5th SASTech
Iran | Mashhad
May 12 - 17 | 2011
5th Symposium on Advances in Science & Technology

Attack graph analysis using parallel algorithm

Dr. Jamali Mohammad (m.Jamali@yahoo.com)

Ashraf Vahid, MA student of computer software, Shabestar Azad University
(vahid.ashraf@yahoo.com)

Ashraf Vida, MA student of General Linguistics, Razi University of Kermanshah
(vida.ashraf@yahoo.com)

Abstract

Each attack graph shows a set of scenarios of penetrating a computer network. A penetration scenario actually defines the order of steps that an intrudor should take to achieve his goal, and each step is characterized to show which host must be abused. A specified weight is attributed to every abuse in each stpes by analyst. The atributted weight is proportional to a required cost to prevent the abuse. In this paper disadvantages of a network are gathered in a graph and analysed by a parallel algorithm. In a way that allthe available path in the graph which represent successful scenarios processed concurrently, and from each path a node with the least possible cost is selected which correspondes to an abuse. Eventually we have minimal set of disadvantages with the least prevention cost, and by preventing the happening of this mininal set no penetration scenario is applicable on our analytical network.

Keywords: attack graph, network graph, parallel algorithm.

1 Introduction

In todays networks most of successful attacks contain sequence of abuses such that in each exploitation of specific abuse, an intruder increases the degree of vulnerabilities for the next steps in order to take the desired steps to achieve their ultimate goal [2]. Philips and Swiler(1998) argued the concept of the attack graph for the first time [3]. Each attack graph is a set of scenarios in a way that each path from beginning node to aimed node in the graph represent a scenario that describe the situation of the intruder from the begining till achieving the aimed point [4]. In order to study and analyse the attacK graph, Sheyner et al.(2002) used a model checker which they called NUSMV. Defects of a network in this model was gather in an xml file by different network software and finally by using the current information they showed the attack graph in a graphical way [5], Subsequently Noel et al.(2005) offered optimum ways of presenting an attack graph [6].

Attack graph analysis help us to make a minimal set of abuses by preventing which no abuses would be exploited. Jha et al.(2002) by considering this problem proved that the solution could be find in the category of NP-HARD problems[10]. They represent an approximate solution to find the almost optimum set.

In order to analyse the attack graph, Abadi and Jalili(2007) used a genetic algorithms [7].

Ashraf(2010), in his proposed solution used the parallel genetic algorithms in a way that evaluate the achieved optimum algorithm of Abadi and Jalili faster [8].

In some of the proposed methods [5],[9] of the attack graph analysis weight of abuses is not mentioned but in this paper, by considering the cost of preventing each abuse, analizer dedicates a weight to each of them and finally the minimal set must have the least possible weight.

The proposed algorithm named parallel-wag study all scenarios in a parallel way and eventually in the minimum time represents the minimal set. In the experiments we used five attack graphs that contained. After doing experiments and analogies, it would be clear that in addition to accelerating performance, compared to other algorithms that give probable and undecisive answers, this algorithm gives a decisive answer.

In this paper we will argue the network security model in the second part, the proposed algorithm in the third part, and in the forth part the algorithm would be implemented and executed by the use of multi pascal simulator and the results will be discussed.

2 Description of the network security model

As networks include lots of software and hardware most of which does not interfere with the attack space, we select six members of the network to make an attack model that contains adequate information for representation of each network and the attack scenario. Any security model of a computer network can be defined as a tuple (H,C,T,E,R,IDS) in which H is a set of hosts connected to the network, C is a relationship, T is a trust link between hosts, E is a set of known abuses, R is a model of intruder, and IDS a model of intrusion detection system [1].

In coming sections we will review the network components.

2.1 hosts

In the intrusion scenrio each steps targets a host and makes use of the defects of the softwares or bad configuration to facilitate the intruder. The purpose of modeling hosts is to obtain information about components that could be useful in exploiting the defects. Each network host $h \in H$ is a foursome (id,svcs,plvl,vuls) in which id is a host's unique identifier, svcs is a set of running services on hosts, plvl shows the degree of intruder access to host. For simplicity, it is assumed that there are only three level of access namely; none, root, user.

2.2 network conectivity

Set of network connectionss defined as a ternary relationship $C \subseteq H * H * p$ in which P is a set of port numbers, each connection $c \in C$ is triad (h_s,h_t,p) in which h_s is a source host, h_t is destination host and p is the destination port number. Note that the connection relationship take into consideration the firewalls and other factors that limit the abilities of the other host.

2.3 trust

Trust is modeled as a twosome relationship in which T$\subseteq H * H$ and T(h_t,h_s) shows that a user may enter to h_t host without confirming the authenticity from h_s host.

2.4 services

Set of services is a list of independent names for each service on the hosts. We separate softwares from services because network services are often the means of exploiting. Services are also associated with connections. Port numbers and information services should be specified in the host model, so that each service name in the list of host services is taken from the name of services in the service model.

2.5 Intrusion Detection System

We relate a logical variable to each action which defines whether IDS has the identification power or not. According to IDS the actions are divided into two categories; if an action is detectable the system will alarm and otherwise it will not realize its running

2.6 actions

Any misuse of $e \in E$ is a pentamerous (pre,h_s,h_t,post,cost) in which pre is the list of preconditions that is defined based on networks connections, services, the existing vulnerabilities and degree of access required for source and destination hosts. h_s is a host starts an abuse. H_t is a host which is a destined to abuse. Post will show the effects of abuses. For simplicity, it is assumed that the required cost for prevention of each abuse is low, medium, high, very high.

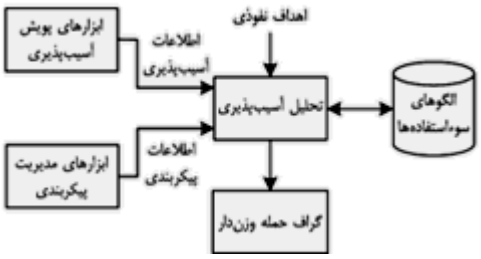

Figure 1. Generation Process of Attack Graph

3 The proposed algorithm

In the proposed algorithm, parallel-wag acts in a way that an array is received from the scenarios and simultaneaously each precessor eliminates an abuse and finds out each scenario that hit with actions and remove them from the set of primary scenarios. The

process will be continued till no executable scenario remains. For this algorithm runs simultaneously each processor should remove abuses such that the ultimate achieved set is not like other processors. For this purpose, in each cycle with the replacement of abuses the mother CPU operates in a way that two processors may not remove one abuse at the same time. The proposed algorithm is given by:

As you can see in the above algorithm, scenario arrays and the number of actions are

```
Global:
Senarioarray : two dimontion array
N : integer
Procedure parallelwag
Input: actionarray,remainsenarioarray,remainsenariocount,level
begin
        If  remainsenariocount=0
        begin
                For i=1 to level-1
                Print actionarray[i]
        end
        Else
        begin
                For i=level to n
                Swap(actionarray[i], actionarray[level])
                For j=1 to remainsenariocount
                Remove each senario[j] that hit with action[level] of remainsenarioarray
                Fork prallelwag(actionarray,remainsenarioarray,remainsenariocount,k+1)
        End
END
```

defined in a parallel form, and the function recursively calls itself and obtain all critical sets.

By having all critical sets at hand one can easily select an optimum critical set and certainly there would be no other optimum set better than this.

4 Experiments

In this section, we describe the cunducted experiments to evaluate the parallelwag algorithm.

It should be noted that experiments are performed on a machin with Intel 2.7 processor, 1 GB memory, and XP operation system, as well as a multi pascal enviroment and PRAM model is used to simulate a parallel machine.

In table 1, five weighted attack graphs that have been randomly generated are shown, and in these attack graphs it is exploited from 20 abuses with the maximum scenario length of 5 and the required weights for preventing abuses are considered 5,1 or 10.

	senarios	sum of weight	avrage weight in senario	avrage cardinality
wag1	10	115	11.5	3
wag2	20	315	15.75	3
wag3	30	435	14.5	3.33
wag4	40	705	17.62	3.75
wag5	50	815	16.3	3.6

Table1. Weighted attack graphs

The results of the execution of a prallelwag algorithm in the above attach graphs are shown in table 2.

	cardinality of critical set	weight critical set
wag1	3	7
wag2	4	8
wag3	5	22
wag4	7	24
wag5	6	32

Table 2. The results of parallelwag

In table 2, the number of a critical set members as well as their weight are shown by preventing which, all attack to the network will be blocked with the lowest cost.

In figure 2, the cycle of the required time for performing an algorithm in the parallel and sequential form, as well as degree of speed up are shown.

To review all three figures together, we divide the resulting number once to 10000 and once to 100 for sequential and parallel mode respectively in order to make the numbers closer to each other to be able to show in one graph.

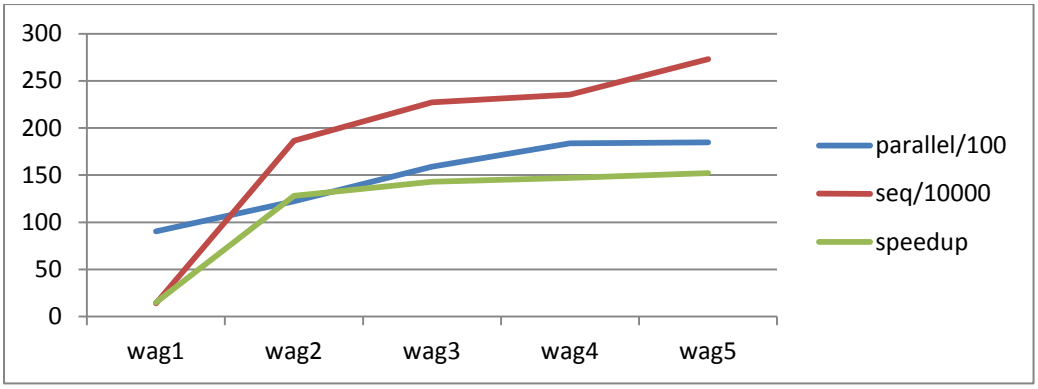

Figure 2. Algorithm requierd cycles

As is clear in figure 2, as the number of scenarios increases, the sequential time grows significantly while in the parallel form the number of cycle grows a little.

5 *conclusions*

In this paper we proposed an algorithm to evaluate an attack graph in parallel form and an optimum set in the entire network by preventing which all the attack to the network would be failed with the lowest cost.

In the past Sheyner et al., and Jalili, Abadi and Ashraf et al. presented a huristic algorithm, the result of which was not a definit answer, while in the proposed algorithm described above the definite answer will be obtained in a minimum time.

As you can see in the result of experiments, parallel performance time of algorithm grows a little as the number of scenarios increases while in the sequential form grows much more, thus in a network with a large number of defined scenarios practically executing a graph in the sequential mode takes so many years time but in parallel mode a little time is reqiured to obtain a definite answer.

Since there is no program to do processing on CPU in this algorithm, this paper paves the ways for further studis to devise an algorithm which considers load_balancing on CPUs.

6 *Refrences*

[1] P. Ammann, J. Pamula, R. Ritchey, and J. Street, "A host-based approach to network attack chaining analysis," in Proc. Annual Computer Security Applications Conf., pp. 72-84, Tucson, US, Dec. 2005.

[2] C. Ramakrishnan and R. Sekar, "Model-based vulnerability analysis of computer systems," in Proc. 2nd Int. Workshop on verification Model Checking and Abstract Interpretation, pp.1998.

[3] C. A. Phillips and L. P. Swiler, "A graph-based system for network vulnerability analysis," in Proc. 1998 New Security Paradigms Workshop, pp. 71-79, Charlottesville 1998.

[4] O. Sheyner, Scenario Graphs and Attack Graphs, Ph.D. Thesis, Carnegie Mellon University, Apr. 2004.

[5] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing, "Automated generation and analysis of attack graphs," in Proc. IEEE Symposium on Security and Privacy, pp. 273-284, Oakland, US, May 2002.

[6] S. Noel and S. Jajodia, "Managing attack graph complexity through visual hierarchical aggregation," in Proc. ACM CCS Workshop on Visualization and Data Mining for Computer Security, pp. 109-118, Washington DC, US, Oct. 2004.

[7] M. Abadi and S. Jalili, "Applying genetic algorithms for minimization analysis of network attack graphs," in Proc. 11th Int. CSI Computer Conf., pp.133-139, Tehran, Jan. 2007.

[8] V. Ashraf and A.Gholami, " Attack graph analysis using genetic-parallel algorithm" . 2010.

[9] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley, Reading, MA, 1989.

[10] S. Jha, O. Sheyner, and J. M. Wing, "Two formal analyses of attack graphs," in Proc. 15th IEEE Computer Security Foundations Workshop, pp. 49-63, Nova Scotia, Canada, Jun. 2002.