Flexible and Robust k-Zero Day Safety Network Security Metrics to Measure the Risk on Different Vulnerabilities

Ms. Suchita D. Pawar
Department of Computer Engineering
JSPM, Hadpsar
Pune, India
E-mail: suchitadpawar18@gmail.com

Prof. H. A. Hingoliwala
Department of Computer Engineering
JSPM, Hadpsar
Pune, India
E-mail:ali_hyderi@yahoo.com

Abstract:- Today's computer systems face sophisticated attackers who combine multiple vulnerabilities to penetrate networks with devastating impact. The overall security of a network cannot be determined by simply counting the number of vulnerabilities. In fact, the security risk of unknown vulnerabilities has been considered as something immeasurable due to the less predictable nature of software flaws. This causes a major difficulty to security metrics, because a more secure configuration would be of little value if it were equally susceptible to zero-day attacks. In this paper, instead of just counting how much such vulnerability would be required for compromising network assets we can also attempting to rank unknown vulnerabilities. We propose a Flexible and Robust k-Zero Day Safety security model to rank the zero-day attacks by using collaborative filtering technique to different (types of) zero-day vulnerabilities and novel security metrics for uncertain and dynamic data.

Keywords:- *vulnerability, zero-day attacks, collaborative filtering*.

*****

I. Introduction

Today Internet connects and enables a growing list of critical activities from which people expect services and revenues. In other words, they trust these systems to be able to provide data and elaborations with a degree of confidentiality, integrity, and availability compatible with their needs. Unfortunately, this trust is often not based on a rational assessment of the risk to which the system could be exposed [2]. Users typically know only the interface of the system and, for example, they have too little information for evaluating the confidentiality of their credit card number: it could be even transmitted on an SSL armored link, but this does not help if on the other side it will be stored on a publicly available database! [2].

The scale and severity of security threats to computer networks have continued to grow at an ever increasing pace [p paper]. One of the main difficulties in securing computer networks is the lack of means for directly measuring the relative effectiveness of different security solutions in a given network, because “you cannot improve what you cannot measure.”[8]

A variety of authors have noted that identifying vulnerabilities in isolation is only a small part of securing a network, and that a significant issue is identifying which vulnerabilities an attacker can take advantage of through a chain of exploits [1]. For example, an attacker might exploit a defect in a particular version of ftp to overwrite the .rhosts file on a victim machine. In the next step, the attacker could remotely log in to the victim. In a subsequent step, the attacker could use the victim machine as a base to launch another exploit on a new victim, and so on [1].

II. Background

Every organization is at risk for zero-day exploits regardless of size. These exploits will often circulate for months until the vulnerability is made public, leaving organizations unprotected.

There were more zero-day vulnerabilities discovered in 2013 than in any previous year according to Symantec’s Internet Security Report of 2014. “The 23 zero-day vulnerabilities discovered represent a 61 percent increase over 2012 and are more than the two previous years combined”

Analysis of zero day vulnerabilities by following methods

A. Statistical-based techniques

Statistical-based techniques for the detection of exploits rely on attack profiles from past exploits that are now publically known. From those known exploits this defense technique adjusts the historical exploit’s profile parameters to detect new attacks. The quality of the detection is directly related to threshold limits set by the vendor or security professional using this technique. This technique determines what normal activity is and anything outside of normal is blocked or flagged.

The system that is utilizing this technique is online, the more accurate the system is at learning or determining what normal is. “Existing techniques in this approach perform static analysis and/or dynamic analysis on the packet payloads to detect the invariant characteristics reflecting semantics of malicious codes (e.g., behavioral characteristics of the decryption routine of a polymorphic worm)

B. Signature-based technique

Signature-based detection is often used by virus software vendors who will compile a library of different malware signatures. They will cross reference these signatures with local files, network files, email or web downloads depending on settings chosen by the user. These libraries are constantly being updated for new signatures that often represent the signatures of new exploited vulnerabilities. The technique is often one step behind a zero-day exploit because this technique requires a signature to be in the signature library for detection. This is the reason virus software vendors are frequently updating their virus definitions.

Signature-based techniques are classified by content-based, semantic-based and vulnerability-based signatures and are somewhat effective against polymorphic worms.

C. Behavior-based technique

The activity of a program can be viewed as malicious or benign based on the requirements of the code. "Behavior-based techniques look for the essential characteristics of worms which do not require the examination of payload byte patterns"

The goal of such techniques is to predict the future behavior of a web server, server or victim machine in order to deny any behaviors that are not expected. Those behaviors are learned by the current and past interactions with the web server, server or victim machine. This technique relies on the ability to predict the flow of network traffic.

D. Hybrid detection-based technique

Hybrid-based techniques combine heuristics with various combinations of the three previous techniques which are statistical-based, signature-based, and behavior-based techniques. Using a hybrid model technique will overcome a weakness in any single technique.

The benefits of their hybrid technique are four fold:

- Proposal of an efficient technique that offers better sensitivity and specificity by identifying zero-day attacks from data collected automatically on high interaction honeypots.
- Strengthening of the basic existing techniques by combining the advantages of existing techniques and minimizing their disadvantages.
- This technique does not need prior knowledge of zero-day attacks and uses Honeynet as an anomaly detector.
- This technique can detect zero-day attacks in its early phase and can contain the attack before major consequences occur.

III. LITERATURE SURVEY

In [1] P. Mell, K. Scarfone, and S. Romanosky (2006), main goal of Common vulnerability Scoring System (CVSS) is "Good enough" for non-expert administrator, Relative Simplicity and efficient representation of vector. They provide security analysts and vendors standard ways for assigning numerical scores to known vulnerabilities that are already available in public vulnerability databases, such as the National Vulnerability Database (NVD). But Temporal/Environmental aspects not well-tested, Requires good documentation. In[2] Common Weakness Scoring System (CWSS) the process of discovering new vulnerabilities, automated and human analysis will find weaknesses.

In [5] R. Lippmann, K. Ingols, C. Scott, K. Piwowarski, K. Kratkiewicz, M. Artz, and R.Cunningham (2006), Defense in depth is a common strategy that uses layers of firewalls to protect Supervisory Control and Data Acquisition (SCADA) subnets and other critical resources on enterprise networks. NetSPA (NETwork Security and Planning Architecture) verifies and, if necessary, provides suggestions to restore defense in depth for large enterprise networks. NetSPA successfully imported vulnerability scanner and firewall configuration information and was able to produce attack graphs and make recommendations in only a few minutes.

In [9] Mohammed, M.M.Z.E.; Chan, H.A; Ventura, N.; Pathan, A-S.K. (2013), Their technique first tries to detect zero-day polymorphic worms and then tries to prevent them. "STF observes all network traffic at an edge network and the Internet. The traffic is passed simultaneously to both Honeynet and IDS/IPS (Intrusion Detection System/Intrusion Prevention System) sensors through a port mirroring switch". Suspicious Traffic Filter (STF) is the first defense layer from zero-day attack. Zero-day Attack Evaluation (ZAE) takes input (malicious traffic) from STF to evaluate and analyze captured zero-day attack. Signature Generator (SG) generates new signature for zero day attack and updates the signature database in STF.

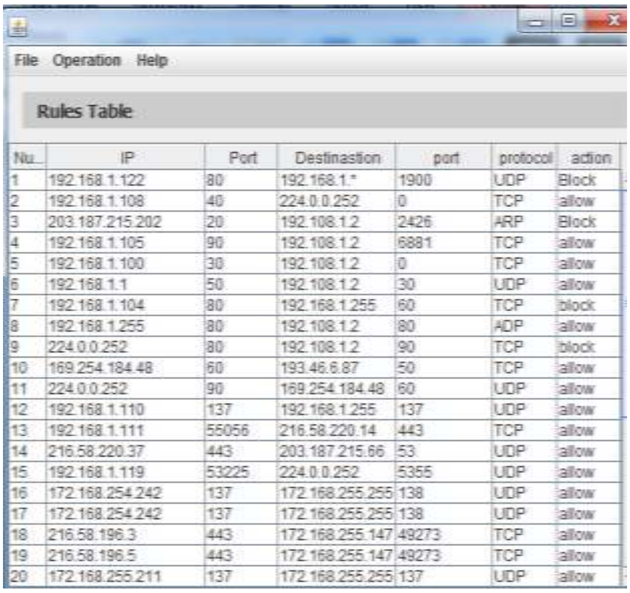
In [3] M. Frigault, L. Wang, A. Singhal, and S. Jajodia (2008), In this paper Explores the causal relationships between vulnerabilities and measuring network security in a dynamic environment. In this module used tool for measuring network security by integrating attack graphs generated by the TVA system with CVSS scores provided by NVD. Tool accuracy is important to get optimal result.

In [5] J. Homer, X. Ou, and D. Schmidt (2009), apply probabilistic reasoning to produce a sound risk measurement. Running time of algorithm depends on size of data sets and interconnection in attack graph. If in attack graph their exist lots of interconnectivity in exploits then it not able to generate optimal result.

IV. PROPOSED SYSTEM

As we studied in the literature, there are many methods were discussed and presented by various authors over the network security, which is important issue in today's life in zero day attack we can detect the known as well as unknown vulnerabilities in network. In [1] author considers three groups of vulnerability, detects vulnerability and measures the score of vulnerability in between 0-10. In [2] author considers weakness of vulnerability and detects them. But in both paper we are unable to rank them. In [6] NetSPA is tool used for vulnerability scanner it also used for just vulnerability detection. In above survey system just provides vulnerability detection but Alleviation of security risk is an important task in enterprise network security management.

Network security is important issue in today's life in zero day attack we can find the known as well as unknown vulnerabilities in network. In vulnerability each triple indicates an exploit <vulnerability, source host, destination host> and a pair indicates a condition <condition, host>. Consider a firewall policy, which consists of a sequence of rules that define the actions performed on packets that satisfy certain conditions. The rules are specified in the form of <condition; action>. A condition in a rule is composed of a set of fields to identify a certain type of packets matched by this rule. Services running on each host are marked beside that host and firewall rules are depicted below each firewall. We will assume different services or firewalls involve different zero-day vulnerabilities.



Rules Table

Nu...	IP	Port	Destinastion	port	protocol	action
1	192.168.1.122	80	192.168.1.*	1900	UDP	Block
2	192.168.1.108	40	224.0.0.252	0	TCP	allow
3	203.187.215.202	20	192.108.1.2	2426	ARP	Block
4	192.168.1.105	90	192.108.1.2	6881	TCP	allow
5	192.168.1.100	30	192.108.1.2	0	TCP	allow
6	192.168.1.1	50	192.108.1.2	30	UDP	allow
7	192.168.1.104	80	192.168.1.255	60	TCP	block
8	192.168.1.255	80	192.108.1.2	80	ADP	allow
9	224.0.0.252	80	192.108.1.2	90	TCP	block
10	169.254.184.48	60	193.46.6.87	50	TCP	allow
11	224.0.0.252	90	169.254.184.48	60	UDP	allow
12	192.168.1.110	137	192.168.1.255	137	UDP	allow
13	192.168.1.111	55056	216.58.220.14	443	TCP	allow
14	216.58.220.37	443	203.187.215.66	53	UDP	allow
15	192.168.1.119	53225	224.0.0.252	5355	UDP	allow
16	172.168.254.242	137	172.168.255.255	138	UDP	allow
17	172.168.254.242	137	172.168.255.255	138	UDP	allow
18	216.58.196.3	443	172.168.255.147	49273	TCP	allow
19	216.58.196.5	443	172.168.255.147	49273	TCP	allow
20	172.168.255.211	137	172.168.255.255	137	UDP	allow

Figure 1: firewall Rule list

Note that the symbol "*" utilized in firewall rules denotes a domain range. For instance, a single "*" appearing in the IP address field represents an IP address range from 0.0.0.0 to 255.255.255.255.

In this model we can able to count known as well as unknown dynamic vulnerabilities and design optimal firewall rule policy to block them. In this model we calculate the risk of vulnerability to affect the security of system. Considering the risk value ranks the vulnerability and reduces the cost of system security.

The overall process of to captured vulnerability, calculate risk and rank them and design new security policy is very complex and costly. So to overcome this problem recently we presented one new system. To find vulnerability in network is NP hard problem to convert it in to NP complete we consider firewall rule policy to detect vulnerability and design optimized firewall rule list for security safety. In this system we monitor network traffic and capture network packet and display the allow packet vulnerability count which are able to attack our system. To reduce the complexity of problem we monitor small network path by using shortest path algorithm we select small network model and reduce vulnerability count and display vulnerability list. In this system calculate the risk of captured vulnerability list by using mathematical formula. Then it ranks the vulnerability according to CVSS dataset and risk values in three categories.

In this model we can able to count known as well as unknown dynamic vulnerabilities and design optimal firewall rule policy to block them. We can also calculate the risk of vulnerability to affect the security of system. Considering the risk value rank the vulnerability to reduce the cost of system security. Collaborative filtering technique is used for ranking the vulnerability

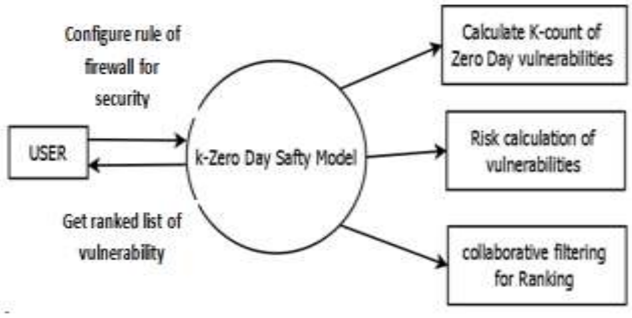

Configure rule of firewall for security
USER
Get ranked list of vulnerability
k-Zero Day Safty Model
Calculate K-count of Zero Day vulnerabilities
Risk calculation of vulnerabilities
collaborative filtering for Ranking

Figure 2. Module Structure

PROPOSED SYSTEM MODULES

A. *Computing k count*

In this model firewall rule list dataset is designed by using network rules and used jpcap and WinPcap software's to capture the data packets travelled in network. After capturing data packets matches their sources and destination ip addresses in the firewall rule list. ip addresses of data packets are allowed can able to attack our system that packets transferring protocol count as vulnerability. Then optimized firewall rule list for security.

Algorithm 1: To computing K count
Module 1: computing K vulnerability count:
Input: Firewall Rule list F, Captured data packet in network.
Output: A non negative vulnerability K count
Method:

```
1. Take firewall Rule list F
2. Capture data packet in network P = {transport layer protocol, source IP address, transport layer source port, destination IP address, transport layer destination port}.
   While
3. Match captured data packet p with firewall rule list F
4. If capture packet match with rule list
5. Then increment vulnerability count of allow packet
6. Return K count
```

B. *Calculate shortest path to reduce the vulnerability count*

In this system to minimize the problem we consider small network model by using dijkstra's shortest path algorithm and firewall security rules for network security. In this we can able to configure optimal firewall rule list after counting vulnerability count and make our system more secure.

Algorithm 2: To find shortest path and generate new K count
Module 2: find shortest path and reduce K count
Input: captured packets and IP's
Output: A non negative vulnerability K count, shortest path from source to destination.
Method:

```
1. Take all node(IP)
2. Send test packet to all node
3. Calculate response time to get acknowledgement packet
4. Response time is less of those nodes are consider as shortest distance between two nodes.
5. Return shortest path
6. Return reduced K count
```

C. Risk calculating and Ranking the vulnerability

In existing system is just find the count of the known vulnerabilities but not able to make system secure in proposed system we are providing risk calculation of known as well as unknown dynamic vulnerabilities. To calculate the risk of vulnerability we considers following formula

Vulnerability Risk = Threat * Vulnerability * Impact

Where;

Threat: - is the CVSS score of vulnerability

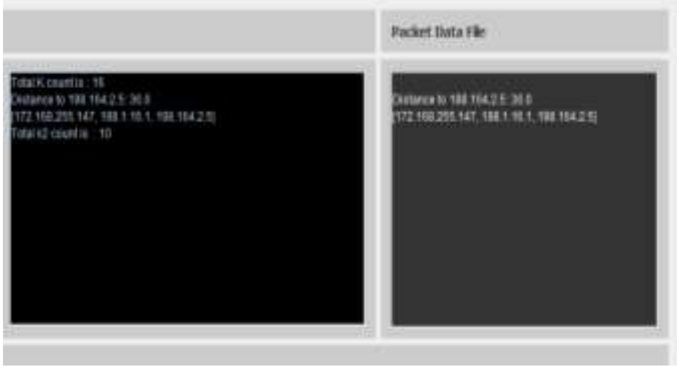
Vulnerability: - is the no of vulnerability count of same type

Impact: - is the importance value of vulnerability

In this system we are also rank the vulnerabilities according to their risk by using collaborative filtering. In ranking we are consider the three categories ie high-low-medium.

V. RESULT ANALYSIS

Output of first algorithm is K count of vulnerability in network to minimize the count we use dijkstra's shortest path algorithm to reduce vulnerability count as shown following window.

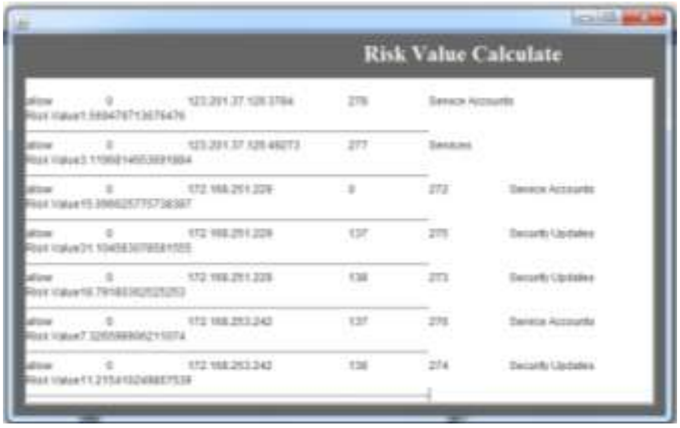


```
Total K count is : 16
Distance to 188.164.2.5: 30.0
[172.168.255.147, 188.1.16.1, 188.164.2.5]
Total k2 count is : 10
```

```
Distance to 188.164.2.5: 30.0
[172.168.255.147, 188.1.16.1, 188.164.2.5]
```

Figure 3. Output window of dijkstra's shortest path algorithm

We calculate the risk of captured vulnerability and show the risk value using risk calculation formula.



```
allow     0     123.201.37.126.3784     278     Service Accounts
Risk Value1.569479713676476
allow     0     123.201.37.126.49273    277     Services
Risk Value3.1196814653691884
allow     0     172.168.251.229         0       272     Service Accounts
Risk Value15.096625777573 8387
allow     0     172.168.251.229         137     275     Security Updates
Risk Value31.104583078581555
allow     0     172.168.251.229         138     273     Security Updates
Risk Value18.791803625 25253
allow     0     172.168.253.242         137     276     Service Accounts
Risk Value7.326598906211074
allow     0     172.168.253.242         138     274     Security Updates
Risk Value11.215410248887538
```

Figure 4. Risk value of captured vulnerability

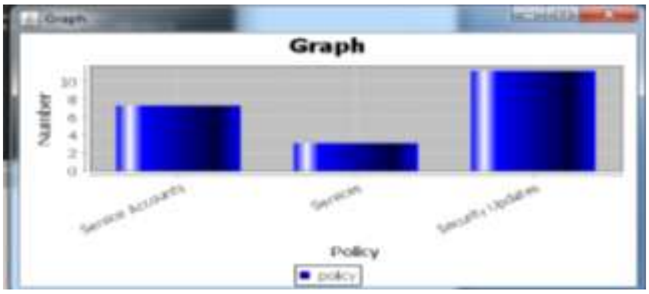
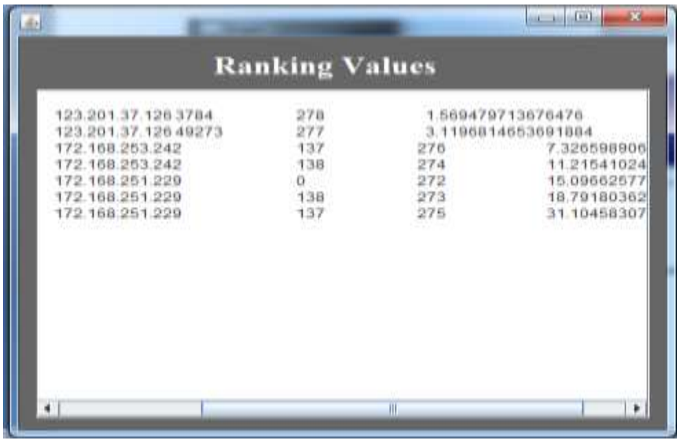


Figure 5.vulnerability risk value graph



```
123.201.37.126.3784         278          1.569479713676476
123.201.37.126.49273        277          3.1196814653691884
172.168.253.242             137          276           7.326598906
172.168.253.242             138          274           11.21541024
172.168.251.229             0            272           15.09662577
172.168.251.229             138          273           18.79180362
172.168.251.229             137          275           31.10458307
```

Figure 6. Ranking of captured vulnerability according to risk value in increasing order

VI. CONCUSION AND FUTURE SCOPE OF INHANCEMENT

In this project we design the security model for zero day attack. We are able to catch the total count of known and dynamic vulnerabilities in network which affect our system security. In previous system we are not able to calculate the risk of vulnerability as well as not able to rank the vulnerabilities for network hardening, this system provide this function. In this model we are using collaborative filtering for ranking vulnerabilities. In this model we are design practical model for firewall system. We configure optimal list of firewall rule list to make our system more secure and find the known as well as unknown and dynamic vulnerabilities in network.

The scope of our metric is limited by the three basic assumptions about zero-day vulnerabilities (the existence of network connectivity, vulnerable services on destination host, and initial privilege on source host). The model will be more suitable for application to the evaluation of penetration attacks launched by human attackers or network propagation of worms or bots in mission critical networks. An important future work is to broaden the scope by accommodating other types of attacks (e.g., a time bomb which requires no network connection).

REFERENCES

[1] P. Mell, K. Scarfone, and S. Romanosky, "Common Vulnerability Scoring System," IEEE Security and Privacy, vol. 4, no. 6, pp. 85-89, Nov./Dec. 2006.(24)

[2] MITRE Corp., "Common Weakness Scoring System (CWSS)," http://cwe.mitre.org/cwss/, 2010.(37)

[3] M. Frigault, L. Wang, A. Singhal, and S. Jajodia, “Measuring Network Security Using Dynamic Bayesian Network,” Proc. Fourth ACM Workshop Quality of Protection (QoP ’08), 2008.(9)

[4] Kaur, R.; Singh, M., "Efficient hybrid technique for detecting zero-day polymorphic worms," Advance Computing Conference (IACC), 2014 IEEE International ,pp.95,100, 21-22 Feb. 2014.

[5] J. Homer, X. Ou, and D. Schmidt, “A Sound And Practical Approach to Quantifying Security Risk in Enterprise Networks,” technical report, Kansas State Univ., 2009.(12)

[6] R. Lippmann, K. Ingols, C. Scott, K. Piwowarski, K. Kratkiewicz, M. Artz, and R. Cunningham, “Validating and Restoring Defense in Depth Using Attack Graphs,” Proc. IEEE Conf. Military Comm. (MILCOM’ 06), pp. 981-990, 2006.(20)

[7] N. Poolsappasit, R. Dewri, and I. Ray, “Dynamic Security Risk Management Using Bayesian Attack Graphs,” IEEE Trans. Dependable Secure Computing, vol. 9, no. 1, pp. 61-74, Jan. 2012.(31)

[8] L. Wang, S. Jajodia, A. Singhal, and S. Noel, “k-Zero Day Safety: Measuring the Security Risk of Networks against Unknown Attacks,” Proc. 15th European Conf. Research Computer Security (ESORICS ’10), pp. 573-587, 2010.(41)

[9] Mohammed, M.M.Z.E.; Chan, H.A; Ventura, N.; Pathan, A-S.K., "An Automated Signature Generation Method for Zero-Day Polymorphic Worms Based on Multilayer Perceptron Model," Advanced Computer Science Applications and Technologies (ACSAT), 2013 International Conference on , vol., no., pp.450,455, 23-24 Dec. 2013

[10] Alosefer, Y.; Rana, O.F., "Predicting client-side attacks via behavior analysis using honeypot data," Next Generation Web Services Practices (NWeSP), 2011 7th International Conference on Next Generation Web Services Practices, pp.31,36, 19-21 Oct. 2011.

[11] D. Balzarotti, M. Monga, and S. Sicari, “Assessing the Risk of Using Vulnerable Components,” Proc. ACM Second Workshop Quality of Protection (QoP ’05), pp. 65-78, 2005.