Applied Mathematics & Information Sciences
An International Journal

Approach to Security Attack Pattern Networks on the Basis of Bayesian Networks

Chao Xu[1], *Guangquan Xu*[2], *Xiaohong Li*[2], *Zhiyong Feng*[2] *and Zhaopeng Meng*[1]

[1]School of Computer Software, Tianjin University, Tianjin, China
[2]School of Computer Science and Technology, Tianjin University, Tianjin, China

Received: 26 Aug. 2012, Revised: 27 Nov. 2012, Accepted: 5 Dec. 2012
Published online: 1 Feb. 2013

Abstract: Researchers are becoming more and more interested in the security issues of software engineering. It will effectively reduce the cost of development and maintenance in order to detect and predict security threats. In this paper, attack patterns are analysed in the field of software engineering, and Bayesian Networks is applied to construct attack networks topology, to find the dependencies of attack patterns. It helps to find the vulnerable points, locate the path of security threats effectively, and predict probable attacks reasonably. We use multi-variant statistical analysis for the attack networks, and factor analysis is applied to reduce the relevance. In Dirichlet distribution, the state transition distribution of each attack node is calculated to detect and predict the security threats. In order to verify the effectiveness and robustness of the approach, buffer flow is chosen as the analysis domain, and 14 attack patterns are selected for the experiments. It shows that attack patterns are effectively modelled based on Bayesian Networks and potential attack patterns are discovered, while threats are predicted and located accurately.

Keywords: Bayesian Networks, Attack Pattern Networks, Attack Path.

1. Introduction

In recent years, security is necessary for software engineering because of a rising of criteria for defects design in software development. Software engineers should understand design model clearly and know security threat caused by design flaw or program technique [1]. Because of the lack of adequate security control of attack threat, how to predict the risk object and locate the origin of the attack is still the most important phrase in the security software engineering [2].

In order to give a significant treatment on security software, experts collect plenty of attack data and deduce attack patterns in vulnerable fields [3]. Management methods are proposed to deal with security issues. For the threats, it is important to design system model and process algorithm so as to handle the vulnerability to security attacks.

Attack patterns are representative attack paths or the strategies, and they are useful models to study the origin and the development of security threat [4]. Not only can it increase the accuracy of detecting and locating the security [5], but also we can predict attack trend and decrease the possibility of security threat [6].

For attack patterns, researchers follow a two-step approach: (a) describe the software active processes, (b) compare with recognized attack patterns and find out attack threat [7]. In order to perform the procedure, some approaches are commonly used to describe the attack patterns with the knowledge representation, comparison patterns with traversal search and location path with backtrack reasoning [8]. The current available attack threat control systems offer the attack descriptions and retrieval of security threat, and some even offer the assistant control of potential security threats [9].

As utility of attack patterns is depended on the experience of software engineers, it is helpful to less-experienced developers through simplifying attack patterns representation and enhancing the ability of comparison patterns with traversal search and backtrack reasoning [10].

In order to give reasonable attack patterns representation [11], Bayesian Networks is applied to construct attack networks. In view of the uncertainty and dynamic characteristics for security attack threats based

* Corresponding author e-mail: xuchao@tju.edu.cn,losin@tju.edu.cn

on complicated model design, we focus on dynamic information flowing between software system and attack pattern diagrams [12]. It is suitable to apply Bayesian Networks to deal with the uncertainty and probabilistic reasoning.

In general, Bayesian Networks is one of most powerful tools for developing and dealing with the uncertainty and probabilistic reasoning [13]. It can describe the relationship between the probabilities based on digraph, combine with information in real-world and sample data, and describe the uncertain logic information by polymorphism logic at the same time. In our case, different attacks patterns of the same security field contain different threat information, and the attack patterns in the field are inevitably related, such as side-by-side or cause-effect relationship.

The contributions in this paper are that it is not based on attack patterns to compare with software process directly, but to follow a new tendency in re-constructing and achieving attack patterns networks with Bayesian Networks. There are two novelties to apply Bayesian Networks: one is that the attack networks topology is intuitive to express qualitative information; the other is that it is quantitative to express most likely mode of attack security threats through the joint probability density.

In the new attack path networks, we analyze the relationships among different kinds of attack patterns including general attack pattern, strategy attack pattern and method attack pattern. And then existing attack patterns or new patterns are integrated in the software system.

In the attack path networks, posterior probability of the conditional dependence is calculated with Bayesian theorem can be applied to a variety of control factors, such as attack mode of the security threats in decision-makings including new attack model prediction and attack diagnosis location.

In this sense, it is convenient to categorize, integrate and recognize the new attack pattern in the attack networks. In order to perform probabilistic reasoning, various types of data are applied to express the joint distribution. And the reasoning process can be summarized as: 1) it determines the attack probabilities based on attack networks topology and priori probability of attack patterns; 2) it recognizes the attack mode to maintain and predict the security threats probability in an attack.

In this paper, the attack pattern network to detect, locate and predict security threat is proposed and discussed. With the uncertainty of security software engineering in attack pattern, Bayesian Networks are applied to realize the relationship between attack patterns, security threats prediction and research. We devote section 2 to explain how to determine attack patterns and design model of attack networks, and devote section 3 to expose the security attack pattern platform and experiments.

2. Bayesian Networks Based Attack Pattern Networks Model

In the research of security software engineering in recent years, we analyze attack patterns and try to find out the dependence among security threats. There are three dependence relationships: they are parallelized between unrelated threats, collaborated between related attack patterns, or to excavate the granularity of attack patterns. Attack patterns can be used as sub-patterns, fragments, methods, or approaches of security threats. We try to find out effective attack paths from the existing software engineering, and give a more effective combination of maneuverable and trustable attack patterns.

The Bayesian Networks model comprehends the characteristics of treatment based on the literature of the attack mode. Those attack patterns can be analyzed in some particular fields. For example, taking the buffer flow as the security domain, a qualitative and quantitative analysis of attack patterns given by experts of the field can be considered as computational issues.

According to the characteristics of Bayesian Networks, the description of attack patterns can be converted to Bayesian Networks, and the network topology is intuitive to show the translative relation of attack patterns. And it even shows the implicated relationships among different granularity of attack patterns. The translative attack path networks based on Bayesian Networks can represent knowledge and reasoning knowledge precisely in uncertainty of the domain.

The uncertainty of the domain mentioned here is buffer flow, and sample data is taken as the realistic data in run-time environment. Logical algorithms are designed to achieve guaranteed and feasible location and prediction. However, it is impossible to know all views of the situation in buffer flow, since each field is related to a wide range of applications. For instance, buffer flow may occur in desktop system protection or in the network security.

Different domains may even cross-attack, such as the buffer flow may be used as a secondary attack in other fields, or simply an attack strategy rather than a real attack process. Therefore, we cannot grasp the true meaning and intent of attack patterns in reality. What we can do is to analyze and reasoning uncertain security domain. And section 2.1 is detailed to construct the model of attack networks based on Bayesian Networks.

2.1. Model Attack Path Networks with Bayesian Networks

A large number of algorithms for reasoning in uncertain environments have been proposed in recent years. With the semantic representation such as the joint probability distribution and the conditional independence

relationships, Bayesian Networks achieve useful reasoning for uncertainty.

Bayesian Networks can be applied to represent the independence of threats in software engineering and simplify the probabilistic representation of security attack path. The proposed method in this paper captures the uncertain knowledge in a natural and effective manner and improves the ability of probabilistic reasoning for attack path networks.

In the process of modeling security software engineering base on Bayesian Networks, the steps are needed to be fulfilled as: 1) reclassify the attack pattern according to the threat performance and dependence described by attack patterns to compute the evaluation weights for the index of attack patterns key points (KP) in Bayesian Networks; 2) standardize the weight-value of attack patterns key points [14]; 3) construct attack path networks based on security threat and find out the optimal topology. Bayesian Dirichlet equivalent (BDe) is applied to evaluate the network, and simulated annealing algorithm is used to search the optimal network topology; 4) compute the joint probability distribution of attack points and the conditional probability of each attack pattern so as to locate the security attacks and to predict the security threats.

The security attack networks with Bayesian Networks and graph theory can analyze probability, express and reasoning the uncertainty of the knowledge. The attack network is acyclic digraph, and the conditional probability of the node and its parent indicate and reflect the dependence intensity of KP accurately in attack networks.

After standardization of KP, define variable set $K = \{KP_i\}$ and its joint probability distribution P for the attack networks N to be constructed. Each variable in K is corresponded to a node in N and satisfy the following probabilistic condition

$$P(KP_1, KP_2, ..., KP_n) = \prod_{i=1}^{n} p(KP_i | Parent_i). \quad (1)$$

where n is the number of variables in K, $Parent_i$ is the parent node of KP_i , then the combination of N and P is Bayesian Networks based attack networks, which needs to satisfy the following condition

$$P(KP_1, KP_2, ..., KP_n) = \prod_{i=1}^{n} p(KP_i | KP_1, KP_2, ..., KP_{i-1}). \quad (2)$$

and each variable in K needs to satisfy Markov independence condition, that is

$$p(KP_i | KP_1, KP_2, ..., KP_{i-1}) = p(KP_i | parent(KP_i)). \quad (3)$$

If the structure of attack networks is known and the data is completed, such as $KP_i = kp$, then the distribution of $p(KP_{j \neq i} | KP_i = kp)$ in each node can be calculated, and the max distribution is as

$$KP_N = arg\{max_{KP_i} P(KP_1, KP_2, ..., KP_n | KP_i) P(KP_i)\}. \quad (4)$$

If the data is not completed, Expectation-Maximization algorithm can be used to study the parameters of the network, and to achieve the joint probability distribution of each variable in KP.

We use the expert knowledge of security attack model and statistical tests to establish the structure of attack path (AP) networks. There are different personally preferred algorithms of the network structure study in different areas of the attack models. Here are the main steps of modeling AP with Bayesian Networks:

Step 1:
Initialize attack networks APN_0 with expert and experience knowledge as, and set the parameters randomly;
Step 2:
Set the stop conditions of searching optimal topology, such as the construction time or the number of iterations;
Step 3:
Apply BDe to learn APN_0 and the score of network as $score_0$ which is the maximum likelihood estimate;
Step 4:
Generate new structure as APN_i by adding, deleting or reversing an edge in APN_0 ;
Step 5:
Calculate the scores of APN_i and find the network with max score as APN_{max} , label its score as $score_{max}$;
Step 6:
If $score_{max}$ is larger than $score_0$, then save the structure of APN_i and APN_{max} ; otherwise turn to **Step 8**;
Step 7:
Set $APN_0 = APN_{max}$ as the new initial attack networks and turn to **Step 3**;
Step 8:
Stop the iteration and output APN_{max} .

APN_{max} is applied to construct the final attack pattern networks, and to compute the joint probability distribution of KP nodes. Location and prediction of attack patterns can be conducted on the basis of superior attack networks APN_{max} . In order to set up a reasoning prediction model, we use a quantitative estimate of the attack probability range prediction in the section.

2.2. AP Goals, Performance Rankings and Standardization with Weight-State Value

The classified and the standardization process of weight-state of attack path data set is introduced in this section, since attack patterns in a specific security field are needed to be classified by attack targets and effects. The weight of attack patterns can be determined based on the classification and the KP can be identified with its weights. KP can be set as the parameters of the research

domain need to be discussed, and the training data are standardized with the weight-state value as the state set.

First of all, attack patterns are divided into three categories according to the attack targets and effects: 1) general attack pattern; 2) strategy attack pattern; and 3) method attack pattern.

The general AP is defined as an overview, which is mainly used to explain what category is the attack target in the composed AP set. It affects only a small portion of the APs key components. If these key goals are met, there will be a real risk, then the domain system will shut down and the AP will directly impact the whole system attack. This attack pattern has the maximum weight value roughly define as $\omega_1 \in (0.4, 0.7)$.

The strategy AP is defined as an instructional AP, which only gives a solution or direction of attack, but it does not include main attack methods. The strategy AP is important for the existence of the key components. A large number of other keys may be affected if these goals are met, this type of the attack pattern has a guidance to the attack threats and has the weight value of $\omega_2 \in (0.2, 0.5)$.

The method AP is defined as a specific AP, which can exist as the sub-fragment of the AP. It is critical to the existence of one AP and possibly the other AP key components. This attack pattern has specific method of attack or threat of attack fragment. It has a direct impact on the attack threat with the weight value of $\omega_3 \in (0.0, 0.3)$.

According to the above definitions of the weight value of the attack patterns in fields of security engineering, each AP is classified to category and is given weight value. As a matter of fact, all the weight values of AP should satisfy $\sum_{i=1}^{3} \omega_i = 1$. The weight-state values of KP in each AP are calculated according to the categories of APs. For example, the j-th weight-state value of the KP in the i-th AP described as $KP(i, j)$ can be calculated as

$$KP(i,j) = \frac{sample(KP_{i,j}) \times \omega_i}{\sum_{k=1}^{3} \sum_m ((sample(KP_{k,m}) \times \omega_k)} \times 2^t. \quad (5)$$

where $sample(KP_{i,j}$ is the j-th data value of the KP in the i-th AP; ω_i is the weight value of the i-th category of AP; 2^t is the value range of the KP.

With the formula (5), data can be standardized with weight-state value, and it is important to find the reasonable argument set which will be the KP node of the Bayesian Networks.

Considering the description of the field of AP, a preliminary set of KP can be selected based on the opinions such as the threat target of the attack pattern and the performance of KP. Since KP is state transition point, each of them maintains the probabilistic distribution of its parent. The analysis variables may be redundant according to selected KP, so we need to screen and refine them.

In order to find the KP nodes of the attack pattern, factor analysis in the multivariate statistical analysis is applied. This method uses a few factors to explain the relationship within the argument set, and can maintain the most structural information of the original AP. Then we can achieve the goals that to simplify and abstract the argument set.

Assume that there are p initial arguments of AP, and each one can be broken up into the linear combination of q common factor KP_j and one specific factor ε_i . The i-th AP can be described as

$$AP_i = \sum_j (x_{i,j} \times KP) + \varepsilon_i. \quad (6)$$

where KP_j is included in each argument, but only the i-th AP has ε_i . $x_{i,j}$ is the weight or factor loading of the j-th common factor in the i-th argument. X is pattern argument matrix. Since the common factor KP_j is independent to the factor ε , Maximum Likelihood is used to select the factors.

Assume $AP_i \quad MN_p(0, \Sigma)$ is satisfied, where $\Sigma = Cov(AP) = X\Phi X' + \Psi$ .With the logarithm of the likelihood function, formula (7) is as

$$\begin{aligned} ln(l) &= -\frac{1}{2} n[ln|\Sigma| + tr(\Sigma^{-1}S)] + c \\ &= -\frac{1}{2} n[ln|X\Phi X' + \Psi| + tr((X\Phi X' + \Psi)^{-1}S)] + c \end{aligned} \quad (7)$$

Calculating the max value of equals to the minimal value of $M = ln|\Sigma| + tr(\Sigma^{-1}S)$.

Solving the above formulas we get the result of factor analysis which is the KP set. Since KP set can be deal with the standardization process, then the weight-state value of each KP node can be calculated for the training set of attack path networks. In the next section, how to maintain the joint probability distribution of KP node and the reasoning process are discussed.

2.3. Joint Probability Distribution of KP

In this section, joint probability distribution of attack networks, effective location and prediction of security threats are introduced.

In the above construction of attack networks, BDe score is used to evaluate the network topology N in a given data set D. Assume each prior probability of parameter set S meets Dirichlet distribution, the maximum posterior probability score and parameter value will be found out in topology N and sample S by prior knowledge as

$$p(S|N) = \prod_{i=1}^{n} \prod_{j=1}^{a_{ij}} \frac{\Gamma(a_{ij})}{\Gamma(a_{ij} + N_{ij})} \prod_{k=1}^{KP_i} \frac{\Gamma(a_{ijk} + N_{ijk})}{\Gamma(a_{ijk})} \quad (8)$$

where a_{ijk} and N_{ijk} are super parameters, and $p(S|N)$ is the Bayesian metric.

In a given topology N, if sample value of KP is ς , the formula (9) can be used to calculate the posterior probability as

$$P(\varsigma|S,N) = \frac{P(S|\varsigma,N)P(\varsigma|N)}{P(S|N)} \tag{9}$$

where $P(\varsigma|N)$ is the prior probability of parameter ς in attack networks topology N.

Assume a_i is the super parameters set, $P(\varsigma|N)$ meets Dirichlet distribution, and the posterior probability of KP can be calculated as

$$P(\varsigma|S,N) = \frac{T(a)}{\Pi_i T(a_i)} \Pi_i (\varsigma^{KP_i})^{-1} \tag{10}$$

Joint probability distribution of each node in attack networks will be obtained formulas above. Attack location and prediction can be analyzed effectively with the analysis of the joint probability distribution, attack networks topology, combined with calculation of evidence dataset.

How to use attack networks and joint probability distribution for verification in the security software engineering will be shown in section 3. Field of Buffer Flow is selected as experimental domain to verify the security attack networks, locate the security attack and predict threats reasonably.

3. Experiments of AP Networks

In order to verify the attack networks approach proposed in this paper, we take a particular field (buffer flow leak) as examples for the experiments.

Access to analysis from the Bayesian Networks and the uncertainty of probability analysis, combined with a single field of the cause-effect model, we analyze and reason attack mode on the access network topology as: 1) node is on behalf of the attack mode key; 2) arc is on behalf of the relationship between key points to attack mode (reflected in the focus of transition probability); 3) relationship intensity between the node and its father is expressed by the conditional probability that the model accurately reflects the attack or the reliance on key points of internal relations, and reflect the attack through the conditional probability model of uncertainty information.

3.1. Experiment Design

The attack networks provide AP keys from internal and external domain fields, increasing collaboration, reducing unreliable components and generating high quality software engineering. Clarify and quantify the possibility that certain events will directly impact application component performance in AP context with key points.

This stage outputs the lists of all the risks and maintains their appropriate weight-values. Then we apply Dynamic Bayesian Networks to define the cost-effective manner. Any suggested activities should be taken into account cost, implementation time, likelihood of success and completeness.

The experiment builds a model to locate and predict attack threats based on attack networks. The aim of the experiment is to find out the basic reason of security attack and to prevent the incoming security threats. Suppose that $DateSet = D_i$ is observed from the corresponding KP output nodes, and it can acquire the posterior probability P of all KP nodes. Select the key point which has the highest posterior probability or the most vulnerable KP, and check its running state. The result is added into the dataset as feedback evidence for location and prediction in the next iteration. The location and prediction algorithm is described as below.

Input:
Attack pattern network topology APN .
Prior probability P_0
Standardized $Dateset = D_i$.
For:
Calculate posterior probability of $DateSet$ Pds.
Select KP_{max} with the highest posterior probability.
If the running state of $KP_{max} = ERROR$ then
$APN = KP_{max} \cup APN$
Add KP_{max} to D_i , and conduct the next iteration.
Output:
Return Attack path and joint probability Distribution.

If the prior probability of the procedure above is unknown, they can be substituted in randomly. KP joint probability distribution maintained by the network will be ended. If the KP state which has the highest posterior probability is abnormal, then the KP is suffering from security attack threats.

Repeat the procedures and it can get the location of the attack path. Otherwise, the KP is vulnerable to attack threats that can acquire the prediction of threats in the attack networks.

3.2. Experiment Data and Preprocessing

The proposed algorithm has been verified and validated in some the security categories with 14 AP patterns: Overflow Buffers, Filter Failure through Buffer Overflow, Buffer Overflow via Environment Variables, Buffer Overflow in an API Call, Buffer Overflow in Local Command-Line Utilities, Overflow Variables and Tags, Buffer Overflow via Symbolic Links, String Format Overflow in syslog, Client-side Injection-induced Buffer Overflow, Overflow Binary Resource File, Buffer Overflow via Parameter Expansion, MIME Conversion, Accessing/ Intercepting/ Modifying HTTP Cookies, and Forced Integer Overflow.

For each type of AP, there are KP [15] described in literatures. All of them are pre-processed by factor analysis and validation. In this way, 32 KPs have been used to train and validate.

For the 14 APs within the experiment, each one has been manually annotated with KP and even the state transition process of recognized AP. The nodes are marked according to the topological order, i.e., ancestors before descendants'Select target data-1', 'Inject malicious data-2', 'Run attacker code-3', 'Program crashed-4', 'Access to privileges or data-5', 'Filter buffer overflow-6', 'Search environment variable-7', 'Manipulate environment variable-8', 'Exposed API interface-9', 'Malicious code calling the API-10', 'Enter code to command line-11', 'Change configuration variables-12', 'Create malicious link-13', 'Modify existing link-14', 'Point to invalid data-15', 'Deceive user to read Links-16', 'Wait for user to read link-17', "Inject illegal parameters-18', 'Create local server-side-19', 'Analyze client data-20', 'Return malicious data-21', 'Create malicious binary file-22', 'Wait to be downloaded passively-23', 'Replace original file-24, 'Load resource file-25', 'Extend program to inject data-26', 'Determine server patch-27', 'Send specific format e-mail-28', 'Intercept cookies sent to client-29', 'Intercept cookies sent to server-30', 'Parse Cookie-31' and 'Send falsified data to server-32'.

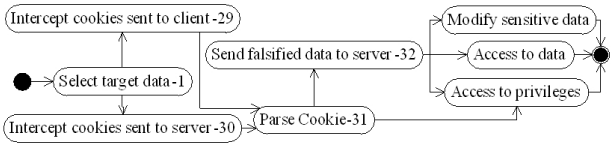

Figure 1 Initial information flow of Accessing/ Intercepting/ Modifying HTTP Cookies

The state transition process of the recognized AP is presented as the form of adjacency matrix. Figure 1 shows the initial information flow of Accessing/ Intercepting/ Modifying HTTP Cookies for the 13th attack pattern [16], and each node is the alternative key point at the beginning [17].

In addition to specify the attack networks structure, sample data is transformed to discrete values with discretization strategy. Experiment data is collected for the multi-factor analysis, and key factors are filtered and integrated for discretization.

Discretization is controlled by data collection. After monitoring discretizations numbers generated by KP, calculate the weight-state value. If the KP is discrete, its size is the number of possible values each node should be.

Attack networks consist of the structures and the parameters. The parameters are represented by Conditional Probability Table (CPT), which define the probability distribution of KP node given its parents. According to the approach proposed above, the weight-state value is described in Table 1.

Table 1 Weight-state value Indicators for KP

KP_i	Weight-State Value Indicators
i=8	Collaborative keys can infect components interface and code input to establish affective attack and threat.
i=7	Important interfaces for keys composition make AP easier which may create valid transactions.
i=6	High enforcement of access control rules allows misuse by inside and outside.
i=5	Keys can make others susceptible to denial-of-keys attacks.
i=4	Keys mechanism or requirement access may cause system crashed or unexplained behavior.
i=3	Keys do cover fault tolerance. Components failures are likely to kill the process.
i=2	Keys for attack detecting do not fully evaluate threat.
i=1	Keys do have access to attack directly.

3.3. AP Network Construction and Discussion

A graphical analyzing platform is implemented to support the approach to attack networks in modeling, analyzing, and evaluating the attack path, see to Figure 2. It is utilized to predict and locate the threaten nodes. The platform can be applied in the areas including desktop application, database development and internet project.

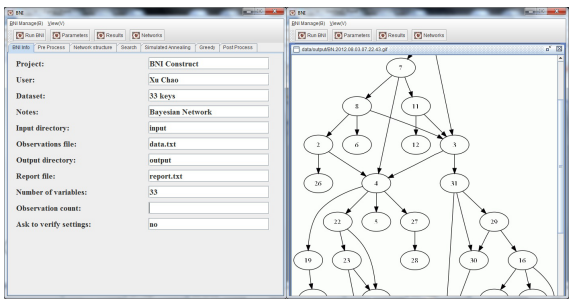

Figure 2 Graphical user interfaces of AP networks model tool

The experiment is built around the attack networks with Bayesian Networks which are taken as the example for some internet projects. From the evaluation results, the possible attack path is introduced in the networks whose probability of the attack sequence can be calculated separately.

In the Figure 3, the attack networks can show the whole situation of the tested project with the CPT results related to each attack path. Since the attack path prediction is very important to analyze and evaluate the internet project, the most important attack path can be pointed out to void the realistic attack and terrible threats.

For instance, we take attack path of 0-19-20-21-9-7-11-12-3-30 as the analysis example which stands for Root - Create local server-side - Analyze client data - Return malicious data - Exposed API interface - Search environment variable - Enter code to command line - Change configuration variables - Run attacker code - Parse Cookie.

From the attack path above, it is clear to see that the main direction to achieve the internet threaten need the path including nodes of 19, 20, 21, 9, 7, 11, 12, 3 and 30 whatever the attack type is. If one node in the path is security enough, the whole attack is useless, and the internet project is safe. For example, we can apply and take some of the methods to prevent the KP such as node 30 which is Parsing Cookie. Thus, the attack path networks based on Bayesian Networks is meaningful and important for software engineers.

As the analysis above, there are three key points for the attack path prediction: 1) the structure of the AP networks should be well-constructed; 2) the CPT should be computed and analyzed as accurately as possible; 3) the engineers should determine which points of attack nodes are critical for the whole software project.

The key point for AP networks is to construct the computational model for the enhancement of the security prediction with the attack path. The AP networks with Bayesian Networks describe and calculate the most possible attack path, which is suitable to help software engineers design and develop more security software project. Especially for the internet attack, the AP model proposed above can provide more safety information.

4. Conclusions

The importance of software security has become more and more profound, since most attacks to software systems are based on vulnerabilities caused by poorly designed and developed software. Furthermore, the enforcement of security in software systems at the design phase can reduce the high cost and effort associated with the introduction of security during implementation.

For these purpose, architectural security patterns have been proposed as the well-known design patterns. In this paper a method to construct security attack patterns in some specific category is presented. In order to find out the relationship among attack patterns, we assess the patterns on Bayesian Networks rules for attack patterns. Due to the uncertainty and complexity of attack patterns, we present a novel method based on Bayesian Networks extended for attack patterns that find out the deeper relationship among patterns.

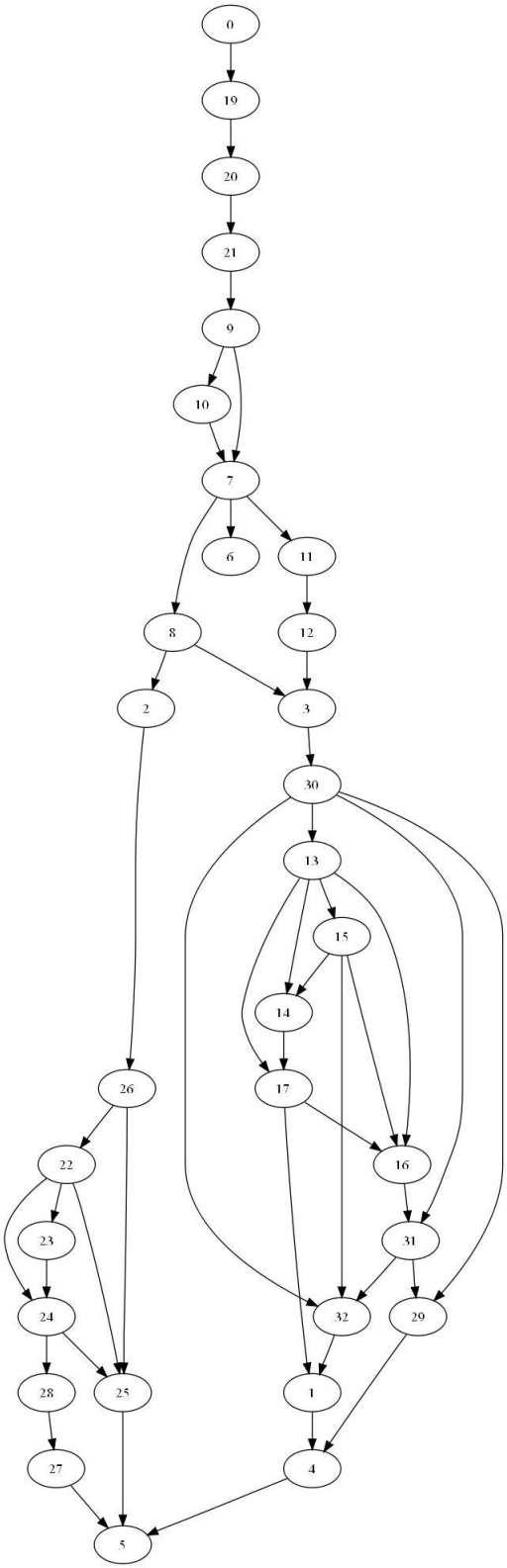

Figure 3 Example of attack path networks

The main goal of this paper is to perform attack patterns analysis. It is to determine what extent security patterns shield from known attacks. The information is fed to a mathematical model based on Bayesian Networks theory and multivariate statistics in order to compute the attack path topology. Within the attack networks, threaten path location and prediction are verified and validated, and the result is presented meaningfully.

Acknowledgement

This work is partially supported by the National 973 Foundation of China (No. 2013CB329304), the National 985 Foundation of China, National Science Foundation of China (No. 61070202, 61272106 and 91118003) and the Tianjin Science Foundation (No. 10ZCKFGX01100).

References

[1] M. Howard and D. LeBlanc. Writing secure code, second edition. Microsoft Press. (2003).
[2] F. Swiderski and W. Snyder. Threat modeling. Microsoft Press. (2004).
[3] A.P. Moore, R.J. Ellison and R.C. Linger. Attack modeling for information security and survivability. Technical Note CMU/SEI-2001-TN-001. (2001).
[4] B.H. Charles, L. Robin, D.M. Jonathan and N. Bashar. Security requirements engineering: a framework for representation and analysis, IEEE Transactions on Software Engineering, **34(1)**, 133-153, (2008).
[5] R. Kazman, L. Bass, M. Klein, T. Lattanze and L. Northrop. A basis for analyzing software architecture analysis methods, Software Quality Journal, **13(4)**, 329-355, (2005).
[6] B. Schneier. Attack trees. Dr. Dobb's Journal of Software Tools, **24(12)**, 21-29, (1999).
[7] D. Xu and K.E. Nygard. A threat-driven approach to modeling and verifying secure software, Proceedings of the IEEE/ACM International Conference on Automated Software Engineering, 342-346, (2005).
[8] G. Elahi, E.S.K. Yu and N. Zannone. A vulnerability-centric requirements engineering framework: analyzing security attacks, countermeasures, and requirements based on vulnerabilities, Requirements Engineering, **15(1)**, 41-62, (2009).
[9] J.P. McDermott. Attack net penetration testing, Proceedings of the 2000 Workshop on New Security Paradigms, 15-21, (2000).
[10] G.C. Dalton, R.F. Mills, J.M. Colombi and R.A. Raines. Analyzing attack trees using generalized stochastic Petri nets, Information Assurance Workshop, 116-123, (2006).
[11] O. Sheyner and J.M. Wing. Tools for generating and analyzing attack graphs, Proceedings of Formal Methods for Components and Objects, Lecture Notes in Computer Science 3188, 344-371, (2004).
[12] I. Hogganvik and K. Stolen. A graphical approach to risk identification, motivated by empirical investigations, 9th International Conference on Model Driven Engineering Languages and Systems, Lecture Notes in Computer Science **4199**, 574-588, (2006).
[13] B. Cheng-Gang. Bayesian network based software reliability prediction with an operational profile Original Research Article, Journal of Systems and Software, **77(2)**, 103-112, (2005).
[14] T.P. Kelly and R.A. Weaver. The goal structuring notation - a safety argument notation, Proceedings of the Dependable Systems and Networks 2004 Workshop on Assurance Cases, DSN 2004, (2004).
[15] K. Goseva-Popstojanova, A. Hassan, A. Guedem, W. Abdelmoez, D.E. Nassar, H. Ammar and A. Mili. Architectural-level risk analysis using UML, IEEE Transactions on Software Engineering, **29(10)**, 946-960, (2003).
[16] H. Chivers and M. Fletcher. Applying security design analysis to a service-based system, Software: Practice and Experience, **35(9)**, 873-897, (2005).
[17] J.jrjens. Secure systems development with UML. Springer Academic Publishers, Germany. (2004).

Chao Xu received his Ph.D at School of Computer Science and Technology, Tianjin University. He is currently a lecturer in Tianjin University. His research interests lie in Knowledge Management, Pattern Recognition, Security Software engineering, and Affective Computing.

Guangquan Xu received his Ph.D at School of Computer Science and Technology, Tianjin University. He is currently an associate professor in Tianjin University. His research interests lie in Security and Privacy, Network and Information Security.

Xiaohong Li received her Ph.D at Tianjin University. She is currently a professor in Tianjin University. Her research interests lie in Trusted Computing, Knowledge Engineering, and Security Software Engineering.

Zhiyong Feng received his Ph.D at Tianjin University. He is currently a professor in Tianjin University. His research interests lie in Knowledge Engineering, Services Computing, Security Software Engineering and Computer cognitive.

Zhaopeng Meng is currently a professor in Tianjin University. His research interests lie in CSCV-based Collaborative systems, Computer networks and applications, Computer distance education.