ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Inspection of Vulnerabilities through Attack Graphs and Analyzing Security Metrics Used For Measuring Security in A Network.

R.Dhaya[1] D.Deepika[2]

Associate Professor, Department of CSE, Velammal Engineering College, Chennai, Tamilnadu, India[1]

M.E, Department of CSE, Velammal Engineering College, Chennai, Tamilnadu, India[2]

ABSTRACT – Cloud computing is now evolving like never before, with companies of all shapes and sizes adapting to this new technology. Industry experts believe that this trend will only continue to grow and develop even further in the coming few years. Storing information in the cloud could make vulnerable to external hack attacks and threats. As we are well aware, nothing on the Internet is completely secure. If any attack is detected the corresponding alert will be generated and it will be matched with the corresponding node in the attack graph. And finally countermeasure is applied to mitigate the attack progress. This paper discusses various attack graph construction and possible countermeasure selection techniques to stop the vulnerability from exploitation. And also identify the security of the network using security metrics such as VEA-bility metric also discussed.

KEYWORDS – Intrusion Detection System (IDS), security, attack graph, security metric.

I. INTRODUCTION

Cloud computing is a general term for anything that involves delivering hosted services over the Internet. These services are broadly divided into three categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). The name cloud computing was inspired by the cloud symbol that's often used to represent the Internet in flowcharts and diagrams.

A cloud service has three distinct characteristics that differentiate it from traditional hosting. It is sold on demand, typically by the minute or the hour; it is elastic -- a user can have as much or as little of a service as they want at any given time; and the service is fully managed by the provider (the consumer needs nothing but a personal computer and Internet access). Significant innovations in virtualization and distributed computing, as well as improved access to high speed Internet and a weak economy, have accelerated interest in cloud computing. Even it has an various advantages in terms of service and economical it is vulnerable to threats and malicious activities. There is a possibility for the virtual machines to be compromised and this leads a network become more and more vulnerable to the attacks.

To manage resources and host efficiently, the cloud servers need an effective mechanism to identify and prevent attacks. There is lot of security mechanisms are deployed to mitigate vulnerabilities. The attack graphs are constructed and each node represents possible attacks in the network. The alerts are generated whenever an scanner finds an vulnerability in the network. The corresponding alert is matched to the nodes in the attack graph. Then according to the identified attack, a possible countermeasure is applied. This will help the network administrator to maintain a securable network system.

This attack graph construction is only efficient in a small scale cloud system. The focus is on deploying this graph based approach for scalable cloud environment effectively and efficiently. It should provide security with less vulnerable to attacks.

The remainder of this paper is organized as follows: section 2 focuses on overview, describes the different approaches used to provide security and section 3 concludes the discussion.

II. OVERVIEW

To assure secure network services, security solutions have to be provided to overcome security issues.

2.1 SPAM DETECTION BY SPOT

In the detection of spam zombies, SPOT scans the outgoing messages from the network. It consists of spam filter which classifies messages as either spam or non spam. It reduces the number of experiments to reach the conclusion. It will automatically detect the vulnerable machines in the network. It is based on the Sequentially Probability Ratio Test (SPRT). It ensures that probability of machine being compromised is greater than probability of machine being normal. SPOT scans the outgoing messages and it keep track of sender IP address. For each recorded IP address, it maintains a log value. The value is calculated using the Bernoulli formula.

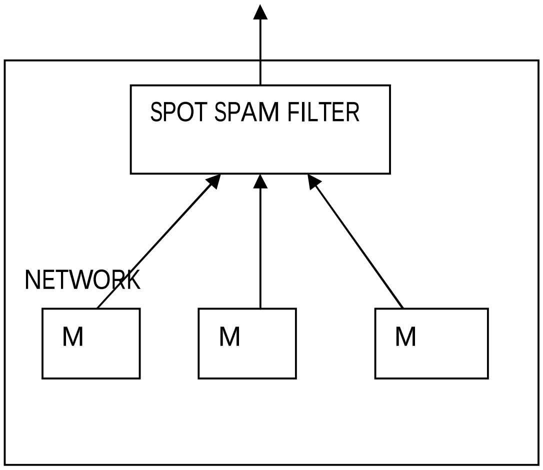


Fig1. SPOT mechanism [1]

In the fig1 the SPOT will record the incoming IP address of the sender machine. The content based spam filter will scan the incoming messages to it. The filter will classify the outgoing message either as a spam or non spam message. If vulnerability found then it will separate the message as spam message.

The log value is being compared to the user assigned constants. According to the comparison, the SPOT will make the machine either as a compromised state or normal state. If it put it in a normal, then it will assign zero to the log value and continue the test with new observations.

2.2 EFFICIENT NETWORK SECURITY ANALYZER

The network security tool should be able to generate formal vulnerability notations from the defects reporting community. It should be able to scale based on the size of the network. In this paper MulVAL is a technique to template the interaction of software defects and network configurations. It uses Datalog as its modelling language.

There is information needed as a input to the MulVAL analysis. The information in the vulnerability database, configuration information of each host which is encoded as Datalog facts. The reasoning engine consists of a collection of Datalog rules that captures the operating system behavior. After collecting all the information needed, the MulVAL analysis can be made. The inputs to the analysis are vulnerability name and configurations of the host which is identified using OVAL scanner, configuration of network, users of the network, type of model employed to make interaction, access permissions. After giving inputs to the MulVAL analysis, the output will be the detection of compromised machines in the network. According to the vulnerability the appropriate countermeasure has been applied.

After that MulVAL scanner will be running on the new inputs and the security procedures is satisfied. With all the information presented in Datalog, MulVAL runs effectively for networks with large amount of hosts. And it identified interesting network security problem in a real world network.

2.2.1 Disadvantages

This model can only reason about privilege escalation vulnerability and the denial of service vulnerability. Other types of vulnerabilities are found in a small number of scales.

2.3 OPTIMIZING COUNTERMEASURE USING OPTMAC

To optimize security solution in the absence of probability assignments to the model. This method helps to optimize the countermeasure needed for mitigating attacks exploitation. This method uses Attack Countermeasure Tree (ACT). ACT takes both attack and countermeasures. The goal of ACT to minimize the number of countermeasures and investment cost to adopt security methods and to maximize the benefits from optimizing a countermeasure set in ACT [2].

The ACT consists of three distinct classes of events as atomic attack events, detection events and mitigation events. It uses three logical gates like AND, OR, NOT to represent the connectivity of the nodes. In ACT graph, the nodes are represented as square, inverted triangle, oval. The square represents the basic attack event, the inverted triangle represents the repeated attack event, the oval represents the mitigation event.

2.3.1 Optimal countermeasure selection without probability assignments [3]

This method first consider the optimal countermeasure selection without probability assignments. Because assigning a probability value to all the events is difficult. To make optimal countermeasure set(OC set) minimal cut-set is found. A cut-set in Attack Tree(AT) represents an attack scenario and in ACT represent an attack countermeasure scenario. A cut-set is minimal if it cannot be reduced further. SHARPE [3] is an software package that contains algorithm for finding all the mincut. In mincut if the attacks occur and the countermeasure fail, then attack will exploited.

2.3.2 Selection of minimum number of countermeasure in ACT [3]

There are many ways to select the minimum number of countermeasure in ACT. Here two ways are discussed detail.

2.3.2.1 Covering all atomic events

The full cover of ACT achieved if the set of countermeasures in the optimal set (OPT) covers all the atomic attack events. In this Matrix T is generated from ACT mincut. Each column represent the countermeasure and each row represents the attack events. To find the optimal countermeasure set by minimizing the number of columns from the matrix T with an condition each row is covered by atleast one column.

2.3.2.2 Covering partial atomic events

System admin is interested in finding the best possible way of detecting only the critical section of their system. It is classified as partial cover with intent and partial cover without intent.

2.3.3 Automated generation of ACT

Once the attack graph is constructed successfully with preconditions and postconditions as nodes. The edges connect these nodes to the destination or targeted node. Given a set of precondition and a specific attack goal, the ACT is generated for that attack goal by pruning the system's exploit graph.

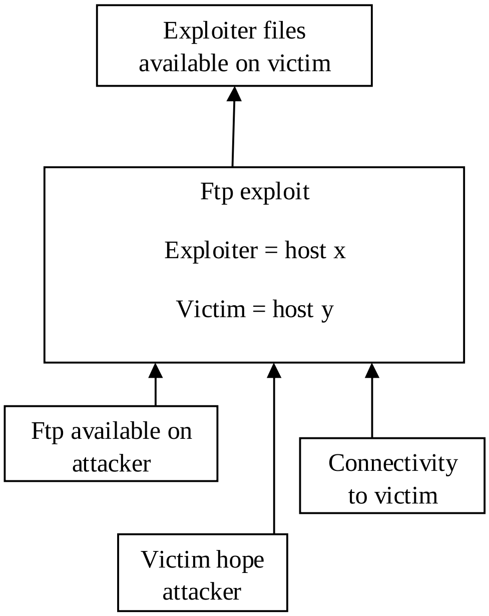
2.4 NETWORK VULNERABILITY ANALYSIS USING MONOTONICITY.

In this attack graph plays a major role in detecting vulnerabilities in the network. The attack graph generation depends on monotonicity. In this nodes are considered as attributes that is pre condition and post condition. The edges represents the set of exploits of interest. In this method concrete exploits are used. A generic attack is divided into many form of attacks called as concrete exploits.

A given exploit can have many numbers of pre conditions and post conditions. The number of edges is the product of number of preconditions and number of post conditions. Each attribute i.e., pre condition has an edge to the post condition. For a network configuration, attributes are partitioned to two sets.

Satisfied in the initial state and require some successful exploit to activate them. Unsatisfied attributes are arranged into two layers. Layer1 can be satisfied by applying a single exploit to the initial state. In Layer2 attributes require atleast two exploit and so on.

Post condition attributes



Pre condition attributes

Fig2. Example Exploit [4]

By applying this layer partition of attributes, the attack graph is constructed successfully. In this assumption of monotonicity is used to develop scalable graph-based representation for encoding attack trees.

2.5 SCORE BASED SECURITY METRIC

In this a security metric is developed to observe the numerous factors that uses the security of a network. This metric is the function of security scores along three dimensions called vulnerability ($V_{N)}$, exploitability($E_{N)}$ and availability($A_{N)}$ [5]. It uses data from network topology, attack graph, scores assigned by the Common Vulnerability Scoring System (CVSS).

In this network dimension is defined as a function of three dimensions scores for each host named as V(host), E(host), A(host) [5].

The vulnerability dimension is a function of two scores assigned by the CVSS. They are impact score measures the impact that a exploit will have on availability, integrity, and accessibility of information resources and the temporal score assigns a value based on the lifetime of the vulnerability. The vulnerability score is the exponential average of the host vulnerability scores or a maximum of 10.

The exploitability dimension is the sum of exploitability scores for each host on the network. This score is function of the exploitability score assigned by the CVSS. It evaluates the likelihood of exploitation.The Attackability dimension is the summation of the Attackability scores of each host.

Usually network comprises of lot of hosts. A host with multiple vulnerabilities is more vulnerable than a host with a single vulnerability. In this the severity, S, is defined as a average of the impact and temporal scores. The temporal score requires a user data. V(host) is an exponential average of the severity scores of the vulnerabilities on a host.

E(host) is the exponential average of the exploitability score for all host vulnerabilities and the value is multiplied by the ratio of network services on the host. A(host) is the ratio of attack paths produced by attack graphs to total number of possible attack paths, and it is also multiplied by 10.

The VEA-bility metric is given by

$$\text{VEA-bility}_N = 10 - ((V+E+A)_N / 3)$$ [5]

This metric is measurable, and it is used to compare the performance of a network configuration to other possible configurations by using score.

The metric allocates a numeric value from 0 to 10 to each network configuration. In this zero indicates the most vulnerable network configuration and the most secure etwork configuration is indicated by the value 10.

2.6 NETWORK SECURITY MEASUREMENT USING ATTACK GRAPH PATHS.

Due to the vast increasing of dependence on network services and information, Security of an network is remain unanswered. To improve the security there is a need of some metric to measure it. Vulnerabilities in a network get increased day by day. An individual low score vulnerability may cause other vulnerability to occur simultaneously. In this paper, to overcome the security problem a model is used based on the Bayesian networks (BN). It models the security states of networks and encode the probabilistic properties of the network vulnerabilities.

The exploit of an vulnerability is designed as a transition between the system states. It uses a attack grap(AG) developed through Topographical Vulnerability Analysis (TVA) [7]. The BNs are a Directed Acyclic Graph (DAG) with nodes represents variables and arcs represent conditional relationship among the variables.

The AG can be represented as a DAG coupled with the Conditional Probability Tables (CPT), encoding the conditional independencies for all nodes will constitute a BN. In this model, the nodes are assigned by score. This score is derived from CVSS[8]. The CPT encode the probability value of each node and its conditional dependencies. In this the nodes are assigned discrete values of T for successful exploit done by the attacker and F for unsuccessful exploit performed.

In this the main thing is to determine the probability of achieving the goal state where successful exploitation has been done.

In this model the conclusion is derived from many cases of network configuration. If a path to a goal state is less then the security of the network considerable. If a path to a goal state is more then the security of the network is more vulnerable to attacks.

III. CONCLUSION

This paper focuses on vulnerabilities and impacts of the vulnerabilities on the cloud server. In this paper have been discussed a several vulnerability detection techniques mitigation security metrics. Each techniques has its own advantages and disadvantages. One vulnerabilities may cause another vulnerabilities to be exploit simultaneously and leads to security problem. In order to overcome this problem, to propose and implement a new approach which is used to detect and mitigate compromised machines and provide security with less resource consumption.

REFERENCES

[1] Z. Duan, P. Chen, F. Sanchez, Y. Dong, M. Stephenson, and J. Barker, "Detecting Spam Zombies by Monitoring Outgoing Messages," IEEE Trans. Dependable and Secure Computing, vol. 9, no. 2, pp. 198-210, Apr. 2012.

[2] X. Ou, S. Govindavajhala, and A.W. Appel, "MulVAL: A Logic- Based Network Security Analyzer," Proc. 14th USENIX Security Symp., pp. 113-128, 2005.

[3] A. Roy, D.S. Kim, and K. Trivedi, "Scalable Optimal Countermeasure Selection Using Implicit Enumeration on Attack Countermeasure Trees," Proc. IEEE Int'l Conf.Dependable Systems Networks (DSN '12), June 2012.

[4] P. Ammann, D. Wijesekera, and S. Kaushik, "Scalable, graph- based network vulnerability analysis," Proc. 9th ACM Conf. Computer and Comm. Security (CCS '02), pp. 217-224, 2002.

[5] M. Tupper and A. Zincir-Heywood, "VEA-bility Security Metric: A Network Security Analysis Tool," Proc. IEEE Third Int'l Conf. Availability, Reliability and Security (ARES'08),pp. 950-957, Mar.2008.

[6] M. Frigault and L. Wang, "Measuring Network Security Using Bayesian Network-Based Attack Graphs," Proc. IEEE 32nd Ann. Int'l Conf. Computer Software and Applications (COMPSAC '08), pp. 698-703, Aug. 2008.
[7] S. Noel, S. Jajodia, A. Singhal. Security Risk Analysis With Attack Graph Metrics, Technical Report, CSIS, George Mason University, 2007.

[8] P. Mell, K. Scarfone, and S. Romanosky, "Common Vulnerability Scoring System (CVSS)," http://www.first.org/cvss/cvss-guide. html, May 2010.

[9] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J.M. Wing, "Automated Generation and Analysis of Attack Graphs," Proc. IEEE Symp. Security and Privacy, pp. 273-284, 2002,

[10] "NuSMV: A New Symbolic Model Checker," http://afrodite.itc.it :1024/nusmv. Aug. 2012.

[11] R. Sadoddin and A. Ghorbani, "Alert Correlation Survey: Framework and Techniques," Proc. ACM Int'l Conf. Privacy, Security and Trust: Bridge the Gap between PST Technologies and Business Services (PST '06), pp. 37:1-37:10, 2006.

[12] S. Roschke, F. Cheng, and C. Meinel, "A New Alert Correlation Algorithm Based on Attack Graph," Proc. Fourth Int'l Conf. Computational Intelligence in Security for Information Systems, pp. 58-67, 2011.

[13] E. Keller, J. Szefer, J. Rexford, and R.B. Lee, "No Hype: Virtualized Cloud Infrastructure without the Virtualization," Proc. 37th ACM Ann. Int'l Symp. Computer Architecture (ISCA '10), pp. 350-361, June 2010.

[14] National Institute of Standards and Technology, "National Vulnerability Database, NVD," http://nvd.nist. gov, 2012.

[15] K. Kwon, S. Ahn, and J. Chung, "Network Security Management Using ARP Spoofing," Proc. Int'l Conf. Computational Science and Its Applications (ICCSA '04), pp. 142-149, 2004.