Journal of Advances in Computer Research
Quarterly pISSN: 2345-606x eISSN: 2345-6078
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 6, No. 4, November 2015), Pages: 139-156
www.jacr.iausari.ac.ir

Network Situational Awareness and Quantitative Threat Assessment Based on Multi Sensor Information Fusion

Amin Sardeh Moghadam[1]✉, Behzad Moshiri[1],Ali Payandeh[2]
(1) Control and Intelligent Processing Center of Excellence ECE, University of Tehran, Tehran, Iran
(2)Department of Information and Communication Technology, Malek Ashtar University of Technology, Tehran, Iran
binaloodir@gmail.com; moshiri@ut.ac.ir; payandeh@mut.ac.ir

Received: 2014/10/28; Accepted: 2015/07/01

Abstract

Threat assessment in the computer networks of organizations can reduce damage caused by attacks and unexpected events. Data fusion models such as the JDL model provide efficient and adequate sensors to gather the right information at the right time from the right components. This information then is refined and normalized to provide situational awareness and assess events that may be intended as a threat. This study suggests a new method based on the JDL model where data collected from different sources is normalized into an appropriate format. After normalization, Data is converted into the information. Threat assessment unit analyzes this information based on various algorithms. We use three algorithms to detect anomaly, one to correlate alerts, and one to determine the successfulness of an attack. The model is then evaluated based on a small simulated network threat to ascertain the efficacy of the proposed method. The results show that the method is an appropriate model for situational awareness and threat assessment.

Keywords: *threat assessment, data fusion, situation awareness, computer networks*

1. Introduction

Every organization has a mission and uses information technology to support that mission. Risk management is critical to protecting the assets of the organization. A primary aspect of the process of risk assessment is threat assessment [1]. Threat assessment can be done at the developmental phase of a system or during its lifetime. Both methods should be used to assess the threats.

During the development of a system, threat is usually assessed by means of threat modeling [2]. Over the lifetime of a system, threat often is assessed based on evidence of attack. The complexity of today's networks makes the process of network monitoring and threat assessment increasingly difficult. The massive volume of data produced by different sensors can be overwhelming.

Different algorithms and methods have been used to assess threats to a network. Data fusion was introduced for military applications, and expanded to other applications. Researchers have used the data fusion approach for network threat assessment by proposing new models and the algorithms to handle uncertainties in the network. An excellent review paper [3] cluster and describe about 100 articles related to the cyber situational awareness. It is a remarkable reference to students who always interested in security of cyber environment. Paper [4] first proposes online algorithm for the alert fusion. This algorithm is similar to alert correlation methods. Then threat priority is calculated based on D-S theory.In the paper [5], IDS alerts after preprocessing and normalization are being verified using NASL script Language. Then their Severity is determined based on CVSS, and finally severity of alerts is multiplied by success rate to calculate threat. Another paper [6] has provided an information fusion framework to assess the threat. This paper examines the information aggregation models and finally suggests a fusion architecture in which Bayesian belief networks are used as a mechanism to assess the threat. Using the information security risk management (ISRM) is one of the solutions to provide security of information resources. Situation aware ISRM is presented in [7] to complement the ISRM process. The paper [8] is similar to the previous article except that the support vector machine is used to assess the threat. Other researchers have used different methods for network threat assessment. One approach uses CVSS and IDS alerts as contextual information and aggregates them to show the threat score [9]. Another uses predefined metrics of network performance to measure the impact of denial of service attacks on service availability [10]. All the studies focused on threat assessment.

The approach proposed in the present study examines aspects of data fusion models such as object, situational, and threat assessment. Existing approaches mainly use alerts sent from an intrusion detection system to assess a threat. This paper introduces a new model based on JDL model with three level. This is the first paper that discusses the model with detail in each level. Other papers mostly present a model without details and focus on the threat assessment part. Therefore a complete description of model is one of the advantages of this paper. This study suggests a new model in which information from alerts, vulnerabilities, and monitoring parameters from network intrusion detectors, vulnerability scanners, and monitoring tools are processed to evaluate the situation and threat real-timely. So using various information resources and algorithm, and being real-time is another advantage of this paper. The model is then evaluated using data from a simulated network. The result can’t be compared with other papers. For there are diverse factors in each simulation. Just we can compare the result of two paper, if all of the conditions of the simulation be the same.

The remainder of this paper is organized as follows. Section 2 brings out the proposed model and its details, while Section 3 discusses the architecture of our simulation and the software is used for that. In Section 4 experiments are discussed based onthreat scenario, and in Section 5 we provide a conclusive summary and suggest future directions.

2. The Proposed Method

Monitoring and evaluation of a computer network situation improve the performance of the network, avoiding attacks, and increasing the availability of all monitored devices. The proposed model monitors computer networks and assesses network threats based on the JDL model, the most prominent model for data fusion. The JDL model and its revisions [27-30] focus on maximizing the automation of fusion. The first part of the model is the object assessment unit that provides the required data through its agents to evaluate the network situation. Then, the situational assessment unit converts the data provided into a reference format that is called information. This new format can be used to visualize and assess threats. The threat assessment unit uses the information to assess threats via various algorithms and formulas.

Specifically, the system has been designed to process data and events from various network resources. Implementing this system requires the following components:

- **Agents:** Software agents located on the equipment for the purpose of gathering the initial data. Some companies claim not to use agents and the evidence of this can be seen from the low efficiency of their tools. Agents provide data determined by the administrator.
- **Management console:** This manages the agents. The management console requests and receives data from agents or determines a scheduling task and receives data automatically.
- **Normalizer:** The normalizer converts the data to an appropriate format (information). Information should be in the form of a reference language.
- **Visualizer**: Large amounts of information can lead to confusion and the lack of proper understanding of the network situation. Visualization provides effective and comprehensive network situational awareness that is appropriate to the reference language.
- **Threat analyzer:** Assesses threat based on the severity of network vulnerabilities, alerts, and anomaly rates that exist in the network monitoring parameters.
- **User interface:** Communicates with all parts of the model and provides interaction between users and the model.

The proposed model employs the following in accordance with the data fusion level of the JDL model [30]:

- **Signal processing/features:** Managed devices on the network receive signals and convert them to features.
- **Entity/object assessment:** Agents gather the data requested by the management console from the managed devices.
- **Situational assessment:** The normalizer converts data to a reference format that can be used for visualization or threat assessment.
- **Threat assessment:** The threat analyzer processes the information provided and aggregates a large amount of information to assess the threat.

2.1. Object Assessment Unit

Figure 1 shows details of the object assessment unit. The unit components are described below.

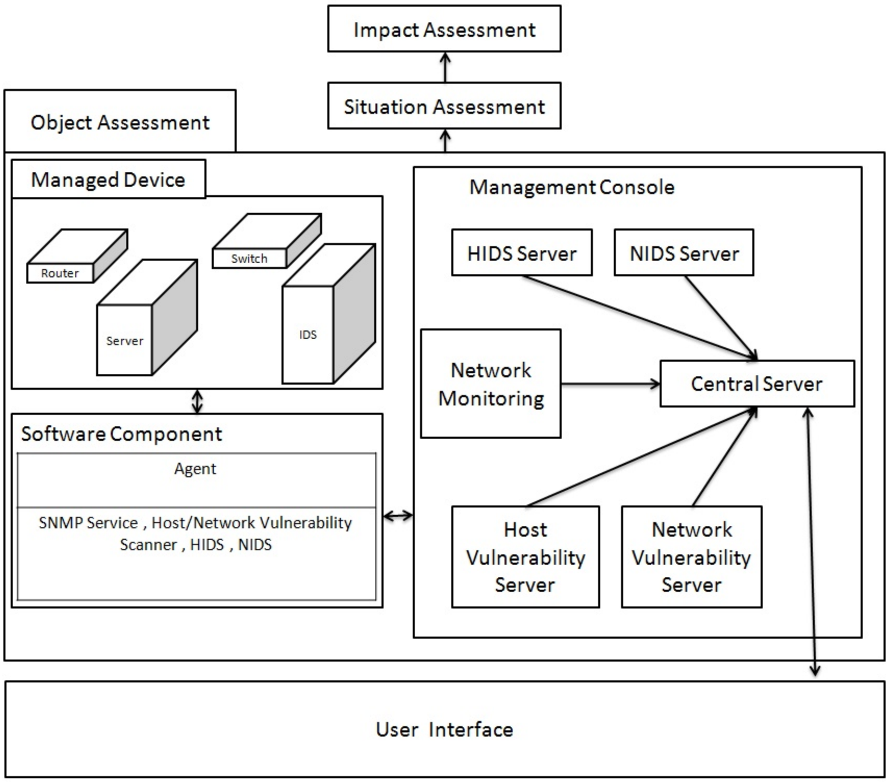

Figure 1. Object assessment unit

2.1.1. Managed devices

This component includes all equipment used in the network, such as routers, switches, and servers. The number and types of managed devices is a network-dependent feature. The network administrator has a crucial role in selecting the best network architecture.

2.1.2. Software agents

Software agents receive data from managed devices and send it to the management console. As shown in Figure 1, these agents can include intrusion detection systems, vulnerability scanners, or the separate software that communicates with them. Because of the large size of a computer network, the proposed model uses a multi-sensor hierarchical architecture [26] to obtain the needed data.

2.1.3. Management console

The console includes several different servers. Each server is specifically designed to collect data from specific heterogeneous sensors.

2.1.4. Central server

The central server connects the network administrator to the first part of the model. The administrator set policies and configures the management console to gather data. The central server also communicates with other servers to transmit the policies determined by administrator and collect the data from them.

2.1.5. Network intrusion detection system server

This server communicates with all the network intrusion detection systems used in the network. Intrusion detection systems can be the same type or different types. The server receives the data from all intrusion detection systems, eliminates the redundant data, and validates any alert sent by them. The output of this server is a collection of useful data such as events, network attack, and alerts.

Many alerts sent by the intrusion detection system are general and are of low value. Alert correlation methods are used to identify high value alerts and increase confidence in the validation of alerts. If there are no alert correlation tools in the network, following algorithm can be used to correlate the alerts [4]. In this algorithm, a raw alert (RA) is invalid and a hyper alert (HA) is valid. The AID is the unique id-number of an attack event; SrcIP and DstIP are the source and destination IP addresses, respectively, of an attack event.SigID is the signature generated by an IDS sensor indicating the type of attack, AT is the time the attack event occurs or is detected, N denotes the number of raw alerts maintained by the hyper-alert; Ts is the timestamp for the first creation; and Te is the time of the latest updating; maxT is the max time span of a hyper-alert; expT is a expire time for outputting a hyper-alert; find is a sign used in the fusion process.

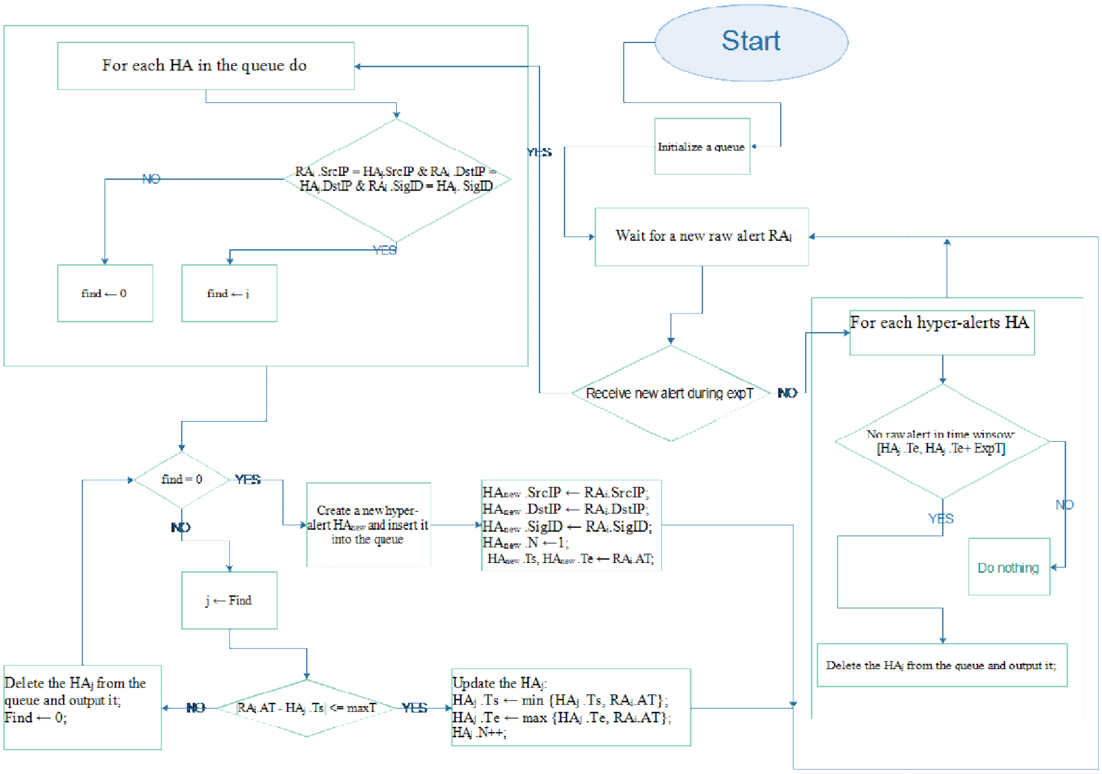
Start
Initialize a queue
YES
Wait for a new raw alert RA_i
Receive new alert during expT
NO
For each HA in the queue do
RA_i .SrcIP = HA_j.SrcIP & RA_i .DstIP = HA_j.DstIP & RA_i .SigID = HA_j. SigID
NO
YES
find ← 0
find ← j
find = 0
YES
NO
Create a new hyper-alert HA_{new} and insert it into the queue
HA_{new} .SrcIP ← RA_i.SrcIP;
HA_{new} .DstIP ← RA_i.DstIP;
HA_{new} .SigID ← RA_i.SigID;
HA_{new} .N ←1;
HA_{new} .Ts, HA_{new} .Te ← RA_i.AT;
j ← Find
|RA_i.AT - HA_j .Ts| <= maxT
NO
YES
Delete the HA_j from the queue and output it;
Find ← 0;
Update the HA_j:
HA_j .Ts ← min {HA_j .Ts, RA_i.AT};
HA_j .Te ← max {HA_j .Te, RA_i.AT};
HA_j .N++;
For each hyper-alerts HA
No raw alert in time winsow [HA_j .Te, HA_j .Te+ ExpT]
YES
NO
Do nothing
Delete the HA_j from the queue and output it;

Figure 2. Correlation algorithm

2.1.6. Host intrusion detection system server

This server is similar to the network intrusion detection system, except that it communicates with the host intrusion detection system installed on the host computer. Alerts from the host intrusion detection system are different from alerts from the network intrusion detection systems and have a lower alert level.

2.1.7. Network vulnerability scanning system server

This server communicates with all network vulnerability scanners. It receives the vulnerabilities from the scanners located in the different segments of the network. Duplicate data may be reported by the sensors because of the multi-sensor architecture. To manage this, an algorithm of redundant vulnerability elimination can be easily applied. If the vulnerability ID and IP address of two reports are the same, one is removed. The output of this server is a set of network vulnerabilities.

2.1.8. Host vulnerability scanning system server

This server is similar to the network vulnerability scanning system server, except that it communicates with host vulnerability scanners installed on the host computer. The output of this server is a set of host vulnerabilities.

2.1.9. Network monitoring

Network monitoring [11] can achieve different information about the network. This tool provides awareness about the performance of the various network segments and data for network threat assessment.

The simple network management protocol (SNMP) is an internet-standard protocol for managing devices on IP networks. This protocol uses a hierarchical data structure (MIB) to collect the required data. The structure of the MIB is laid out in an SNMP-related standard (RFC 1155) that defines how MIB information is organized, what data types are allowed, and how resources within the MIB are represented and named. The MIB contains name, object identifier (a numeric value), data type, and indication of whether the value associated with the object can be read from and/or written to. While the top levels of the MIB are fixed, specific sub-trees are defined by IETF, vendors, and other organizations.

The MIB is an extensible structure; SNMP can be used to gather all necessary information. All servers can communicate with the agents using this protocol and receive the data determined by network administrator.

2.2. Situational Assessment

A situation in the data fusion domain is a relationship between objects. An alternative definition is “a series of events occurring in a time step” [24]. Each event can be expressed using several sentences. A situational assessment unit consists of two basic components. The first component normalizes the collected data into a reference format. Each situation in this model is shown in structured extensible markup language (XML) documents. The second component visualizes the XML documents to better understand the network situation.

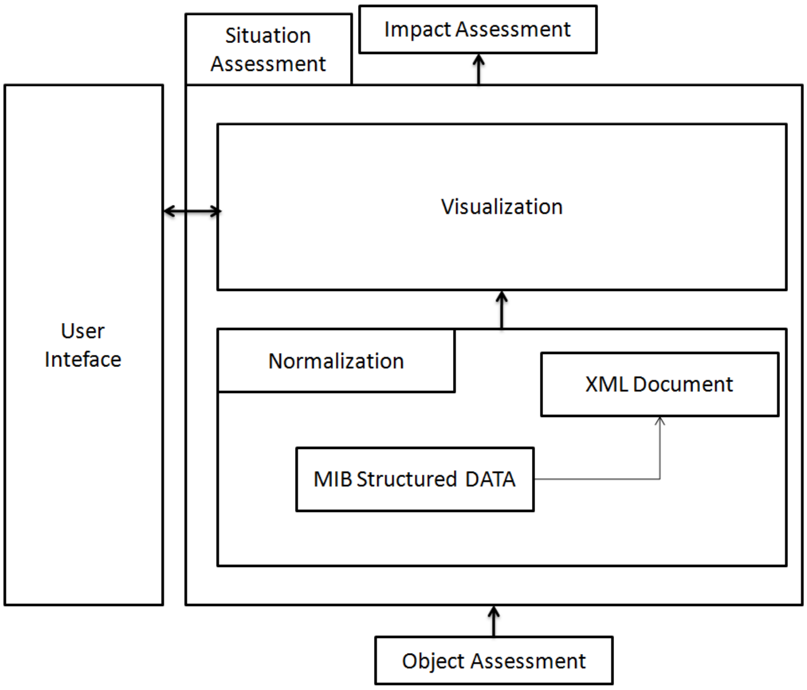

Figure 3.Situational assessment unit

2.2.1. Normalization

Data collected in the form of SNMP protocols and MIB structured data cannot be used to visualize and provide situational awareness. MIB structured data should be converted to a suitable format to have a data structure that can be used to visualize or assess the threat.

XML is a standard language that is used to define a set of rules for encoding documents in a form that is understandable to humans and machines. The goal of this standard was originally simplicity of language, comprehensiveness, and usability in the internet. Over time, it has proven useful in a wide range of applications. The language supports unicode for all languages around the world.

XML is an extensible language; because the labels of XML are not predefined, which allows an author to define individual labels and structures. XML documents can be defined in a tree structure with a root directory as the parent of all other elements. The tree starts from the root and continues to the lowest level with branches and leaves. All elements in the tree can have sub-elements and each of them can have contents and attributes.

Figure 3 represents the network as a tree. The first level represents a network that should be monitored by the model. The second level represents the types of managed devices, including routers, switches, servers, and other devices in the network. Information about the each device is displayed in the third level. Information needed for network situational awareness is shown in the fourth level. The types and number of leaves on the tree are determined by organizational policies and set by the network administrator, but the trunk of the tree is formed by the network structure. Network scanners can be used to build the trunk of the tree.

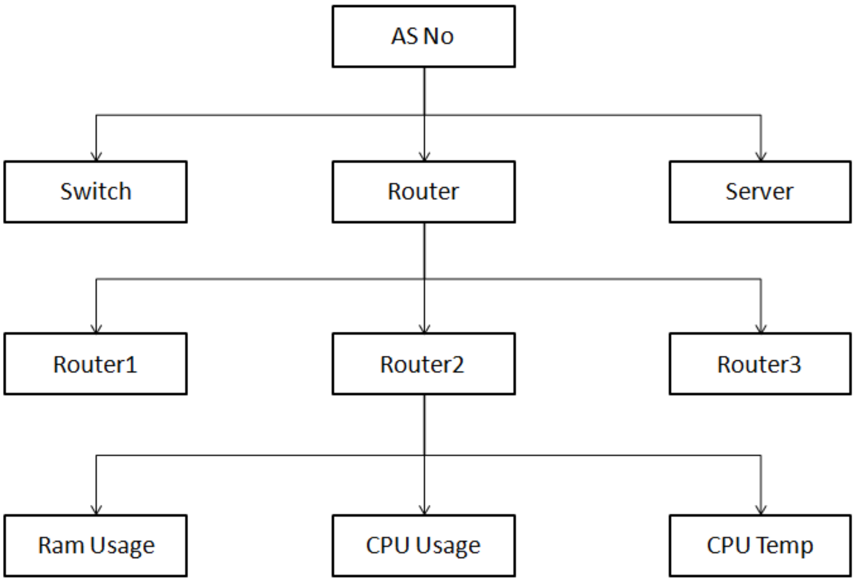

Figure 4.Example of network structure

The parameters proposed in this study are presented in table format for the leaves of the tree for intrusion detection systems (Table 1), vulnerability scanning systems (Table 2), and monitoring tools (Table 3) that are used for situational awareness.

Table 1. Parameters of intrusion detection systems

Parameter	**Description**
SensorID	Identifies the sensor
Time	Shows the time that the alert is reported
Alert name	Specifies the name of the alert
SourceIP	Gives the source IP address of the packet related to the alert
Sourceport	Gives the source port of the packet related to the alert
DestinationIP	Gives the destination IP address of the packet related to the alert
Destinationport	Gives the destination port of the packet related to the alert
Class	Shows the class to which the alert belongs
Completion	Determines whether or not the alert is issued for a successful attack
Severity	Determines the severity of the alert

Table 2. Parameters of vulnerability scanners

Parameter	Description
CVEID	Identifies the vulnerability
Description	Describes the vulnerability
Score	Determines the rate of severity of the vulnerability
Risk factor	Shows the severity of the vulnerability on four levels
Affected host	Gives the IP address of host with the vulnerability
Solution	Gives solutions to resolve the vulnerability

Table 3. Monitoring parameters

Parameter	Description
Resource usage	Specifies the resource usage (e.g. CPU, memory, disk)
Availability	Determines the availability of the monitored device
Bandwidth	Represents the amount of bandwidth consumed
Response time	Shows the time of response to a request

2.2.2. Visualization

Visualization is any technique that creates images, diagrams, or animation to communicate a message. Visualization through visual imagery is an effective way to communicate abstract and concrete ideas. The XML language is easy to understand, but expanding networks and increasing the amounts of equipment makes awareness of the network situation difficult. To better understand it, visualization should be used. An internet search easily produces tools for the visualization of XML documents. This study proposes using 3D-XV language (3D XML visualizer) [25] for the visualization of XML documents. This tool is a graphical interface that accesses large structured documents. It is a geometric model designed to combine sequential organization and hierarchical structure.

2.3. Threat assessment unit

The threat assessment structure is shown in Figure 4. Second part delivers information from the XML document format to the threat assessment unit in time steps. Vulnerabilities and their severity, alerts and their severity, and monitoring parameters are used to assess the threat.

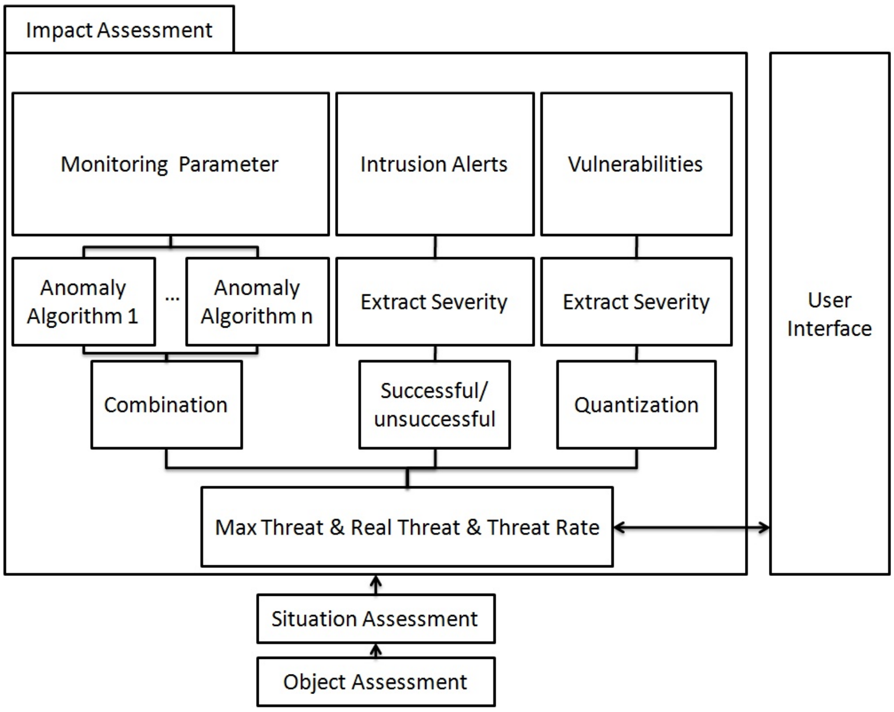

Figure 5. Threat assessment unit

Network behavior is central to situational awareness and threat assessment. The parameters that define network behavior are mainly provided by network monitoring tools. Examples of these are CPU load, memory usage, bandwidth usage, response time, and packet loss. Parameters with analogous behavior indicate that problems exist in the network and each problem in the network can lead to a threat. Past network monitoring parameters are considered to be network behavior. Current information having the same behavior as past information is considered to be normal behavior.

One aspect of real threat is calculated based on the anomaly rate of parameters. Anomaly detection algorithms [12] compare current and past information. The lack of similarity between current and past data can indicate incorrect behavior and a threat to the network. Anomaly rates are calculated by dividing the number of parameters showing anomalies into the total number of parameters. For example, if 4 parameters out of a total of 10 show anomalies, the anomaly rate is 0.4. All anomaly detection algorithms give false positives and negatives. Classification combiners [18] can be used to increase the accuracy of the results. The number and type of anomaly detection methods used is a function of the available resources and type of data.

Total real threat is a combination of the anomaly rate and successful alerts. The alert rate is the sum of severity of successful alerts sent by intrusion detection systems. Intrusion detection systems mainly determine the severity of alerts. CVSS can also be used to determine severity by calculating a default severity from the severity of the class of the alert. This is the average severity of alerts that belongs to one class.

The success of an attack determines the real threat to a network. NASL[1] can be used to determine the successfulness of an attack. The total real threat is obtained by multiplying the anomaly rate by the sum of severity of successful alerts in the time steps. The following formulas show the threat to a network:

$$MT = \sum \left(V^{*}S \right) \tag{1}$$

$$RT = \left(\sum A^{*}S \right)\left(1 + AR \right) \tag{2}$$

$$TR = RT / MT \tag{3}$$

WhereMT is maximum threat, V is vulnerability, S is severity, RT is real threat, A is alert, AR is anomaly rate, and TR is threat rate.

2.3.1. Synchronization

Security applications, real time services, and network management require the synchronization of devices and computers. Threat in the time steps is also assessed in the time steps. The time of a device can change because of inherent instability, environmental factors, user manipulation, or device error, thus it must constantly be set using a timing source. The NTP protocol is an accurate protocol for sending the coordinated universal time (UTC) via a packet switching network. This protocol compensates for the delay between servers and computers and is accurate up to a few milliseconds. The UDP port number in this protocol is 123.

NTP uses a tree structure to prevent a loop. Tree structure synchronizes all devices in an autonomous system. Root of the tree is an atomic clock and the tree can be expanded to 16 levels. The root in the AS is the gateway router; other devices connected to this router create the next level of the tree. The number of levels depends upon the size of the network. Devices close to the root show more accurate times than those further away, meaning that the most critical devices should be close to the gateway router.

[1] Nessus Attack Scripting Language

3. Evaluation of the Proposed Method

Virtualization was used to simulate a network to evaluate the proposed model. One host operates as a switch between other hosts. VirtualBox [23] was used to simulate the network. As shown in Figure 5, OS1 at the top is the host OS and other hosts run through VirtualBox. The OS installed for this simulation was Windows 7, Windows Server 2003, Windows Server 2008, windows XP, and Ubuntu 11.10.

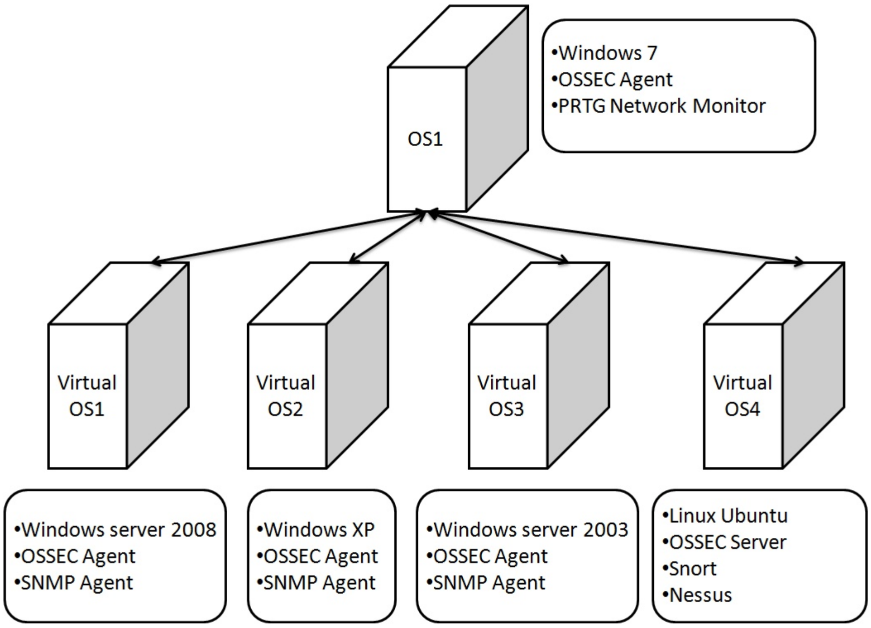

Figure 6. Simulated network

The network intrusion detection system used for the simulation was Snort [19] installed on Ubuntu Linux (Virtual OS4). Nessus [21] and OSSEC Server [20] were installed on the virtual OS as a network vulnerability scanner and host intrusion detection system. OSSEC Server communicates with OSSEC agents installed on the Windows OS. Protector Plus [22] was the host vulnerability scanner and the DARPA 98 dataset [17] was replayed as simulated network traffic. Virtual OS4 operated as a switch with an IP address of 172.16.112.1. The IP addresses of the other hosts were 172.16.112.10, 172.16.112.20, 172.16.112.50, and 172.16.112.149. PRTG network monitoring software [13] was used to select the CPU usage, RAM usage, and traffic information such as the volume of WWW traffic, FTP, mail, infrastructure, remote control, and total. Six 15-min time steps were used to simulate network behavior.

Replicator neural networks [14] were used for one-class anomaly detection. In this network, a multi-layer feed forward neural network is constructed. The number of input and output neurons corresponds to the features in the data. Training was done by compressing data into three hidden layers. The testing phase reconstructed each data instance (x_i) using the learned network to obtain the reconstructed output (o_i). The reconstruction error (δ_i) for x_i is then computed as:

$$\delta_\mathrm{i} = \frac{1}{n}\sum_{j=1}^{n}\left(x_{ij} - o_{ij}\right)^2 \tag{4}$$

where n is the number of features over which the data is defined and δ_i is directly used as an anomaly score for the test instance.

Figure 6 shows the results of applying this method to the collected data. As seen, the average anomaly rate of the parameters is approximately 8%in the normal state of the network based on the neural network algorithm. The anomaly rate increased to 18% at times because of changes in the total number of bytes. Iperf was used to change this parameter in the sender and receiver computers.

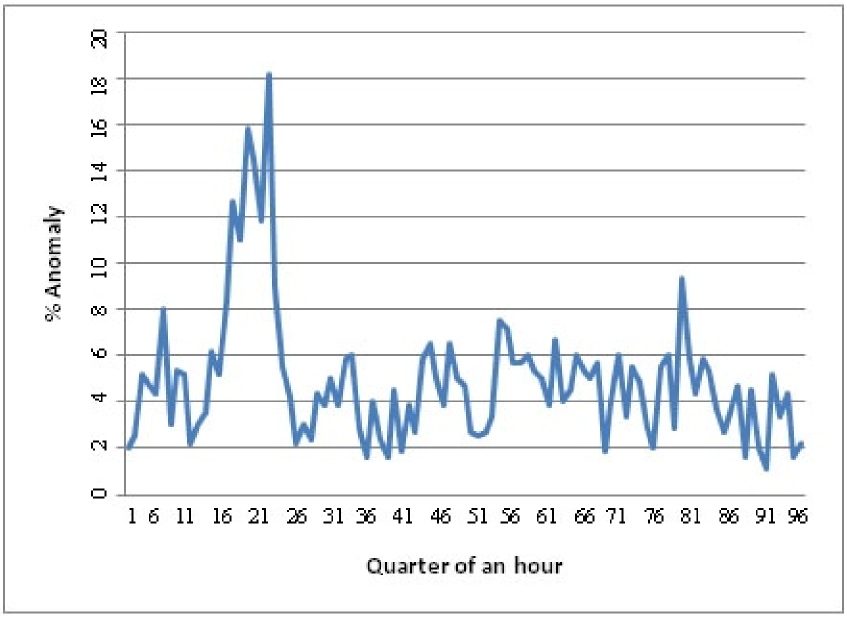

Figure 7. Anomaly rate based of neural network

Another anomaly detection method used was the local outlier factor (LOF) [15]. For any given data instance, the LOF score is equal to the ratio of average local density of k nearest neighbors of the instance and the local density of the data instance itself. If *k-distance*(A) is the distance of object A to k, the set of k nearest neighbors is $N_k(A)$. This distance was used to define the reachability distance:

$$reachability - distance_k(A, B) = max\{k - distance(B), d(A, B)\}$$

The reachability distance of object A from B is the true distance of two objects at least the k-distance from B. Objects that belong to the k of B are considered to be equally distant. Note that this is not a distance in the mathematical definition, since it is not symmetric. The local reachability density of A is defined by:

$$lrd(A) = 1 \Bigg/ \left(\frac{\sum_{B \in N_k(A)} reachability - distance_k(A, B)}{|N_k(A)|} \right) \tag{5}$$

The local reachability density is then compared with its neighbors is:

$$LOF_k(A) = \frac{\sum_{B \in N_k(A)} \frac{lrd(B)}{lrd(A)}}{|N_k(A)|} \tag{6}$$

The anomaly rate calculated using this method was less than 4% and at times equaled zero. At abnormal times, the anomaly rate was 12%.

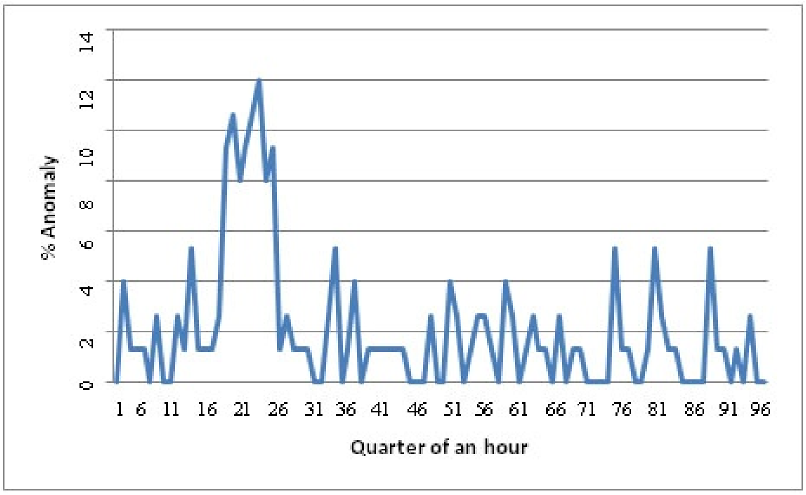

Figure 8. Anomaly rate based on local outlier factor

Parzen window estimation is the subset of statistical methods that detects anomalies. This technique is based on the probability density function (PDF). Desforges et al. [16] proposed a semi-supervised statistical technique to detect anomalies that uses kernel functions to estimate the PDF for normal instances. A new instance that lies in the low probability area of this PDF is declared to be anomalous. For this test, *p*(*x*) was the density function to be estimated. The Parzen-window estimate of *p*(*x*) is:

$$\hat{p}(x) = \frac{1}{n}\sum_{i=1}^{n}\delta_i(x - x_i) \tag{7}$$

where δ_n is the kernel function with localized support whose exact form depends on *n*. Gaussian kernel functions are mainly used in the Parzen windows estimations as a kernel function. Thus *p*(*x*) can be expressed as a mixture of radially symmetrical Gaussian kernels with common variance σ^2:

$$\hat{p}(x) = \frac{1}{n(2\pi)^{d/2}\sigma^d}\sum_{i=1}^{n}\exp\left\{-\frac{\|x - x_i\|^2}{2\sigma^2}\right\} \tag{8}$$

where *d* is the dimensional feature space.

The anomaly rate based on this method is shown in Figure 8. The average anomaly rate was approximately 15% in the normal state of the network, which is high compared to other methods. The abnormal state of the network produced a 30% anomaly rate.

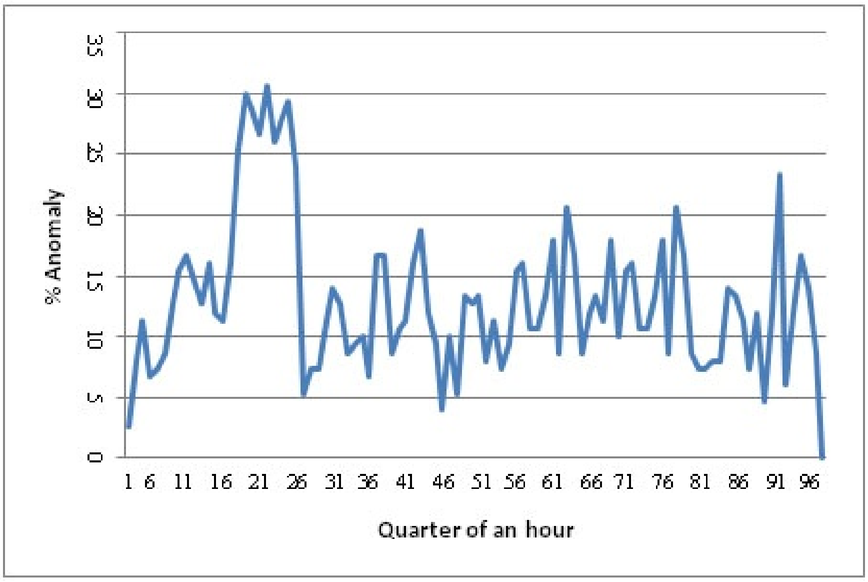

Figure 9.Anomaly rate based on Parzen window

In this study, classification combiners were used to combine the results of the anomaly detection algorithms. The majority vote method of combining the classification results will give an accurate class label if at least ⌊L/2⌋+1 classifiers give correct answers, where L is the number of classifiers. The classifiers used in this study were anomaly detection algorithms. Figure 9 shows the result of the combination of classifiers.

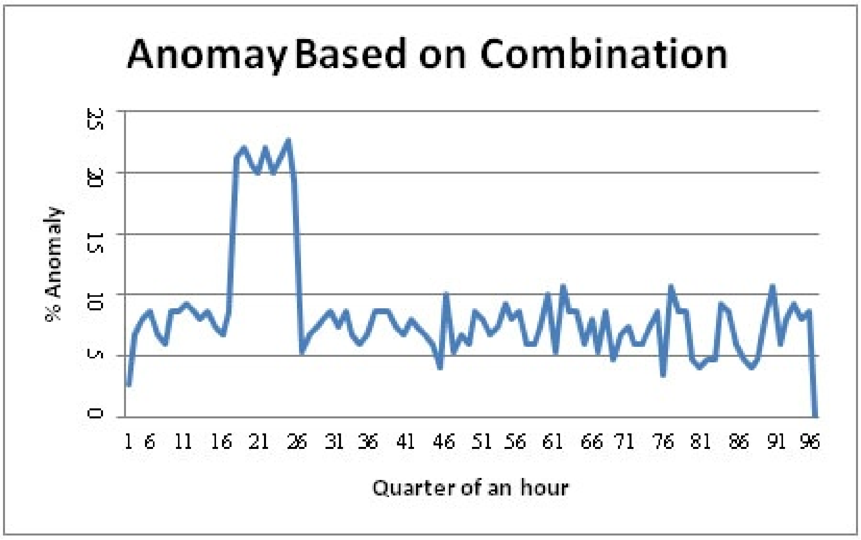

Figure 10. Anomaly rate based on combination of results

As seen, the anomalies totaled approximately 8% in normal times and 21% in the abnormal times of the network. Figure 10 shows that when anomaly algorithms were viewed individually, changes in the anomalies were high. After using a classification combiner, the changes were low, approximately 8%.

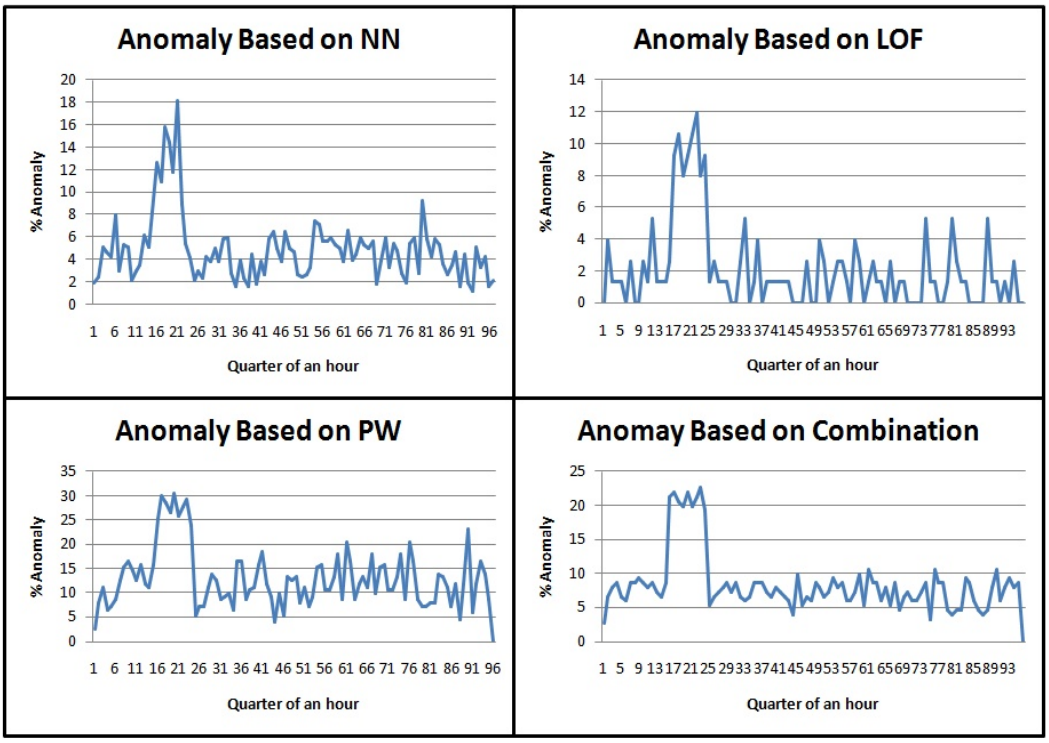

Figure 11. Comparison of anomaly detection algorithms

Anomaly rate is only one aspect of the real threat to the network. Real threat is calculated by multiplying the anomaly rate by the sum of the severity of successful IDS alerts. Because the DARPA network is different from the simulated network, application of a verification algorithm will be unsuccessful. For the sake of the test, it was assumed that all alerts were successful. Figure 11 shows the real threat based on the alerts and anomaly rate.

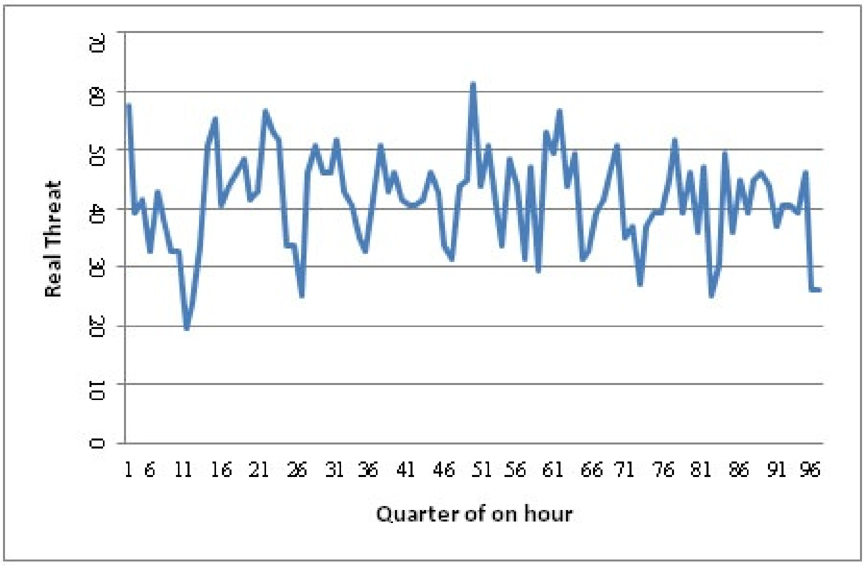

Figure 12.Real threats

Although Figure 11 provides awareness about threats to the network, relative threat can provide additional awareness. The maximum threat is the sum of the severity of vulnerabilities. Figure 12 shows the relative threat and Figure 13 shows the percentage of relative threat.

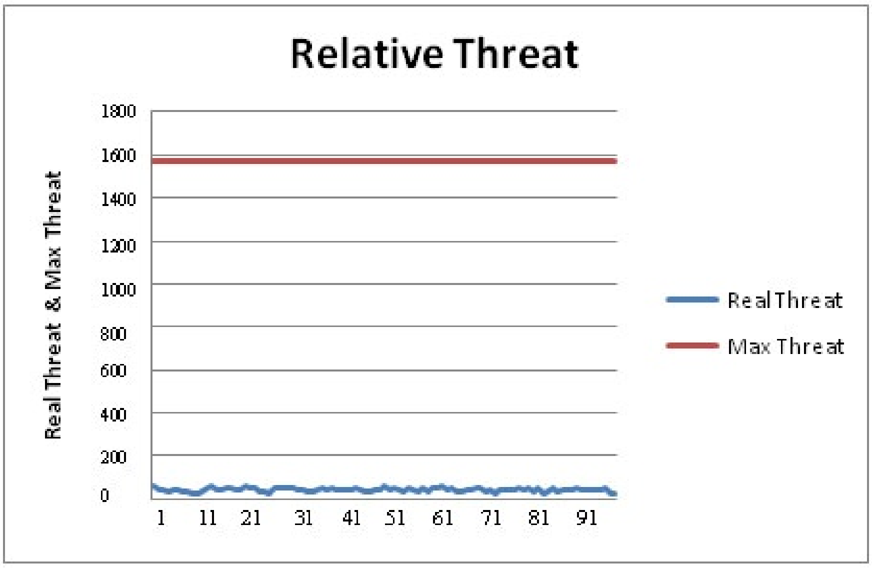

Figure 13. Relative threat

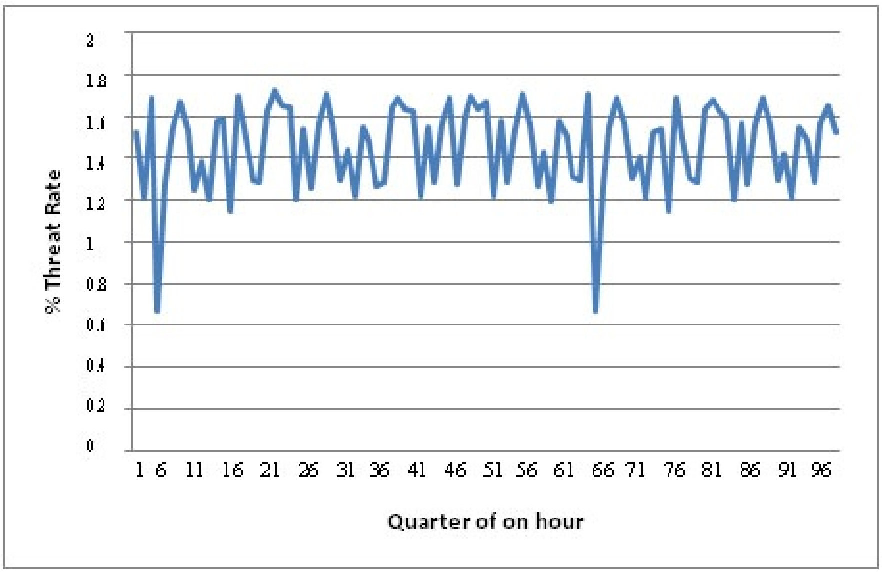

Figure 14. Threat rate

As seen, the relative threat of the network was very low; the average relative threat is approximately 1.5%. This is because the dataset used in the experiment was old and the severity of alerts was low. In addition, much of the vulnerability discovered by vulnerability scanners showed high severity.

4. Conclusion and Future Works

Research in threat assessment has traditionally focused on the development of methods, tools, and standards. Threat modeling is no longer adequate to assess threat in today’s increasingly complex environment. This study presents an approach for real time network threat assessment that determines the threat rate of a network as the combination of the anomalies of parameters, IDS alerts, and vulnerabilities. It provides a precise and fine-grained model for situational awareness and threat assessment.

Three anomaly detection algorithms were used and combined using the majority vote method. The anomaly rate of the parameters was multiplied by the sum of the severity of successful alerts from successful attacks. The maximum threat was the sum of the severity of vulnerabilities; the relative threat is the real threat divided into the maximum threat.

Process refinement might be applied to our model to reduce the overhead of the model. The parameter of network monitoring was selected staticallyfor this study. The approach should be applied to assess all parameters and assign a weight to each parameter or remove unnecessary parameters.

Three methods were used to detect anomalies in network parameters. Each method showed different rates of anomalies. Other anomaly detection methods can be used to improve the results. A comparison of the results identifies the best algorithms.

The proposed model can be used for risk assessment in computer networks. Each parameter, alert, and vulnerability is associated with an asset. The relative threat of assets can be considered to be the probability of threat and multiplying these values specifies their risk.

5. References

[1] International Standard ISO/IEC 27001. Information technology, Security techniques, Information security management systems, Requirements, 2005.

[2] Jangam E. Threat Modeling and its usage in mitigation security threats in an application,http://isea.nitk.ac.in/publications/ThreatModeling.pdf; Thesis Submitted in partial fulfillment of the requirements for the degree of master of technology, 2009.

[3] U. Franke, J. Brynielsson, “Cyber situational awareness – A systematic review of the literature,” Computers & Security, Volume 46, October 2014.

[4] Ma J, Li Z, Zhang H. An Fusion Model for Network Threat Identification and Risk Assessment, International Conference onArtificial Intelligence and Computational Intelligence, AICI '09, 2009.

[5] Xiaowu L, Jiguo Y, MaoLi W. Network Security Situation Generation and Evaluation Based on Heterogeneous Sensor Fusion,. 5th International Conference onWireless Communications, Networkingand Mobile Computing, WiCom '09, 2009.

[6] Beaver JM, Kerekes RA, Treadwell JN. An information fusion framework for threat assessment, InformationFusion, 12th International Conference onInformationFusion,FUSION '09, 2009.

[7] J. Webb, A. Ahmad, S. B. Maynard, G. Shanks, “A situation awareness model for information security risk management,” Computers & Security, Volume 44, July 2014.

[8] Chen X, Zheng Q, Guan X, Lin C, Sun J. Multiple behavior information fusion based quantitative threat evaluation, International journal of computer & security, Volume 24, Issue 3, May 2005, Pages 218–231.

[9] Xi R, Yun X, Jin S, Zhang Y.Network Threat Assessment Based on Alert Verification, 12th International Conference onParallel and Distributed Computing, Applications and Technologies (PDCAT), 2011.

[10] Chen X, Li S, Ma Jin, Li J.Quantitative threat assessment of denial of service attacks on service availability, International Conference onComputer Science and Automation Engineering, CSAE 2011.

[11] Danapardaz company, “Network Monitoring concepts”, http://www.danapardaz.net/ws/Pages/fa/Bina/Article.aspx?ID=44

[12] Chandola V, Banerjee A, Kumar V. Anomaly Detection : A Survey, in: ACM Computing Surveys, 2009.

[13] Kuncheva LI. Combining Pattern Classifiers: Methods And Alghorithms, John Wiley & Sons, INC. Publication, 2004.

[14] Hawkins S, He H, Williams, GJ, Baxter RA. Outlier detection using replicator neural networks, in: Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery. Springer-Verlag, 170-180, 2002.

[15] Markus Breunig , Hans-Peter Kriegel , Raymond T. Ng , Jörg Sander, “LOF: Identifying Density-Based Local Outliers”, Proceeding of the 2000 ACM Sigmod International Conference on Management of Data.

[16] Chow, C. and Yeung, D.-Y. 2002. “Parzen-window network intrusion detectors” In Proceedings of the 16th International Conference on Pattern Recognition. Vol. 4. IEEE Computer Society, Washington, DC, USA, 40385.

[17] Lincoln Laboratory; DataSet; http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/ [accessed 12.23.14]

[18] PAESSLER, the network monitoring company; http://www.paessler.com/prtg [accessed 12.23.14]

[19] Snort, open source network intrusion prevention and detection system (IDS/IPS),https://www.snort.org/downloads/snort/snort-2.9.7.0.tar.gz [accessed 12.23.14]

[20] OSSEC, Open Source Host-based Intrusion Detection System, http://www.ossec.net/ [accessed 12.23.14]

[21] Tenable Network Security, Nessus vulnerability scanner, http://www.tenable.com/products/securitycenter-continuous-view/evaluate [accessed 12.23.14]

[22] Protector Plus, Windows Vulnerability Scanner, www.pspl.com/download/winvulscan.htm [accessed 12.23.14]

[23] VirtualBox, A powerful virtualization product, http://download.virtualbox.org/virtualbox/4.3.20/VirtualBox-4.3.20-96997-Win.exe [accessed 12.23.14]

[24] Lambert DA. Assessing situations, Information, in: Proceedings of Decision and Control, IDC 99, 1999.

[25] Jacquemin C,Jardino M. Multi-dimensional and Multi-scale Visualizer of Large XML Documents, the Eurographics Association, 2002.

[26] Xiong N, Svensson P. Multi-sensor management for information fusion: issues and approaches, International Journal of Information fusion, Volume 3, Issue 2, June 2002, Pages 163–186.

[27] Data Fusion Subpanel of the Joint Directories of Laboratories. Data fusion lexicon, in: technical panel for C3, United states of America Department of Defence, 1991.

[28] Steinberg AN, Bowman C, White F. Revision to JDL Data fusion Model, NATO/IRIS Conference, October 1998.

[29] Linas J, Bowman C, Revora G, Steinberg A, Waltz E, White F. Revision and extentions of JDL Data fusion model II, 7th International Conference on Information Fusion, pp. 1218-1230, 2004.

[30] Llinas J, Liggins ME, Hall DL.Handbook of Multisensor Data Fusion, Theory and Practice, Second Ed. 2008.