoming *TCP* packet rate, the attacker node is operating normally. While combined with the total *TCP* activity on this node, we can conclude anomaly happened that the attacker is using spoofing to attack.

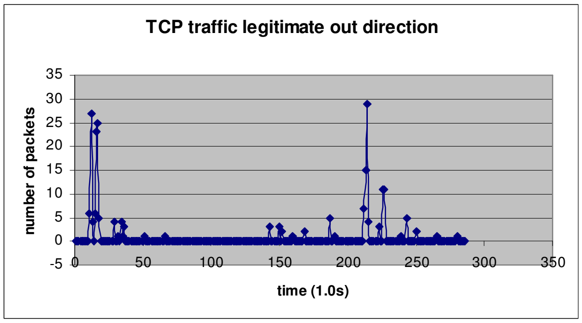

Figure 7: outgoing TCP traffic on the attacker computer

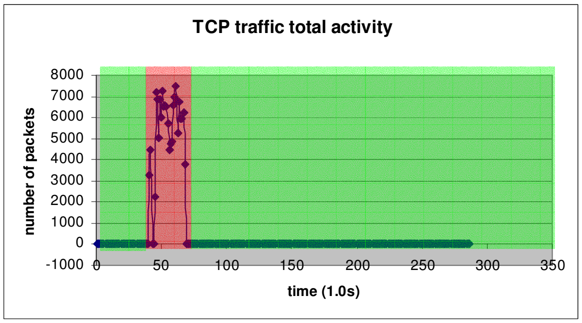

Figure 8: total TCP activity on the attacker computer

Figure 9 shows the result of *AD* metric after applying *MANA* algorithm on the measurement attributes for *TCP* protocol and total *TCP* activity. The upper and lower control limits are dynamically updated. By checking the *AD* metric with respect to *TCPout* and *TCPtotalRate* - $AD_{tcpout, \ totaltcp}(t)$, the occurrence of the spoofing *TCP SYN* attack can be easily detected.

5. Conclusion & Future Work

In this paper, we develop an efficient real time multivariate analysis algorithm - *MANA* based on Hotelling's T^2 multivariate statistical technique to analyze the behaviors of network protocols under a wide range of well known network attacks. The new algorithm builds an adaptive behavior profile of system resources that can be used to detect any abnormal behavior caused by network attacks in real time. We have tested this algorithm and validated its capability to detect several well-known attacks such as Distributed Denial of Service, *SQL* Slammer Worm, and email spam attacks. Also this algorithm can be used to accurately characterize the operating state of each network or system resource or service. We are investigating self protection algorithms based on our online monitoring and analysis of network attacks.

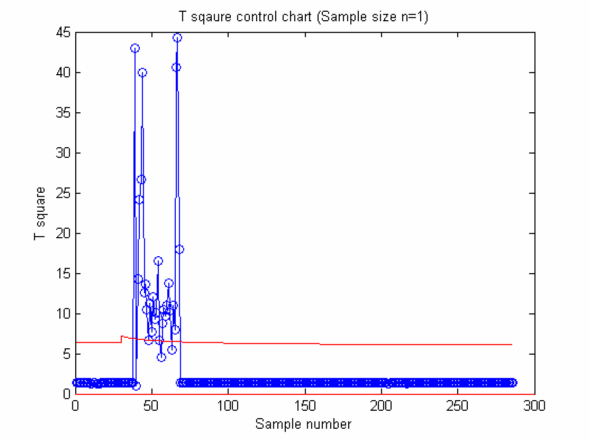

Figure 9: Abnormality distance metric for TCP legitimate outgoing and TCP total activity measurement attribute

6. References

[1]. Q1Lab3 (2003) "The SQL Slammer Worm Incident".http://www.q1labs.com/resources/documents/q1_slammer_whitepaper.pdf

[2] S. Hariri, L. Xue, H. Chen, M. Zhang, S. Pavuluri, S. Rao (2003) "AUTONOMIA: An Autonomic Computing Environment". Submitted to *International Performance Computing and Communications Conference*.

[3] S. Hariri, G. Qu, T. Dharmagadda, M. Ramkishore, C.S. Raghavendra. (2003) "Impact Analysis of Faults and Attacks in Large-scale Networks". *IEEE Security & Privacy*, Sep/Oct Volume 1, Number 5. pp. 49-54.

[4] S. Gaudin (2003). "2003 Worst Year Ever for Viruses,Worms".http://www.internetnews.com/infra/article.php/3292461

[5] Symantec and Ian Poynter, Jerboa Inc. (2000) “Quantifying Vulnerabilities in the Networked Environment: Methods and Uses Char Sample”.
[6] Montgomery, D. C. (2000). Design and Analysis of Experiments, 5th Edition.

[7] R.P.Lippmann, D.J.Fried, I.Graf, J.W.Haines, K.P.Kendall, D.McClung, D.Weber, S.E.Webster, D.Wyschogrod, R.K.Cunningham, and M.A.Zissman, Evaluating Intrusion Detection Systems: The 1998 DARPA off-line Intrusion Detection Evaluation. *Proceedings DARPA Information Survivability Conference and Exposition (DISCEX) 2000,* Vol. 2, pp. 12-26, IEEE Computer Society Press, CA, 2000.

[8]A.Lazarevic, L.Ertoz, V.Kumar, A.Ozgur, and J.Srivastava, “A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection,” *Proceedings of the Third SIAM Conference on Data Mining,* May 2003.

[9]Wired News: A Frenzy of Hacking Attacks. (2000) http://www.wired.com/news/business/0,1367,34234,00.html

[10]Wired News: Was Yahoo smurfed or Trinooed? (2000)http://www.wired.com/news/business/0,1367,34203,00.html

[11] Womrs Threat Aanlysed (2002), retrived from http://www.uksecurityonline.com/threat/worms.php

[12] LinuxProx, (2004). “News: New Worm Spreading Rapidly Across Internet”
http://www.overclockersclub.com/newscomment.php?article=7518375

[13] J.Jin, J.Shi (2001). “Automatic feature extraction of waveform signals for in-process diagnostic performance improvement”, Journal of Intelligent Manufacturing 12, 257-268.