International Journal of Scientific Research in Computer Science and Engineering
Vol.6, Issue.3, pp.35-38 , June (2018)
Research Paper
E-ISSN: 2320-7639

An Implementation of an Vulnerability Management in Complex Networks and Defining Severity

Rikam Palkar[1*], Swati Chopade[2]

[1,2]Department of Master of Computer Applications, Veermata Jijabai Technological Institute,Mumbai, India

Available online at: www.isroset.org

Received: 03/Jun/2018, Revised: 11/Jun/2018, Accepted: 21/Jun/2018, Online: 30/Jun/ 2018

***Abstract*—** Complex systems are found in an extensive variety of areas from technological and social to biological environments. In spite of this scope of uses and settings in which complex systems are utilized as models, examines propose that numerous genuine systems are represented by a comparable elements. An important characteristic is that in general such networks are robust against failures but vulnerable against targeted attacks. Impossible to miss trademark: resilience. In this technological era business endeavor to stay aware of evolving technology, it implies that they have a high capacity to absorb changes. However, resilience mechanisms are not present per se in technological networks. Thus, this work presents a framework for vulnerability assessment, vulnerability analysis and vulnerability management in versatile technological networks

Keywords- Network, VMP

I. INTRODUCTION

We are surrounded by innovation and systems which may be classified or modelled as complex networks. As complex networks are present in several domains in environments which are usually large, complex, highly dynamic and heterogeneous.

A vulnerability is loophole in the network that can be exploited by one or more threats.

In layman terms vulnerability is a loophole associated with a network or a device within an organization which can be exploited by intruder/malicious user. Where vulnerability management is a passive technique under cyber security which takes care of patching of such loopholes.

Vulnerability management may seems like a single term but it is a 2 stage process.

1. Vulnerability scans
2. Remediation/ mitigation of vulnerabilities or Risk acceptance

Vulnerability management is the process in which vulnerabilities in organization's network are identified by scanning hosts/ machines through network and applications running within a network and then the risks of these vulnerabilities are evaluated. Vulnerabilities classified as a critical, high, medium and low type of severity. This leads to correcting the vulnerabilities and removing them from network which leads to protection of organization in passive way.

II. LITERATURE REVIEW

Managing large-scale networks is a complex task. Both humans and automated entities make errors when configuring them, potentially increasing their own security exposure. Under this perspective, vulnerability management constitutes a crucial activity. The CVE3 language, introduced by the MITRE Corporation, is an effort for standardizing the enumeration of known information security vulnerabilities. Nevertheless, it only provides means for informing about their existence and not for their assessment. In order to cope with these problems, MITRE has developed the OVAL language as an effort to standardize the process by which the state of a computer system can be assessed and reported. OVAL is an XML-based language that allows the expression of specific machine states such as vulnerabilities, configuration settings, and patch states.

Following metrics are used to define CVSS score for each vulnerability.

1. Base Score Metrics:

Exploitability Metrics			
*Attack Vector (AV)**			
Local (AV:L)	Adjacent Network (AV:A)	Network (AV:N)	Physical (AV:P)
*Attack Complexity (AC)**			
High (AC:H)	Low (AC:L)		
*Privileges Required (PR)**			

None (PR:N)	Low (PR:L)	High (PR:H)	
*User Interaction (UI)**			
None (UI:N)	Required (UI:R)		
Score (S)*			
Unchanged (S:U)	Changed (S:C)		
Impact Metrics			
*Confidentiality Impact (C)**			
None (C:N)	Low (C:L)	High (C:H)	
*Integrity Impact (I) **			
None (I:N)	Low (I:L)	High (I:H)	
Availability Impact (A)			
None (A:N)	Low (A:L)	High (A:H)	

* - All base metrics are required to generate a base score.

2. Temporal Score Metrics:

Exploitability (E)	Remediation Level (RL)
Not Defined (E:X)	Not Defined (RL:X)
Unproven that exploit exists (E:U)	Official fix (RL: O)
Proof of concept code (E:P)	Temporary fix (RL:T)
Functional exploit exists (E:F)	Workaround (RL:W)
High (E:H)	Unavailable (RL:U)

Report Confidence (RC)
Not Defined (RC:X)
Unconfirmed (RC:U)
Uncorroborated (RC:R)
Confirmed (RC:C)

3. Environmental Score Metrics:

Base Modifiers				
Attack Vector (AV)				
Not Defined (MAV:X)	Network (MAV:N)	Adjacent Network (MAV:A)	Local (MAV:L)	Physical (MAV:P)
Attack Complexity (AC)				
Not Defined (MAC:X)		Low (MAC:L)	High (MAC:H)	
Privileges Required (PR)				
Not Defined (MPR:X)	None (MPR:N)	Low (MPR:L)	High (MPR:H)	
User Interaction (UI)				
Not Defined (MUI:X)	None (MUI : N)	Required (MUI : R)		
Scope (S)				
Not Defined	Unchanged	Changed (MS:C)		
(MS:X)	(MS:U)			
Impact Metrics				
Confidentiality Impact (C)				
Not Defined (MC:X)	None (MC:N)	Low (MC:L)	High (MC:H)	
Integrity Impact (I)				
Not Defined (MI:X)	None (MI:N)	Low (MI:L)	High (MI:H)	
Availability Impact (A)				
Not Defined (MA:X)	None (MA:N)	Low (MA:L)	High (MA:H)	
Impact Sub score Modifiers				
Confidentiality Requirement (CR)				
Not Defined (CR:X)	Low (CR:L)	Medium (CR:M)	High (CR:H)	
Integrity Requirement (IR)				
Not Defined (IR:X)	Low (IR:L)	Medium (IR:M)	High (IR:H)	
Availability Requirement (AR)				
Not Defined (AR:X)	Low (AR:L)	Medium (AR:M)	High (AR:H)	

CVSS v3 Equations:

1. Base

The Base Score is a function of the Impact and Exploitability sub score equations. Where the Base score is defined as,

If (Impact sub score <= 0) 0 else,

Scope Unchanged4 $Roundup\ (Minimum\ [(Impact + Exploitability), 10])$

Scope Changed $Roundup\ (Minimum\ [1.08 \times (Impact + Exploitability), 10])$

And the Impact sub score (ISC) is defined as,

Scope Unchanged $6.42 \times ISC\text{Base}$

Scope Changed $7.52 \times [ISCBase - 0.029] - 3.25 \times [ISCBase - 0.02]15$

Where,

$ISCBase = 1 - [(1 - ImpactConf) \times (1 - ImpactInteg) \times (1 - ImpactAvail)]$

And the Exploitability sub score is,

$8.22 \times AttackVector \times AttackComplexity \times PrivilegeRequired \times UserInteraction$

2. Temporal

The Temporal score is defined as,

$Roundup(BaseScore \times ExploitCodeMaturity \times RemediationLevel \times ReportConfidence)$

3. Environmental
The environmental score is defined as,
If (Modified Impact Sub score <= 0) 0 else,

If Modified Scope is Unchanged Round up (Round up (Minimum [(M.Impact + M.Exploitability), 10]) × Exploit Code Maturity × Remediation Level × Report Confidence)

If Modified Scope is Changed Round up (Round up (Minimum [1.08 × (M.Impact + M.Exploitability), 10]) × Exploit Code Maturity × Remediation Level × Report Confidence)

And the modified Impact sub score is defined as,

If Modified Scope is Unchanged 6.42 × $[ISCModified]$

If Modified Scope is Changed 7.52 × $[ISCModified - 0.029]$-3.25× $[ISCModified - 0.02]$ 15

Where,
$ISCModified = Minimum$ $[[1 - (1 - M.IConf \times CR) \times (1 - M.IInteg \times IR) \times (1 - M.IAvail \times AR)], 0.915]$

The Modified Exploitability sub score is,
$8.22 \times M.AttackVector \times M.AttackComplexity \times M.PrivilegeRequired \times M.UserInteractio$n
4 Where "Round up" is defined as the smallest number, specified to one decimal place which is equal to or higher than its input. For example, Round up (4.02) is 4.1; and Round up (4.00) is 4.0.

III. PROPOSED SOLUTION

Vulnerability Assessment Process:

Roles and responsibilities

When building a Vulnerability Assessment Process, the following roles should be established within the organization:

a) Security Officer: The security officer is the owner of the entire process. This person designs the process and makes sure it is implemented as designed.
b) Vulnerability Engineer: This role is responsible for configuring the vulnerability scanner and scheduling the various scans.
c) Asset Owner: The asset owner of the IT asset that is scanned by the vulnerability management process. This role should decide whether identified vulnerabilities are mitigated or their associated risk acceptance.
d) IT System Engineer: The IT system engineer is typically responsible for implementing remediating actions defined as a result of detected vulnerabilities.

Vulnerability Management Process (VMP)

Vulnerability management process consists of five phases:

a) Preparation
b) Vulnerability scan
c) Define remediating actions
d) Implement remediating actions
e) Rescan

Preparation
The preparation phase is mainly the responsibility of the Security Officer in an organization. The first step is to define the scope of the vulnerability management process. It is important to obtain an agreement which systems will be included or excluded from the vulnerability management process. Besides the in scope systems, an organization should also determine the type of scans

Vulnerability scan
Once the preparation phase is complete, the next phase of the process begins and the initial vulnerability scans are performed. Any issues which occurs during the scans, for example systems becoming unavailable or poor application response, should be recorded since this may happen again in the future. In this case, actions may be defined to reduce the impact of future scans on the stability or performance of the target systems.

c. Define remediating
Actions In the next phase, the asset owners, with the cooperation of the security officer and the IT department, will define remediating actions. The security officer will analyze the vulnerabilities, determine the associated risks and will provide input on risk remediation. The IT department will analyze the vulnerabilities from a technical perspective and answer questions such as if patches are available or whether the configuration can be hardened? The IT department recommendation also includes the feasibility of the possible remediating action such as whether installing a certain patch will result in the application no longer be supported by the vendor.

Risk Acceptance
In case asset owners decide to accept the risk, it should be documented through a risk acceptance process. A risk acceptance or waiver process is a formal process in which an exception to the security policies can be requested. This request is analyzed with regards to risks the organization would be exposed to if the exception is granted. If possible, compensating controls to remediate these risks are proposed. In the final step of a risk waiver process, the asset Common Vulnerability Scoring System owner analyses the risks, whether or not compensating controls can be foreseen. This allows the asset owner to make thoughtful decisions with regards to accepting the risk. The ability to signoff is determined based on the level of risk. Usually high risks can

only be accepted by management of an organization, whereas small risks can be accepted by asset owners

d. Implement remediating actions
The planned remediating actions should be executed in line with the agreed timeframes. If a problem occurs with implemented remediation, it should be recorded. Alternative actions should be defined by the asset owner based on recommendations by the security officer and the IT department. These new or other remediating actions should then be implemented. The security officer should track the status of the remediating actions.

e. Rescan
Once a vulnerability is remediated, a rescan has to be scheduled to verify the remediating actions have been implemented. This scan will be performed using the same vulnerability scanning tools and identical configuration settings as the initial scan. This step is very important to prevent inaccurate results due to configuration errors. Typically a rescan is scheduled after the deadline for implementing remediating actions. For these scans, the same types of reports generated during the initial scan are created. For follow-up, management and asset owners will be interested to know whether the remediating actions have been effectively implemented and whether any residual risk remains. The IT department will be interested in how effective the remediating actions have been implemented.

IV. EXPERIMENT

The CWE Common Weakness Enumeration is a software community and a formal list of software weaknesses. Its definitions and descriptions support the finding of these common types of software security flaws in code prior to fielding. The NVD uses CWE as a classification mechanism that differentiates CVEs by the type of vulnerability that they represent. When we gathered the CVEs from the CWE-ID, we reviewed their descriptions and found that 75% of the descriptions have similar structures. We also realized the name of the vulnerability type can be found in the explanation. The keyword of the summary is the second step to decide the vulnerability types of the CVEs. In the second step, we searched using keywords from the description to further filter the remaining CVEs. The majority of the keywords are found in the beginning or middle of the descriptions. Once step one and two were completed to gather and sort the data, we allocated the vulnerability types to the rest of the CVEs from their references. One of the references of CVE is the Open Source Vulnerability Database (OSVDB) – Open source Vulnerability Database, which offers the attack types, CVE-ID and the attack type of vulnerability. We looked for the CVEs which did not have assigned vulnerability types, but the attack types are in the OSVDB. In the select 15 vulnerability types. We were now able to yes

Following the three steps of filtering and sorting the data as we mentioned before, we found that roughly 80% of all CVEs were CVSS scores, and CVSS base metrics for the four years selected. The CVSS is an open framework to measure the relative severity of software vulnerabilities. It offers a structured approach by the standardized vulnerability scores and prioritized risk. There are three metric groups in CVSS: Base, Temporal, and Environmental.
Access complexity (AC), Access vector (AV), Authentication (AU), Confidentiality impact (CI), Integrity impact (II), Availability impact (AI).

V. CONCLUSION

Algorithm will defined score for vulnerability through which we can distinguish vulnerability as per their severity. It can be classified into 4 categories

1. Critical
2. High
3. Medium
4. Low

For critical score should be more than 8, for high score should be more than 6, for medium score should reach 4 and for low score should cross 2.

If vulnerability is classified as a critical then organization should take necessary action to mitigate vulnerability as soon as possible. It should be added into company's policy. It helps the IT professionals to predict threats and protect organizations. Bringing focus on the vulnerability analysis, it is hoped that this research will serve as a reference to guide a wide cross section of people in the IT and security field. By analyzing vulnerabilities score, IS professionals will be better informed in developing policies that more closely reflect the vulnerability threat landscape. Software developers can use these trends to guide them in development of better coded software, and making them resilient to these vulnerabilities. It is also expected that knowledge of analysis can influence the development of security strategies developed by IS professionals.

REFERENCES

[1] "CWE- Common Weakness Enumeration",Available: http://nvd.nist.gov/
[2] "NVD Data Feed and Product Integration", National Vulnerability Database, Available: http://nvd.nist.gov/download.cfm
[3] N. W. Adger,"Vulnerability," Global Environmental Change, no. 3, pp. 268–281, August.
[4] A. Reka, J. Hawoong, and B. Albert-Laszlo,"Error and attack tolerance of complex networks," Nature, no. 6794, pp. 378–382, July.
[5] "CVE, Common Vulnerabilities and Exposures." http://cve.mitre.org/
[6] http://www.metasploit.com/
[7] http://www.tenable.com/products/nessus
[8] M.E.J. Newman "Analysis of weighted networks", The American Physical Society.
[9] "The OVAL Language." http://oval.mitre.org/ - Open vulnerability and assessment language.

