Journal of the Nigerian Association of Mathematical Physics
Volume27, (*July, 2014*), pp485 – 494

An Anomaly Based Statistical Intrusion Detection Model

Konyeha Susan and Onibere Emmanuel

Department of Computer Science, University of Benin.

Abstract

Security is of utmost importance in creating robust and reliable computer networks. The critical challenge is to defend computer systems against network based intrusive activities. In this paper, we present a statistical model for intrusion detection that builds on the knowledge of normal traffic behaviour of the protected computer network. The deviation from the normal traffic pattern is used to detect anomalies in the network traffic. This statistical method is able to detect previously unknown attacks (novel attacks). Our intrusion detection system is able to achieve high detection accuracy with a low false positive rate for a variety of simulated attacks on the network.

Keywords: Intrusion detection, anomaly-based detection, signature based detection.

1.0 Introduction

Computers and computer networks have become indispensible to modern day organizations, business institutions and governments. People are completely dependent on computer networks, as the networks play a major role in their daily operations. The necessity for protecting their networked systems has also increased. Services which rely on computer networks such as email, web-browsing, social networks, remote connections, online payment, and online chatting are being utilized by many users every day. A single intrusion into the service supporting networks can result in denial of service, content alteration, and theft or loss of very privileged and important data. Therefore network intrusions pose a severe threat to the privacy and safety of computer users and this has become a global and urgent problem. The use of antivirus, user authentication and firewalls is no longer enough protection for these networks as these can be bypassed easily using easily accessible tools. Intrusion Detection Systems (IDSs) can be introduced as an added layer of defense for these networks. IDSs can be deployed to identify intrusions and initiate the mitigation of their resultant effects.

The Internet, a driver for online transactions is an essential tool as well as an important component of business operation strategies, hence it is important to ensure network security and to provide secure information channels [1]. Intrusion detection is a major research problem area in network security and is based on the concept that the behaviour of intruders is different from legal users [2]. The goal of intrusion detection systems (IDSs) is to identify unusual access mode or attacks to secure internal networks [3].

When planning, building, and operating a network, the importance of a strong security system should be emphasized and made the basic component of every network design process while yet balancing security with user friendly experience. Unlike in the past when hackers were highly skilled programmers who understood the details of computer communications and how to exploit vulnerabilities, today almost anyone can become a hacker by simply using freely downloadable network penetration tools from the Internet. These complicated attack tools can generally open networks and assess their vulnerability. With the development of large open networks, security threats have increased significantly in the current decade. This has increased the need for network security and dynamic security policies.

2.0 Anomaly-based IDS Model

Many distinct techniques are used depending on type of processing related to behavioural model. They may be statistical anomaly based detection, operational or threshold metric model, Markov Process or Markov Model, statistical moments or mean and standard deviation model, neural network model, fuzzy logic model, outlier detection model, computer immunology based models, user intention based model [4]. In [5], it is observed that a wide variety of techniques including data mining, statistical modeling and hidden Markov models have been explored as different ways to approach the anomaly

Corresponding author: Konyeha Susan, E-mail: susan.konyeha@gmail.com, Tel.: +2348060826547

detection problem [5]. Anomaly based IDSs observe network traffic or computer activities and detect intrusions by identifying activities distinct from a user's or a system's normal behaviour. Anomaly detection is concerned with identifying events that appear to be anomalous with respect to normal system behaviour. Most commonly used approaches for anomaly based detection models are statistical methods [6], e.g. Statistical Packet Anomaly Detection Engine (SPADE) [7], intrusion detection based on the use of agents [8], ontology rule in anomaly behaviour detection [9], use of the Hierarchical Hidden Markov Models (HHMM) [10], and the construction of user action classifier based on the Markov models [11]. Anomaly detection may be host or network based.

Statistical model based Anomaly IDSs use a profile describing the normal network traffic, and any abnormal behaviour observed to deviate from the model is identified. Data relating to the behaviour of legitimate users is collected over a period of time, and then statistical tests are applied to the observed behaviour to determine whether that behaviour is legitimate or not. It has the advantage of detecting attacks which have not been found previously in contrast to signature-based systems. It requires a training phase to develop the profile of legitimate activities and a careful setting of threshold level for the detection of anomalous activities. A framework for the holistic approach for network security strategy of an organization is presented in Table 1.

Table 1: Output of the IDS

		Actual situation	
		Attack	No Attack
Prediction by the IDS	Attack	True positive	False Positive
	No Attack	False Negative	True negative

2.1 Proof of Statistical Formulae for Mean and Standard Deviation Model

We used the mean and standard deviation model as a statistical tool for implementing our anomaly detector component of SKONI_IDS. The model is based on sample mean and variance formula used in statistics. Formulas for computing variance may involve sums of squares, which can lead to numerical instability as well as to arithmetic overflow when dealing with large values. The proof of the equivalence of the formulae used for sample mean and variance formula used, and derivation are presented below. A formula for calculating the variance of an entire population of size N using a naïve algorithm is:

$$\sigma^2 = \frac{\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i\right)^2 \Big/ N}{N} \tag{1}$$

A formula for calculating an unbiased estimate of the population variance from a finite sample of n observations is:

$$s^2 = \frac{\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i\right)^2 \Big/ n}{n-1} \tag{2}$$

Because $\left(\sum_{i=1}^{N} x_i^2\right)$ and $\left(\sum_{i=1}^{N} x_i\right)^2 \Big/ n$ can be very similar numbers, the precision of the result can be much less than the inherent precision of the floating-point arithmetic used to perform the computation. This is particularly bad if the standard deviation is small relative to the mean. However, the algorithm can be improved by adopting the method of the assumed mean. A two pass algorithm is an alternative approach which is more reliable than the method of the assumed mean for large sets of data. This requires using a different formula for the variance. The formula for computing the sample mean is:

$$\bar{x} = \sum_{i=1}^{N} \frac{x_i}{n} \tag{3}$$

To compute the sum of the squares of the differences from the mean:

$$s^2 = \sum_{i=1}^{N} \frac{(x_i - \bar{x})^2}{n-1} \tag{4}$$

The two pass algorithm implemented using this formula is often more numerically reliable than the naïve algorithm for large sets of data, although it can be worse if much of the data is very close to but not precisely equal to the mean and some are quite far away from it. It is often useful to be able to compute the variance in a single pass, inspecting each value x_i only once; for example, when data are being collected without enough storage to keep all the values, or when costs of memory

access dominate those of computation. This kind of computation is carried out using a statistical recurrence relation from which the required statistics can be calculated in a numerically stable fashion. The formula used for updating the mean and variance for an additional element x_{new}, where $\bar{x}_n$ denotes the sample mean of the first n samples $(x_1,\ldots,x_n)$, and s_n^2 is the sample population variance are presented in equations (5) and (6).

$$\bar{x}_n = \bar{x}_{n-1} + \frac{x_n - \bar{x}_{n-1}}{n} \tag{5}$$

$$s_n^2 = \frac{(n-2)s_{n-1}^2 + (x_n - \bar{x}_n)(x_n - \bar{x}_{n-1})}{n-1}, n > 1 \tag{6}$$

The above formulae can be derived using the formula for sample mean, equation (3), and formula for sample variance, equation (4). Straight forward translation of the equation for sample mean into code suffers from loss of precision because of the difference in magnitude between an additional element and the sum of all elements in the sample. The procedures for the derivation of equations (5) and (6) from equations (3) and (4) are shown below.

$$\bar{x}_n = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\bar{x}_n = \frac{1}{n}(x_n + \sum_{i=1}^{n-1} x_i) \tag{7}$$

Putting n = n-1 into equation (3) we get:

$$\bar{x}_{n-1} = \frac{1}{n-1}\sum_{i=1}^{n-1} x_i$$

Hence $\sum_{i=1}^{n-1} x_i = (n-1)\bar{x}_{n-1}$ (3*)

Putting equation (3*) into equation (7) we get:

$$\bar{x}_n = \frac{1}{n}(x_n + (n-1)\bar{x}_{n-1}) \tag{8}$$

This equation (8) is then re-arranged to arrive at equation (5) which is the updated mean for an additional element of the sample

$$\bar{x}_n = \bar{x}_{n-1} + \frac{1}{n}(x_n - \bar{x}_{n-1})$$

This formula also provides us with some useful identities in equations (9) and (11):

$$x_n - \bar{x}_{n-1} = n(\bar{x}_n - \bar{x}_{n-1}) \tag{9}$$

$$x_n - \bar{x}_n = n(\bar{x}_n - \bar{x}_{n-1}) - \bar{x}_n + \bar{x}_{n-1} \tag{10}$$

$$x_n - \bar{x}_n = (n-1)(\bar{x}_n - \bar{x}_{n-1}) \tag{11}$$

Starting with the sum of the squares of the differences from the means of samples, equation (12), we derive equation (6)

$$s_n^2 = \sum_{i=2}^{n} \frac{(x_i - \bar{x}_n)^2}{n-1} \tag{12}$$

$$(n-1)s_n^2 = \sum_{i=2}^{n} (x_i - \bar{x}_n)^2 \tag{13}$$

$$(n-1)s_n^2 = \sum_{i=2}^{n} ((x_i - \bar{x}_{n-1}) - (\bar{x}_n - \bar{x}_{n-1}))^2 \tag{14}$$

$$(n-1)s_n^2 = \sum_{i=2}^{n}(x_i - \bar{x}_{n-1})^2 - \sum_{i=2}^{n}(\bar{x}_n - \bar{x}_{n-1})^2 + 2\sum_{i=2}^{n}(x_i - \bar{x}_{n-1})(\bar{x}_n - \bar{x}_{n-1}) \quad (15)$$

Simplifying the first summation:

$$\sum_{i=2}^{n}(x_i - \bar{x}_{n-1})^2 = (x_n - \bar{x}_{n-1})^2 + \sum_{i=1}^{n-1}(x_i - \bar{x}_{n-1})^2 \quad (16)$$

Now from equation (12), putting n = n – 1,

$$s_{n-1}^2 = \sum \frac{(x_i - \bar{x}_{n-1})^2}{n-2} \quad (17)$$

Substituting equation (17) in equation (16),

$$\sum_{i=1}^{n}(x_i - \bar{x}_{n-1})^2 = (x_n - \bar{x}_{n-1})^2 + (n-2)s_{n-1}^2 \quad (18)$$

From equations (9) and (18), we obtain equation (19)

$$\sum_{i=1}^{n}(x_i - \bar{x}_{n-1})^2 = (n-2)s_{n-1}^2 + n^2(\bar{x}_n - \bar{x}_{n-1})^2 \quad (19)$$

Simplifying the second summation using equation (9), we get:

$$\sum_{i=1}^{n}(\bar{x}_n - \bar{x}_{n-1})^2 = n(\bar{x}_n - \bar{x}_{n-1})^2 \quad (20)$$

Simplifying the third summation:

$$\sum_{i=1}^{n}(x_i - \bar{x}_{n-1})(\bar{x}_n - \bar{x}_{n-1}) = (\bar{x}_n - \bar{x}_{n-1})\sum_{i=1}^{n}(x_i - \bar{x}_{n-1}) \quad (21)$$

$$\sum_{i=1}^{n}(x_i - \bar{x}_{n-1})(\bar{x}_n - \bar{x}_{n-1}) = (\bar{x}_n - \bar{x}_{n-1})\left(x_n - \bar{x}_{n-1} + \sum_{i=1}^{n-1}(x_i - \bar{x}_{n-1})\right) \quad (22)$$

$$\sum_{i=1}^{n}(x_i - \bar{x}_{n-1})(\bar{x}_n - \bar{x}_{n-1}) = (\bar{x}_n - \bar{x}_{n-1})\left(x_n - \bar{x}_{n-1} - (n-1)\bar{x}_{n-1} + \sum_{i=1}^{n-1}(x_i)\right) \quad (23)$$

$$\sum_{i=1}^{n}(x_i - \bar{x}_{n-1})(\bar{x}_n - \bar{x}_{n-1}) = (\bar{x}_n - \bar{x}_{n-1})\left(x_n - \bar{x}_{n-1} - (n-1)\bar{x}_{n-1} + (n-1)\bar{x}_{n-1}\right) \quad (24)$$

$$\sum_{i=1}^{n}(x_i - \bar{x}_{n-1})(\bar{x}_n - \bar{x}_{n-1}) = (\bar{x}_n - \bar{x}_{n-1})\left(x_n - \bar{x}_{n-1}\right) \quad (25)$$

$$\sum_{i=1}^{n}(x_i - \bar{x}_{n-1})(\bar{x}_n - \bar{x}_{n-1}) = n(\bar{x}_n - \bar{x}_{n-1})^2 \quad (26)$$

Now we substitute equations (19), (20) and (26) back into equation (15)

$$(n-1)s_n^2 = (n-2)s_{n-1}^2 + n^2(\bar{x}_n - \bar{x}_{n-1})^2 + n(\bar{x}_n - \bar{x}_{n-1})^2 - 2n(\bar{x}_n - \bar{x}_{n-1})^2 \quad (27)$$

$$(n-1)s_n^2 = (n-2)s_{n-1}^2 + n^2(\bar{x}_n - \bar{x}_{n-1})^2 - n(\bar{x}_n - \bar{x}_{n-1})^2 \quad (28)$$

$$(n-1)s_n^2 = (n-2)s_{n-1}^2 + (n-1)(\bar{x}_n - \bar{x}_{n-1})^2 \quad (29)$$

Dividing both sides by $n-1$ produces equation (6) which gives us the updated sample variance

$$s_n^2 = \frac{(n-2)s_{n-1}^2 + (n-1)(\bar{x}_n - \bar{x}_{n-1})^2}{n-1}$$

The statistical formula stated in equations 5 and 6 have been derived and can be used to update the mean and variance of the sequence, for an additional element X_{new}.

3.0 Methodology

These statistics formulae were used to implement an anomaly detection algorithm in Java programming language for our IDS based on the traditional statistical determination of the normalcy of an observed data element which is a member of a clustered data set relative to some specified confidence range. The adaptive engine component checks for anomaly in network traffic volume for an IP:PORT pair and then computes an anomaly score given by $x_n - (\mu + 3(s_n) + T)$. Where x_n is number of packets counted in the nth cycle (e.g IP:PORT pair), μ is mean count of packets, s_n is standard deviation and T is Threshold level, which sets the sensitivity of the detection system. The network traffic is examined within a 30 seconds window, for which packets that enter the network during the analysis cycle are stored in a buffer and analyzed at the end of the cycle. During the analysis cycle, the analysis engine computes $x_n - (\mu + 3(s_n) + T)$ for each packet. If the statement $x_n - (\mu + 3(s_n) + T) < 0$ is true, the packet is normal, else the packet is anomalous. We used standard deviation instead of the variance because the area under the normal distribution curve is obtained using the values of the mean and standard deviation for a normal distribution curve [12]. The normal distribution curve is shown in Figure 1.

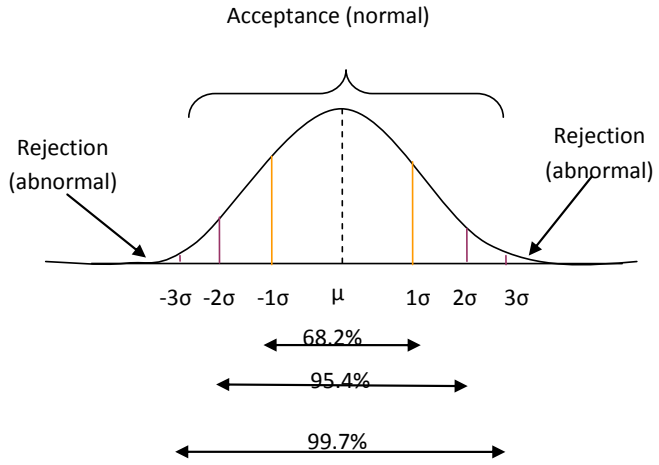

Figure 1: Normal distribution curve

We took our baseline as the upper bound for the third standard deviation from mean plus a pre-assigned threshold value. Thus our baseline is given as $\mu + 3(s_n) + T$. Using the upper bound, we can define a bound of $x_n - (\mu + 3(s_n) + T) < 0$ as a normal region, where the proportion of observed packet scores falling in the region is at least 99.7%. Beyond this normal region, the observed packets (IP:PORT pair) are anomalous.

Our system design methodology is the object oriented design using the unified modeling language (UML). The design of the database model for the system used the object oriented approach which is a modeling and development methodology based on object oriented (OO) concepts [13]. Objects are described by their attributes known as instance variables.

3.1 Set Up of Test Bed

For the experiments, we set up a Local Area Network (LAN) consisting of a server with Internet connection and 10 clients (hosts). SKONI_IDS was configured on one of the host on the LAN. Static IP address was configured manually on the server with IP address 192.168.0.1 and subnet mask 255.255.255.0. Static IP addresses were configured manually on the hosts using the following IP address 192.168.0.x (where "x" ranges from 2 to 11) with the subnet mask 255.255.255.0. We collected several network data traces from a cyber café to use for our experiment. The normal activity in a cyber café consists of accessing e-mails, browsing, authentication, uploading and downloading of files etc requiring the following network protocols: tcp, http, ack, arp, dns, https, and arp etc. A log or record of this normal data trace was made and used for our experiments. The set up of the test bed for data collection and testing is shown in Figure 2.

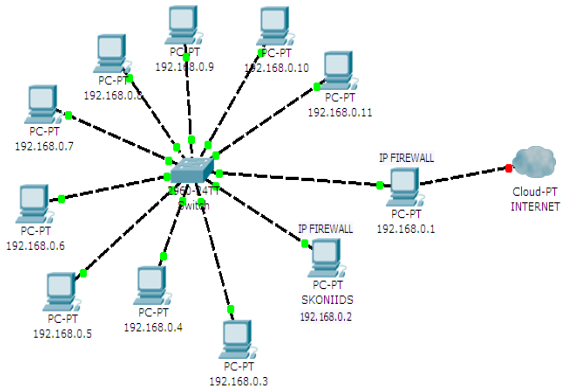

Figure 2: Set up of test bed

3.2 Procedures for Anomaly Detection

Anomaly detection is effective against previously unknown attacks but the system first has to be trained with data containing the normal network traffic of the network which requires monitoring. This training is used by the system, SKONI_IDS, to learn the normal profile of the network. Training and Detection are the two processes carried out here. In the training mode the system learns, whereby the learning algorithm uses a statistical formula to compute the mean and standard deviation. The result of the computation is used to update the previous mean count of packet and the standard deviation whenever an additional packet enters the network. This result of the computation is used to generate a profile or database table for the monitored network connections. During the detection process, the adaptive engine checks for anomaly in the network traffic by comparing incoming network connection statistics against the profile or database table. The code that automates the training and detection process is presented below:

```
void detect(){
packets = getPackets();
                }
voiddoAnalysis(Integer val, Stats sta) {
l.entering("Task", "doAnalysis");
try {
if (!checkVolume(val,sta)) {
sta.setAdditional(val);
facade.merge(sta);
AlertPlayer.getInstance().play();
new AlertChecker().displayAlertAD(sta.getStatsPK().getDest()+":"+sta.getStatsPK().getDestPort(),packets);
alert = true;
            }
   @Override
public void receivePacket(final Packet packet) {
     //l.info("Received a packet");
keepPacketForAnomalyDetector(packet);
voidkeepPacketForAnomalyDetector(Packet packet){
            packetList.add(packet);
   }
void train() {
packets = getPackets();
if(packets.size()>1)
l.info("First packet time is: "+packets.get(0).sec+" and last packet time is: "+packets.get(packets.size()-1).sec);
else
l.info("No packet at this time");
        Stats stat;
HashMap<StatsPK,Integer>sortedPackets = sort(packets);
Object[] ids = sortedPackets.keySet().toArray();
for (inti = 0; i<ids.length; i++) {
```

```
facade = StatsFacade.getInstance();
stat = null;
stat = (Stats) facade.find((StatsPK)ids[i]);
if(stat==null){
stat = new Stats((StatsPK) ids[i]);
double mean = sortedPackets.get((StatsPK)ids[i]);
doublesd = 0.0;
stat.setMean(mean);
stat.setStandardDev(sd);
stat.setNumber(1);
facade.persist(stat);
}else{
            Integer prevNo = 0;
if(stat.getNumber()==null)
            prevNo = 1;
else
            prevNo = stat.getNumber();
double mean = findMean(prevNo,stat.getMean(),sortedPackets.get((StatsPK)ids[i]));
            double                                                    sd                                    =
findStandardDev(prevNo,stat.getStandardDev(),mean,stat.getMean(),sortedPackets.get((StatsPK)ids[i]));
prevNo++;
stat.setNumber(prevNo);
stat.setMean(mean);
stat.setStandardDev(sd);
facade.merge(stat);
          }
        }
    }
public static Double findMean(intprevNo,double mean, int score){
doubleans = (prevNo*mean + score)/(prevNo+1);
returnans;
    }
public static double findStandardDev(intprevNo,doublesd,doublenewMean, double oldMean, int score){
doubleans;
ans = ((prevNo*sd*sd)+(score-newMean)*(score-oldMean))/(prevNo+1);
returnMath.sqrt(ans);
    }
```

4.0 Results and Discussions

A log trace comprising of connections observed during operation of the network being monitored by SKONI_IDS is analyzed. This log trace, comprising a mix of connections observed during training and other connections not observed during training, was then analyzed by SKONI_IDS during detection cycle. These connections flagged alerts which was recorded and presented in Figure 3. We observed two of our host computers which received anomalous packets.

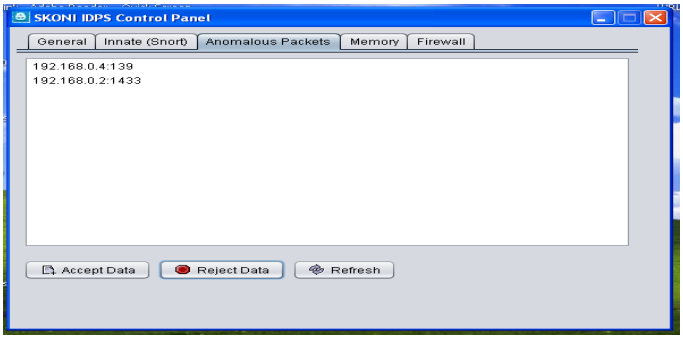

Figure 3: IP address of hosts which received anomalous packets

We viewed the list of anomalous packets by clicking on the host's IP and then clicking on list tab. A close investigation will filter out the culprit packets.

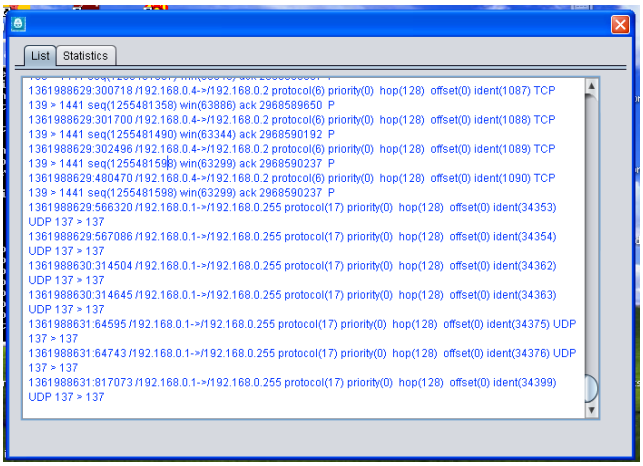

Figure 4: Anomalous packets received by the host.
A graph plotted indicating the baseline alongside the count of packets at a glance shows the anomalous packets detected.

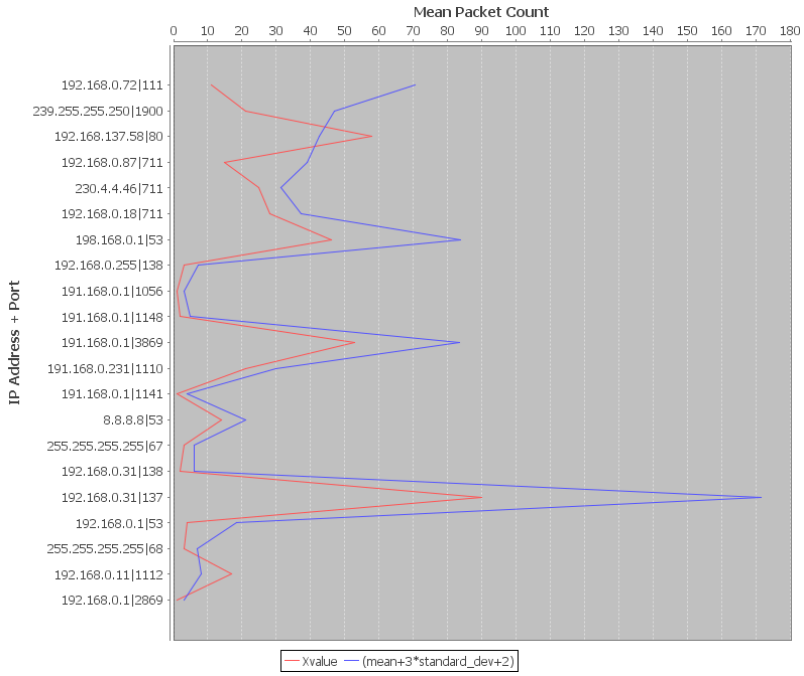

Figure 5: Anomalous packets from 192.168.137.58:80 and 192.168.0.11:1112

5.0 Findings

In testing for anomaly using live traffic, SKONI alerted on several instances of anomalous packets from 192.168.0.231:445, 192.168.0.72:32751, 192.168.0.72:1555, 192.168.0.72:6, 192.168.0.72:8620, 192.168.0.72:4319, 192.168.0.71:44511, 192.168.0.72:2431, 192.168.0.72:36801, 192.168.0.72:52789 and 192.168.0.72:44509. This is interpreted as an intense port scan attack.

6.0 Summary and Conclusion

Network based attacks are widespread and have become less tractable as new technologies emerge in computer development. Firewalls and security policies, which have been the main flagship of computer security systems, can no longer handle these

attacks because the attack methods are continuously changing. Many attackers are able to evade these security measures by taking advantage of new undiscovered bugs and hidden vulnerabilities. Our system can detect attacks including those that are previously unknown at the application and network layer using attack profile of network packets. Our work combined signature detection with anomaly detection for an effective computer security system.

7.0 References

[1.] Shon, T. and Moon, J. (2007): A hybrid machine learning approach to network anomaly detection. Information Sciences, vol.177, pp. 3799-3821

[2.] Stallings, W. (2006): "Cryptography and network security principles and practices", 2nd Edition, pp. 33, Prentice Hall, USA

[3.] Tsai, C., Hsu, Y., Lin, C. and Lin, W. (2009): "Intrusion detection by machine learning: A review", Expert Systems with Applications, vol. 36, pp.11994-12000

[4.] Jyothsna, V., Prasad, R. V. V., and Prasad, M. K. (2011): A Review of Anomaly based Intrusion Detection Systems. International Journal of Computer Applications (0975 – 8887), Volume 28– No.7, pp 26-35,

[5.] Modi, C., Patel, D., Patel, H., Borisaniya, B., Patel, A. and Rajarajan, M. (2012): A survey of intrusion detection techniques in Cloud. Journal of Network and Computer Applications, doi:10.1016/j.jnca.2012.05.003

[6.] Denning, D. E. (1987): An Intrusion Detection Model. IEEE *Transactions on Software Engineering*, 13(2):222–232

[7.] Biles, S. (2006). "Detecting the Unknown with Snort and the Statistical Packet Anomaly Detection Engine (SPADE)", Technical Report, 01.01.2006, available via web address http://www.computersecurityonline.com/spade/SPADE.pdf

[8.] Dasgupta, D., Gonsalez, F., Yallapu, K., Gomez, J., Yarramstetti, R. (2009): CIDS: An agent-based intrusion detection system. Computers & Security, 24:387- 398

[9.] Isaza, G.,Castillo, A.,López, M., and Castillo, L. (2009): Towards Ontology-Based Intelligent Model for Intrusion Detection and Prevention. In Computational Intelligence in Security for Information Systems - CISIS 09, 2nd International Workshop, Burgos, Spain, 23-26 September 2009 Proceedings. Volume 63 of Advances in Intelligent and Soft Computing, pages 109-116, Springer, DOI: 10.1007/978-3-642-04091-7_14

[10.] Chunfu, J. and Deqiang, C. (2009): Performance Evaluation of a Collaborative Intrusion Detection System. College of Information Technology and Science, NankaiUniversity , Tianjin 300071, China cfjia@nankai.edu.cn. pp 409 – 413

[11.] Jha,S., Tan, K., and Maxion, R.A. (2001): Markov Chains, Classifiers and Intrusion Detection //Computer Security Foundations Workshop (CSFW), page 206

[12.] Lipschutz, S. and Schiller, J. (1998): Schaum's Outline Series of Introduction to Probability and Statistics. McGraw Hill Companies, Inc., page 183.

[13.] Peter, R. and Carloc, C. (2007): "Database systems: Design, implementation and management" Thomas Course technology, Seventh Edition, 0-4188-3593-5.