Using Genetic Algorithm for Optimal Security Hardening in Risk Flow Attack Graph

Fangfang Dai, Kangfeng Zheng, Binwu and Shoushan Luo
Information Security Center, Beijing University of Posts and Telecommunications
Beijing, 100876, P. R. China
[e-mail: daiff.bupt@gmail.com]
*Corresponding author: Fangfang Dai

Received March 6, 2014; revised March 4, 2015; accepted April 19, 2015; published May 31, 2015

Abstract

Network environment has been under constant threat from both malicious attackers and inherent vulnerabilities of network infrastructure. Existence of such threats calls for exhaustive vulnerability analyzing to guarantee a secure system. However, due to the diversity of security hazards, analysts have to select from massive alternative hardening strategies, which is laborious and time-consuming. In this paper, we develop an approach to seek for possible hardening strategies and prioritize them to help security analysts to handle the optimal ones. In particular, we apply a Risk Flow Attack Graph (RFAG) to represent network situation and attack scenarios, and analyze them to measure network risk. We also employ a multi-objective genetic algorithm to infer the priority of hardening strategies automatically. Finally, we present some numerical results to show the performance of prioritizing strategies by network risk and hardening cost and illustrate the application of optimal hardening strategy set in typical cases. Our novel approach provides a promising new direction for network and vulnerability analysis to take proper precautions to reduce network risk.

Keywords: Genetic algorithm, security hardening, attack graph, vulnerabilities, network risk

The authors would like to thank the anonymous reviewers for their detailed reviews and constructive comments, which help improve the quality of this paper. This work was supported in part by National Natural Science Foundation of China under Grant No.61101108 and No. 61121061.

http://dx.doi.org/10.3837/tiis.2015.05.021 **ISSN : 1976-7277**

1. Introduction

Information security environment has been experiencing tremendous shift over the last decade. The fact that capability of brute force and scale of botnet dominate network attack effect has been brushed into background. Adversaries tend to adopt complicated invasive actions to achieve goal with an information-driven precision instead of blindness of target selection [1]. All these changes have led to emerging threats and sophisticated attackers such as Advanced Persistent Threat (APT) and Determined Human Adversaries (DHA). When dealing with these sophisticated environment and determined adversaries, precise environmental model, interactive adversary pattern and advisable risk analysis are required for the sake of potential prevention.

Since manual security analysis is error-prone and tedious, and gradually becomes infeasible for large and complicated networks. Paradigms like attack graphs [2,3] and attack trees [4,5] have been commonly adopted by researchers to build security model and determine attack scenarios that could lead to damage. With the aid of such methods, security analysts are able to obtain concise representation of all the paths an attacker may follow to compromise a security goal through leveraging dependencies among known vulnerabilities [6]. However, while attack graphs can reveal threats, they do not directly provide solutions of security hardening. In practice, it is almost infeasible to remove all identified threats since the system administrator always has to work within a given set of fixed budget constraints. Moreover, under no circumstances should security mitigation measures affect the normal operation of network infrastructure. Therefore, the crucial question in defending against those nontrivial invasions is that: *which of the vulnerabilities should be removed to mitigate security risk to acceptable levels without causing a breakdown of normal services, where such removal incurs the least cost?* [7]

To make this specific problem from intractable to approachable, efforts have been spent in this context. Researches have been performed to seek for a trade-off between the cost of securing chosen vulnerability subsets and the residual damage caused to network if certain weak points are left unpatched [7,8].

The above research works motivate us to study the optimal security hardening problem. In particular, we first introduce a series of risk metrics and augment the attack graph to risk flow attack graph. This extension not only preserves the advantages of traditional attack graph to represent network state and vulnerabilities, but also depicts the adversary's status transition sequence. By encoding the attack pattern and defending strategy into binary sequences, we employ a multi-objective genetic algorithm in deriving hardening solutions. Performance of a solution is measured by a pair of risk function and cost function, the value of which is related to the risk flow attack graph metrics. This implementation enables us to revisit network risk by following the risk path of originating, transferring, redistributing and converging. We observe that this approach is able to achieve an optimal solution set of security hardening strategies which takes full account of residual risk and enhancement cost.

The rest of this article is organized as follows. Section 2 gives an overview of some related work. Section 3 describes a risk flow attack graph model to illustrate our method. Section 4 presents our approach of using multi-objective genetic algorithm for optimal security hardening in detail. The experimental results are presented in Section 5. Finally, Section 6 summarizes this paper and discusses future work.

2. Related Work

The network security hardening problem has been extensively studied in all manner of ways. Among the massive explorations, different variants of attack graphs have been applied by researchers. In [7,9], the exploit dependency graph was utilized to make assignments of initial network conditions, represent given set of critical resources as a logic proposition of these initial conditions, and compute actual sets of hardening measures to guarantee the safety of given critical resources. Their approaches worked from a new perspective of initial conditions rather than independent exploits. Ref [10,11] focused on achieving cost-effective security controls by exploring the logical attack graph to represent network observations. They identified vulnerabilities existed in network and explored their causal relationships. Similarly, [12-14] concentrated on accurately measuring risk for enterprise networks. They considered from the perspective of security defenders and made efforts to select the most effective countermeasures against multi-step network penetration such as zero-day exploits, client-side attacks, etc.

Ref [15] on the other hand, took not only the defender's cost, but also the attacker's strategy into account. Their approach modeled the attacker-defender interaction as an arms-race, and explored how security controls can be placed in a network to induce a maximum return on investment. They also developed a multi-objective approach to formulate the vulnerability-patching problem, taking advantage of an attack tree model and using an evolutionary algorithm to search for the solution.

Because the security hardening issue is influenced by many real-world elements, such as residual damage, network reliability, enhancement payoff, etc. It is feasible to model it into a multi-objective problem. Ref [16] provided such a formulation and emphasized that the settlement of specific vulnerabilities may introduce additional potential damage to network. Moreover, a genetic algorithm was adopted to choose the minimal-cost security profile providing the maximal vulnerability coverage. Ref [17] demonstrated a model of quantitative risk analysis, and deplyed a genetic algorithm to search for the best countermeasure combination, while multiple risk factors are considered. Apart from genetic algorithm, efforts have been spent to seek for more possibilities of solving the problem. Frigault et. al. [18] introduced security metrics into attack graph for measuring network security risks using dynamic Bayesian network, and Xie et.al. [19] extended their work by capturing uncertainties in attacker action, intrusion alerts, etc. Zhang et. al. [20] adopted Hidden Markov Model (HMM) instead. They constructed a quantitative model to specify cost factors and design heuristic algorithms for automated inference.

In summary, researchers have explored various possibilities on modeling vulnerabilities and analysing security. Yet there are still problems that remain unsettled. For example, some approaches define risk functions of static network metrics, which are always based on empirical statistics. Furthermore, most of previous works tend to work out an optimal solution of security hardening, which will limit the application of such methods. Because when facing diversified security goals, these methods have to be modified to meet the security hardening targets.

Our work has fundamental differences with previous works because it adopts risk flow attack graph to represent and simplify network observations rather than using complicated attack trees or attack graphs. It also develops a different way of calculating network risk by following the risk state transitions of two attacker prototypes in the attack graph. This calculation can dynamically relate network risk to the implementation of security precautions. Then, by relating the objective functions of a multi-objective genetic algorithm to the risk

function, our approach can arrive at a Pareto optimal set of hardening strategies. This optimal set can help security analysts to filtrate most inferior solutions and reduce the amount of alternatives by orders of magnitudes. Moreover, it's convenient for the optimal set to cooperate with different security goals and hardening constraints, because defenders can choose specific solutions flexibly from the optimal set to address practical security problems. Compare with previous works which also adopt genetic algorithms to work out the problem, such as [8][17] and [20], our method have three major differences. First, our multi-objective model is constructed on the basis of simplified risk flow attack graph rather than complex attack trees or attack graphs. Second, we introduce a fitness based crowd distance sort scheme to facilitate global convergence of genetic algorithm. Third, we employ an elite strategy in our method, which can help maintain diversity of individuals and guarantee the efficiency of algorithm.

3. Model and Background

As has been noted, network environment is becoming more and more sophisticated and attackers tend to adopt multiple actions to achieve goal. Among these prominent changes, which remains unaltered is the fact that vulnerabilities have always been one of the most favorable means for attackers to utilize when penetrating a network. In this section, we describe how to model and quantify network vulnerabilities as well as attack patterns using risk flow attack graph. For simplicity, we focus only on the application rather than the generation process of the graph. We also introduce some basic definitions and the approach of measuring security risk in risk flow attack graph. These elaborations are requisite for Section 4 where we show how to find efficient hardening method.

We consider the hypothetical network shown in **Fig. 1.** The setup consists of three servers and an internal host. The Database Server (DS) and the File Server (FS) are located inside the internal firewall, as well as the internal user on Host 1. Each dashed box on the server icon depicts services an external user can utilize to communicate with these servers. Other unauthorized accesses will be blocked by the external firewall policy. In addition, we assume that the adversary on Host0 intends to compromise the DS.

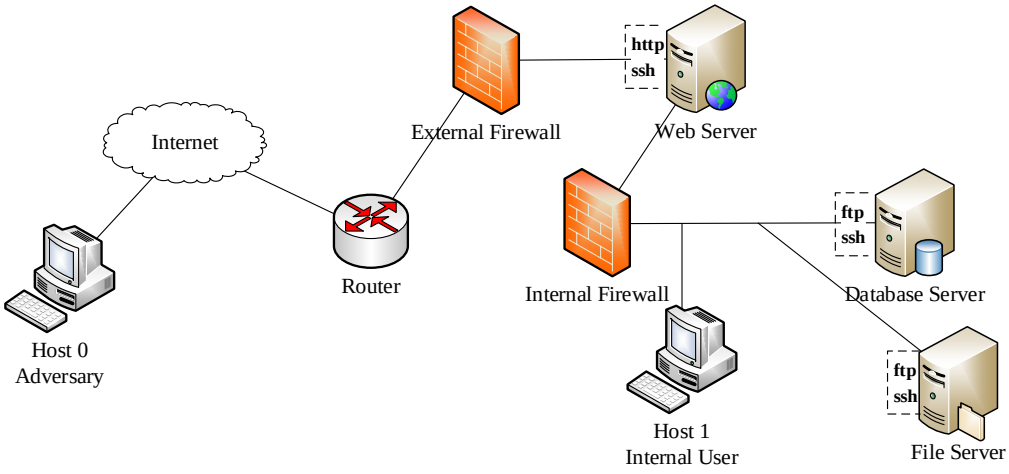


Fig. 1. Hypothetical network model

Formally, a risk flow attack graph can be described as a tuple,

$$RFAG = \{V, E, \tau, \mu, f\}, \text{ where}$$

1) $V : V = V_s \cup V_g \cup V_m$ constitute the set of nodes. V_s and V_g indicate initial capabilities and the ultimate goal of an attacker, respectively. V_m is the intermediate node set representing the status of individual network assets. Each element in the node set has a value of true or false, indicating whether this asset is compromised by an attacker.
2) $E : E \subset ((V_s \cup V_m) \times (V_m \cup V_g))$ is the set of edges in the graph. Each edge can be mapped to a vulnerability which can be exploited. The binary values of {0,1} can be assigned to indicate whether the corresponding vulnerability is utilized by an attacker to penetrate the network. Specifically, '1' stands for a successful penetration and '0' otherwise.
3) $\tau : \tau \subseteq (V \times V)$. An ordered pair $(V_{pre}, V_{post}) \in \tau$ if there exists an edge ε that $(V_{pre} \in pre(\varepsilon)) \wedge (V_{post} \in post(\varepsilon))$.
4) $\mu : \mu \subset (E \to Vuls)$ is a mapping from an edge to its corresponding vulnerability. The metrics of the vulnerability will help determine the risk on edge and of the network.
5) $f : f$ is the risk function defined on exploit edges, the calculation of f depends on specific application scenarios. Our definition of risk function will be given later in this section.

For better illustration, we exemplify the attack graph of hypothetical network model in **Fig. 2**, which is similar to that in [20]. Some modeling specifications have been made, such as defining two typical attack prototypes, emphasizing exploit edges and constructing risk function.

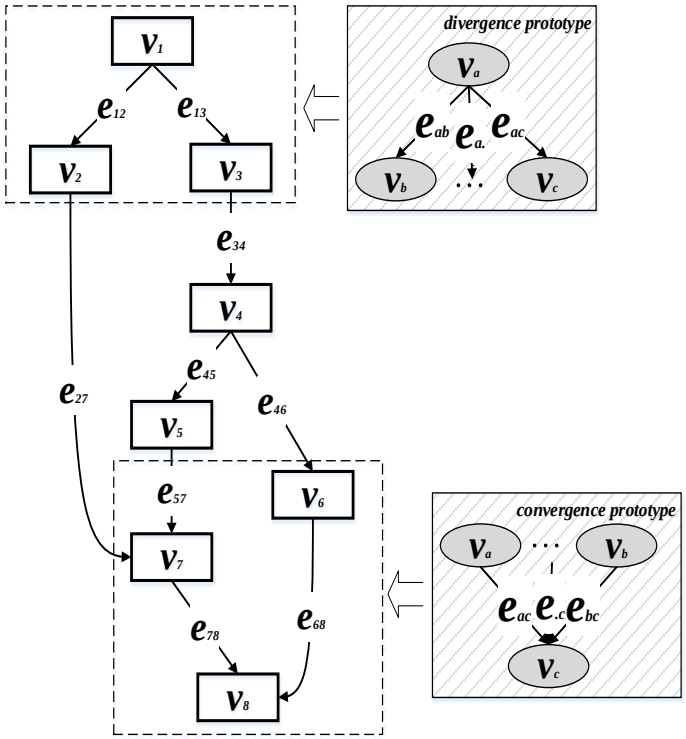


Fig. 2. Risk flow attack graph for hypothetical network

The corresponding nodes and edges are defined in **Table 1** to amplify the sample scenario.,

Table 1. Definition of nodes and edges

Node	Notation	Node	Notation
v_1	*user(0)*	e_{12}	*atk(h0,http,WS)*
v_2	*root(*WS)	e_{13}	*atk(h0,ssh,WS)*
v_3	*guest(*WS)	e_{34}	*atk(WS,ssh,h1)*
v_4	*user(1)*	e_{27}	*atk(WS,BoF,DS)*
v_5	*user(FS)*	e_{45}	*atk(h1,ftp,DS)*
v_6	*root*(1)	e_{46}	*com(h1,BoF)*
v_7	*user(*DS)	e_{57}	*atk(FS,ssh,DS)*
v_8	*root(*DS)	e_{78}	*com(DS,BoF)*
		e_{68}	*com(h1,ftp,DS)*

As depicted in **Fig. 2**, exploits appear as edges, and network conditions as nodes. As an example of attack paths, a path $AP_1=[e_{13},e_{34},e_{46},e_{68}\,|\,v_1,v_3,v_4,v_6,v_8]$ consists of four exploits and five conditions, including the initial condition v_1 and the ultimate goal v_8. Following this attack path, an attacker (host 0) can first establish a trust relationship and gain root privilege on WS by exploiting a ssh vulnerability on it, then gain user privilege on host 1 via a remote ssh attack. After that, a local buffer overflow vulnerability makes the attacker able to get a root privilege on host 1. Finally, the goal of root privilege on DS is achieved by compromising a remote ftp connection vulnerability. In total, there are three feasible attack paths which can be generated using existing algorithms [2]:

- $AP_1=[e_{13},e_{34},e_{46},e_{68}\,|\,v_1,v_3,v_4,v_6,v_8]$
- $AP_2=[e_{12},e_{27},e_{78}\,|\,v_1,v_2,v_7,v_8]$
- $AP_3=[e_{13},e_{34},e_{45},e_{57},e_{78}\,|\,v_1,v_3,v_4,v_5,v_7,v_8]$

Generally, security risk will always conceal in these feasible attack paths. However, when an exploit occurs, it will bring the latent risk to the table, causing a series of safety problems. By modeling of RFAG, we represent attackers' behavior by the binary value of exploit edges. Under normal circumstances, the states of exploit edges are set to '0' and security risk is implicit. When the network is under attack, adversaries will choose certain attack paths on their own preferences. These attacks will activate relevant exploit edges and set their states to '1', which means that the implicit risk will become explicit and do actual harm to network. Moreover, this kind of risk will always originate, transfer, redistribute and converge along the attack path.

Although the RFAG can give intuitive analysis of attack paths, an optimal solution to harden the network is still not apparent from the attack graph itself. To address this problem, we give several requisite definitions in this section to help find an efficient hardening method.

3.1 Attack Path

Generally, there is at least one feasible attack path in an attack graph, pointing from the adversaries' initial status to their ultimate attack goals. Formally, an attack path AP_k is an ordered set of condition nodes and exploit edges where $AP_k=[e_{12}^k,e_{23}^k,...,e_{mn}^k\,|\,v_1^k,v_2^k,...,v_n^k]$.

3.2 Attacker Prototype

As depicted in **Fig. 2**, there are two prototypes in the risk flow attack graph, the ***divergence prototype*** and the ***convergence prototype***.

– ***Divergence prototype***: a node is *true* if and only if its parent node and the edge between them are both *true*. The logic truth table of a divergence prototype is shown in **Table 2**.

Table 2. Logic truth table of divergence prototype

v_a	e_{ab}	v_b
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

As inferred from the logic truth table, the logical relationship between a node v_b and its parent node v_a in a divergence prototype is that $v_b = v_a e_{ab}$. Accordingly, the potential risk *PR* of v_b can be calculated as:

$$PR(v_b) = PR(v_a) + PR(e_{ab}) \tag{1}$$

In (1), we take the Common Vulnerability Scoring System (CVSS) base score to represent $PR(e_{ab})$. The detailed calculation is omitted here and can be found in the Common Vulnerability Scoring System in [21].

– ***Convergence prototype***: a node is *true* once an arbitrary pair of its parent node and their connection edge is *true*. Similarly, the logic truth table of a divergence prototype is shown in **Table 3**.

Table 3. Logic truth table of convergence prototype

v_a	e_{ac}	v_b	e_{bc}	v_c
1	1	0	0	**1**
1	1	0	1	**1**
1	1	1	0	**1**
1	1	1	1	**1**
0	0	1	1	**1**
0	1	1	1	**1**
1	0	1	1	**1**

Similar with the divergence prototype, the logic truth values in **Table 3** depicts a $v_c = v_a e_{ac} + v_b e_{bc}$ relationship in the convergence prototype. The potential risk *PR* of v_c can be calculated by:

$$PR(v_c) = \max\{PR(v_*) + PR(e_{*c})\} \tag{2}$$

where v_* is the parent node of v_c connected by e_{*c}.

3.3 Hardening Strategy

For a given set of *h* exploit edges, the hardening strategy $HST = (ST_1, ST_2, ..., ST_h)$ is a Boolean vector indicating which strategy ST_i is implemented on exploit edge e_i. Particularly, $ST_i = 1$ if hardening strategy for e_i is chosen, otherwise $ST_i = 0$.

Specifically, countermeasures that are frequently-adopted by defenders can be divided to the following types according to an OSVDB (Open Source Vulnerability Database) [22] classification:

- *Patch():* patch the corresponding vulnerability using patches released by vendors;
- *Configure()*: alteration of default configurations, such as blacklist, whitelist and access control table;
- *Upgrade()*: upgrade relative software or operating system to a safer version;
- *Disconnect()*: disconnect the vulnerable services from Internet or Local Area Network;
- *Disable()*: disable vulnerable services or shutdown corresponding systems.

The implementation of security countermeasures will incur different security control cost, including installation cost, operation cost, system downtime, incompatibility cost, etc. In practical applications, these data can be obtained by statistics. For simplification, in this work we omit the acquisition of the costs and use a decimal $C_i \in [0,1]$ to represent the cost of ST_i. And the overall cost *C(HST)* of harden strategy *HST* can be formulated as:

$$C(HST) = \sum_{i=1}^{h} (C_i ST_i) \tag{3}$$

3.4 Risk Function

The value of risk function *f* can not only measure the security status of network, but also indicate the validity of enhancement measures. Suppose there are *l* attack paths in a given risk flow attack graph *G*, the attack path set $AP = \{AP_i \mid i \le l, i \in Z^+\}$, and the hardening strategy $HST = (ST_{e_1}, ST_{e_2}, ..., ST_{e_h})$, the risk polynomial of an attack path can be formulated as:

$$R_{AP_i}(HST) = \sum_{v_j, e_{*j} \in AP_i} (PR(v_j) * \overline{ST_{e_{*j}}}) \tag{4}$$

We take the accumulation risk of network assets to define the risk of an attack path. This risk value is an accumulative function of harden strategy *HST* and independent node risk. In this formulation, the risk value $PR(v_j)$ of an independent node v_j is calculated recursively by (1) and (2) following the exploit sequence of the path.

Based on common threat behavior, it's a reasonable assumption that higher risk paths are likely to have a larger selection probability. Under this assumption, we measure the probability of attack path selection as:

$$p_{AP_i} = R_{AP_i}(HST_0) \Big/ \sum_{i=1}^{l} (R_{AP_i}(HST_0)) \tag{5}$$

Here, HST_0 stands for the harden strategy of {000…00}, considering that adversaries always hold the hypothesis that the conditions are available to exploit. Apparently, our

definition of p_i satisfies the basic requirements of non-negativity, normalization and countable additivity. The properties of p_{AP_i} are listed as follows:

- $p_{AP_i} \in [0,1)$;
- $\sum_{i=1}^{l} p_{AP_i} = 1$;
- $p_{(AP_1 \cup AP_2 \cup ... \cup AP_k)} = p_{AP_1} + p_{AP_2} + ... + p_{AP_k}$;

In practice, adversaries tend to choose m of l attack paths at a time to penetrate the network. Hence, the risk value of the whole network can be calculated as:

$$PR(HST) = \oplus \sum_{i=1}^{m} R_{AP_i}(HST) = \sum_{i=1}^{m} (R_{AP_i}(HST) * p_{AP_i} * \prod_{k \in [1,m] \setminus \{i\}} (1 - p_{AP_k})) \quad (6)$$

4. Using GA for Optimal Hardening

Our approach to seek for efficient hardening strategies is to formulate it into a multi-objective genetic algorithm problem that:

Given a risk flow attack graph G, find an optimal vector HST , which minimize the overall security control cost C(HST) and network risk PR(HST).

The solver starts with an initial population of chromosomes, representing possible combination of hardening strategies *HST*. In each generation, every strategy in a population is evaluated by the fitness selection based on harden cost *C(HST)* and network risk *PR(HST)*. The selection is determined by a fitness based crowd distance metric to preserve the diversity of individuals. Those individuals will experience several rounds of selection, crossover, and mutation until a set of Pareto optima is created. Particularly, we adopt an elitist preservation strategy to preserve the best individual in every genetic process and directly copied it to the next generation. This strategy can protect the elitist individual from being decomposed by the crossover and mutation operator. The elitist preservation strategy is also able to maintain the global convergence as has been proved by Rudolph [23]. The procedure of proposed approach is shown in **Fig. 3.**

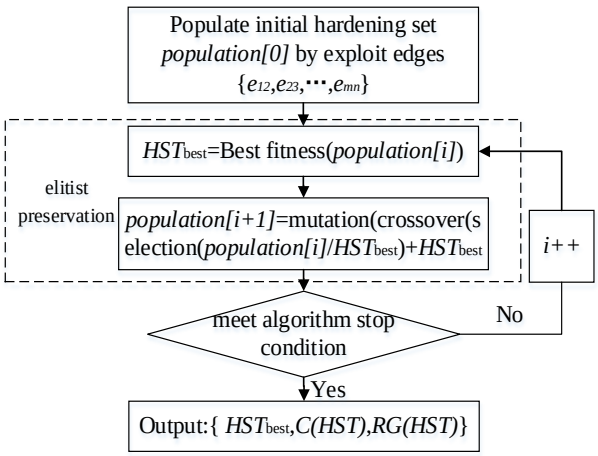


Fig. 3. Block diagram of our approach

4.1 Chromosome Coding

The algorithm starts by generating an initial population of chromosomes. We adopt the binary encoding format and define the implementation of security countermeasures on exploit edges as the genes of chromosomes. **Fig. 4** shows a sample chromosome of the attack graph in Section 3.

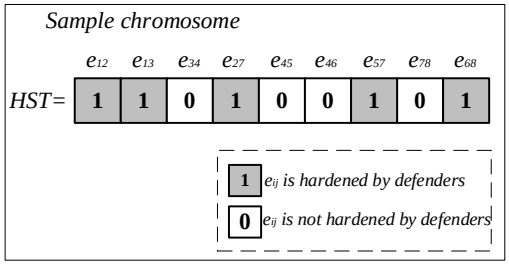


Fig. 4. Sample chromosome of nine exploit edges

This sample chromosome is encoded as *HST*= {110100101} for the risk flow attack graph of nine exploit edges. Those genes encoded by '1' represent edges that are hardened by defenders. On the contrary, the '0' genes stand for unhandled edges.

4.2 Initial Population

We populate the initial population population[0] by harden strategy set $\mathbf{HST} = \{HST_1, HST_2, ..., HST_l\}$, where $\text{sizeof}(population[0]) = \text{card}(\mathbf{HST}) = l$. Here l is an input parameter of genetic algorithm. And each chromosome in population[0] is encoded as a binary string by the above chromosome coding scheme, where $HST_i = (ST_1, ST_2, ..., ST_h), 1 \le i \le l$. The values of binary strings are initialized randomly without loss of generality.

4.3 Objective Function

As mentioned before, defenders are always faced with the challenge to reduce network risk as much as possible within a fixed budget. Thus, the two objective functions consist of security control cost *C(HST)* and network risk *PR(HST)*:

$$obj = \begin{cases} \min C(HST) = \sum_{i=1}^{h} (C_i ST_i) \\ \min PR(HST) = \sum_{i=1}^{m} (R_{AP_i}(HST) * p_{AP_i} * \prod_{k \in [1,m] \setminus \{i\}} (1 - p_{AP_k})) \end{cases} \tag{7}$$

4.4 Fitness Selection

In order to guarantee the diversity of individuals and the uniformity of non-inferior solutions, we performed a crowd-distance selection by a 2-Tournament strategy in our work. The selection procedure can be divided into three steps:

- *rank the individuals*: for individual x in $population[t]$, $rank[x,t] = 1 + d_x^t$, where d_x^t stands for the count of individual y, that $\forall i \in \{1,2\}, obj_i(y) \le obj_i(x)$ and $\exists j \in \{1,2\}, obj_j(y) < obj_j(x)$.

● *sort within a rank*: sort individuals in the same rank by descending order of objective function *PR(HST)*;

● *crowd distance selection*: calculate the crowd distance of individual x where $dist(x)=\sum_{i=1}^{2}|obj_i(x-1)-obj_i(x+1)|$. And the selection of individuals abides by the principle that if $rank[x,t]<rank[y,t]$, or $rank[x,t]=rank[y,t]$ but $dist(x)>dist(y)$, then x will be selected.

It's worth noting that the non-inferior individuals in each population will be preserved as elite individuals, whose selection probabilities are '1'.

4.5 Crossover

Taking L as the length of chromosome, we perform a two point crossover process by generating two random integers m and n within the interval [1, L]. Suppose there are two parent chromosomes to perform a two point crossover. The function selects:

- vector entries numbered less than or equal to m from the first parent chromosome;
- vector entries numbered from $m+1$ to n, inclusive, from the second parent chromosome;
- vector entries numbered greater than n from the first parent chromosome;

As depicted above, three intervals are selected respectively from two parents to form a single gene as the child chromosome. The two point crossover process is illustrated in **Fig. 5**:

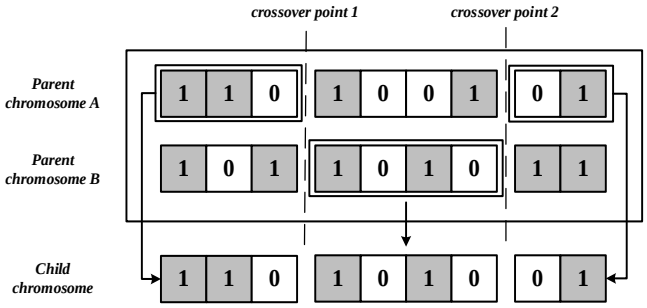


Fig. 5. Two point crossover process

For example, the parent chromosomes in **Fig. 5** are {110100101} and {101101011}. If the crossover points {m, n} are set to {3, 7}, the child chromosome of {110101001} is achieved.

4.6 Mutation

We perform a two-step Uniform Mutation process. First of all, a fraction of the vector entries of an individual is selected for mutation, where each entry has a probability rate r of being mutated. The default rate is set to r=0.01 and can be adjusted according to the performance of algorithm. In the second step, genes on each selected entry is replaced by the opposite value.

5. Experiments and Analysis

Taking the example network given in Section 3 as a scenario, we present some numerical results about the proposed method to evaluate its feasibility and effectiveness. The experimental implementation of our approach mainly uses a genetic algorithm toolbox *gatbx*

[24] developed by University of Sheffield for Matlab of version R2011b. The experiments are performed on a PC with 2.3GHz Intel(R) Core(TM) i5-2410M CPU with 4G RAM running Windows 7 Ultimate Operating System. The performed experiments include: A) the evolutionary process of harden cost and network risk; B) average distance between individuals (harden strategies) during evolution; C) rank histogram of individuals in each Pareto tier; D) Pareto frontier of optimal harden strategy for cost and risk and E) Scalability of the multi-objective genetic algorithm.
The simulation parameters are listed in **Table 4**.

Table 4. Simulation parameters

Problem Formulation	objective function	*Formula (4)*
	number of variables	*9*
Population	population type	*bit string*
	population size	*100*
Selection	selection function	*tournament*
	tournament size	*2*
Crossover	crossover function	*two-point*
	crossover fraction	*0.6*
Mutation	mutation function	*uniform*
	mutation rate	*0.01*
Multi-objective Setting	distance measure function	*distance crowding*
	pareto front population fraction	*0.35*
Stopping Criterion	number of generation	*100*

The sample risk flow attack graph is a 8-node graph, with 9 valid exploit edges. Our simulation take an 8×8 weighted matrix $R=[r_{ij}]_{8\times 8}$ to represent the sample attack graph, where

$$obj=\begin{cases}\min C(HST)=\sum_{i=1}^{h}(C_i ST_i)\\ \min PR(HST)=\sum_{i=1}^{m}(R_{AP_i}(HST)*p_{AP_i}*\prod_{k\in[1,m]\setminus\{i\}}(1-p_{AP_k}))\end{cases}$$

For the convenience of simulation, the nonzero values of r_{ij} are generated randomly within the interval of (0, 1). Upon initialization, our simulation encode the harden strategy *HST* into a 9-bit string with the population size of 100. And other essential parameters of this multi-objective GA approach can be found in **Table 4**.

5.1 Value Trend of Objective Functions

After 100 generations of selection, crossover and mutation, we can obtain the value trend of two objective functions in **Fig. 6.**

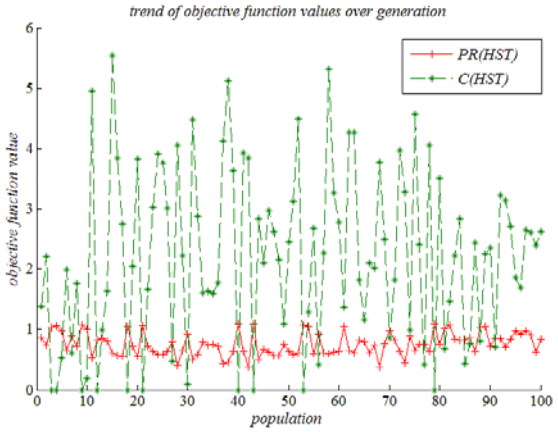


Fig. 6. Value trend of objective functions

Fig. 6 depicts the trend of the two objective functions: *PR(HST)* and *C(HST)* over generation. As shown in **Fig. 6**, the solid line with marker '+' represents risk value *PR(HST)*, and the dotted line marked by '*' stands for harden cost *C(HST)*. In the 100-generation evolutionary process, these two objective functions are optimized by the genetic process described in Section 3. It's worth noting that, every time when the harden cost reaches a peak value, the corresponding risk value falls to a low point, and vice versa. Obviously in a real case, the more effort defenders spent, the less risk will remain in network. It can be learned from this trend that the determination of optimal harden strategy is not a simple maximize or minimize problem. Therefore, we adopt a Pareto optima set approach to work out the optimal solutions.

5.2 Average Distance between Individuals

When determining defense strategies, security analysts always need to prioritize the alternatives and apply the most efficient ones. Due to the high similarity of possible hardening strategy combinations, our approach enforces a diversity-preserving mechanism based on a crowding distance metric. This metric for an individual is the sum of the average side-lengths of the cuboid generated by its neighboring individuals in objective space. The value of average distance metric over generation is shown in **Fig. 7.**

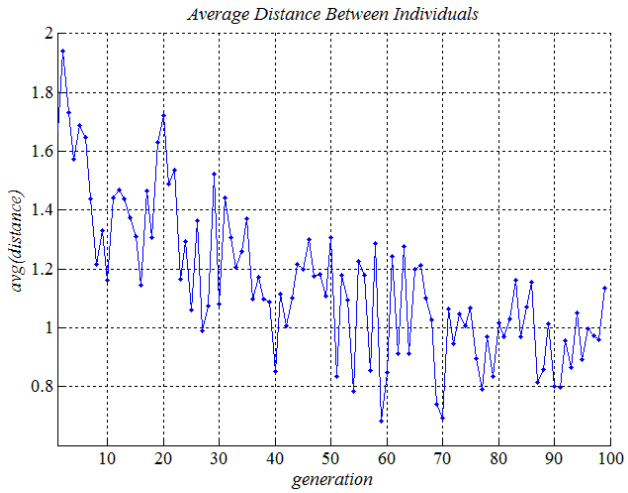


Fig. 7. Average distance between individuals

As depicted in **Fig. 7**, the average distance of individuals varies within the ranges of 0.3 to 1.9, and slowly converges from around 2 to 1.15 in 100 generations. At the beginning of simulation, the average distances change around 2, by this time, there's only few non-inferior solutions, i.e. optimal hardening strategies in the solution space. With the enforcement of crowd distance selection, individuals with lower rank and larger crowd distance are given more preference. Thereby forcing the mechanism to search in the area with lesser density in the solution space. After 100 generations of evolution, the hardening strategies with better performances have been obtained. Meanwhile, the average distance between individuals converges to 1.15 at the stopping criterion of 100 generation.

5.3 Distribution of Optimal Solutions

As the consequence of generations of selection, crossover and mutation, the hardening strategies in the solution space are divided into several ranks according to their performances. These ranks are known as Pareto tiers, which give an intuitive view of the distribution of individuals. The fraction of individuals in each Pareto tier is shown in **Fig. 8** as a rank histogram.

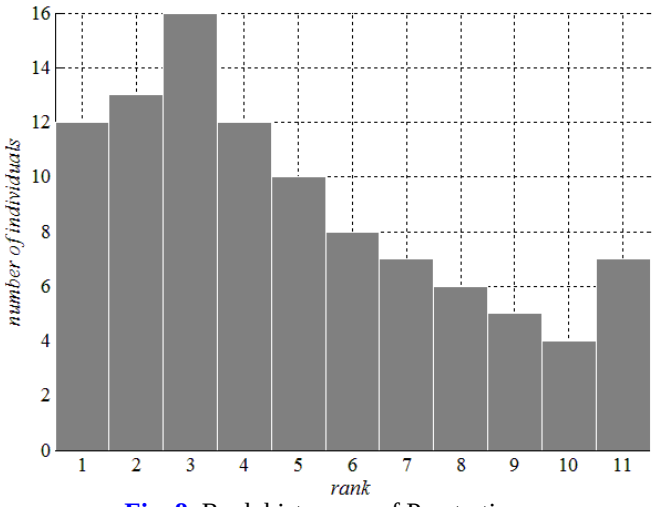


Fig. 8. Rank histogram of Pareto tiers

As depicted in **Fig. 8**, the individuals are partitioned into 11 ranks according to their performances. For example, in **Fig. 8**, there are 10 individuals in Rank 5, which are dominated by Rank $\{i|i<5\}$, meaning that given arbitrary hardening strategy HST_x and HST_y in Rank 5 and Rank (i<5), respectively, the following relation holds:

$$\begin{cases} PR(HST_y) \le PR(HST_x) \\ C(HST_y) \le C(HST_x) \\ PR(HST_y) < PR(HST_x) \, or \, C(HST_y) < C(HST_x) \end{cases}$$

Those individuals in Rank 1 are best, which are also known as the non-inferior solutions. In our experiment population, there are 12 individuals in Rank 1. And these 12 non-inferior harden strategy solutions form the Pareto frontier of our multi-objective optimization. **Fig. 9** plots the function values for all non-inferior individuals.

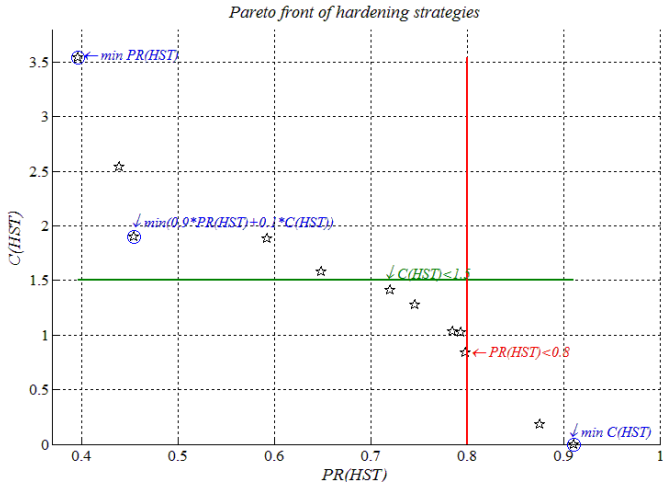


Fig. 9. Distribution of optimal hardening strategies (Pareto front)

As is shown in **Fig. 9**, the 12 optimal strategies are denoted by marker '☆'. These individuals are in Rank 1 that is not dominated by any rank. The hardening strategies in Pareto front provide security analysts with an optimal decision set to choose from, depending on the actual optimization goals. For example, the five annotations in **Fig. 9** correspond to the following optimization goals, from left to right:

- ***Goal 1***: minimize network risk without cost constraint
- ***Goal 2***: minimize the weighted sum of harden cost and network risk, $\alpha * PR(HST) + \beta * C(HST)$, where $\alpha = 0.9$ and $\beta = 0.1$;
- ***Goal 3***: given cost budget of 1.5, minimize network risk;
- ***Goal 4***: given risk constraint of 0.8, minimize harden cost;
- ***Goal 5***: minimize harden cost without risk constraint.

The corresponding bit string value of harden strategies and numeric result of risk and cost can be seen in **Table 5**.

Table 5. Pareto non-inferior solutions

HST	*risk*	*cost*	e_{12}	e_{13}	e_{34}	e_{27}	e_{45}	e_{46}	e_{57}	e_{78}	e_{68}	*Goal*
1	0.5922	1.8802	1	1	1	1	0	0	0	0	1	
2	0.7847	1.0360	0	1	1	1	0	0	0	1	1	
3	0.9113	0	0	0	0	0	0	0	0	0	0	5
4	0.7199	1.414	0	1	1	0	1	1	1	1	0	
5	0.7462	1.2724	0	0	0	0	1	1	0	0	1	
6	0.3959	3.5415	1	1	1	0	1	1	1	1	0	1
7	0.8758	0.1799	0	0	0	0	0	0	1	0	1	
8	0.4387	2.5381	1	1	1	0	1	1	0	1	0	
9	0.4545	1.9006	1	1	1	0	1	1	1	1	0	2
10	0.7933	1.0225	0	0	1	1	1	1	0	0	0	
11	0.6484	1.5796	1	0	0	0	1	1	1	1	1	3
12	0.7987	0.8373	0	1	1	0	0	1	1	0	0	4

5.4 Scalability

As stated in Section 3, we use the risk flow attack graph in **Fig. 2** to illustrate our multi-objective genetic algorithm. The graph model is relatively simple but practical. Because

taking the intruders' attack pattern into consideration, an adversary always tend to limit the attack process in as few steps as possible to avoid exposure. Moreover, we conduct a series of experiments to verify the scalability of our method. The result is shown in **Fig. 10**.

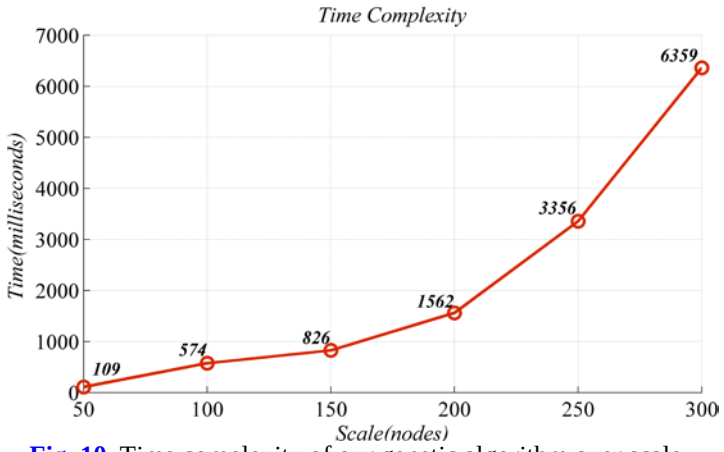


Fig. 10. Time complexity of our genetic algorithm over scale

As depicted in **Fig. 10**, we computed the execution times for working out optimal hardening strategies in 6 risk flow attack graphs with similar structures from 50 to 300 nodes. The solid line in **Fig. 10** show the tendency of increasing time complexity of our multi-objective genetic algorithm. The time complexity experiences an acceptable trend of modest increasing over scale, which verifies the feasibility and scalability of our method.

5. Conclusion

The risk flow attack graph based approach presented in this paper models vulnerabilities and quantifies network risk by a multi-objective GA solver. By means of this strategy, network security risk is calculated by an iterative process according to two attacker prototypes defined in the attack graph. The metrics of network risk and harden cost are taken as objective functions to be optimized, which are two non-ignorable elements on security analysts' side.
Compared with existing security hardening methods, this work has the following differences:

1) The method of quantifying risk using the risk flow attack graph in our approach is different from those using traditional quantization methods;
2) The approach handles security hardening problem in a multi-objective optimization way than simple weighting combination of statistical security data, such as harden cost, network risk, network reliability, etc.

As we all know, that risk will always conceal in a network. Once an exploit occurs, the latent risk will be brought to the table, causing defenders to take preventive actions. These measures will in turn prompt adversaries to make improvements of their ways of attacks. Thus, in future work, the dynamic relationship between security hardening strategies and attack behaviors will be researched on the basis of this work. More network factors which could affect security risk to will be studied to improve and refine our approach, such as concealment of attackers and risk threshold. Furthermore, we would be interested in analyzing large-scale attack scenarios to test the performance and scalability of proposed approach.

References

[1] A. Espenschied, "A Discussion of Threat Behavior: Attackers & Patterns," last accessed 2014. Article(CrossRefLink)

[2] A. Paul, W. Duminda, K. Saket, "Scalable, graph-based network vulnerability analysis," in *Proc. of the 9th ACM Conference on Computer and Communications Security*, pp. 217–224, 2002. Article(CrossRefLink)

[3] S. Oleg, H. Joshua, J. Somesh, L. Richard, W. Jeannette, "Automated generation and analysis of attack graphs," in *Proc. of the IEEE Symposium on Security and Privacy*, pp. 273–284, 2002. Article(CrossRefLink)

[4] P. Andrew, J. Ellison, C. Linger, "Attack modeling for information security and survivability," *Technical Report of Carnegie Melon University/Software Engineering Institute CMU/SEI-2001-TN-001*, 2001. Article(CrossRefLink)

[5] R. Indrajit, P. Nayot, "Using attack trees to identify malicious attacks from authorized insiders," in *Proc. of the 10th European Symposium on Research in Computer Security*, pp. 231–246, 2005. Article(CrossRefLink)

[6] C. Pengsu, W. Lingyu, J. Sushil, S. Anoop, "Aggregating CVSS base scores for semantics-rich network security metrics," in *Proc. of International Symposium on Reliable Distributed Systems*, pp. 31-40, 2012. Article(CrossRefLink)

[7] W. Lingyu, N. Steven, J. Sushil, "Minimum cost network hardening using attack graphs," *Computer Communications*, vol. 29, no. 18, pp. 3812-3824, 2006. Article(CrossRefLink)

[8] D. Rinku, R. Indrajit, P. Nayot, W. Darrel, "Optimal security hardening on attack tree models of networks: A cost-benefit analysis," I*nternational Journal of Information Security*, vol. 11, no. 3, pp. 167-188, 2012. Article(CrossRefLink)

[9] S. Noel, S. Jajodia, B. O'Berry, M. Jacobs, "Efficient minimum-cost network hardening via exploit dependency graphs," in *Proc. of the 19th Annual Computer Security Applications Conference*, pp. 86–95, 2003. Article(CrossRefLink)

[10] S. Diptikalyan, "Extending logical attack graphs for efficient vulnerability analysis," in *Proc. of the 15th ACM Conference on Computer and Communications Security*, pp. 63-73, 2008. Article(CrossRefLink)

[11] Z. Zhang, S. Wang, "Boosting logical attack graph for efficient security control," in *Proc. of the 7th International Conference on Availability, Reliability and Security*, pp. 218-223, 2012. Article(CrossRefLink)

[12] I. Kyle, C. Matthew, L. Richard, W. Seth, B. Stephen, "Modeling modern network attacks and countermeasures using attack graphs," in *Proc. of the 25th Annual Computer Security Applications Conference*, pp. 117-126, 2009. Article(CrossRefLink)

[13] J. Sushil, N. Steven, "Topological vulnerability analysis," *Cyber Situational Awareness Issues and Research*, vol. 46, pp. 139-154, 2009. Article(CrossRefLink)

[14] K. Nizar, C.-Boulahia Nora, C. Frédéric, D. Hervé, "A service dependency model for cost-sensitive intrusion response," in *Proc. of the 15th European Symposium on Research in Computer Security*, pp. 626-642, 2010. Article(CrossRefLink)

[15] D. Rinku, P. Nayot, R. Indrajit, W. Darrell, "Optimal security hardening using multi-objective optimization on attack tree models of network," in *Proc. of the 14th ACM Conference on Computer and Communications Security*, pp. 204-213, 2007. Article(CrossRefLink)

[16] G. Mukul, R. Jackie, C. Alok, C. Jie, "Matching information security vulnerabilities to organizational security policies: a genetic algorithm approach," *Decision Support Systems-Special Issue: Intelligence and security informatics*, vol. 41, no. 3, pp.592-603, 2006. Article(CrossRefLink)

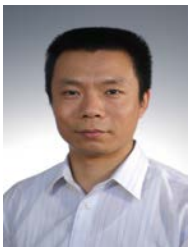
[17] V. Viduto, C. Maple, H. Wei, A. Bochenkov, "A multi-objective genetic algorithm for minimizing network security risk and cost", in *Proc. of 2012 International Conference on High Performance Computing and Simulation (HPCS)*, pp. 462-467, 2012. Article(CrossRefLink)

[18] F. Marcel, W. Lingyu, S. Anoop, J. Sushil, "Measuring network security using dynamic Bayesian network," in *Proc. of the 2008 ACM Conference on Computer and Communications Security*, pp. 23-29, 2008. Article(CrossRefLink)
[19] X. Peng, L. Jason, O. Xinming, L. Peng, L. Renato, "Using bayesian networks for cyber security analysis," in *Proc. of 2010 IEEE/IFIP Conference on Dependable Systems and Networks*, pp. 211-220, 2010. Article(CrossRefLink)
[20] W. Shuzhen, Z. Zonghua, K. Youki, "Exploring attack graph for cost-benefit security hardening: a probabilistic approach," *Computers & Security*, vol. 32, pp. 158-169, 2013. Article(CrossRefLink)
[21] P. Mell, K. Scarfone, S. Romanosky, "Common vulnerability scoring system," *IEEE Security and Privacy*, vol. 4, no. 6, pp. 85-89, 2006. Article(CrossRefLink)
[22] Opensource Vulnerability Database, last accessed 2014. Article(CrossRefLink)
[23] R. Gunter, "Convergence properties of canonical genetic algorithms," *IEEE Transaction on Neural Networks (special issue on Evolutionary Computation)*, vol. 5, no. 1, pp. 96-101, 1994. Article(CrossRefLink)
[24] Genetic Algorithms Toolbox, last accessed 2014. Article(CrossRefLink)

Fangfang, is currently a Ph.D. candidate at the School of Computer, Beijing University of Posts and Telecommunications, Beijing, China. She received her B.E. degree in information security from Beijing University of Posts and Telecommunications in 2006. Her research interests include network security, cyber-attack modelling, vulnerability analysis and risk assessment. *The corresponding author. Email:daiff.bupt@gmail.com

Kangfeng Zheng, is a vice-professor, Ph.D supervisor at the School of Computer, Beijing University of Posts and Telecommunications, China. He received his Ph.D. in 2006 from School of Information Engineering in Beijing University of Posts and Telecommunications. His research interests are network security, intrusion detection, malicious code analysis, honey-net system, cyber-attack modeling and Advanced Persistent Threat.

Bin Wu, Ph.D. in signal and information processing, lecturer at the School of Computer, Beijing University of Posts and Telecommunications, China. He received his Ph. D. in 2008 from Beijing University of Posts and Telecommunications. His research is centered on active defense technology in various computer and communication networks, including intrusion detection, honey-net systems and secure gateway.



Shoushan Luo, is a professor, Ph.D supervisor at the School of Computer, Beijing University of Posts and Telecommunications, China. Luo received his Ph.D. in signal and information processing from Beijing University of Posts and Telecommunications in 2001. His research interests include coding cryptography, grid computing, neutral networks and Boolean function theory.