International Journal of Engineering & Technology Sciences (IJETS) 1 (4): 177-191, 2013
ISSN 2289-4152
© Academic Research Online Publisher

Research Article

Malicious Insider Attacks Based Colored Petri Nets Approach

Abdelali EL BOUCHTI, Abdelkrim HAQIQ

Computer, Networks, Mobility and Modeling laboratory
e-NGN research group, Africa and Middle East
FST, Hassan 1st University, Settat, Morocco
E-mail address: {a.elbouchti, ahaqiq}@gmail.com

ARTICLE INFO

Accepted:
01 July213

Keywords:
Vulnerability
Attack modeling
Attack tree
Colored petri net
Malicious insider attacks

Abstract

In this paper, we propose Colored Petri Net (CoPNet) modeling approach by extending the attack trees with new modeling constructs and analysis approaches. CoPNet based attack model is flexible enough to model Internet intrusion, including the static and dynamic features of the intrusion. The process and rules of building CoPNet based attack model from AT are also presented. In order to evaluate the risk of intrusion, some cost elements are added to CoPNet based attack modeling. We show how attack trees can be converted and analyzed in CoPNets. Finally, we provide a malicious insider attacks as case study that illustrates the CoPNet approach.

1. Introduction

A secure computer system provides guarantees regarding the confidentiality, integrity and availability of its objects (such as data, processes or services). However, systems generally contain design and implementation flaws that result in security vulnerabilities. An intrusion takes place when an attacker or group of attackers exploit security vulnerabilities and thus violate the confidentiality, integrity, or availability guarantees of a system or a network. Intrusion Detection Systems (IDSs) [10] detect some set of intrusions and execute some predetermined action when an intrusion is detected.

Some literatures show a comprehensive taxonomy of Internet attack [4, 21]. Other common intrusion database such as [5] 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 [7]. Other research [6] uses the software fault tree approach to analyze the design and implementation of intrusion detection system. Schneier [2] 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. This modeling tool has proved to be simple, easy to use and easy to analyze results, and yet powerful in its modeling capability. Besides modeling attacker behavior ATs are found to be useful for modeling system vulnerabilities and points of access. However, the capabilities of ATs are limited, because of their limited construct set and static nature. Our effort uses Petri Net constructs to augment and extend existing principles that are already proven useful in ATs. Although some Petri Net and CoPNet based models have existed, they are only used to model the intrusion detection system itself [8].

Our choice of a CoPNet formalism to address the design of security policies is motivated by the following reasons: Petri Nets are well known for their graphical and analytical capabilities for the specification and verification of concurrent, asynchronous, distributed, parallel, and nondeterministic systems. Various features contributing to such a success include graphical nature, the simplicity of the model, and the firm mathematical foundation. It also provides modularity in design. Hence, a Petri-net-based policy is more flexible when it is embedded into a system.

The purpose of our proposed approach, called Colored Petri Net Attack Modeling approach (CoPNet) [17, 18 and 19] is to provide intuitive modeling approach for modeling attacker behavior in vulnerable systems from security perspective, based on the concepts of ATs and modeling abilities of Petri Nets. Some cost elements are added to CoPNet based attack modeling to evaluate the risk of intrusion. We choose malicious insider attacks (MIA) networks as a case study that illustrates the CoPNet approach.

The remainder of this paper is organized as follows. An overview of AG and AT is presented in Section 2. In Section 3, we present and define Petri Nets and CoPNet. In Section 4, we show how to build CoPNet attack model from AT. We show the extended CoPNet model in Section 5. CoPNet based attack model of malicious insider attacks is described in Section 6. Finally, we conclude the paper and give an overview of future work in Section 7.

2. Related work

Generally, approaches for attack modeling can be divided into two kinds: graph structure and tree structure. The tree structure use a tree to present the situation of vulnerabilities exploited to attack. Approaches based on those two structures are presented in the following.

2.1. Attack graph (AG)

AG is suitable and effective to describe the sequence of the vulnerabilities exploited. In recent years, many research institutes have devoted time and energy into this field, such as Carnegie Mellon University and George Mason University [11]. AGs are used to determine if designated goal states can be reached by attackers attempting to penetrate computer networks from initial starting states. For this use, they are graphs in which the starting node represents an attacker at a specified network location. Nodes and arcs represent actions the attacker takes and change in the network state caused by these actions. Actions typically involve exploits or exploit steps that take advantage of vulnerabilities in software or protocols. A full AG will show all possible sequences of attackers' actions that eventually lead to the desired level of privilege on the target.

2.2. Attack tree (AT)

Attack models are used frequently in the context of computer networks and power control systems. Traditionally ATs have been the most common type of model for representing known cyber-attacks [12]–[13]. In an AT, the root all possible sequences of action steps towards the goal. An attacker might be imagined proceeding up the tree, reaching a new sub goal at each node. Thus, the modeling approach implemented in an AT visualizes an attack as a hierarchy of sub goals leading to the ultimate goal. The basic AT may be made more complicated in various ways, for example, nodes might have associated values or logical "and/or" conditions [14].

Ten et al. proposed to use ATs for modeling cyber intrusions in existing power control systems [15]. ATs were shown to offer a systematic way to identify vulnerabilities of SCADA systems [9] and quantify different vulnerability scenarios. McLaughlin, Podkuiko, and McDaniel presented an AT to illustrate potential ways to commit energy theft in the smart grid [16]. Their AT shows three classes of attacks, depending on how demand data is tampered with.

ATs are a popular modeling approach because they are good at describing an attack in an intuitive visual way; show all attack paths within a broad picture; and can lead to useful mathematical analyses (e.g., risk assessment, vulnerability analysis) if nodes are assigned values. On the other hand, ATs are somewhat limited in their view of attacks only proceeding in sequential steps. Also, they tend to focus on vulnerabilities, a single goal, and a single attacker. In this paper, we are concerned with Petri nets because they do not have the limitations of ATs.

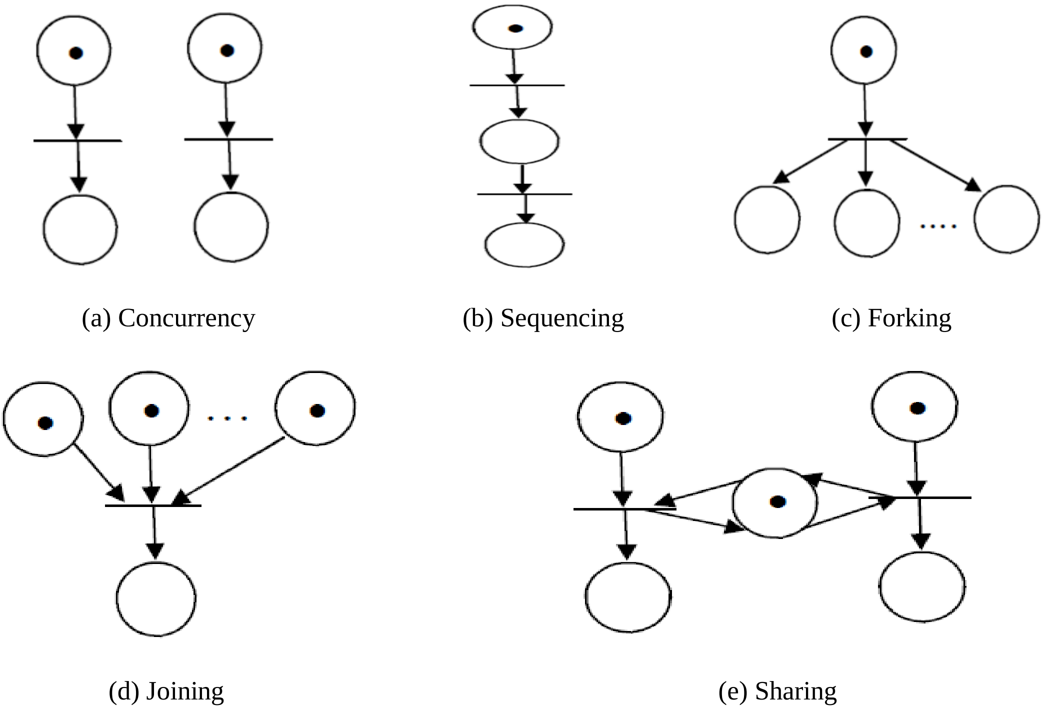
3. Petri nets and colored petri

3.1. Petri Nets

Petri Nets (PN) was created in August 1939 by Carl Adam Petri for the purpose of describing chemical processes. A PN is a mathematical modeling language. It consists of places, transitions, and arcs that connect them. Input arcs connect places with transitions. Output arcs start at a transition and end at a place. Places can have tokens; the current state of the modeled system is given by the number and type of tokens in every place.

PN's are good enough for describing and studying systems that are characterized as being concurrent, asynchronous, distributed, parallel, nondeterministic, and stochastic. Since PN's are a graphical tool, they can be used as a visual communication aid similar to flow charts, block diagrams, or networks. Moreover, tokens are used in these nets to simulate the dynamic and concurrent activities of systems. In a PN is possible to set up state equations, algebraic equations, and other mathematical models governing the behavior of systems.

A set of PN models of event is given in Figure 1. In case (a) two processes are mutually independent and can be enabled and executed concurrently. In case (b) two processes occur in sequence. (c) and (d) show forking and joining, and (e) sharing.



(a) Concurrency

(b) Sequencing

(c) Forking

(d) Joining

(e) Sharing

Fig. 1. Petri net constructs for event modeling.

3.2. Colored Petri Nets

CoPNets were introduced as a full-fledged language for the design, specification, simulation, validation and implementation of large software systems. CoPNets combine the strength of Petri nets with the strength of programming languages and are widely employed in both academical and industrial areas for software system design, implementation, simulation and validation. CoPNets have a series of good features that make it suitable for modeling and analyzing complex systems.

Each CoPNet has a set of declarations, which we position by convention in a box with dashed lines. The declarations introduce a number of color sets, functions, operations, variables and constants, which can be used in the net inscriptions of the corresponding CoPNet, particularly in the guards, arc expressions and initialization expressions. The declarations of a CoPNet can be made in many different languages, e.g., by means of standard mathematical notation or by means of many sorted sigma algebras. Each color set declaration introduces a new color set, whose elements are called colors. Every color set declaration implicitly declares a set of constants (the colors of the color set). Moreover, the color set declaration can implicitly declare some standard functions and operations which can be used on the colors of the color set. A declared color set can be used [1]:

- In the declaration of another color sets.
- In the declaration of variables (having the color set as type).
- In the declaration of functions, operations and constants (e.g., a function may map from one color set into another color set).
- In the color set inscription of a place (indicating that all tokens on the place must have token colors which belong to the color set).

Definition 1. A CoPNet can be represented as a tuple $CPN = (\sum, P, T, A, N, C, G, E, M_0)$, where:

- $\sum$ is a finite set of color sets, P, T and A are finite sets of places, transitions and arcs, respectively. $P \cap T = P \cap A = T \cap A = \phi$, and $A \subseteq P \times T \cup T \times P$. There are two types of arcs for a transition: incoming arc and outgoing arc. When a transition takes place, incoming arcs indicate that the input places shall remove specified number of tokens, while outgoing arcs mean that the output places should add the specified number of tokens. The exact number is determined by the arc expression, defined by the expression function E.
- N is a node function $N : A \to P \times T \cup T \times P$ and it specifies the source and destination of an arc.

- *C* is a color function, and $C:P \to \Sigma$. *C(p)* specifies the set of allowed colors for any token of place *p*. A token element is a pair (*p*, *c*), where $p \in P$ and $c \in C(p)$. The set of all token elements is denoted as $TE = \{(p,c) / p \in P \wedge c \in C(p)\}$.
- *G* is a guard function, mapping each transition t to a boolean expression *G(t)*. Let IB stand for boolean type, which contains the elements {*true*, *false*}. Let $Type\ (v)$ denote the type of the variable v, and *Type (expr)* stands for the type of the expression *expr*. *Var(expr)* denotes the set of variables in expression *expr*. So $\forall t \in T : Type\ (G(t)) = \mathrm{IB} \wedge Type\ (Var(G(t))) \subseteq \Sigma$.
- *E* is an arc expression function, mapping each arc into an expression with type *C(p)*, where *p* is the place of the given arc. That is $\forall a \in A : Type\ (E(a)) = C(p) \wedge Type\ (Var(E(a))) \subseteq \Sigma$
- M_0 is the initial marking of CoPNet, and $M_0 \in (TE)$.

Definition 2. A transition $t \in T$ is firable (or enabled) at a marking *M* if and only if $\forall p \in P : (M(p) \geq W(p,t))$, where $W(p,t)$ is the weight of the arc to transition *t* from its input place *p*. Firing (or executing) transition *t* results in changing marking *M* to a reachable marking *M'*, where $\forall p \in P : (M'(p) = M(p) - W(p,t) + W(t,p))$.

Definition 3. The pre-incidence matrix PRE of a net N is $|P| \times |T|$ matrix row *p* and column *t* is the weight $W(p,t)$ of the arc from place *p* to transition *t*. The post-incidence matrix POST of *N* is a $|P| \times |T|$ matrix whose element is the weight $W(t,p)$ of the arc from transition *t* to place *p*. *V=POST-PRE* is called the incidence matrix of N.

For example, in Figure 2, the initial marking is $M_0 = ((2,1),(0,0),(0,0),(0,0))$, both t_1 and t_2 are firable, and after firing t_1, a reachable marking is $M_1 = ((1,0),(1,0),(0,0),(0,0))$.

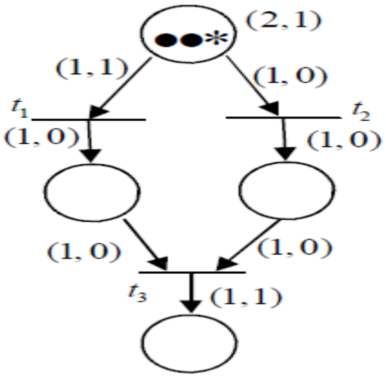


Fig. 2. A colored Petri net example.

4. Building CoPNets from attack tree

Although CoPNet 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 AT into CoPNet based attack model is analyzed. Some other extended features of this model are also expressed in this section.

The CoPNet based attack model can be defined from AT to reduce the cost of modeling. It is because that some attack models have been built with ATs [2, 3]. To build a CoPNet based model from an AT, the mapping rules between them should be determined. The root node of AT 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 AT can map to transitions. The relationship among nodes could be regarded as the arc of CoPNet. The node value in ATs is expressed as arc expression of CoPNet. And the logics of ATs can map to event relationship of CoPNet.

4.1 Root node mapping

In AT, root node is the goal and result of attack. In CoPNet 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 CoPNet. Their maps are shown in Figure 3.The node with OR gate maps to event's conflict relation of CoPNet. This means only when one event occurs, will the attack take a place. The node with AND gate maps to event's sequential relation implies that only when all events occur, then the attack will take place.

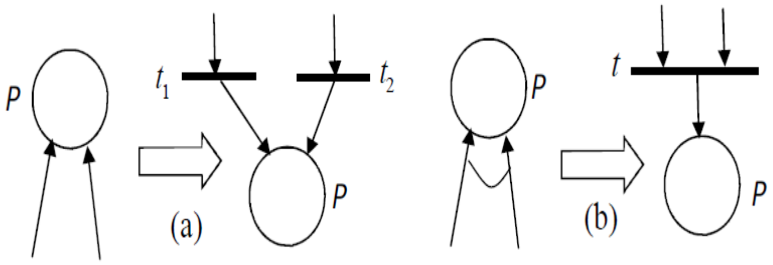


Fig. 3. Root nodes and their mapping in CoPNet. (a) is OR gate of root node, and (b) is AND gate of root node.

4.2 Leaf nodes mapping

Leaf nodes in ATs are attacker's actions breaking into the victim's system. It is clearly that leaf nodes can map to the transition of CoPNet. But in an AT, 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. Ruis's analysis of intrusion divided attacks into seven stages: Reconnaissance, Vulnerability Identification, Penetration, Control, Embedding, Data Extraction & Modification, and Attack Relay [4]. Each stage can also be divided into some or several sub-stages. So we can model attacks' stages and sub-stages as intermediate states when translating ATs into CoPNet based attack models. Figure 4 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 Figure 4, the value of place p can be derived from function $f(\mathrm{t})$, where $t \in T$, and $f(\mathrm{t}) \in \sum$. And the output arc of place *p* is the input arc of next transition. In Figure 4, the IP place in an Initialization Place whose means depend on the transition.

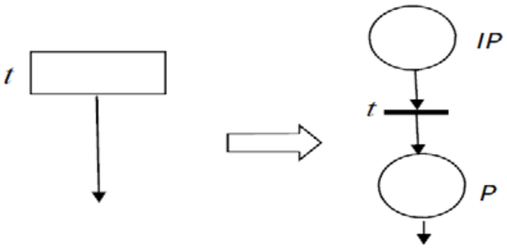


Fig. 4. Leaf node and its mapping in CoPNet

4.3 Intermediate Nodes Mapping

Intermediate nodes of ATs are sub-actions or sub-goals of attackers. It is more difficult to translate these nodes into CoPNet 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 follow:

- The intermediate node itself maps to transition, *t*, of CoPNet.
- Input arc of an intermediate node maps the input arc of *t*, OR and AND gates are translated to confliction relation and sequence relation respectively.
- Intermediate place is added in the same way as the translations of leaf nodes.

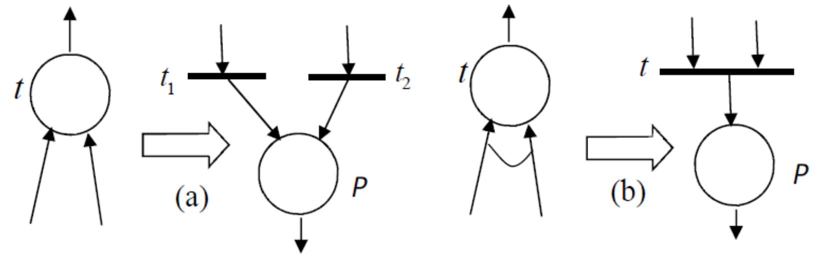


Fig. 5. Intermediate nodes and their mapping in CoPNet. (a) is OR gate of intermediate node, and (b) is AND gate of intermediate node.

By using above rules, the intermediate node can be mapped into CoPNet attack model. Figure 5 shows the mapping relations 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 transition and a place is added to connect newly added place with the corresponding transition.

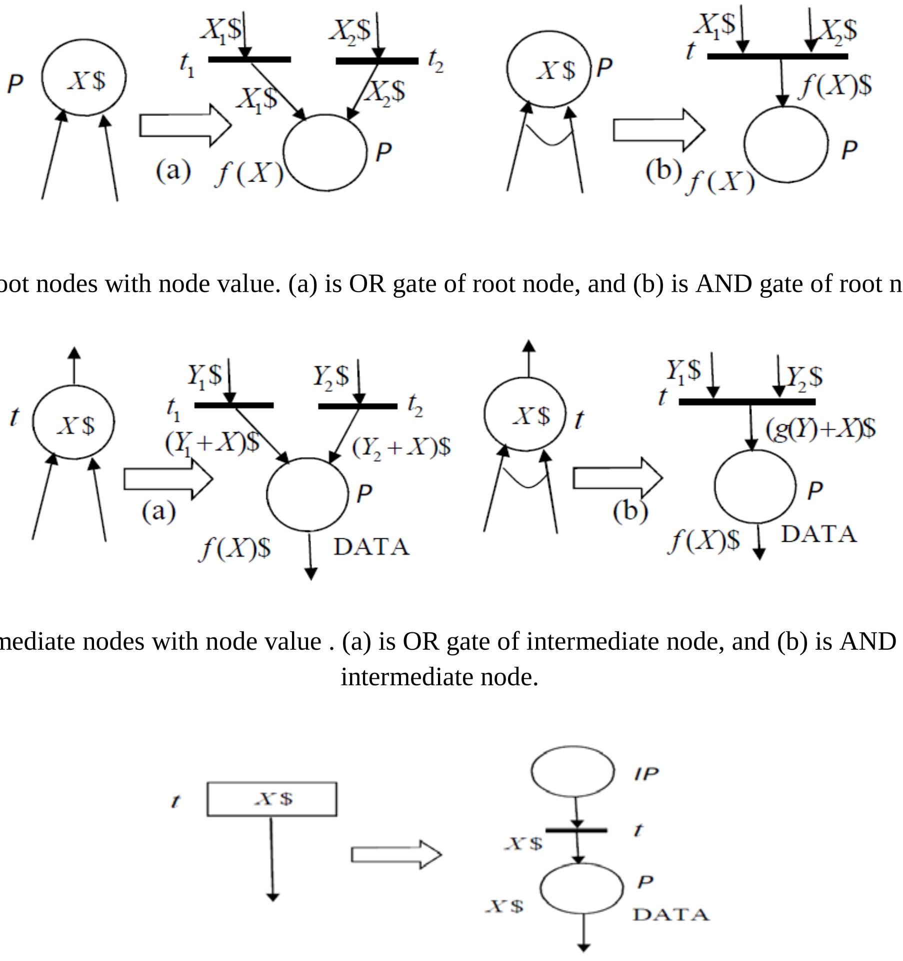
4.4 Temporal Logic Mapping

As to building template CoPNet from AT, an important issue is how to deal with temporal sequence of attack. 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 AT. Even relations of CoPNet can depict temporal logics in an AT. Event relations of CoPNet can depict temporal logics in an AT. In fact, only sequence relation and conflict relation are used in CoPNet attack models. Although intermediate modes have multiple out arc, concurrence relation may also be used. In this paper, concurrence relations are note used.

4.5 Node Value Mapping

The node value in AT is used to perform risk assessment. This special feature extends its application scope and usage for quantifying intrusions. In CoPNet, there is no node value function, but it can be expressed by color value of place of CoPNet. During the translation of CoPNet model, after transition is fired, some value will be added to arc expression of this transition t, 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 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 $g(Y)=\{y/y=Y_1\wedge y\neq Y_2$ or $y=Y_2\wedge y\neq Y_1\}$; 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 OR gate, it can be calculated from the input arc expression of transition added during translation by function $g(X)=\{x/x=X_1\wedge x\neq X_2$ or $x=X_2\wedge x\neq X_1\}$; if root node has an AND gate, node value ($ denotes) $f(X)$ includes two parts, $X_1\$$ and $X_2\$$ where $f(X)=X_1\$+X_2\$$.



(1) Root nodes with node value. (a) is OR gate of root node, and (b) is AND gate of root node.

(2) Intermediate nodes with node value . (a) is OR gate of intermediate node, and (b) is AND gate of intermediate node.

(3) Leaf node with node value.

Fig. 6. Node with node value and their mapping.

5. Extended CoPNet Model for Intrusion and Response

AT only depicts the process of an attack. No mechanism in AT is provided to allow active response and defense, partially due to limits of tree model. But CoPNet 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 CoPNet, transition can fire only when all its bindings occurs. So we can model the defense and response actions as follows: 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. But readers should be aware that this model is not derived from AT, but extends directly from CoPNet attack model.

6. Malicious Insider Attacks MIA

In this section, we describe the attack selection process for a case study process control network, and we use this case study to illustrate the usages of CoPNet based attack modeling approach. We have chosen malicious insider attacks (MIA) as our case study.

The basic structure of the AT for malicious insider attack (MIA) was proposed in [20]. Figure 7 illustrates a sample brief network-level AT for the process control network described above. The top event is chosen to be “Malicious Insider attack success”. As shown in Figure 7, 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 Figure 7 have propagated up the tree, the cheapest attack has been highlighted and so the tree in this figure is called minimum cost AT. Hence, it is difficult to depict all cost features of an attack can be attained in one tree.

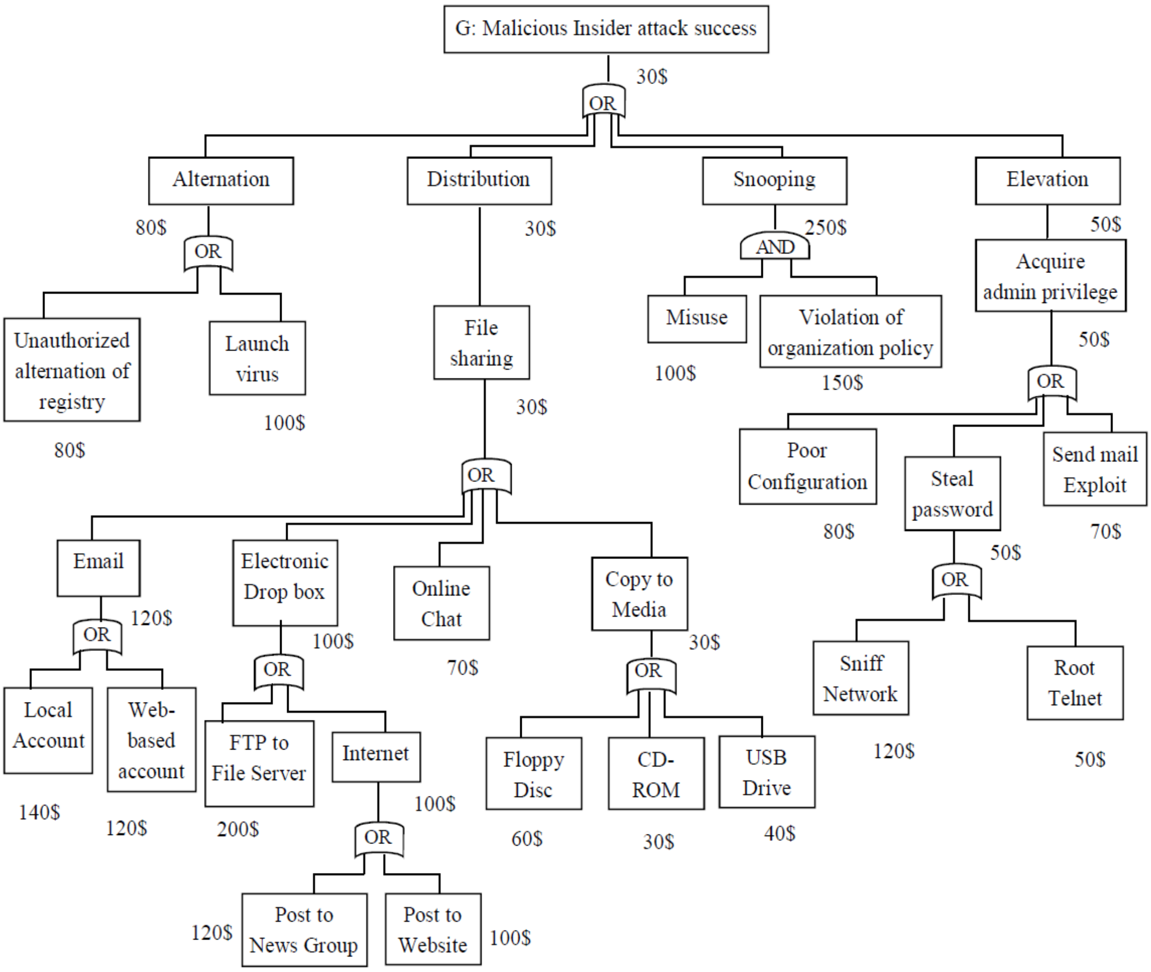
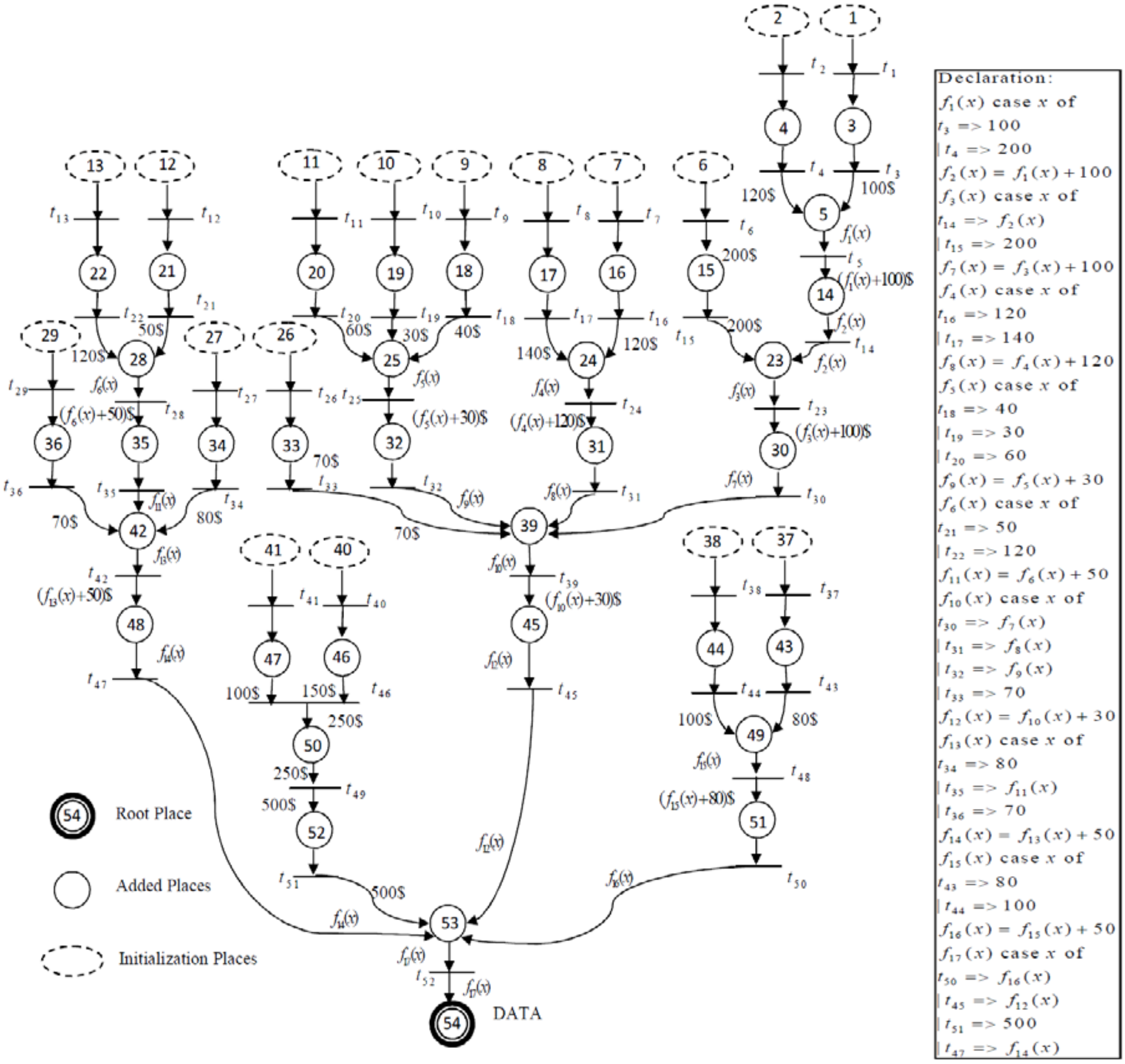


Fig.7. An AT for Malicious Insider Attack (MIA)

Figure 8 shows CoPNet based attack model of MIA. The $f_1(x)$ is a function whose value depends on the fired transitions: t_3 and t_4. ML language is adopted to define functions and variables. In CoPNet 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 CoPNet model. Additionally, from CoPNet based model, the attack process and states of victim can be clearly attained through state space analysis of CoPNet. If some time-related features are added to such CoPNet model, it is also easy to test and verify the temporal logic and its performance of CoPNet attack model.



```
Declaration:
f1(x) case x of
t3 => 100
| t4 => 200
f2(x) = f1(x)+100
f3(x) case x of
t14 => f2(x)
| t15 => 200
f7(x) = f3(x)+100
f4(x) case x of
t16 => 120
| t17 => 140
f8(x) = f4(x)+120
f5(x) case x of
t18 => 40
| t19 => 30
| t20 => 60
f9(x) = f5(x) + 30
f6(x) case x of
t21 => 50
| t22 => 120
f11(x) = f6(x) + 50
f10(x) case x of
t30 => f7(x)
| t31 => f8(x)
| t32 => f9(x)
| t33 => 70
f12(x) = f10(x) + 30
f13(x) case x of
t34 => 80
| t35 => f11(x)
| t36 => 70
f14(x) = f13(x) + 50
f15(x) case x of
t43 => 80
| t44 => 100
f16(x) = f15(x) + 50
f17(x) case x of
t50 => f16(x)
| t45 => f12(x)
| t51 => 500
| t47 => f14(x)
```

Fig.8. CoPNet based attack model of MI attacks

7. Conclusions

In this paper, we have presented CoPNet based attack modeling approach to model the attacks. The objective of our modeling approach is to provide more precise quantitative parameterization and advanced modeling capabilities compared to ATs. 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, so the most effective controlling actions can be determined. But the practical experiment shows the CoPNet based attack model has a more complicated form than the graph-like model, especially AT. So it is necessary to condense the CoPNet based attack model. This goal can be achieved by using the ML language. Afterward we will further explore some CoPNet place reducing methods to simplify CoPNet attack model. We have provided MI attacks as a case study that illustrates the CoPNet approach. We have showed that CoPNet based attack model of MIA has many unique characters which AT model has not. Simulation approach of CoPNet based attack model is our future work.

References

[1] Jensen K. Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use [C]. Springer-Verlag, Berlin, Germany/ Heidelberg, Germany/ London, UK/ ect., 1992, 1.

[2] Schneier B. Attack Trees [J]. Dr. Dobb's Journal of Software Tools 1999, 24(12):21-29.

[3] Cunningham W. The WikiWikiWeb [DB/OL]. http:// c2.com/cgibin/ wiki, 2002.

[4] Ruiu D. Cautionary tales: stealth coordinated attack how to [DB/OL]. http:// www.nswc.navy.mil/ISSEC/CID/Stealth_coordinated_Attack .html, July 1999.

[5] Bugtraq Vulnerability Database [DB/OL]. http:// www.securityfocus.com, 2003.

[6] Helmer G, Wong J., Slagell M, et al. A Software Fault Tree Approach to Requirements Analysis of an Intrusion Detection System [C]. In Proceeding of symposium on requirements engineering for information security. Center for education and research in information assurance and security, Perdue University, March 2001.

[7] Phillips C, Swiler L P. A Graph-based System for Network- Vulnerability Analysis [C]. In proceeding of new security paradigms workshop, Charlottesville, VA, USA, 1998, 71-79.

[8] Helmer G, Wong J., Slagell M, et al. "Software Fault Tree and Colored Petri Net based specification and implementation of agent- based intrusion detection systems" Int. J. Information and Computer Security, Vol. 1, No. 1/2, 2007.

[9] "The center for SCADA security," The Center for SCADA Security, Sandia National laboratory. [online]. Available http://www.sandia.gov/scada/home.htm.

[10] R. Rehman. Intrusion Detection Systems with Snort. Prentice-Hall, 2003.

[11] R. Lippmann and K.Ingols. An annotated review of past papers on attack graphs. Technical report, MIT Lincoln Laboratory, March 2005.

[12] S. Mauw and M. Oostdijk, "Foundations of attack trees," in Proc. 8th Annu. Int. Conf. Inf. Security Cryptol. (ICISC), Seoul, Korea, Dec. 2005, pp. 186–198.

[13] P. Khand, "System level security modeling using attack trees," in Proc. 2nd Int. Conf. Comput., Control, Commun. (ICA), Karachi, Pakistan, Feb. 2009, pp. 1–6.

[14] K. Schneider, C.-C. Liu, and J.-P. Paul, "Assessment of interactions between power and telecommunications infrastructures," IEEE Trans. Power Sys., vol. 21, pp. 1123–1130, Aug. 2006.

[15] C.-W. Ten, C.-C. Liu, and M. Govindarasu, "Vulnerability assessment of cybersecurity for scada systems using attack trees," in IEEE Power Eng. Soc. Gen. Meet., Tampa, FL, Jun. 2007, pp. 1–6.

[16] S. McLaughlin, D. Podkuiko, and P. McDaniel, “Energy theft in the advanced metering infrastructure,” in Proc. 4th Int. Workshop Crit. Inf. Infrastruct. Security (CRITIS 2009), Bonn, Germany, Sep. 2009, pp. 176–187.

[17] A. El Bouchti and Abdelkrim Haqiq,“Performance Modeling of Attack Countermeasure Using Colored Petri Nets”, International Symposium on Security and Safety of Complex Systems, Agadir, Morocco, May 25 - 26, 2012.

[18] A. EL BOUCHTI and Abdelkrim HAQIQ, “Modeling Cyber-Attack for SCADA Systems using CoPNet Approach’’ ICCS'12, Agadir, Morocco, November 5-6 2012.

[19] A. EL BOUCHTI and Abdelkrim HAQIQ, “Cyber-Attack for BGP Systems Using CoPNet Model” INTECH'12, Casablanca, Morocco, September 18-20 2012.

[20] Butts J, Mills R, Baldwin R. Developing an insider threat model using functional decomposition. Computer Network Security 2005; LNCS(3685):412–417.

[21] Tidwell T, Larson R, Fitch K, Hale J. Modeling internet attacks. Proceedings of the 2001 IEEE Workshop on Information Assurance and security, vol. 59, IEEE, 2001.