Vulnerability Analysis in Attack Graphs Using Conditional Probability

Pankaj Chejara, Urvashi Garg, Gurpreet Singh

Abstract*— Computer networks have become an essential part of almost every organization. These organizations spend a lot of time and money to secure their networks from intruders and attackers. As the need of computers increased, need for network security increased correspondingly. Attackers are always trying to find weakness in network which can be used to break into the network known as vulnerability. So network administrator needs to patch vulnerabilities to thwart attacker from achieving their goal. As new vulnerability are discovered daily, it is very hard to patch every vulnerability in network but if riskier vulnerabilities get patched, risk level can be reduced significantly. Vulnerability score gives insight into the behavior of vulnerability. These scores make security analyst's work easier to some extent. But these scores do not include collective effect of vulnerabilities. A number of vulnerability scanners are available, which provide complete vulnerability details about host. These vulnerability details give analyst a good idea about to which extent the network security can be compromised, but does not give complete view of network vulnerability. Attack graph provides solution to this problem. Attack graph is set of nodes and edges where node represents attacker's state and edge represent possible transition among attacker's state. This technique gives path that can be followed by attacker to gain network's resources. In the network attack graph depict how vulnerability affect network in conjunction with other vulnerabilities. Some vulnerability may not be riskier alone but when chained with some other, it can compromise the security of network. These attack graphs are important security tools to find out such vulnerabilities also. In this paper, we have developed an technique to provide scores to each path in attack graph so as to analyze, which path is to be patched first to remove the risk of attack. These scores are based on conditional probability method.

***Index Terms*— Attack Graphs, Attack Model, Vulnerability Score, Attack Sequence**

I. INTRODUCTION

Almost in every field, security of computer system is a major issue. It is very important as well as difficult to protect a system from network attacks. In this paper we have presented a new technique to analyze network attacks and can be used by anyone who has little knowledge of security and vulnerability. A lot of work has been done in this field which will be discussed later in this paper in section 2. In our technique first all nodes that can be directly reached from attacker node are selected and their privilege level is considered.

Manuscript received December, 2013

Pankaj Chejara, Department of Computer Science, Sharda University, Noida, India.

Urvashi Garg, Department of Computer Science, Lovely Professional University, Jalandhar, India.

Gurpreet Singh, Department of Computer Science, Lovely Professional University, Jalandhar, India.

After that the node that can be exploited by previous nodes at that privilege level are found until finally reaching the goal system. Wyss [6] uses Probabilistic Risk Analysis (PRA) technique for generation of attack graph. But limitation of this technique is that it will generate a network attack graph and determination of attack path for a particular node is difficult. Also, it does not consider possibility of multiple attacks.

There are a number of vulnerability scanners available, which can scan network and gives details of hosts and vulnerabilities present in the network. These scanners do not give any idea about how these vulnerabilities are interdependent or can be chained to perform attacks on complete network hosts. Attack graph solve this problem by giving information about interdependence among vulnerabilities. Thus attack graph is an important tool to find out vulnerable points in the network and to make decisions by patching high vulnerability points and provides means to stop attacker from reaching his goal. The major extension in our method is that it can find multiple attack attempts for a complete network hosts. Our method considers the complete network for attack graph generation.

We have used a scoring scheme for providing score to each path, so as to get most suitable path. Cynthia Philips [3] proposed a technique to get the most cost effective path based on shortest path algorithm but sometimes path may be the shortest one but launching exploits along that path can be more difficult than longer paths and in that case their algorithm will not be useful. It compares weight of each edge from a given node but sometimes one edge may costs more, but still attack path's overall weight can be less than the one over which it was chosen and can be more difficult to exploit than the selected paths. But in our technique we have considered all these issues. We are using optimized path method by providing scores to each path.

For calculating scores for attack paths, first we have to create attack graph which requires preprocessing of network to gather information. Preprocessing involve vulnerability scanning of hosts in the network, creation of vulnerability dataset [7]. Vulnerability scanning gives complete list of all vulnerabilities in each host. Vulnerability dataset consist of preprocessing conditions like user privileges that are to be present on target machine to exploit it, application to be run for a particular attack etc. and post processing conditions like user privileges acquired after exploit etc. which would become preprocessing condition for other next exploit. Attacker's profile specifies which system can be directly accessed by attacker.

The organization of this paper is as follows: section I specifies related work extending to section III elaborating about CVSS details. Starting with section IV giving conditional probability, section V continues the calculation of attack path scores concluding with example calculation

shown in section VI. Finally section VII gives conclusions with proposed future work mentioned in section VIII.

II. RELATED WORK

To generate attack graphs Cynthia Philips [3] proposed another approach which takes three types of inputs i.e. attack template, configuration file and attacker profile. Attack template specifies conditions for exploiting vulnerabilities, i.e. operating system version, user access level etc. Network topology information is fed through configuration file and attacker profile includes attacker's capabilities in exploiting vulnerabilities. An attack graph node represents attacker's state and edge represents state transition. Their approach also assigns some weight to edge known as success probability, time to succeed or cost of attack. Attack graph is generated backward from goal state to attacker state. Firstly attack goal is specified and then algorithm checks for attack template to find matching attack state for transition. Each attack path's total cost is the sum of all edge's weight occurring in path. Then low cost attack path is determined in order to provide defense mechanism. Success probability considers only one aspect of attack, either time to succeed or cost.

A new type of attack graph called multiple prerequisite graph is proposed by Richard Lippmann [4] and developed NetSPA [12] tool to generate attack graph in less time but this attack graph has the problem of scalability. This tool first calculates reachability matrix that is generated for groups and not for individual system, thus reducing the size of resulting matrix. Generated attack graph have three kinds of nodes i.e. Privilege node represents privilege level on particular host, State node represents reachability group or credential and vulnerability instance represent s particular vulnerability. NetSPA [12] tool considers means by which attacker can gain access to network's resources. NetSPA [12] generates attack graph in no time. Although high speed is achieved, but no scoring is provided for attack paths in this method.

A model checking technique to generate attack graph is proposed by Oleg Sheyner [4]. They used NuSMV [11] model checker to generate attack graph. Attack graph analysis involves finding minimal critical set. Minimal critical set is a set of states which are necessary to reach the attack's goal state. They used probability to compute state transition likelihood. In their approach, edges are assigned probability thus known as probabilistic attack graph. Problem with their approach is that only some edges have probability score, some of them remain without any score and they does not give complete score for attack path.

In [6] proposed a model that describes Probabilistic Risk Analysis (PRA). It will first find all the nodes that are directly reachable from the attacker's node, considered as child node. The child node is then further divided into sub nodes that will be directly reachable from child node. But they are unable to model multiple attack attempts.

Nirnay [2] proposed a technique that uses CVSS [9] score for probability of exploits. CVSS [9] base and temporal scores are used to calculate probability of vulnerability and then cumulative score known as probability security metric is calculated. Attack resistance metric is also calculated and based on these metrics, network assessment is carried out. Attack resistance metric is reciprocal of probability security metric, and this is a measure of efforts required in order to perform successful exploit.

Cynthia Philips [3] have also used CVSS [9] scoring framework in attack graphs to calculate attack path's score. They have considered CVSS [9] metrics score for individual attack and compare this metrics value to previous attack value in the path and choose minimum value. For attack path's score, only base group metrics is taken into consideration. For each host value of access vector score, access complexity and authentication score in CVSS [9] is set to minimum value of previous exploited vulnerabilities metrics. They have also calculated damage done by host in attack path.

III. COMPLETE VULNERABILITY SCORING SCHEME (CVSS)

Complete Vulnerability Scoring Scheme [9] is an open framework for assigning scores to vulnerabilities. These scores will be calculated by proper formulas developed by FIRST organization [9]. These scores are built upon three metric groups. But we will consider only Base metrics group. Metric groups are as follows

a. Base Metrics
b. Temporal Metrics
c. Environmental Metrics

a. *Base Metrics*: Base Metrics contains metrics which are time independent. These metrics signify characteristics of vulnerability which do not change over time. CVSS score mainly depends on Base Metrics and this score is considered as total score of vulnerability. Base Metrics score calculation formula depends upon some sub group values which are described in table (see Table I).
    a. *Access Vector*: Access vector specify vulnerability access method. These methods can be local, remote and adjacent network. If exploitation of vulnerability requires local access to target machine, then value of access vector become local, when it is compulsory for attacker to be present in the local network then adjacent network value is used; Remote value is for remotely exploitable vulnerability.
    b. *Access complexity*: This characteristic specifies complexity of attack required to exploit the vulnerability on target machine. If a vulnerability require victim to interact with attack mechanism then access complexity is set to level of interaction required, it can be low, medium or high. In case of high, attacker needs to wait for some kind of action to be performed by victim before launching attack. Higher access complexity makes it difficult to attack victim's system. Medium value specifies that attacker has to put some efforts to attack the system and low value specifies that attack can be lunched very easily.
    c. *Authentication*: To start an attack what level of authentication is required, is represented by this metric. Values that can be set for this metrics are multiple, single or none. If authentication is required as many times as attacker launches attack then “multiple” value is set for metric. In similar manner other values also set according to different

scenarios.

d. *Confidentiality impact (ConfImpact):* Confidentiality impact is the impact on confidentiality of victim's resources. Its value can be complete, partial or none. If attacker gains access to all system file then value is complete, if it gains access to some files then it is partial otherwise it is none.

e. *Integrity impact (IntegImpact):* Integrity impact measures how much integrity of system is affected by attack. Integrity is trustworthiness of information. This metrics value can be complete, partial or none.

f. *Availability impact (AvailImpact):* Availability impact is impact on availability of files, services etc. This metrics value can be complete, partial or none. If attack stops a service then value of availability impact become complete, if it reduces the performance of service then value is partial and in case of no effect, value is none.

b. *Temporal Metrics*: These metrics show characteristics of vulnerability which changes over time. Its sub group metrics are:

a. *Exploitability*: Exploitability of vulnerability is capability of vulnerability to attack the system. If exploit code is available then chances of attack are increased for particular vulnerability. If exploit code is available but some changes are required in it in order to function, then proof of concept (POC) value is set for metrics. Functional value is assigned to metric if available exploit code works on some system only.

b. *Remediation level*: As the new vulnerability is discovered, initially these vulnerabilities are unpatched. Unpatched vulnerability increases chances for successful exploitation. So release of patch also makes it difficult for attacker and set score downwards. This metric is considered if vulnerability is patched officially, third party patch is available or unpatched.

c. *Report confidence*: This metric is measure of how many systems exist with a particular vulnerability i.e. degree of confidence in existence of vulnerability.

c. *Environmental Metric*: Characteristics of vulnerability related to user's environment are stored in environmental metric group. Its sub group metrics are:

a. *Collateral damage*: Damage of victim's asset inflected by attacker is specified in this metric.

b. *Target distribution*: It implies how many systems are affected by that vulnerability.

c. *Security requirement*: Security requirement mainly concentrate on three kinds of requirements confidentiality, availability and integrity. For example if a web server is attacked by attacker and organization owning that web server want server running anyhow. So security requirement for this case is availability.

Base score: It can be computed from following equations [9]:
Base Score *= round_to_1_decimal (((0.6 * Impact) + (0.4 * Exploitability) - 1.5) * f(Impact))*

Impact *= 10.41 * (1 - (1 - ConfImpact) * (1 - IntegImpact) * (1 - AvailImpact))*
Exploitability *= 20 * Access Vector * Access Complexity * Authentication*
f(impact) *= 0 if Impact=0 , 1.176 otherwise*

IV. CONDITIONAL PROBABILITY

We have used conditional probability method to calculate interdependent vulnerability scores. If two events A and B occur one after another then probability of second event is known as conditional probability, given probability of event A. Conditional probability is the probability of event B when sample space restricted to event A. CVSS independent score does not fulfill purpose of attack analysis. In attack chaining each vulnerability contribute in exploiting next vulnerability in attack. Therefore to calculate complete score of an attack path, we have used conditional probability.

$$P(A/B) = \frac{\mathrm{P}(\mathrm{A} \cap \mathrm{B})}{\mathrm{P}(\mathrm{B})}.$$

Table I: Base Sub group Metrics values

Sub group Metrics Name	Metrics value 1	Metrics value 2	Metrics value 3
Access Vector	Local Access : 0.395	Adjacent Network : 0.646	Remote : 1
Access Complexity	High : 0.35	Medium : 0.61	Low : 0.71
Authentication	Multiple : 0.45	Single : 0.56	None : 0.704
Confidentiality Impact	None : 0.0	Partial : 0.275	Complete : 0.660
Integrity Impact	None : 0.0	Partial : 0.275	Complete : 0.660
Availability Impact	None : 0.0	Partial : 0.275	Complete : 0.660

V. ATTACK PATH SCORE CALCULATION

Since CVSS provides a way to calculate score of a particular vulnerability. But sometimes it is not possible to attack the target machine directly. Therefore attacker can follow attack path which involves first the machine which is directly accessible followed by machine which can be compromised after exploiting first machine and so on which may lead to target. Since one machine can have large number of vulnerabilities and which can be compromised at a time. In that case we cannot determine which path is more vulnerable because CVSS gives only one machine score and we have to calculate multiple machines collective score. For this purpose we have used conditional probability of occurrence of vulnerabilities. Calculation of $P(A/B)$ as discussed in previous section is problematic. In order to resolve this situation, if v1 and v2 are two vulnerabilities exploited in succession in attack path, then probability $P(v1/v2)$ is ratio of attack path having both vulnerability v1 and v2, and total number of attack paths.

$$P(v1 \cap v2) = \frac{\text{Attack paths having both vulnerabilities}}{\text{Total no. of attack paths}}.$$

$$P(v1/v2) = \frac{P(v1 \cap v2)}{\mathrm{P}(\mathrm{v2})}$$

VI. EXAMPLE

To understand the concept of score calculation (discussed in previous section), let us consider an example, there are three machines. Vulnerability v2 on machine C can be affected by the attacker only after exploiting vulnerability v1 on machine B and machine A can be infected directly by exploiting vulnerabilities having CVSS scores as shown in fig. I. To calculate the final score, first convert all scores to base point 1 i.e. score 7 becomes 0.7 and so on. Then calculate $P(v1 \cap v2)$. For that purpose let us suppose there are total 8 attack paths having vulnerability 1 and 2 and out of which there are total 5 attack paths in which vulnerability v1 and v2 come in succession. Then

$$P(v1 \cap v2) = 5/8 = 0.62.$$

$$P(v1/v2) = \frac{P(v1 \cap v2)}{P(v2)} = 0.62/0.7 = 0.85$$

Hence final attack path score is 0.85.

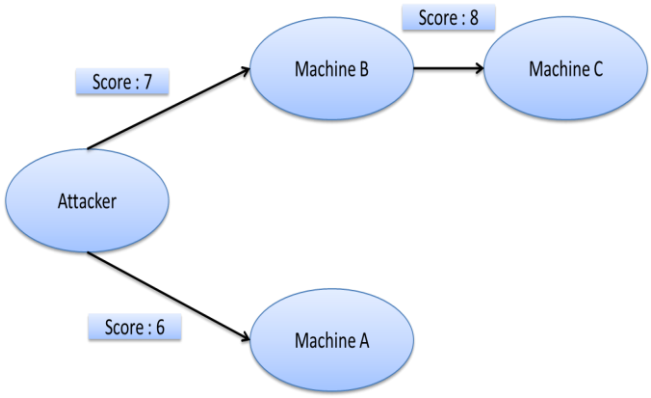

Fig I: Attack Path Example

VII. CONCLUSION

Vulnerability analysis alone is not enough in order to find complete network's vulnerability. Analysis of vulnerability considers vulnerability independently, but in real scenario attacker uses more than one vulnerabilities to break into a network. Vulnerability chaining is a very important technique for breaking the network security. Attack graph represents network's severity level in the form of picture showing nodes representing host and edges representing ways to make transition among these hosts. Attack graph have a number of attack paths. It's not necessary that every attack path is vulnerable so providing vulnerability score to these attack paths becomes very helpful in analyzing network security and these attack path's score gives administrator powers for prioritizing patching of vulnerabilities.

This paper proposed a new technique based on conditional probability to generate attack graph with the attack path score for each possible attack path in the network. This technique has been shown to be better suited to this task as it considers shortcomings in various techniques being used and try to remove these while generating attack graphs.

VIII. FUTURE WORK

Defense mechanism for coping with attacks depend on some parameters i.e. cost of deploying new technique, required time in defending and security requirements. Future work will concentrate on how administrator can set a threshold value for attack path's score. This threshold will be used in prioritization of attack paths. Attack path having score greater than threshold will be considered. This threshold will depend upon defense mechanism and resources requirement. Since our current state of work is restricted to only some vulnerabilities which are much less than required to build a completely secure network. As the dataset will scale, we can get more accurate attack path to the target host. To get complete set of attack paths, dataset should have information of all vulnerabilities that are present in target node and intermediate nodes as well. It is very difficult to maintain the accuracy of this graph, because as the new vulnerabilities come, dataset requires updating of vulnerabilities to keep the attack graph up to date. This limitation will be considered in future work to automatically update the dataset.

REFERENCES

[1] Sushil Jajodia and Steven Noel, "Topological Vulnerability Analysis. A Powerful New Approach For Network Attack Prevention, Detection and Response", World Scientific Press, 2007.

[2] Nirnay and S K ghosh., " An approach for Security Assessment of Network Configurations Using Attack Graph", IEEE Conference on network and communication , 2009, pp. 283-288.

[3] Cynthia Phillips and Laura Painton Swiler. Nspw '98, " Proceedings of the 1998, Workshop on New Security Paradigms", ACM Transaction Program on Language System, 1999.

[4] Richard Lippmann, Oleg Sheyner and Somesh Jha, " Automated Generation and Analysis of Attack graph", IEEE Symposium on Security and Privacy , 2002, pp. 273-284.

[5] Pavan Vejandla, Dipankar Dasgupta, Aishwarya Kaushal, and Fernando Nino, "Evolving Gaming Strategies for Attacker-Defender in a Simulated Network Environment", IEEE International Conference on Privacy, Security, Risk and Trust , 2010, pp. 889-896.

[6] Wyss, Schriner and T Gaylor, " Probabilistic Logic Modeling of Hybrid Network Architectures" , IEEE Conference on Local Computer Networks, 1996, pp. 404-413.

[7] Urvashi Garg, "Attack Graphs for Cyber Warfare", Unpublished master's thesis, Malaviya National Institute of Technology, Jaipur, India.

[8] Cve Details, Available: http://www.cve.mitre.org/.

[9] CVSS, Available: http://www.first.org/cvss/.

[10] Nessus, Available: http://www.nessus.org/.

[11] NuSMV, Available: http://nusmv.fbk.eu/.

[12] NetSPA, Available: http://dspace.mit.edu/handle/1721.1/29899.