An Attack Modeling Based on Colored Petri Net*

ZHOU Shijie    QIN Zhiguang    ZHANG Feng    LIU Jinde
(School of Computer Science and Engineering, UESTC  Chengdu  610054  China)

Abstract  A color petri net (CPN) based attack modeling approach is addressed. Compared with graph-based modeling, CPN based attack model is flexible enough to model Internet intrusions, because of their static and dynamic features. The processes and rules of building CPN based attack model from attack tree are also presented. In order to evaluate the risk of intrusion, some cost elements are added to CPN based attack modeling. This extended model is useful in intrusion detection and risk evaluation. Experiences show that it is easy to exploit CPN based attack modeling approach to provide the controlling functions, such as intrusion response and intrusion defense. A case study given in this paper shows that CPN based attack model has many unique characters which attack tree model hasn’t.

Key words  petri net;  color petri net (CPN);  intrusion detection and response;  attack modeling

1  Attack Modeling

A typical attack contains the following elements:
1) Objects attacked. These objects belong to the victims or can be regarded as public resource, such as networking bandwidth.
2) Attacker. These objects contain the hacker’s information, attacking tools and other states of attacker.
3) Attack processes. The stages of an attack and attack processes are used to depict the attacking action.
4) Control actions. These actions can be classified into response actions and defensive actions. The former will be fired if the intrusion is detected by system. While the latter is the controlling flow used to prevent intrusion happening.

An ideal attack model should describe all these features. But currently no approach can completely attain this goal. Among the present attack modeling methods, attack tree is the most popular one. Attack trees can capture the steps of an attack and its interdependencies[3]. Attack trees are also used to represent and calculate probabilities, risks, cost or other weightings. The main building blocks of attack tree are nodes. Every node is used to model one step of an attack or one attacker’s action. Every tree has a single root node that represents the ultimate goal of an attack. Tree hierarchy models the temporal logics of stages and goals. In bottom-up attack tree, child node means attack stage has to be successfully performed before another step occurs. AND or OR logical gate can be set to each node. A node with OR gate occurs only when any of its child node occurs. For occurrence of a node with AND gate, all of its child events are necessary. Node can be set with a value or probability. This kind of attack trees is defined as weighted attack tree. Similarly, node with value can also be defined as valued node. In this way attack tree can model cost feature of an attack and can be used to perform risk assessment of information security.

But control actions cannot be modeled with attack trees. So there are some limits to the application of attack tree to intrusion response. Some extended model should be used to depict this important aspect of intrusion detection and response. Color petri net is such an approach.

2  Colored Petri Net Based Attack Models

Although CPN is a powerful modeling technique,
attacks can also be modeled by it. The definition of color petri net is addressed in this section firstly. Then the mapping from attack tree into CPN based attack model is analyzed. Some other extended features of this model are also expressed in this section.

2.1 From Attack Tree to CPN Attack Model

The CPN based attack model can be defined from attack tree to reduce the cost of modeling. It is because that some attack models have been built with attack trees \(^3\)[4]. To build a CPN based attack model from an attack tree, the mapping rules between them should be determined. The root node of attack tree is the result of an attack, and the leaf nodes are actions attacker exploiting to break into system. It is clearly that the root node of an attack tree can map to place of CPN and leaf nodes can map to transitions. The relationship among nodes could be regarded as the arc of CPN. The node value in attack trees is expressed as arc expression of CPN. And the logics of attack trees can map to event relationship of CPN.

2.1.1 Root Nodes Mapping

In attack trees, root node is the goal and result of attacks. In CPN attack models, the root node can map to place: node maps to a place, node inputs map to arcs of place. This kind of place is called Root Place. The OR gate and AND gate will map to the event relationship of CPN. Their maps are shown in Fig.1. The node with OR gate maps to event’s conflict relation of CPN. The node value in attack trees is expressed as arc expression of CPN. And the logics of attack trees can map to event relationship of CPN.

2.1.2 Leaf Nodes Mapping

Leaf nodes in attack trees are hacker’s actions breaking into the victim’s system. It is clearly that leaf nodes can map to the transition of CPN. But in an attack tree, all leaf nodes are connected directly. So it is difficult to do straightforward maps. Some intermediate states must be defined so that the mapping can be performed. Ruiu’s analysis of intrusion divided attacks into seven stages: Reconnaissance, Vulnerability Identification, Penetration, Control, Embedding, Data Extraction & Modification, and Attack Relay\(^5\). Each stage can also be divided into several sub-stages. So we can model attacks’ stages and sub-stages as intermediate states when translating attack trees into CPN based attack models. Fig.2 shows how to deal with such translation. These newly added places (including the places added during translation of intermediate nodes) are called Added Place. In Fig.2, the value of place \(p\) can be derived from a function \(f(t)\), where \(t \in T\), and \(f(t) \in \Sigma\). And the output arc of place \(p\) is the input arc of next transition. In Fig.2, the PS place is an Initialization Place whose means depend on the transition.

2.1.3 Intermediate Nodes Mapping

Intermediate nodes of attack trees are sub-actions or sub-goals of hackers. It is more difficult to translate these nodes into CPN models because intermediate nodes have not only input arc(s) and output arc(s), OR and AND gate logics, but also the same problems confronted in leaf node translating. Mapping rules of leaf nodes are listed as follows:

1) The intermediate node itself maps to transition, \(t\), of CPN;

2) Input arc of an intermediate node maps the input arc of \(t\), OR and AND gates are translated to
conflict relation and sequence relation respectively;

3) Intermediate place is added in the same way as the translations of leaf nodes.

By using above rules, the intermediate node can be mapped into CPN attack model. Fig.3 shows the mapping relations between them. It shows that the translations of a root node and a leaf node are mostly similar, other than that the latter has an output arc. But essentially, they are different from each other. In root node translation, the node itself maps to a place and there is no place added. But as to intermediate node translation, the node itself maps to a transition and a place is added to connect newly added place with the corresponding transition.

2.1.4 Temporal Logic Mapping

As to building template CPN from attack tree, an important issue is how to deal with temporal sequence of attacks. From above discussing, we know that an attack comprises many stages and sub-stages that all have temporal logical relations. One occurring sequence of many stages and sub-stages means an intrusion while all occurring sequences comprise the attack trees. Event relations of CPN can depict temporal logics in an attack tree. In fact, only sequence relation and conflict relation are used in CPN attack models. Although intermediate nodes have multiple output arc, concurrence relation may also be used. In this paper, concurrence relations are not considered. Fig.4 shows the typical temporal logics in CPN attack models.

2.1.5 Node Value Mapping

The node value in attack tree is used to perform risk assessment. This special feature extends its application scope and usage for quantifying intrusions. In CPN, there is no node value function, but it can be expressed by color value of place of CPN. During the translation of CPN model, after transition is fired, some value will be added to arc expression of this transition, and a color function maps each place, \( p \), to a type \( C(p) \) that expresses the node value. So each token must have a data value to evaluate risk.

It is easy to translate the weighted leaf node: node value maps to output arc expression of transition and the output expression of newly added place is also evaluated to this value. For intermediate node with value, if the node has OR gate, the value should be mapped to value of output arc of transition by mapping function \( f(x) = g(Y) + X \), where \( g(Y) = \{ y \mid y=Y_1 \land y \neq Y_2 \) or \( y=Y_2 \land y \neq Y_1 \} \); if the node has AND gate, node value will be mapped to output arc expression whose value is evaluated by \( f(X) = g(Y) + X \), where \( g(Y) = Y_1 + Y_2 \). Translations of leaf nodes and intermediate nodes are more difficult than that of root node. As to root node with node value, if root node has an OR gate, it can be calculated from the input arc expression of transition.
2.2 Extended CPN to Intrusion and Response Model

Attack tree only depicts the process of an attack. No mechanism in attack tree is provided to allow active response and defense, partially due to limits of tree model. But CPN based attack modeling can give administrators such means to control the hacker’s action or carry out some effective response. Based on the definition of CPN, transition can fire only when all its bindings occur. So we can model the defense and response actions as follow: for each transition, an input arc is added to allow control, and an output arc is added to allow response.

Additionally, if there are many control and response actions, many arcs can also be added. This paper only considers the simple effective response. Fig.6 shows an example that extends CPN attack model to support intrusion detection and its response mechanism. But readers should be aware that this model is not derived from attack tree, but extends directly from CPN attack model.

3 Case Study

In this section, we will study a practical case to illustrate the usages of CPN based modeling approach. This simple attack against physical safe will give us some experiences about attack tree model and CPN model. The goal of physical safe is to open the safe. Then attackers can pick the lock, learn the combination, cut open the safe or install the safe improperly so that they can easily open it later. To learn the combination, they have to either find the combination written down or get the combination from the safe owner and so on. In its attack tree, each node...
becomes a sub-goal and children of that node are ways to achieve that sub-goal.

Fig.7  The attack tree of physical safe

Fig.7 shows the tree with different costs assigned to the leaf nodes. The “$” is the cost of attack. Like Boolean node values, these processes can propagate up the tree as well. OR nodes have the value of their cheapest child; AND nodes have the value of the sum of their children. Obviously, the costs in Fig.7 have propagated up the tree, the cheapest attack has been highlighted and so the tree in this figure is called minimum cost attack tree. Hence, it is difficult to depict all cost features of an attack in attack tree model. Only one cost of attack can be attained in one tree.

The $f_1(x)$ is a function whose value depends on the fired transitions: $t_8$, $t_9$, $t_{10}$, and $t_{11}$. ML language is adopted to define functions and variables. In CPN based attack modeling, all cost features are described in one model. If different values are given, different cost model can be derived from the same CPN model. Additionally, from CPN based model, the attack process and states of victim can be clearly attained through state space analysis of CPN. If some time-related features are added to such CPN model, it is also easy to test and verify the temporal logic and its performance of CPN attack model.
4 Related Works

Some literatures show a comprehensive taxonomy of Internet attacks\(^5\)\(^-\)\(^7\). Other common intrusion database such as Bugtrap in Ref\(^6\) also creates a common namespace for all vulnerabilities and exploits. Taxonomy of attacks fails to formally express their dynamic properties. Some graph-based attack models also provide means for modeling intrusion\(^8\). Other researches the software fault tree approach to analyze the design and implementation of intrusion detection system\(^7\). Although some petri net and CPN based models have existed, they are only used to model the intrusion detection system itself\(^9\). Bruce Schneier was the first one to associate the term “attack tree” with the use of fault tree for attack modeling which made this approach more widely known\(^3\). Attack tree, together with ideas from other reports, enlightens us to make use of color petri net method to model attack graphically and quantitatively\(^8\)\(^-\)\(^10\).

5 Conclusions

It is easy to map attack tree into CPN attack model. After other features are added to this model, it can be used to model the intrusion detection and response. Another important feature of this model is that intrusion can be quantified. But the practical experiment shows the CPN based attack model has a more complicated form than graph-like model, especially attack tree. So it is necessary to condense the CPN based attack model. This goal can be achieved by using the ML language. Afterward we will further explore some CPN place reducing methods to simplify CPN attack model. Simulation approach of CPN based attack model is also an important researching topic in our future work.

References


Brief Introduction to Author(s)

ZHOU Shijie (周世杰) was born in 1971. He is now pursuing Ph.D. degree in UESTC. His research interests include: P2P computing, information security, and distributed computing. E-mail: sjzhou@uestc.edu.cn

QIN Zhiguang (秦志光) was born in 1957. He is now a professor, director of PhD student in UESTC. His research interests include: information security, ITS and middleware. E-mail: qinzg@uestc.edu.cn

ZHANG Feng (张 峰) was born in 1978. He is now pursuing Ph.D. degree in UESTC. His research interests include: information security, data mining and e-business. E-mail: ibmcenter@uestc.edu.cn

LIU Jinde (刘锦德) was born in 1930. He is now a professor and Ph.D. director in UESTC. His research interests include: opening distributed computing and middleware technology, mobile agent technology. E-mail: jdliu@uestc.edu.cn