An Intrusion Plan Recognition Algorithm Based on Max-1-Connected Causal Networks

Zhuo Ning , Jian Gong

Department of Computer Science and Engineering, Southeast Univ.
Nanjing, Jiangsu, 210096 China
{zhning, gjian}@njnet.edu.cn

Abstract. Intrusion plan prediction and recognition is a critical and challenging task for NIDS. Among several approaches proposed so far, probability inference using causal network seems to be one of the most promising mechanisms. Our analysis shows that the polytree is limited in its expressiveness, and belief updating in max-k-connected networks is hard for all $k \geq 2$ [12]. To find a tradeoff between expressive power and inference efficiency, this paper extends the structure of causal network from polytree to max-1-connected Bayesian network, and proposes a new intrusion plan prediction algorithm IPR on it. We evaluate the approach using LLOS1.0, and the results demonstrate that IPR can predict the occurrence probability of DDOS when Sandmind attack occurs to gain root privilege, and then confirm the prediction in the beginning of Syn flooding.

Keywords: NIDS, Max_1_connected Causal Network, belief updating, MAP

1 Introduction

The wide spread of use of NIDS and the increasing complain about its high volume alerts have led to more and more intensive exploratory works on it. People begin to realize that the high false positive ratio is due to short of understanding of inner logical relations in the attack flows. So it is not reasonable to detect intrusion only based on single-packet signature and to view the attack flow separately. Among the recent researches focusing on how to accumulate the alert logical relations from the context [1], inference using causal network is one of the most promising approach for its powerful expression of causality and belief propagation consistent with ongoing evidence. [2] designed a two-level Bayesian tree model to discover novel attack strategies by correlating alerts. The expensive cost of this method leads to its poor behavior on practice. [3] proposed an abnormal IDS called eBayes TCP to detect some TCP abnormal behaviors, but the false positive ratio is also high. Comparatively more mature work was done by [4]. [4] proposed an abuse detecting approach based on polytree. Relying on the library of attack plans (defined as polytree), belief updating algorithm is used to calculate the new belief of each node when a new evidence entered, and then the node with the highest score is considered as the most

possibly occurring attack. But the causal network used in [4] is a polytree, as shown in the sequel its expressive power is too limited in illustrating attack plan.

A Bayesian Network (BN) G=(V, P) is represented as a directed acyclic graph G where V is a set of nodes, and each one of V stands for a variable. P is a set of edges and each one of P denotes a causal relationship between a couple of variables. A polytree topology is defined as BN where for every pair of nodes (x_1, x_2), there is at most one path from x_1 to x_2 in the *underlying undirected graph*[6]. When defining the attack tree, security analysts decompose the final goals into subgoals iteratively until those of the lowest level are exercisable penetration points. In the above process, a causal network is expanded and its branches are built to identify the different subgoals. So it is common that x_1 and x_2 have two same children. Unfortunately in this case the path between them will be two (as shown in Fig.1). In order to escape the limitation of polytree in expression, there are naturally two ways to try. One is to remove nodes to make it polytree-structured by conditioning algorithms[5,6] or cluster schemes[7,8]. However, all these reductions are usually exponential in some aspect of the problem instance and not efficient. Another way is to increase the number of paths between pairs of nodes and broad the known classes of tractable Bayesian networks. [9] proved that Belief updating in max-k-connected networks is hard for all $k \geq 2$, even with no evidence. This paper puts forward max-1-connected Baysian Network(M1CBN) as the tradeoff in balancing its expressive power and inference efficiency. Unlike Polytree, M1CBN is defined as networks where for each couple of nodes (x_1, x_2) in DAG, there is at most one *directed path* form x_1 to x_2. The difference between them is illustrated in Fig.1. Clearly, all polytrees are M1CBN, but not vice versa. Based on M1CBN, an intrusion plan recognition algorithm (IPR) is introduced which not only benefits more powerful expressive power, but also retains polynomial performance. In practice many attack graphs are expressed by multiple connected BN, and it is more efficient to transform them to M1CBN and apply IPR directly than to transform them to polytree and apply Pearl's algorithm as [4].

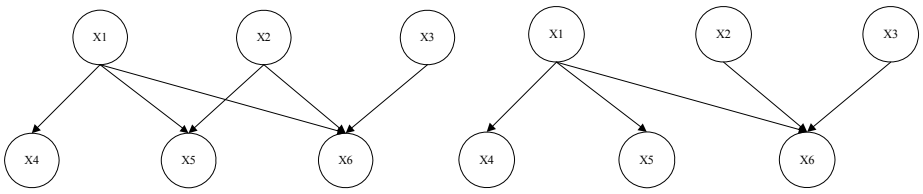

Fig. 1. Max_1_connected BN Polytree

Intuitionally the task of plan recognition is to find an explanation for the observed evidence e. The explanation is usually composed of a set of hypotheses, and what these hypotheses are and the values of them are particularly concerned. In a BN G=(V, P) E denotes nodes that have been trigged by evidence e, and W=V-E denotes all variables without evidence. Any assignment of W that is consistent to e is called an explanation of e. The task of recognition is to find an assignment that makes

$$P(w \mid e) = \max_{W} P(W \mid e) \qquad (1)$$

and it is called the most possible explanation (MPE). Usually people tend to ignore attack details and focus only on key steps and the final goal(that is, to focus on some specific variables of W). When the assignment w is a partial one the task of (1) becomes a MAP problem (to find a most probable instantiation of a set of variables given evidence). Generally the MPE and MAP problem over Bayesian Network are NP hard，but in some known tractable subclasses such as polytree, polynomial algorithms have been found for MPE problem, whereas MAP remains hard.

The rest of the paper is organized as follows. Section 2 shows some important results in inference in M1CBN. Based on section 2, respectively a new approximate belief updating algorithm BeliefUpdating and intrusion plan recognition algorithm IPR are addressed in section 3 and 4. Section 5 applies IPR in LLDOS1.0, a data set of DARPA 2000, and reports our experiment results. Finally, we conclude the paper and discuss future research directions on this topic in section 6.

2 Intrusion Plan recognition on M1CBN

For the sake of convenience, we introduce some notations. Given a M1CBN B and any node X in B, we denote by $\prod(X)$ the parents of node X，by $D(\prod(X))$ the set of all possible assignment on $\prod(X)$, and by $\prod^*(X)$ the set of all the ancestors of X, including X (* is reflexive and transitive closure to $\prod$). The ancestor graph $G^*(X)$ of a node X, induced by X, is composed of all the nodes in $\prod^*(X)$ and the arcs connecting them in B.

Proposition 1 MPE/MAP when restricted to M1CBN is NP-hard, even with no evidence.

Sketch of Proof. Clearly a 2_level BN is a M1CBN, as is shown in Fig.1(a). [10] proved MPE/MAP restricted to 2_level BN with evidence is NP hard, and MPE/MAP on 2_level BN without evidence remains NP hard [11]. And then MPE/MAP on 2-level M1CBN is NP hard. Therefore, MPE/MAP restricted to m-level (m≥2) M1CBN is NP hard as well.

Proposition 2 Belief updating in M1CBN is NP-hard.

This conclusion directly follows from the above proof.

Theorem 1 Top-down belief updating in M1CBN without evidence can be performed in time linear in the size of the network.

Proof. Given a BN without evidence, the marginal probability in any node $X_i = x_{i,j}$ is:

$$P(X_i = x_{i,j}) = \sum_{A \in D(\prod(X_i))} P(X_i = x_{i,j} \mid A) P(A) \qquad (2)$$

In a max-1-conneted BN, any predecessor of X is d-separated, and thus equation (2) can be evaluated as Equation (3):

Equation (3) just formalizes the case of passing just π messages in polytree without evidence. So top-down belief updating in M1CBN without evidence can be performed

$$P(X_i = x_{i,j}) = \sum_{A \in D(\prod(X_i))} P(X_i = x_{i,j} \mid A) \prod_{X_m \in \prod(X_i)} P(X_m = A(X_m)) \quad (3)$$

in time linear in the size of the network.

Theorem 2 $G^*(X)$ forms an *X*-oriented polytree for every $X \in V$.

Proof. The proof of the proposition is constructive. According to the definition of ancestor graph, given node X, there is one and only one path from X to its parent Y. Likewise, there is an exclusive path from Y to its parent Z. So the exclusive path between X and Z is fixed. Iteratively following this way, we can get an exclusive path between X and any of its ancestors. Hence, $G^*(X)$ forms an X-oriented polytree.

Proposition 1 shows MAP is hard in M1CBN. Belief updating is a practically useful inference algorithm for approximating MAP for a number of reasons and has proven to be very effective and efficient in a variety of domains. Unfortunately, even belief updating is hard in M1CBN (shown in proposition 2). Based on theorem 1 and 2, a new belief updating algorithm restricted to M1CBN is described in the next section.

3 Belief Updating Algorithm in M1CBN

Based on the above discussion, local belief updating in M1CBN can be accomplished in two steps. *Step 1 Bottom-up propagation:* all nodes that have direct causal relations with X compose the ancestor tree of X. And as is proved in theorem 2, the ancestor tree is a polytree. So when a new evidence e triggers node X, belief propagation in $G^*(X)$ is the same as that in polytree [6]. *Setp 2 Top-down propagation:* the belief changes of nodes Y in step 1 will affect the likelihood of their children as effect. This step updates the believes of Y' s offspring using equation (3) while breadth first searching M1CBN. One note worth to mention is that belief updating in M1CBN propagates only twice and does not adopt iterative updating mechanism used by polytree, since most of causal influence can be evaluated in two propagations. Moreover, the simplified algorithm directly leads to polynomial performance. In algorithm BeliefUpdating N is a node in M1CBN, and each N has an evaluator J_N, which is responsible for evaluating the condition matching. To measure uncertain information, $J_N \in [50\%,100\%]$. All messages are initialized to 1.

Algorithm BeliefUpdating(B, event$_i$)

```
Input: M1CBN B(V, E), hyper alert event_i
Output: updated believes in B, denoted as a vector
UpdatedBelief(aim:                          likelihood,
key_attack_step_1:likelihood,          …            ,
key_attack_step_i:likelihood,
key_attack_step_n:likelihood)
{
  If (node X is triggered by event_i) then {
    sign X as an observed node, and J_X = current belief
    of X,
    // Bottom-up propagation
```

```
    Breadth-first search G*(X) starting from X, for all
    node Y∈G*(X) do {
    //updating with Pearl's formulation[6]
      receive π_Y(U_i) from every Y's parent node U_i;
      receive λ_Ci(Y) from every Y's child node C_i;
       compute Belief(Y) and sign Y;
       compute λ_Y (U_i) for every Y's parent node U_i;
       compute π_Ci(Y) for every Y's child C_i;
    }
    // Top-down propagation
    Breadth-first search B, for any node N do
    If (the parent of N has been signed) then {
        update the believes of N with equation (3);
        sign N;
      }
    For i=1 to the number of key attack nodes of B do{
      UpdatedBelief [i]. key_attack_step= B[i].node;
      UpdatedBelief [i].likelihood = B[i]. belief;
    }
    Output UpdatedBelief;
  }
}
```

4 Intrusion Plan Recognition algorithm IPR

For raw alerts generated by NIDS, we aggregate and cluster them based on different srcIP or dstIP, and then prioritize them as [14]. The redundancy of resulting alerts is reduced, while important alert attributes retains. We denote each attack flow in the same cluster by a time series-based event vector Event($event_1$, $event_2$, … $event_i$, … , $event_n$), and each $event_i$ is called a hyper alert.

The complexity of IPR heavily relies on BeliefUpdating. BeliefUpdating propagates the diagnostic influence of ongoing evidence in its ancestor tree and thus reduces the problem's difficulty. In other wards, it tries to propagate causal influence as wide as possible. We denote by X the evidence node, by n the number of nodes in BN, and by m the average number of nodes in $G^*(X)$. As shown in [13], generally $m \prec\prec n$. If we measure the complexity of BeliefUpdating by the nodes it visits, then the complexity of Bottom-up propagation is O(m), and Top-down propagation is O(n). So the complexity of BeliefUpdating is O(m+n). Suppose the average number of matching Bayesian Network is k, the complexity of IPR is O(k(m+n)), which is polynomial in the size of attack graph.

```
Algorithm IPR(Event)
  Input: Event(event_1, event_2, … , event_i …)
  Output: the aim of attack with the probability and
  corresponding parameters, such as srcIP and dstIP.
  {
    i=1;
```

```
While (event_i is not NULL){
  Search attack plan library, and trigger BNs that
  include event_i;
  j = the number of triggered BN;
  for k=0 to j do{
  newBelief[k] ← call(BeliefUpdating(BN_k, event_i));
  If (newBelief[k].aim > threshold) then {
      output newBelief[k];
      output corresponding parameters, such as srcIP,
      dstIP, port and time;
      predict newBelief[k].aim as the aim of attack;
    }
  }
  i++;
 }
}
```

5 Experiment

To evaluate the effectiveness of approximation made by these two algorithms, we applies them to LLDOS1.0, the first DDOS attack scenario created for DARPA to evaluate IDS. IPR is implemented on Monster3.0, a GNIDS developed by Southeast University. LLDOS1.0 includes 5 attack phases over the course of which the adversary probes, breaks in, installs trojan mstream DDoS software, and launches a DDoS attack against an off site server. Fig.2 illustrates the DDOS attack graph stored in Monster3.0, where the key parameters, such as CPT and prior probabilities, are listed on arcs and nodes respectively. Table 1 shows the hyper alerts of the attack flow in 172.16.115.0/24 after aggregating, clustering and eliminating redundancy, while the same alerts of other three subnets (172.16.112.0/24~ 172.16.114.0/24) are omitted for its clarity. In fact the alerts are far more than that listed in the table 1, for example, the hyper alert ICMP_PING_SWEEP represents 256 raw alerts. A denotes the aim DDOS, and B, C denote the other two key attack steps, Controlling a group of hosts and Launching attack respectively. In this case, Fig.2 and the alerts in column 2 are the input of algorithm IPR and the likelihood value of A, B and C are listed in each row as the output. The belief of the trigged node which is computed by its evaluator J, is listed in column 3. And the source IPs are 202.77.162.213, except for those that are spoofed and randomly generated by Syn flooding.

Table 1. The experiment results

Dst IP	Hyper alert	Value of Evaluator	A(%)	B(%)	C(%)
172.16.115.0/24	ICMP_PING_SWEEP	$J_Q = 1$ $J_p = 1$	32.5 41.8	50 40.4	32 42.6
172.16.115.0/24	RPC sadmind UDP PING	$J_N = 0.8$	42.8	59.5	43
172.16.115.20	RPC portmap sadmind request UDP	$J_N = 1$	44.8	65.3	46
172.16.115.20	RPC sadmind query with root credentials attempt UDP	$J_N = 1$	44.8	65.3	46
172.16.115.20	RPC_sadmind_UDP_NETMGT_ PROC_SERVICE_CLIENT_DOMAIN overflow attempt	$J_N = 1$	44.8	65.3	46
172.16.115.20	ATTACK-RESPONSES directory listing	$J_E = 1$	62.8	73.2	59.2
172.16.115.20	RSERVICES rsh root	$J_B = 1$	68.9	100	63.4
131.84.1.31	Syn flooding	$J_F = 1$	80.4	100	80.9

A clear attack track is shown in table 1 and believes of A, B and C increase stably in the course of attack. In the beginning, ICMP_PING_SWEEP comes, and the evaluators of node P and Q activate them with a probability of 100% (the adversary probes the subnet). The values of A, B and C suggests that ICMP_PING_SWEEP doesn't contributes much to DDOS, though the belief of A rises from 30% to 32.5%. Secondly, RPC_portmap_sadmind_request_UDP and RPC_sadmind_UDP_PING are probe steps to determine which hosts are running the remote administration tool, "sadmind". At this time, 172.16.115.20 is founded to be vulnerable. During the previous process, J_N gradually increase its belief from 80% to 100%, and A increases from 42.8% to 44.8% accordingly. In the third step, the adversary uses sadmind, buffer-overflow attack, to remotely break in 172.16.115.20, and three different stack pointer values are attempted, generating alerts RPC_portmap_sadmind_request_UDP, RPC_sadmind_query_with_root_credentials_attempt_UDP, and RPC_sadmind_ UDP_NETMGT_PROC_SERVICE_CLIENT_DOMAIN_overflow_assttempt. When 172.16.115.20 responses the adversary by listing the directory (172.16.115.20 is sure to be conquered), J_E = 100% and A and B climb rapidly and reach 62.8%, 73.2% respectively. As soon as the real Syn flooding is launched, A soars to 80.4%, which is greater than threshold (70%), and then IPR outputs the attack vector as ((A, 80.4%), (B, 100%), (C, 80.9%)). From the circumstance variables, such as srcIp(172.16.115.20) and DstIp(131.84.1.31), one can conclude that the source IP of DDOS are not the forged ones appeared in the packets, but 172.16.115.20, which is remotely controlled by the adversary. In LLDOS1.0, 172.16.112.20 and 172.16.112.10 was conquered in the same way. So we can cut down the communication of these three hosts and prevent DDOS from happening.

6 Conclusion

This paper proposes an algorithm IPR with polynomial complexity O(k(m+n)) to predict and recognize attack plan, which exceeds [4] in expressive power and performance. The main improve relies on the following: 1) IPR broads the structure of attack plan depicted by Bayesian Network from polytree to max-1-connected Baysian Network, and thus expressive power becomes rich. 2) IPR gives up the iterative updating mechanism used by polytree, and adopts approximation to

propagate causal influence as wide as possible. The approximation leads to polynomial performance and is effective as the experience shown. Moreover, IPR bears several advantages: firstly, it is able to detect multiple concurrent goals and partially ordered plan; Secondly, it has default reasoning ability and can deal with uncertain information. However, as a method based on predefined attack graph, it can not recognize unknown attacks. So how to recognize the new attack by correlation is a challenge in our future work.

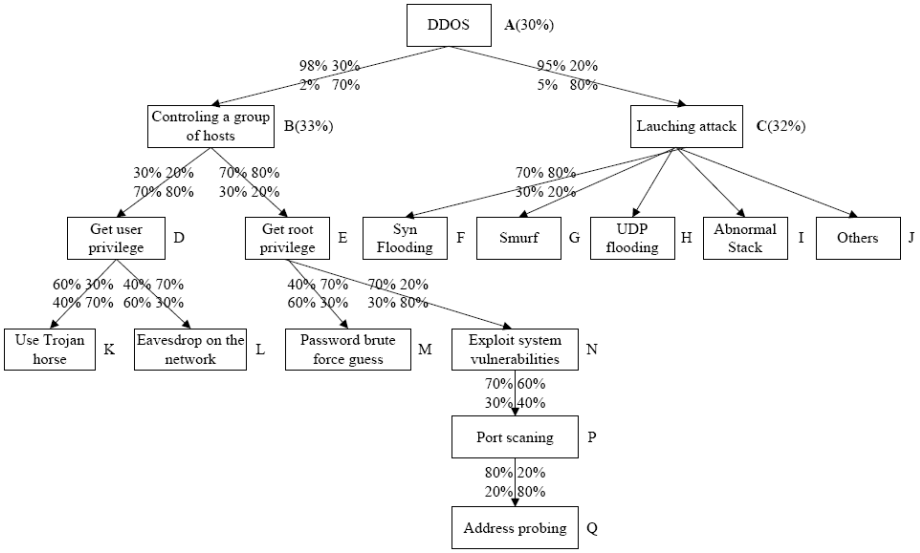

Fig. 2. Attack graph of DDOS

Acknowledgments. This research is partially support by the National Basic Research Program (973 Program) No.2003CB314803, Jiangsu Province Key Laboratory of Network and Information Security BM2003201 and the Key Project of Chinese Ministry of Education under Grant No.105084.

References

For the sake of the space, references will be available whenever you ask the authors.