Cooperation Forensic Computing Research*

Youdong Zhang
Department of Computer Engineering
Huaiyin Institute of Technology, Huaian, China
z.yd@163.com

Abstract

The network forensic computing is faced with the question of the complex network intrusion analyses. So a new concept of cooperation forensic computing is defined. Through to extend the theory of function dependency, a new method called probability function dependency relationships is proposed. Combined it with the Bayesian network and K2 algorithm, the network forensic computing algorithm called CFA is proposed. For the complex network attack, CFA is able to synthesize the various forensic data resource to reappearance the crime scenario intuitionally and realize the network forensic analysis effectively.

1. Introduction

For complex network attack, the attack behaviors are usually stepping stone, levity and synthetically. In order to reappearance the crime scenario and to cognizance the crime evidence, we must to associate analysis to data captured form various network safety equipments. For complex law case, in fact, the court is more attention to the association among evidence. So the independence evidences with cause relationship and cognizance each other are benefit to reconstruct the attack process in network crime. We define the concept of cooperation forensic computing as following.

Definition 1. Cooperation forensic computing indicates to discovery, correlate, explain and analysis information from all available system resource in order to confirm cause relationships among evidence, and reappear the network crime scenario, and form the chain of custody to prosecute.

Network forensic system need to save massive data from various resources. These data are not means the forensic evidence. Generally, it may consider as the suspect evidence. The cooperation forensic computing is a post forensic analysis after event. Through to analysis the suspect evidence, it reappears the crime scenario to find the really crime evidence.

This paper proposes a cooperation forensic computing algorithm called CFA based on Bayesian network. Based on probability function dependency (PFD) we proposed, CFA improves to K2 algorithm to fit the network evidence analysis. CFA is capable to synthesize multi data resource to reappear the crime scenario.

2. The basic ideas of CFA

We know that the current network attack is not an independence behavior. It is usually a multi-step attack finished by many attack steps. A really attack process is shown in figure 1 [1]. This process has address probe, port scanning, password file getting, password broken and system login five attack steps.

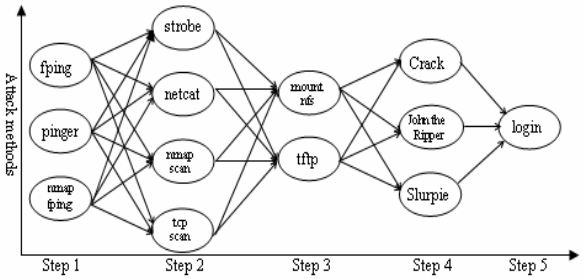

Figure 1. Example of the multi-step attack

Each node in figure 1 denotes one attack method. Each attack step has many methods to be chosen by attacker who will implement the step by one kind of it.

* This paper is supported by the Academic Natural Science Research Project of Education Department of Jiangsu Province, China (Grant Number 06KJD520019)

e-Forensics 2008, January 21-23, 2008, Adelaide, Australia.

The forensic to this kind of complex attack is a new challenge to the network forensic. The network forensic hasn't studied this question yet.

Indeed, Figure 1 can be explained as a cause relationship graph. We need causal data mining in massive information in order to discovery it. But almost all data mining algorithms, such as association rules mining, aggregation, classification and so on, are mining statistical relationship. None of them considers the causal architecture of objects.

The development of causal Bayesian network learning theory provides a chance for causal knowledge discovery. The knowledge discovery based on causal Bayesian network can reveal the matter's causal essentials. It discovers causal knowledge more high level than statistical data. In fact, figure 1 is a classical hidden Bayesian network with three latency variables between step 1 and step 2, step 2 and step 3, step 3 and step 4.

CFA is a cooperation forensic computing algorithm for causality analysis based on Bayesian network. Through to analysis to the meta-alert event synthesized with multi primary alert or other meta-alert, it can construct an alert Bayesian network to reveal the causality among meta-alerts, consequently to construct the crime scenario and to prosecute the post forensic after event. The setup of CFA is:

S1: Data preprocessing, the primary events from different data resource will be normalized to IDMEF format after data preprocessing.

S2: Clustering the normalized primary alert to meta-alert.

S3: Causality analysis to the meta-alert with Bayesian network.

S4: Reconstructing the crime scenario.

3. CFA algorithm analysis

CFA is based on Bayesian network and probability function dependency theory. Pearl proposed Bayesian network firstly. It includes Bayesian network structure (G) and conditional probability (CPD). Each arc of G indicates a probability dependency relation. Each node of G indicates an attribute variable. The variable may be continued or discrete. It's usually expressed with conditional probability table (CPT) when it's discrete.

3.1. Learning Bayesian network

Learning Bayesian network is to find the most approximate network to fit the training dataset. Learning is easy when the structure has been provided by the domain experts, unless it had to be induced from training dataset. The network derived from training dataset is an isomorphic graph of really Bayesian network.

Let D be n dimensions training dataset, $V=\{V_1,V_2,...,V_k\}$ is nodes set of Bayesian network. The state v_i of discrete variable V_i corresponds the v_i possible model structure. Let p (v_i) be conditional probability of it. The CPT of V_i denotes the probability distribution of p ($V_i|D$).

Definition 2. If there is an edge from V_i to V_j, V_i is the parent node of V_j; V_j is a successor of V_i.

Assumption 1. Conditional independency assumes. Every variable in Bayesian network is independency to the non successor node. That is to say, given the parent of V_i, each node V_i is independency to any nodes subset which is congregated by non successor nodes of V_i.

Let A (V_i) be a node set of not V_i successor in the graph, Pa (V_i) be direct parents of V_i, then:

$p(v_i|A(V_i), Pa(V_i))= p(v_i|Pa(V_i))$

Definition 3. A Bayesian network is defined as three truples (G,D,P), here, $G=(V,E)$ is a direct acyclic graph with node $V=\{V_1,V_2,...,V_k\}$ and edge E, P is a probability distribution, D is instance space.

Appling chain rules and Markov condition, the union probability of all nodes yields the following equation under assume 1.

$$p(V_1,V_2,\cdots V_k)=\prod_{i=1}^{k} p(V_i \mid Pa(V_i)) \qquad (1)$$

Definition 4. If node V is no parent, its probability is independence to any other nodes, and p (V) is called prior probability of it.

Learning Bayesian network has two kinds of algorithms which are research based estimated and independency verification. The classical algorithm based on research is K2 algorithm proposed by Cooper [2]. In the case of the node order given, K2 algorithm utilizes the Bayesian probability as marking to estimate the match degree between model and data, and to find the best network structure with greed search methods through adding edges continuously.

Let $Z=\{x_1,x_2,...,x_k\}$ be the variables set of D, B_S represents an arbitrary Bayesian network structure containing just the variables in Z. Parents of x_i denote as π_i, x_i has r_i possible value assignments: $(v_{i1},\ldots,v_{ir_i})$. Let ω_{ij} denotes the jth unique instantiation of π_i relative to D, there are q_i such unique instantiations of π_i. According K2 algorithm, then:

$$P(B_S,D)=P(B_S)\prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{(r_i-1)!}{(N_{ij}+r_i-1)!}\prod_{k=1}^{r_i}N_{ijk}! \qquad (2)$$

here, N_{ijk} is the number of cases in D, $N_{ij}=\sum_{k=1}^{r_i}N_{ijk}$

K2 algorithm is to search the maximum network structure of $P(B_S, D)$, but as the function of the number of nodes, the number of possible structures grows exponentially [3]. Obviously, an exhaustive enumeration of all network structures is not feasible in the most case. This paper proposes a method to reduction the variable set based on PFD, then we search the approximate Bayesian network structure in heuristic methods.

3.2. PFD theory analysis

People usually have some prior cognition to the function dependency (FD) of the attributes in the data set. Function dependency relationship is an important component of database normalization theory, and it has a mature theory system now. We will propose a probability function dependency (PFD) methods in this section. Relative to the PFD, the FD can be regarded as the PFD which the probability is equal to 1.

Definition 5. Let $R(U)$ be a relation pattern, X and Y are the subset of the attribute variables set U. The PFD is a predicate with format $X \xrightarrow{P} Y$, if r is the current relation of R, for arbitrary two instances t and s of r, $t[X]=s[X]$ implicates $t[Y]=s[Y]$ in P probability. Then, the PFD $X \xrightarrow{P} Y$ comes into existence in relation pattern $R(U)$.

Here, $t[X]$ denotes the value of instance t in variable set X. The concept of instance is the same of truple in relation database theory. In order to narrate easy in this section, we use attribute variable instead of variable.

Definition 6. Let F be the set of PFDs. The set which all element implicated by F is called clouser of F and signed with F^+.

$$F^+ = \{ X \xrightarrow{P} Y \mid F \mapsto X \xrightarrow{P} Y \}$$

Theory 1. If $\{A_1, \ldots, A_n\}$ was the attribute variables set of relation pattern R, the sufficiency and necessary condition which $X \xrightarrow{P} A_1, \ldots, A_n$ comes into existence is $X \xrightarrow{P} A_i$ ($i=1, \ldots, n$).

Definition 7. Let F be the PFDs set on attribute set U, X be a subset of U, and then the clouser of X signed with X^+ denotes the set which all element A_i are satisfied $X \xrightarrow{P} A_i$ and reasoned with the inference rules . It can be expressed as following equation:

$$X^+ = \{A_i \mid X \xrightarrow{P} A_i \subseteq F^+\}$$

Algorithm 1. To resolve the clouser X^+ of attribute variable X on F.

Input: The attribute variables set U of relation pattern R.

The probability function dependency F on U.

The subset of X.

Output: the clouser X^+ of attribute variable X on F

S1: $i=0$, $X(i)=X$;

S2: In F, searching the unused element which the left item is the subset of $X(i)$:

$Y_j \rightarrow Z_j$ ($j=1, \ldots k$), $Y_j \subset X(i)$;

In Z_j , searching the attribute variable set A not emerged in $X(i)$, let $X(i+1)=X(i)A$. If no such A then loop S4.

S3: IF $X(i+1)=X(i)$ Then loop S4 otherwise loop S2.

S4: input $X(i)$，that is X^+.

In algorithm, there are four quit conditions for step S3.

- $X(i+1)=X(i)$.
- When $X(i)$ includes all attribute variables.
- In the left item of every PFDs in F, we can't find any attribute variable not emerged in $X(i)$.
- In unused attribute variable set of the left item of each PFDs in F, there are no subset of $X(i)$.

Definition 8. Suppose there two PFD sets F and G on relation pattern $R(U)$, if $F^+ = G^+$, then F and G are called an equivalence PFD set and signed with $F \equiv G$.

Theory 2. Suppose F is a PFD set on relation pattern $R(U)$. F_{min} is the minimum PFD set if F_{min} was satisfied the following four conditions:

- $F_{min}^+ = F^+$.
- The right item of every PFDs is a single attribute variable.
- There are no redundant attribute variables in the left item of every PFDs. That is to say, if X had a proper subset W made $F-\{ X \xrightarrow{P_1} Y \} \cup \{ W \xrightarrow{P_2} Y \}$ equal to F, then there are inexistence PFDs $X \xrightarrow{P_1} Y$ in F.
- There are no redundant PFDs in F_{min}. That is to say, there are no such probability function dependence $X \xrightarrow{P_1} Y$ made F equal to $\{F-\{ X \xrightarrow{P_1} Y \}\}$.

Obviously, each PFD set at least exists a minimum PFD set if F_{min}, and $F \equiv F_{min}$.

Algorithm 2. To resolve the PFD set F_{min} of F.

Input: A PFD set F.

Output: The minimum PFD set F_{min}.

S1: Appling the resolution rules in F, make the left item of every PFDs simplification.

S2: Checking the non single attribute PFDs of left item one by one in F, take out the redundant attribute variables of left item of every PFDs.

S3: Taking out the redundant PFDs. That is to say, taking out the first PFDs supposed as $X \xrightarrow{P_1} Y$ in

F, and resolving the X^+ in remain PFDs. If X^+ includes Y, then takes out $X \xrightarrow{P_i} Y$, otherwise keeps it.

According analysis in this section, when we know the prior PFD F of D, we can resolve the minimum PFD with algorithm 1 and 2. Thus, the attribute variables not in this set can't become parent's node of the variables in the set. So the computing complexity of the Bayesian network shall reduce largely. In fact, prior PFD is existence all over, for example, the association rules set derived from data mining approach generally contain a PFD set.

3.3. Heuristic search methods

Through analysis the PFD relation of D, we can appoint the order for n variables. Such that, if x_i precedes x_j in the order, then the arc from x_j to x_i is prohibited, at the same time, the attribute variable not in F^+ can't become parent's node of the variable in F^+. Given such an order as constraint, there are only $2^{\binom{n}{2}} = 2^{n(n-1)/2}$ possible Bayesian network structures for n variables.

When n is larger, we apply a greedy search approach to modify the equation (2) to maximize $P(B_S, D)$. We assume that a node has no parents when the algorithm action, and then we incrementally add the parents which make the probability of the resulting structure increases maximization to nodes set. When the addition of no single parent can increase the probability, we stop adding parents to the node. The function of greedy search is following:

$$g(i,\pi_i) = \prod_{j=1}^{q_i} \frac{(r_i-1)!}{(N_{ij}+r_i-1)!} \prod_{k=1}^{r_i} N_{ijk}! \qquad (3)$$

The heuristic search algorithm K2_C combined with algorithm 1 and 2 shows in algorithm 3:

Algorithm 3. The heuristic search algorithm K2_C

Input: Training dataset D and The output of algorithm 2 (node numbers $l \le n$)

Output: Parents of each node.

For i = 1 **to** l

$\pi_i = \varnothing$;

$P_{old} = g(i, \pi_i)$; // Function g is computed by equation (3)

OKToProceed= true;

While OKToProceed and $|\pi_i| < u$ **do**

Let z be the node in Pred(x_i)-π_i that maximizes $g(i, \pi_i \cup \{Z\})$;

$P_{new} = g(i, \pi_i \cup \{Z\})$;

If $P_{new} > P_{old}$ **Then**

$P_{new} = P_{old}$;

$\pi_i = \pi_i \cup \{Z\}$;

Else OKToProceed = false;

End While;

Write (x_i ,π_i); // Input node x_i and its parents set π_i

End For;

In algorithm, if all factorial of equation (3) have been computed and stored to an array, we may compute 1 to (m+r-1) integer factorial and store the results to an array in O (m+r-1) time. Since N_{ij} is impossible more than m, there are no factor is more than (m+r-1)! in equation (3). Because parents of node x_i are less than l-1, function g shall be called is less than l-1 also, and then P_{new} shall perform all operations in O ($murl$) time. In addition, other sentences in While cycle operate in O (1) time, the times of every cycle is O (u), and For cycle needs l times. Consider all these factors, the time complexity of algorithm 3 is O (m+r-1) + O ($marl$) O (u) l= O (mu^2r^2l), and in the badly, it is O (mn^4r) when u=n and l=n.

4. Experiment and analysis

4.1. Meta-alert aggregation

We use the alert aggregation method to merge primary alert to meta-alert based on attribute similarity and context required. The format of primary alert is: (AlertID, Classificaton, SrcIP, DseIP, detectime, hyperID), the format of meta-alert is (HyperID, Classification, SrcIP, DesIP, StartTime, EndTime, Count). The process of alert aggregation shall remove the repeat alerts and result the meta-alert remained the importance information granularity. Each meta-alert has a one-to-many relationship with the primary alerts. In primary alert, 'Count' field records the numbers of primary alerts merged into meta-alert. 'HyperID' field records the unique identification code of the meta-alert it merged into.

Meta-alert aggregation algorithm is based on Leader-Follower model [4]. The aggregation criterions is that the alert which has the same source address and destination address are merged into a same class, and the timestamp falls in a self-extending time windows. When the primary alert timestamp is over the limit of windows, it will self-extend within the limit T predefined. This is means of the continuous duplicate primary alerts can be merged into a proper meta-alert with time windows T.

4.2. Results analysis

We use two datasets to test the algorithm. One is ALARM network dataset which is used to experiment compared with K2 algorithm [5]. It has 10000 cases used by Cooper. Another is DARPA2000 dataset for intrusion scenario correlation evaluation which is used to experiment compared with forensic computing [6]. The computer is DELL OptiPlex GX280, and the OS is Windows Vista in experiment.

ALARM contains a total of 37 nodes and 46 edges, and each node has from two to four possible values. In order to easily compare and verify the algorithm in the experiments, the training dataset has used 100、500、1000、2000 and 3000 five sizes respectively. For every dataset, we compare the performance which measured in means and standard deviations with the structure of ALARM about adding edges, omitting edges and computation time of network structure. The experiment results show in table 1. In the experiments, we don't consider to compare the union probability computation time, because it is difficult obviously to select a standard to measure. We can see from the results that the accuracy and the computation time are improved in evidence. But according analysis of previous section that the computation time is correlation with the prior cognition, so the more accuracy about the prior cognition, the more less computation time it is.

Table 1. The results of comparison analysis

Algorithm	Dataset sizes	Adding edges		Omitting edges		time（secs）
		m	s. d.	m	s. d.	
K2	100	0.75	1.28	0.22	0.48	321.00
K2_C		0.19	0.40	0.62	0.86	130.00
K2	500	0.22	0.42	0.11	0.31	2213.00
K2_C		0.19	0.40	0.22	0.48	1077.00
K2	1000	0.11	0.31	0.03	0.16	6783.00
K2_C		0.24	0.49	0.22	0.48	4909.00
K2	2000	0.05	0.23	0.03	0.16	9147.00
K2_C		0.19	0.40	0.11	0.31	6658.00
K2	3000	0.00	0.00	0.03	0.16	1848.00
K2_C		0.16	0.37	0.05	0.23	1249.00

2000 DARPA intrusion detection scenario specific data sets are presented by MIT Lincoln Laboratory. It contains data packets monitored in DMZ and Inside by Tcpdump. It has LLDOS1.0 and LLDOS2.0.2 two attack scenarios. In LLDOS1.0, attacker compromise the three hosts of Eyrie AFB utilizing Solaris sadmind vulnerability, upload the Mstream and attack a government Web site in DDoS method. LLDOS2.0.2 is similar to LLDOS1.0. However, the host vulnerability searching and Mstream uploading are stealthier than those in LLDOS1.0. Consequently, when the underlying intrusion detection system has vulnerability, we can verify the correlation algorithm effectively.

Our forensic experiment is in LLDOS1.0 dataset. We merge primary alert to meta-alert event and learn Bayesian network with algorithm 3. The results show in figure 2.

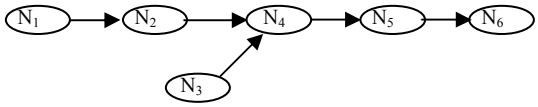

Figure 2. The Bayesian network of meat-alert for LLDOS1.0

Here, node N_1 to N_6 denotes meta-alert Sadmind_Ping, Sadmind_Amslverify_Overflow, Email_Almail_Overflow, Rsh, Mstream_Zombie, and Strem_DoS respectively.

In fact, LLDOS1.0 contains a multi-step attack process which attack sequence has five attack steps: (1) host probe with IPSweep; (2) Service port scan with Sandmin Ping to discovery the host which has Sadmind vulnerability; (3) intrude system utilizing host vulnerability and obtaine root control rights; (4) install Mstream on compromised host to prepare for DDoS attack; (5) launch DDoS attack in controlled host. The Bayesian network shown in figure 3 reconstructs this crime process. But we discover that node 3 is not a single attack step, it only has the probability function dependency relationship with Rsh, and how to remove such noise node is the problem of our next studies.

Excepting the function of reconstructing the crime scenario, the algorithm is good at in alert event detecting. We test two datasets in the same environment as the alert correlation algorithm proposed by Ning [4], the results shown in table 3 narrate our algorithm improves the detection rate and the false alert rate.

Table 2. The results for the intrusion detection

Dataset		Obser_vable attacks	Tool	Alerts	Detect_ed attacks	Detect_ion rate	True alerts	False alert rate
LLDOS1.0	DMZ	89	Real Secure	891	51	57.30	57	93.60
		89	CFA	56	52	58.43	55	1.79
	Inside	60	Real Secure	922	37	61.67	44	95.23
		60	CFA	42	38	63.33	41	2.38
LLDS2.0.2	DMZ	7	Real Secure	425	4	57.14	6	98.59
		7	CFA	47	5	71.43	45	4.26
	Inside	15	Real Secure	489	12	80.00	16	96.73
		15	CFA	59	13	86.67	56	5.08

5. Conclusion

All we know, the first widely DDoS attacking cause the research to alert correlation analysis technology in 2000. It clusters the alerts from the

same IDS according to the similarity of the alert attribution values in the early [7]. This method can't reveal the causality among the alert information. Further more, researchers studies the intrusion event correlation method [8]. It extends the detection objects of IDS especial to the huge switching network system.

Recently years, for complex attack in Internet, a graphics method, called reconstruction attack scenario [9] or attack graph [10], is proposed to narrate the attack process. In fact, these kinds of algorithms are to construct the correlation graphic of hyper alert based on data fusion technology, the scenario so-called refers to the correlation graphic but not the crime scenario in network forensic. Reference [11] presents a network forensic method with the evidence graph according the network forensic requires. But it is only a visual method to the classification alert events.

In this paper, through combined the alert correlation technology, Bayesian network leaning methods and probability function dependency theory with network forensic, we present an algorithm that is able to cooperate forensic computing. The computer forensic is a professional work, it will be intuitionist with Bayesian network to reconstruct the crime scenario. We will visualize the algorithm in the next work so that to use it widely.

6. References

[1]BAO Xuhua, DAI Yingxia, FENG Pinghui, ZHU Pengfei, and WEI Jun, "A detection and forecast algorithm for multi-step attack based on intrusion intention", Journal of Software, 2005, 16(12), pp.2132-2138

[2]G. F. Cooper, "A Bayesian method for the induction of probabilistic networks from data", Machine Learning, 1992, 9(4), pp.309~347

[3]R. W. Robinson, "Counting unlabeled acyclic digraphs", In C.H.C. Little (Ed.), Lecture notes in mathematics,622: Combinatorial mathematics V. New York: Springer-Verlag, 1977

[4]P. NING, C. YUN, and S. Reeves. Douglas, "Constructing attack scenarios through correlation of intrusion alerts", In Proc. of the 9th ACM conference on Computer and communications security, 2002, pp.245-254

[5]I. A. BEIULICH, H. J. SUERMONDT, R. M. CHAVEZ, and G. E. COOPER, "The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks", In Proc. of the Second European Conf. on Artificial Intelligence in Medicine , London, England, 1989, pp.247-256

[6]MIT Lincoln Laboratory , "2000 DARPA Intrusion Detection Scenario Specific Data Sets", http://www.ll.mit.edu/IST/ideval/data/2000/2000_data_index.html

[7]H. DEBAR, and A.WESPI, "Aggregation and correlation of intrusion detection alerts", In Proc. of the 4th International symposium on Recent Advances in Intrusion Detection(RAID). Leture Notes in Computer science 2212. Springer-Verlag,2001,pp.85-103

[8]D. ANDERSSON, M. FONG, and A. VALDES, "Heterogeneous sensor correlation : A case study of live traffic analysis", The 2002 IEEE Information Assurance Workshop , West Point , NY, US , 2002, pp. 186-204

[9]O. DAIN, and R. CUNINGHAM, "Building scenarios from a heterogeneous alert stream", In Proc. of the 2001 IEEE Workshop on Information Assurance and Security, 2001, pp.231-253

[10]P. NING, D. XU, C. HEALEY, and R. St. AMANT, "Building attack scenarios through integration of complementary alert correlation methods", NDSS, February 2004, pp. 285-314

[11]W. WEI, and T. E. DANIELS, "Building evidence graphs for network forensics analysis", Computer Security Applications Conf., 21st Annual (ACSAC), 2005: pp.254-266