Research on Prediction Technique of Network Situation Awareness

Juan Wang，Zhi-Guang Qin，Li Ye
School of Computer Science and Engineering
University of Electronic Science and Technology
Chengdu, China
{wangjuan,qinzg}@uestc.edu.cn, forefell@gmail.com

***Abstract*—In this paper we study on the prediction technique of network situation awareness. It has two levels: the high-level situation and the low-level next attack step. The first one includes the indexes and the evaluation results of the network security situation, they are figure form, we use the RBF network to predict them for RBF's self-learning character. Then we use the weighted attack graph to predict the next attack step. The weights represent the probability; the biggest weights indicate the most possible next attack step. The simulations show these prediction methods can offer different prediction capability to satisfy the prediction need of the network situation awareness.**

***Keywords*—network situation awareness,RBF neural network, alert analysis**

I. INTRODUCTION

Now, we are in "information age", various information around us, but more information doesn't mean more knowledge. Because, this information is mixed up and out-of-order, people hard to find out useful knowledge they want, this called an "information gap" [1].

Then the Situation Awareness (SA) is proposed to solve this problem. SA pays attention to the information transform technology between the abstract data and the knowledge understood by person. It has three levels to transform the data to knowledge: perception, comprehension and prediction.

Its application in network security is called Network Situation Awareness (NSA), first proposed by Tim Bass, 1999[5]. Traditional network security devices such as Intrusion Detection Systems (IDS), firewalls, and security scanners work independently， with no knowledge of the network resource they are defending. This lack of information results in numerous ambiguities when interpreting alerts and making decisions on adequate responses.

We design a near-real-time monitoring system that converts NetFlow source data and network security equipment (e.g., IDS, Firework, IPS, etc) alerts to visualizations of individual and summary statistics, in graphic and tabular formats. This novel network monitoring system provides situational awareness of traffic activity by geographical entity (e.g., location, organization, city, state, etc), then auto-response by pre-configure. The system can decrease the response time of network decision-making and action. One important module of the system is prediction; include the situation prediction, indexes prediction and the attack step prediction.

Supported by National Computer Security Center Project of China (242 project)under Grant No.2006C27，and the China Scholarship Council.

Accurate prediction is the foundation of the early warning, the support of the decision-making. We apply normal prediction technique like RBF neural networks, to predict the situation and the indexes; and the Weighted Attack Graph (WAG, a directed graph) to predict the next attack step that can help network manager or the system itself change the network security strategy timely. The simulations show these prediction methods can offer different prediction capability to satisfy the prediction need of the network situation awareness.

The remainder of this paper is organized as follows: Section 2 contains a brief introduction of situational awareness and some related works. Section 3 gives a detailed description of prediction methods. Section 4 gives the simulation of our prediction methods of the NSA system. We end with a summary and conclusions in Section 5.

II. NETWORK SITUATIONAL AWARENESS

The term Situational Awareness (SA) comes from the military pilot field, it means the ability to identify, and comprehension the critical elements [2]. SA is the foundation of decision-making it can help people in many fields such as air traffic controllers, driving, and military operations [3].

Ensley [4] defined Situational awareness as "the perception of the elements in the environment with a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future" The following three levels constitute the formal definition of SA [2]:

- Level 1 – perception of the elements in the environment.
- Level 2 – comprehension of the current situation.
- Level 3 – projection of future status.

The mental model and the decision-making show in the Fig.1 describe the relationship of the three levels.

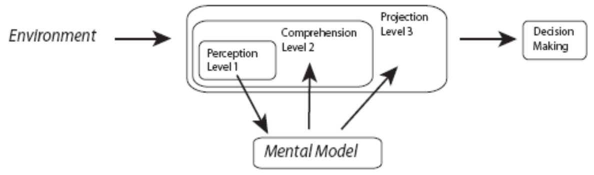

Figure 1. Situational Awareness

As mention above, Tim Bass bring it into the network security research field [5].He first use this NSA idea in

Intrusion Detection System (IDS) to fuse data that come from multi-sensor, and get the information of the rate of intrusion, attackers of intrusion, and the object of the attack, at last offer the evaluation function of attack threaten, it is also the situation awareness of attack to the user. But he didn't offer the prediction function. Other researchers and groups in this field include Stephen Lau (Lawrence Berkeley National Labs) [6], NetSA(The CERT Network Situational Awareness Group) of Carnegie-Mellon University[7], National Center for Advanced Secure Systems Research(NCASSR)[8]. These groups put the data fusion technique into practice, and use the visualization technique improves the idea of Tim Bass to get awareness of the network situation. These systems offer only the information of the net flow and also lack capability of prediction, or only predict the net flow.

III. PREDICTION OF NETWORK SITUATION AWARENESS

The comprehension process of our NSA system generates the following information:

a) The indexes get from the alerts of the network security devices such as SNORT [12]. And the netflow data of the network connected devices such as CISCO ROUTER.

b) The attack graphs get from alerts that are displayed on the user interface to help network manager easily locate the source and object of the attacks, and give manager a good idea of attack strategy.

The evaluation results of the network situation that are floating point numbers get from all of indexes(how to get we will report in another paper), The result represent the risk rank from the low (green) to the highest severe (red), the risk rank defined as [9].

The indexes and the evaluation values are figures, they have similar character, we use RBF neural network to predict both of them, for RBF's self-learning character and the complex relations of the indexes and evaluation results. For the attack graphs, we add weights to the attack graph to predict the next attack step. The weight represents the probability of which attack will be the next attack. The details of these methods are given below.

The predication process of the indexes is similar to the predication of evaluation values. We describe the latter one for example

A. RBF neural network based prediction of the situation of the NSA

Firstly, I simply introduce the indexes and the evaluation values. One index is a feature of network such as the rate of netflow (783m bit per hour) ,we define it as I_1. We totally have thirty-one indexes (I_1, I_2…I_{31}), all of indexes are processed to figure form. As time flies, we can get a time serial values for each index, the same to evaluation values which are figure from 0 to 1. We use past time values to train the RBF neural network, when it ready we use it to predict future values of both indexes and evaluation results.

Radial basis function (RBF) neural network are based on supervised learning. The input into an RBF network is nonlinear while the output is linear, due to the nonlinear approximation properties. RBF network are able to model complex mappings. The transformation functions used are based on a Gaussian distribution.

We use a three-layer n-h-m RBF network to predict, as the Fig.2.The n is the number of inputs, h is the number of hidden units, and the m is the number of outputs.

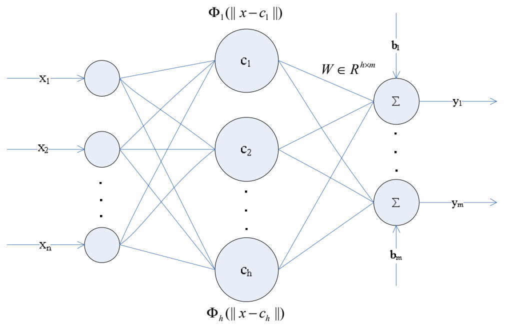

Figure 2. Three-layer RBF neural network

The $x=(x_1,x_2\cdots,x_n)^T \in R^n$ is the input feature vector, C_i(i=1,2,…,h) are the center of the hidden data, $W \in R^{h\times m}$ is the output matrix, b_0 ,…,b_m are the offset of output unit, $y=[y_1,\cdots,y_m]^T$ is the output of the network, $\Phi_i(\cdot)$ is the activation function of the *i*th hidden node. Before using the RBF networks to predict, we need optimize the network parameters in order to fit the network outputs to the given inputs. The fit is evaluated by means of a cost function, usually assumed to be the mean square error. We take the following training process:

The output of the *i*th hidden unit of the RBF network is:

$$q_i = \Phi(\| x - c_i \|) \tag{1}$$

.

The output of the *k*th output node is:

$$y_k = \sum_i \omega_{ki} q_i - \theta_k \tag{2}$$

Algorithm:

1) Use k as iterative times, initialize k=1.Then use test data $X=\{X_1,X_2\cdots,X_m\}$ to find the centre of the data, markedC_1^1, C_k^l delegate the centre of *l*th, *k*th cluster.
2) Suppose the sum of node i to each cluster center distance is A(i),and the distance of i to its cluster centre is B(i),then suppose $R_k = X_i, i \in \{i \mid \max(A(i)\times B(i)), i=1,\cdots,m\}$.
3) Eject R_k from its cluster, re-count that cluster center, and suppose new center is C_j
4) set k=k+1, that means increase the number of cluster from k to k=1, initialize the cluster center is $(R_{k-1}, C_1^{k-1}, \cdots, (C_j)', \cdots C_{k-1}^{k-1})$.
5) Calculate the distance of all nodes to the center in the set X, and divide the nodes to different cluster which center is the most close to them.

6) then re-count the center $C^k = (C_1^k, \cdots, C_k^k)$,judge whether C_k have changed or not, changed, then turn to step(5), otherwise continuing to next step;
7) calculate variance of all cluster: $D^k = (D_1^k, \cdots, D_k^k)$;
8) suppose the distance of R_k to its cluster center is $D_{\max}$,the variance of each cluster is D_{mean} ,if $D_{\max}$ <max(D_{mean}),then clustering is over, go to next step, if not go to step 2;

Now, clustering is over, each cluster center is fixed. Then use the least square method to find out each weigh value, then the prediction module is ready.

When we got a set of situation evaluation values, we take these values as the input of the RBF network, and then the RBF network will give us the values of subsequent m time.

B. Weighted Attack Graph based prediction of the next attack step

As mention above, in the comprehension process we get attack graph from the alerts, such as Fig.3 shows:

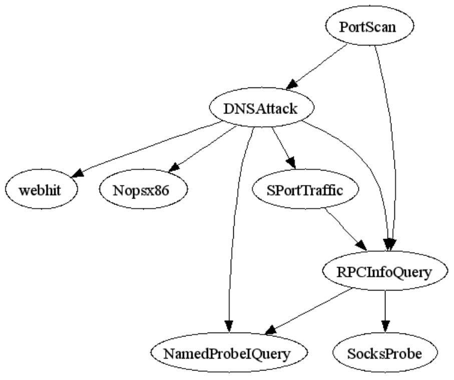

Figure 3. Attack graph

How do we get this graph is not the point in this paper, but you can get some idea from [10] which is the base of our method. The main idea is before the hackers intrude into victim, there are series attack steps, and early steps prepare for the later ones. So we can define the prerequisites and consequences of attack (by experience or generated by algorithm), then we connect the prerequisites and consequences produce the attack graph. We integrate the similar alerts into hyper-alert. One hyper-alert is a set of alerts which have the same prerequisites and consequences. The alerts in the attack graph are hyper-alerts.

But when we use the attack graph to predict next attack step, there is a problem that in an attack graph one attack may has many next steps, such as the node "DNSAttack" in Fig.3, the next step of it may be "webhit" or "SPortTraffic" or "Nopsx86".We don't know which one is the best choice. So in order to predict the next attack, we calculate the like hood of two direct relation attacks. We call the "two direct relation attack" as a "prepare-for pair", the former one contributes to the latter one. All of the prepare-for pairs and their weights (like hood) are stored in a chain table, as the Fig.4 shows:

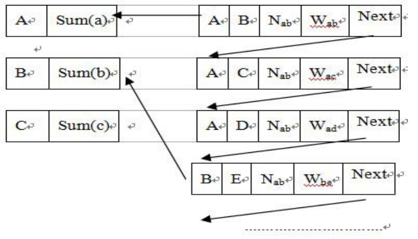

Figure 4. The weighted prepare-for pair chain tale

A, B, C, D, and E represent a kind of hyper-alert. For example, DNSAttack is the hyper-alert of dns-zone-transfer, DNS-version-query, and so on. All of them use DNS to get information of object, we think them have the same effect in an real attack so we use a hyper-alert DNSAttack to delegate all of them .Also the webhit is the hyper-alert of web-cgi-phf, web-cgi-test-cgi, web-cgi-handler, web-cgi-perl-exe, web-cgi-nph-test-cgi, and so on. The N_{ab} in the chain table represent how many times the pair (A, B) appearanced. When we observe a pair (A,B) from the alert stream, the N_{ab} update to N_{ab} +1.This information get from the alerts stream in training stage. The Sum (*) is sum of certain hyper-alert appearance numbers. The W is the weight of prepare-for pair .Its computation algorithm as follows:

Algorithm 1 prepare-for pair weight computation

```
Let i, j be a certain hyper-alert;
Let (i,j) be a prepare-for pair;
Let N (i, j) be the appearance numbers of pair (i,j);
Let W (i, j) be the weight of pair (i,j);
For all hyper-alert in chain table do
Compute the sum of hyper-alert's appearance numbers:
Sum (i) =∑N (i,*);
W (i, j) = N (i, j)/sum (i);
End for
```

After training stage, the weight will change when the prediction is hit or not. Its update algorithm as follows:

Algorithm 2 prepare-for pair weight update

```
Let (i,j) be a prepare-for pair;
Let N (i, j) be the appearance numbers of pair (i,j);
Let W (i, j) be the weight of pair (i,j);
Let Sum (i) is sum of i hyper-alert appearance times;
For all prepare-for pair do
If the prediction of pair (i, j) is hit then
Update the N (i, j) = N (i, j) +1;
Update the Sum (i) = Sum (i) +1;
Update the W (i, j) = N (i, j)/ Sum (i);
End if
End for
```

We don't update the weight by update all the N (i, j), Sum (i, j), and get W=N/Sum. If the alerts and pairs are thousands, it's a time- consume work, don't fit our near-real-time need. On the other side, one hit compare thousands pairs and alerts is too few, it cannot contribute to the weight change obviously, this also mean it cannot help to get new attack strategy quickly.

Lastly, the prediction of next attack algorithm as follows:

```
---------------------------------------------------------------------
Algorithm 3 attack prediction
---------------------------------------------------------------------
Let I be the attack just happened;
Let W (i, j) be the weight of pair (i, j);
For all W (i, *) in chain table do
  If the W (i, j) is the max one
  Return the attack j as the possible next attack
End if
End for
```

IV. SIMULATION

A. RBF neural network based prediction of the situation of the NSA.

Our simulated data are the net flow data of Chongqing (one city of China) offered by the Chinese National Network Security Centre (CNNSC), we transformed the data into indexes, and got the situation evaluation values (one hour get one value) from 2007-10-19 to 2007-10-25, total 7 days. We used 19-24 evaluation values as training data, and the last day as the test data. Our RBF network is 24-*h*-1 structure, has 24 inputs (one day has 24 hours), 1 output (the next hour).

We got the prediction values of 25 whole day. Be limited by space we show the former 7 hours values , the table 1 and the Fig.5 show the prediction values (P) , the real values (R),and error between them(E), E = P – R.

TABLE 1: PREDICTION VALUES AND REAL VALUES

P	0.2286	0.1862	0.1436	0.2154	0.1476	0.4468	0.2815
R	**0.2167**	**0.1834**	**0.1584**	**0.1917**	**0.1534**	**0.4751**	**0.2667**
E	0.0119	0.0028	-0.0148	0.0237	-0.0058	-0.0283	0.0148

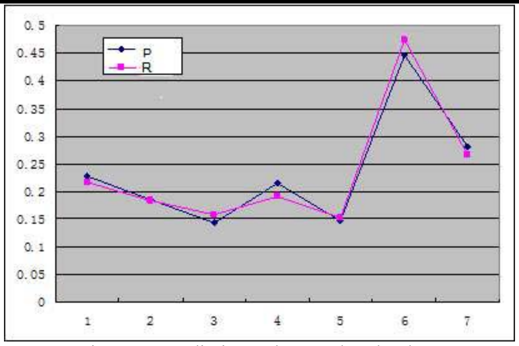

Figure 5. Prediction values and real values

The mean error of prediction is 0.0146, the prediction offset less than 5%, and the trend of the prediction is the same to the real one, the simulation shows our RBF network based prediction method of situation can meet the need.

B. Weighted Attack Graph based prediction of the next attack step

We use the Honeypot data [11] to prove the method, for the alerts of CNNSC aren't public. We use the 2000 April, May data to training our weight (like hood of prepare-for pair).And use the weighted attack graph to predict the attacks in Jun data. We get the attack graph Fig.3 from the 2000 April, now we calculate the weight of each edge, and then we get the weighted attack graph as Fig.6 show.

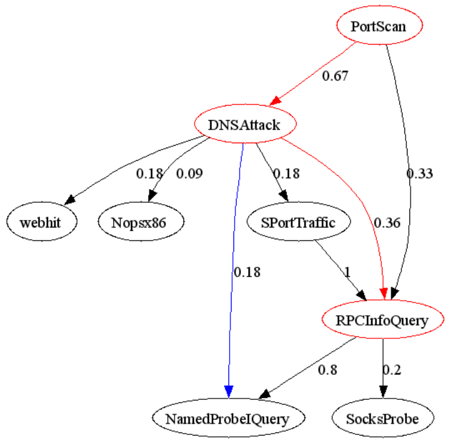

Figure 6 . The weighted attack graph

We can get the information that the intrusion begin with "port-scan", for hackers have to find who is online first. Then the "DNSAttack" follows, the result reveals that the victim host has the DNS exploits bug which can be used intrude into the victim. Let's see following alert of Jun 6:

Jun 6 22:58:56 lisa snort [8804]: IDS198/SYN FIN Scan: 194.247.87.235:53 -> 172.16.1.101

There was a port-scan attack happened。Our weighted attack graph shows the next attack may be DNSAttack or RPCInfoQuery, the former one happen probability larger then the latter one. Actually, the DNSAttack happen latter, named DNS-version-query (Fig.6.red line from portscan).As mentions above,we integrate the similar alerts into hyper-alert. DNS-version-query belongs to hyper-alert DNSAttack. And the weight (port-scan, DNSAttack) update to 0.77. The alert is this:

Jun 6 22:59:12 lisa snort [8804]: IDS/DNS-version-query:194.247.87.235:3722->172.16.1.107

The continuous 4 days (Jun 6-9) prediction as the table 2 shows. We can see the intrusion start with a port-scan, then follows a DNSAttack, and a NamedProbeIQueery.We make some wrong prediction of NamedProbeIQueery(from the "DNSAttack" node, the follow red line are our prediction, blue one is the real one).But when the weight update, we can make correct prediction later. In training data the weight of (DNSAttack, RPCInfoQuery) is too high. Actually, before the Jun 6 the attack came from 194.247.0.1 network, and after Jun 6 the attack came from 211.62.0.1 network, they were two different attacks. Although our prediction method makes some mistake, the method can predict rightly after the weight changed. Our method has adaptability

TABLE 2. FOUR DAYS PREDICTION

Day	Happened attack	Prediction attack	Real next attack	Update weight(0.1,0.05)
6	Port-scan	DNSAttack	DNSAttack	(Port-scan,DNSAttack,0.77)
6	DNSAttack	RPCInfoQuery	NamedProbeIQueery	(DNSAttack, NamedProbeIQueery,0.23)

				(DNSAttack, RPCInfoQuery,0.31)
7	DNSAttack	RPCInfoQuery	NamedProbeIQueery	(DNSAttack, NamedProbeIQueery,0.28) (DNSAttack, RPCInfoQuery,0.26)
8	DNSAttack	NamedProbeIQueery	NamedProbeIQueery	(DNSAttack, NamedProbeIQueery,0.38)
9	DNSAttack	NamedProbeIQueery	NamedProbeIQueery	(DNSAttack, NamedProbeIQueery,0.48)

V. CONCLUSION

The simulations show our prediction method can predict the high-level situation and the low level next attack step. Our future research are training the RBF network as set the prediction time unit from an hour to a day ,a week and a month, to see the algorithm's performance, and improve the weighted attack graph by considering some other information such as source IP.

ACKNOWLEDGMENT

The work reported in this paper was supported in part by National Computer Security Center Project of China (242 projects) under Grant No.2006C27, named "The Network Situation Awareness and Analysis System" The organization give us direction and experimental data. Thanks them for their help. And also thanks to the support of China Scholarship Council.

REFERENCES

[1] M. R. Endsley (2001). Designing for situation awareness in complex systems. Proceedings of the Second intenational workshop on symbiosis of humans, artifacts and environment, Kyoto, Japan

[2] M. R. Endsley, B. Bolt´e, and D.G. Jones. Designing .For Situational Awareness, An Approach to User-Centered Design. Taylor & Francis, 2003.

[3] Y. Livnat,J. Agutter,S. Moon and S. Foresti.Visual Correlation for Situational Awareness.IEEE Symposium on Information Visualization 2005.

[4] M. R. Endsley. Design and evaluation for situation awareness enhancement. In Proceedings of the Human Factors Society 32nd Annual Meeting, pages 97–101, Santa Monica, CA, 1988. Human Factor Society.

[5] T.Bass, D. Gruber .A glimpse into the future of id.http://www.usenix.org/publications/login/1999-9/features/future. html, 1999

[6] S. Lau.The spinning cube of potential doom.Communications of the ACM,2004,47(6):25 26

[7] Carnegie Mellon' s SEI. System for Internet Level Knowledge (SILK). http://silktools.source forge.net,2005

[8] W .Yurcik. Visualizing NetFlows for Security at Line Speed:The SIFT Tool Suite.In:19th Usenix Large Installation System Administration Conference(LISA),San Diego,CA USA,Dec.2005

[9] http://state.ak.us/admin/info/security/ThreatIndicatorSOA.shtml

[10] P. Ning and D. Xu. Learning attack stratagies from intrusion alerts. TR-2003-16, Dept. of Comp. Sci., NCSU, 2003.

[11] Honey net project know your enemy statistics[EB/OL].2001-07-22.http : //www.HoneyNet.org/papers/stats/.

[12] SNORT. http://www.snort.org/