Ontology Based Cooperative Intrusion Detection System *

Yanxiang He, Wei Chen, Min Yang and Wenling Peng

Computer School, State Key Lab of Software Engineering
Wuhan University, Wuhan 430072, China
{yxhe,chenwei,myang,wlpeng}@whu.edu.cn

Abstract. As malicious intrusions span sites more frequently, network security plays the vital role in internet. Intrusion detection system(IDS) is expected to provide powerful protection against malicious behaviors. However, high false negative and false positive prevent intrusion detection system from practically using. After survey of present intrusion detection systems, we believe more accurate and efficient detection result can be obtained by using multi-sensor cooperative detection. To aiding cooperative detection, an ontology consisting of attribute nodes and value nodes is presented after analysis of IDSs rules and various classes of computer intrusions. On the basis of ontology, a matchmaking method is given to improve flexibility of detection. Cooperative detection framework based on the ontology is also discussed. The ontology proposed in paper has two advantages. First, it makes the detection more flexible and second it provides global locality information to support cooperation.

1 Introduction

Since intrusion detection was introduced in the mid-1980s [1], intrusion detection system(IDS) has developed for almost thirty years to enhance computer security. However high false negative and false positive prevent ID system from practically using. After analyzing the reason of high false alarms rate, we think the inefficient detection is partly caused by insufficient audit data sources and lack of cooperating multi-sensors data. Many of IDSs depend on only one kind of sources: network data or host based data. However many intrusions can shows character in both of these two data sources. If more sensors data can be utilized to perform intrusion detection, the alert will be more accurate.

The key problem is how to correlate multi-sensor information to evaluating the security state of monitored system. In this paper, we argue a cooperating detection framework based on ontology. Ontology can provide detection system with the ability to share a common conceptual understanding and provide relationships among heterogeneous audit data. Based on ontology we present our cooperative framework to correlate the information from multi-sensor. A flexible and efficient matching algorithm is also given to perform detection.

* Supported by the National Natural Science Foundation of China under Grant No. 90104005

The remainder of the paper is organized as follows: Section 2 presents related work in cooperative detection by using multi-sensor. In Section 3 we present the ontology established from host and network data feature, give the matchmaking algorithm and the ontology based cooperating function. Some experiment results will be present in Section 4. In Section 5, we conclude our work and discuss future research.

2 Relate Work

Some of IDSs have used both of host and network data to perform detection. DIDS[3] accepts the notable event records from each of the host and LAN monitors and sends them to the expert system. The expert system is responsible for evaluating and reporting on the security state of the monitored system. The detection model is the basis of the rule base and consists of 6 layers, each layer representing the result of a transformation performed on the data. EMERALD[4] is a scalable distributed intrusion detection system that operates on three different levels in a large enterprise network. EMERALD introduces a recursive framework for coordinate analyses from distributed monitors to provide a global detection. However neither of these two systems addresses data sharing between host and network data.

In [5] Peng addresses that most intrusions are not isolated but related as different stages of attack sequence, with the early stages preparing for the later ones. Proposed approach correlated alerts using prerequisites of intrusions. It can discover high-level intrusion scenario and reduce the impact of false alert. But the detection rate is constrained by the low-level IDSs. Frincke in[6] proposes principles for a framework to support cooperative intrusion detection across multiple domains and describes a prototype cooperative data sharing system illustrating many of those principles.

An ontology is an explicit specification of the concepts and relationships and is widely used in many research areas, such as semantic web, knowledge management, AI etc. There are not much literature reports about applying ontology to IDS. In [7] Pinkston gives a target-centric ontology to descript the concepts within ID domain and relations between them. They implement their ontology model by DAML+OIL language. But they do not seem to make full use of correlating and inheritance relationship to perform detection.

3 The Ontology Based Cooperative Detection

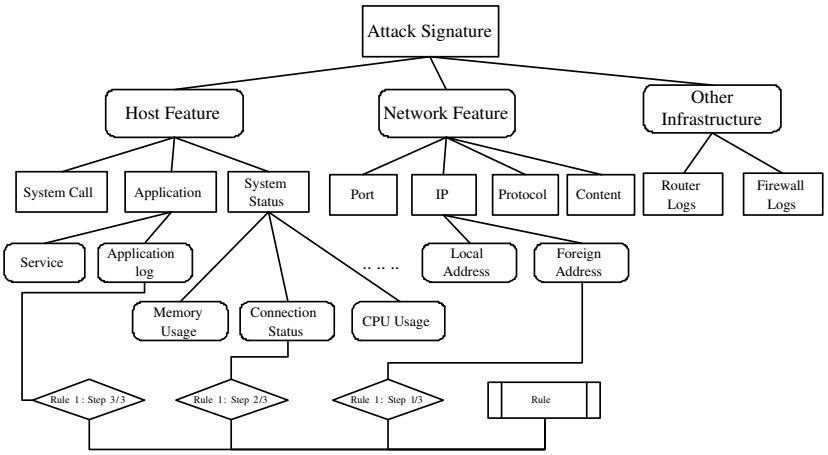
3.1 Ontology

The term ontology means a specification of a conceptualization. An ontology is a description (like a formal specification of a program) of the concepts and relationships in a specific domain. Using ontology in intrusion detection system is to provide powerful constructs that can guide cooperating detector to exchange

machine interpretable message. Understanding other cooperating detectors' message and description of current status is vital in cooperating work. We design an ontology after analysis some IDSs rules and the security vulnerabilities published by Common Vulnerabilities and Exposures (CVE)[8]. Compare to the ontology provided by [7], our ontology focuses on the features that can be observed by each sensor. We not intend to estimate the motivation of the intruder as [7] have done.

The complete ontology includes two kinds of nodes: value nodes and attribute nodes. Attribute nodes describe all the features that can be observed by multi-sensor and value nodes are the children of some attribute nodes which represent the possible value of their parent attribute node. Fig. 1 illustrates a part of our ontology which only has attribute nodes. The complete ontology not is given because it would make the illustration clumsy. At the root of the ontology is attack signature. The subclass of root is attribute nodes that represent different features from heterogeneous sensors including host sensors, network sensors, router logs etc. The higher level node of ontology means more abstract feature and is the locality of the lower level nodes. By the ontology, we can know on which sensor we can find concerned information. For example, if we require the memory total usage information. We can learn from the ontology that, this value can be obtained from the system status sensor on the host. This is useful in cooperative detection process because it help us to locate the required information.

Fig. 1. The Overview of Ontology

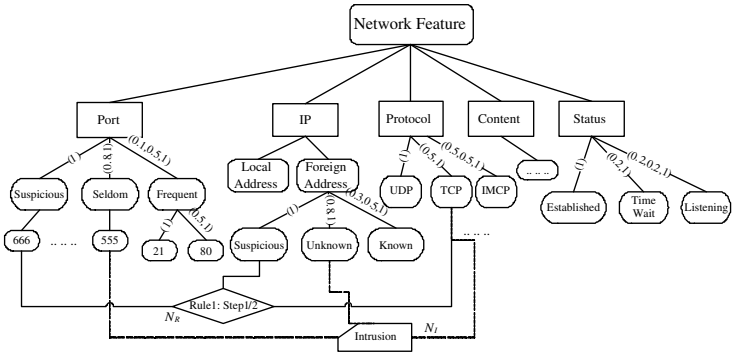


3.2 Ontology based Matching

Signature intrusion detection system often uses string matching or more powerful expert system to perform matchmaking. String matching system, such as snort [9,10], is a simple substring matching of the characters in the text. Such a mechanism of course is not of considerable flexibility. If attack signature changes a little, the system will neglect it. For example, a backdoor server uses port '666' to connect with client. If the server changes communication port to '555', the string matching system will not detect it until another rule added to rules base. The expert system are more powerful and flexible, however execution efficiency is influenced by more complex mechanism. In our approach we give matchmaking method based on above ontology that is a little similar to the powerful expert system but has high execution efficiency.

Each node in the above ontology (in Fig. 1) is feature nodes that show the attributes of collection information. In the complete ontology will also include the value nodes which show the value distribution. The parent of value nodes is the attribute that has these values. If the attribute has a continuous value, some discrete preprocessing should be taken.In the above given example, the intruder changes backdoor program communication port form suspicious '666' to a seldom used port '555' and establishes the connection from an unknown foreign address other than the known suspicious one. To illustrate this scenario, a rule node 'Rule1: Step 1/2' and the observed suspicious data node 'intrusion' are attached to some value nodes in the ontology shown in Fig. 2. 'Step 1/2' suggests that this rule matchmaking has two steps and here it is the first step and it will be illuminated in detail in the next section. Although the observed phenomena is not exactly matching the rule , we know it is still the same backdoor intrusion.

Fig. 2. Ontology based Matchmaking



To evaluate the relationship and similarities among the rules and data, each edge between value node and its parent node is assigned a weight. The weigh

from 0 to 1 shows the relationship among the different values that has the same parent node. The 1 means the two values nodes has the maximum similarity while 0 means the minimum. For example, the 'Port' node has the values of 'Suspicious', 'Seldom used', 'Frequent used' and we store the weight of node N by vector $V_N(w_1, w_2, ..w_n)$. w_i means the weight between N and its brother node i. In above example, the three vectors are $V_{\text{suspicious}}(1)$, $V_{\text{seldom}}(0.8, 1)$, $V_{\text{frequent}}(0.1, 0.5, 1)$ and 0.8 in $V_{\text{seldom}}(0.8, 1)$ means the weight between 'Suspicious' and 'Seldom used' is 0.8.The weight is empirically assigned by expert and will be adjusted according to feedback of result. The autonomous adaptive weight assigning method has more advantage [11], which will be our future work. The weight between 'Suspicious' and 'Seldom Used' is 0.8 because a seldom used port is always utilized by intruder to perform malicious communication. However, the weight between 'Suspicious' and 'Frequent used' is only 0.1 as the frequent used port does not seem used by intruder.

There exists a set of paths, marked as $path\_set(N_R, N_I)$, that from node 'Rule'(N_R)to 'Intrusion'(N_I) which will be utilized to evaluate the similarity between two nodes. It is not necessary to traverse all paths that between N_R and N_I and only path between two attached nodes that have the same attribute parent node are considered. In the above example, only three paths, path from '666' to '555', path from 'Suspicious' to 'Unknown' and path from 'TCP' to 'TCP', are considered. If N_R and N_I are attached to the same node in one path, the detector will give the maximum score 1. If not, the similarity score depends on the path weight between the two nodes N_i, N_j which they are attached to separately. Then the total score is calculated by sum up all the path similarity score.

$$\text{path\_simi\_score}_{\text{path}(i,j)} = V_i(w_1, w_2, ..w_n) * (0_1, 0_2, ..e_j.., 0_n)' \quad (e_j = 1) \tag{1}$$

$$\text{total\_score}(N_R, N_I) = \frac{\sum_{path(i,j) \in path\_set(N_R, N_I)} \text{path\_simi\_score}_{\text{path}(i,j)}}{\text{Count}(\text{pathset}(N_R, N_I))} \tag{2}$$

The total score in the above example is:

$$\text{total\_score} = \frac{((0.8, 1) * (1, 0)' + (0.8, 1) * (1, 0)' + 1)}{3} = 0.8667$$

If the total_score exceeds the threshold, it means the observed data matches the rule and is suspicious intrusion. During the intrusion detection process, lots of rules will be loaded and attached to ontology. The similarity score of each audit data toward the relevant rules is evaluated by total_score function. To make the matchmaking more efficient, we do not need to calculate all total_score between audit data node N_{data} and each rules. One prior fact is that if the total_score can exceed the threshold, at least one path_simi_score belong to the same path set exceeds the threshold. By utilizing this priori fact we filter the rules before calculating the total_score function which influence the execution speed mostly. The matchmaking algorithm is given as below.

```
Boolean:IntrusionMatchMaking(Var RuleSet:Rst; AuditData Node:Ndata)
 {
  Queue: Q;
// AttachedNodeSet function gives all the nodes that Ndata is attached to
  For each node N in AttachedNodeSet(Ndata)
    FOR each rule R in Rst
     IF path_simi_score path(N,Rj) > Threshold THEN
     insert(Q, R) ;  // If the path_simi_score exceed the threshold,
     The rule R will be insert to queue
        continue ;     // Quit inner loop
    END FOR
  END FOR
//Calculate the total score of the Ndata and rule in Q
   FOR each R in Q
     IF total_score(Ndata,R) > Threshold THEN RETURN TRUE;
   END FOR
  RETURN FALSE;
 }
```

3.3 Apply Ontology to Cooperating Detection

The relationship between value nodes and attribute node can be applied to more accurate and efficient matchmaking method for signature detection. At the same time, the relationship between the attribute nodes provides fundamental information to perform cooperating detection. The parent node indicates the locality of the children nodes. For example, if the memory total usage information is wanted during cooperating detection process, the ontology tells the detector to obtain this information from the system status sensor on the host. Ontology provides the machine interpretable knowledge in cooperating detection process.

To give more accurate alert, detectors try to collect as more information as possible before drawing the conclusion. Rules become more complicate when signatures information in rule may distribute on different sensors. For example, a complete rule in Fig. 1 is composed by three sub-rules. When detector finishes first detection on network sensors, it will examine connection status on host. The ontology indicates the detector where to get the desired information, and then the detector will send a query request to system status monitor on host. Before beginning the third step, the detector again learns from the ontology about the locality of application logs. This scenario is common in cooperating detection. If the backdoor intruder communicates with the backdoor server in encrypted commands, the network sensor can only detect some connection using strange port and unable to decrypt the content of connection. So the decrypted commands can only be obtained from application logs on host.

Facing the multi-step rule, the detectors perform sub-rule detections individually and the detection result is stored in local database temporarily. Each sub-alert has a TTL (Time to Live) tag. When the sub-alert expires, it will be deleted from database. When the last step sub-rule detection is finished, the

detector cooperates and queries each sub-rule detection result on relevant local database.

4 Experiment

An experiment is designed to test our approach. The edges and nodes of the ontology are stored in relational database. Because of the infrastructure limitation, we only implement cooperating detection among the host and network. A WinPcap based tool was used to collect network packets and Strace for NT[12] to trace NT system call on Windows 2000 Server operation system. Some host logs, such as memory usage and network connecting status are also utilized in detection. According to the rules of the snort and CVE, we design 21 rules which are mostly R2L(Remote to Local) and U2R(User to Root) detection rules because R2L and U2R intrusion is not easy to detect while our cooperating approach is suitable to detection these intrusion by using both host and network audit data. Most of these rules are two-step rule including sub-rule on network and host.Then we ran intrusion tool Netbus to evaluate our intrusion detection prototype. During the simulation we change the communication port of Netbus to test our ontology-based matchmaking algorithm. The default port for Netbus is 20034 and we change the default port to 20038 and 80 separately. We also ran snort 2.1.1 to detect simulated intrusion and compared the detection rate with our prototype

Table 1. Detection Results Comparison between Snort and Our Prototype

Netbus	Detected by Snort	Detected by Prototype	False Alerts by Snort	False Alerts by Prototype
Default Port	Y	Y	38	12
Port:20038	N	Y		
Port:80	N	N		

Table 1 compares the detection result and false alerts between snort and our prototype. The experiment result shows the superiority of our prototype over snort. From the result in the second line, we can find our approach is more flexible than Snort because ontology-based matchmaking algorithm is employed in our approach. However, if the attack has been changed too much, our method shows its limitation. Because of the co-operation of network and host detector, the false alarm rate decreases in our method. More complicated experiments are on designing, such as intruder's communication in encrypted commands and new intrusion evolved from old one. We expect more conspicuous improvement shown by our prototype.

5 Conclusion and Ongoing Efforts

In this paper, we present an ontology to describe relationship among features observed by multi-senor. There exist two kinds of nodes in ontology: value nodes and attribute nodes. By assigning the weight to the edge between values nodes and their parent attributed node, we provide a more flexible matchmaking method for intrusion detection. At the same time, the relationship between attribute nodes and their parent can indicate the locality of desired information. An ontology based cooperative detection function is also given in the paper.

We will employ more sensors to perform multi-sensor cooperating detection in our prototype and more complicated experiment will be designed to evaluate our approach. Our future work will focus on the improvement of ontology in intrusion detection domain and the ontology based matchmaking and cooperation method. In our future work, we also intend to design autonomous adaptive method to adjust the weights by feedback from detect result.

References

1. D.E. Denning, An Intrusion Detection Model, IEEE Transactions on Software Engineering, vol. SE-13, pp. 222-232, February 1987.
2. Stefan Axelsson. Intrusion Detection Systems: A survey and Taxonomy. Technical Report 99-15, Dept. of Computer Engineering, Chalmers University of Technology, Sweden, March2000
3. Steven R Snapp, Stephen E. Smaha, Daniel M Teal, and Tim Grance. The DIDS (distributed intrusion detection system) prototype. In Proceedings of the Summer USENIX Conference, pages 227-233, San Antonio, Texas, 8-12 June 1992.
4. Philip A Porras and Peter G Neumann. EMERALD: Event monitoring enabling responses to anomalous live disturbances. In Proceedings of the 20th National Information Systems Security Conference, pages 353-365, Baltimore, Maryland, USA, 7-10 October 1997.
5. Peng Ning, Correlating Alerts Using Prerequisites of Intrusions. Department of Computer Science, NC State University. http://www.mts.jhu.edu/ marchette/ID04/ Papers/CorrelationModel.pdf
6. Deborah Frincke, Don Tobin etc. A Framework for Cooperative Intrusion Detection. Proceedings of the 21 st National Information Systems Security Conference, pp. 361-373, October 1998
7. John Pinkston, Jeffrey Undercoffer etc. A Target-Centric Ontology for Intrusion Detection . University of Maryland, Baltimore County Department of Computer Science and Electrical Engineering
8. CVE, Common Vulnerabilities and Exposures ,http://www.cve.mitre.org/
9. Snort. Open Source Network Intrusion Detection System. http://www.snort.org
10. M. Roesch, Snort - lightweight intrusion detection for networks, 13th Administration Conference, LISA'99, Seattle, WA, Nov 1999
11. W. Lee, S.J. Stolfo, and K. Mok. A data mining framework for adaptive intrusion detection. the Proceedings of the 1999 IEEE Symposium on Security and Privacy, IEEE Computer Society Press.
12. Strace for NT.http://razor.bindview.com/tools/desc/strace