*International Journal of Engineering Science Invention*
*ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726*
*www.ijesi.org Volume 3 Issue 4 ǁ April 2014 ǁ PP.26-29*

# A Survey On Cyber Security Modeling Language To Enterprise System Architectures

Rasika Gawande
*(Computer Science And Engineering, P. R. Pote(Patil)College Of Engineering/ S.G.B Amravati University, India)*

***ABSTRACT :*** *The Cyber Security Modeling Language (CySeMoL) could be a tool for quantitative cyber security analyses of enterprise architectures and additionally a modeling language for enterprise-level system architectures coupled to a probabilistic logical thinking engine. If computer systems of an enterprise area unit modeled with CySeMoL, this inference engine will assess the likelihood that attacks on the systems can succeed. The idea used for the attack-probability calculations in CySeMoL could be a compilation of analysis results on variety of security domains and covers a spread of attacks and counter measures. A check indicates that the reasonableness and correctness of CySeMoL assessments compare with the reasonableness and correctness of the assessments of a security skilled.*

***KEYWORDS :*** *Computer security, expert systems, risk analysis, supervisory control and data acquisition (SCADA) systems.*

## I. INTRODUCTION

*Security problems associated with information technology (IT) still be a priority in today's society. The IT environments of the many enterprises area unit composed of an oversized variety of systems connected to make advanced system of systems. Security is additionally a posh downside that's troublesome to master. To totally estimate the protection of associate degree enterprise's system design, an oversized variety of problems should be thought of. Enterprise systems security managers should be ready to assess however vulnerabilities in one system influence vulnerabilities in alternative systems. additionally, security managers should be ready to assess however individual vulnerabilities influence the protection of the whole system of systems, given the protection solutions that area unit utilized in totally different locations within the design. Enterprise systems security managers generally have a basic understanding of their organization's design and systems and also the losses incurred if assets area unit compromised. However, the managers' understanding of however vulnerabilities depend upon one another within the system of systems and the way vulnerabilities are often exploited is usually hazy.*

## II. THE CYBER SECURITY MODELING LANGAUGE

The main objective of CySeMoL is to allow users to create models of their architectures and make calculations on the likelihood of different cyber attacks being successful. Since the model includes theory on how attributes in the object model depend on each other security expertise is not required from the user of CySeMoL. Users must only model their system architecture (e.g., services, operating systems, networks, and users) and specify their attributes (e.g., if encryption is used and if software is well patched) in order to make calculations possible.The classes in CySeMoL includes various IT components such as Operating System (e.g., Windows XP) and Firewall, processes such as Security Awareness Program, and Persons that are users. Each entity has a set of attributes that can be either attacks steps made against the entity or countermeasures associated to it. These attributes are related in various ways. For example, the passwords of password account can be social engineered – but the likelihood of this attack being successful depends on whether the person owning the password account is in a security awareness program. Each attribute in CySeMoL can have the value True or False and represents either the likelihood of an attack being successful or the likelihood of a countermeasure being functional.

## III. ATTACKER

In CySeMoL, an Attacker constitutes an individual who is determined to compromise assets of a depicted object model. Naturally, the characteristics of this attacker will influence what attacks that are possible, and how likely different activities are to succeed . In CySeMoL, it is assumed that the attacker is a professional penetration tester with access to publicly available tools and techniques. Consequently, the attack steps and estimates within CySeMoL need to be viewed in the light of this attacker profile.An Attacker can be connected to any class that has an attack step; connecting the attacker to an attack step within a class denotes the source attack vector.

This particular attack step always evaluates it to TRUE. The Attacker has one attribute - Time. This specifies how many workdays an attacker has to spend on each attack step for an object model. Computationally, the probability of each attack step in an object model being TRUE is evaluated with respect to the number of work days specified for any modeled attackers.

## IV. OPERATING SYSTEM

An Operating System (OS) is a collection of software that manages computer hardware resources and provides common services for computer programs. CySeMoL makes a clear distinction between software programs and OSs. Software programs are often tightly coupled to an operating system; for instance, Windows environments are tightly coupled to a service message block (SMB) service that, for example, enables file sharing and remote printing. However, in CySeMoL any optional software program that come with operating systems should be represented through the classes Application Client and Application Server, and only "mandatory" core functionality (e.g. the TCP/IP stack implementation) should be seen as a part of the Operating System class. Example Oss include Windows 8, Mac OS X . Any OS running inside a hypervisor such as VMWare VSphere 5 would also be considered an Operating System.An Operating System can be connected to ten different assets in thirteen different means (see Figure1 ). An Application Client or Application Server should be connected to an OS that enables its execution. Here, an Application Server can act as either a terminal to its core services (e.g., telnet, SSH, VNC or RDP) or expose only application-specific functionality (e.g., an HTTP or FTP server). An OS must be connected to a Software Product; this denotes what type of OS that is .System might have 400 computers that run the same type of configuration, e.g., Windows XP SP2. This enterprise would model a single Software Product (Windows XP SP2) and connect this to 400 Operating Systems.Connection to a Network denotes that the OS has an IP address on this network. Connection to Physical network means that an attacker has physical access to the machine running the OS. An Access Control Point depicts the means of logical access of the content of an OS. Data Store is kind of database located on the OS. An OS can be protected by one or more Intrusion Detection Systems (IDS Sensors) or Intrusion Prevention Systems (IPSs). Finally, an OS can be connected to a Network Vulnerability Scanner (e.g., Nessus or Qualys Guard), denoting either that the OS is analyzed through an authenticated scan, an unauthenticated scan, or not at all .

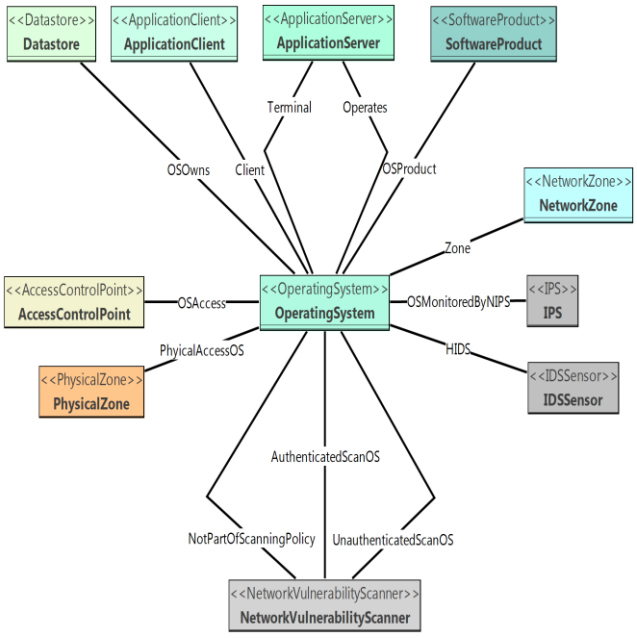


Figure 1: An overview of the connections for operating system

### 4.1. Defenses

4.1.1. Has All Security Patches

Has All Security measure denotes whether the OS has had all software security measure implemented . For instance, the Microsoft security up-date KB-2861561 address the vulnerabilities in Windows Kernel-Mode drivers that could allow remote code execution in various Windows OSs. If a security measure such as KB-2861561 exist, then if is not installed on a vulnerable OS, then Has All Security Measure should be FALSE.The default state of the defense is specified as follows: If an OS is connected to a Network Zone, that in turn it is connected to a Zone Management Process, then the default state of Has All Security Patches is dependent on the

state of Zone Management Process. If an OS, or a Network Zone connected to the OS, is connected to a Network Vulnerability Scanner (and the OS is depicted to be part of the scan), then the default state of the defense is dependent on the type of scan that is designated .

**4.1.2. Static ARP Tables**

Static ARP Tables involves if the OS has functioning static Address Resolution Protocol (ARP) tables. The ARP tables map logical IP addresses to physical addresses in the operating system. The default state of this defense is FALSE; it's not dependent on the existence of any other defense.

Table 1: Defences Affecting Likelihood Of Has All Security Patches.

| PM | ANS | UNS | Data |
|---|---|---|---|
| TRUE | TRUE | TRUE | bernoulli(0.79) |
| TRUE | TRUE | FALSE | bernoulli(0.79) |
| TRUE | FALSE | TRUE | bernoulli(0.79) |
| TRUE | FALSE | FALSE | bernoulli(0.79) |
| FALSE | TRUE | TRUE | bernoulli(0.637) |
| FALSE | TRUE | FALSE | bernoulli(0.637) |
| FALSE | FALSE | TRUE | bernoulli(0.302) |
| FALSE | FALSE | FALSE | 0 |

**4.1.3. Host Firewall**

A Host Firewall , is assumed to allow all knowledge Flows from/to the OS and any software used on it. It serves to block services that are unknown to the modeler (Operating System. find Unknown Service). The default state of this defense is TRUE; it's not hooked in to the existence of the other defense.

**4.1.4. Address Space Layout Randomization**

The purpose of Address space Layout randomization (ASLR) is to introduce randomly into memory addresses used by a given software module . This will make a class to exploit a techniques to fail with a quantifiable probability and also allow their detection since failing makes an attempt can possibly crash the attacked task. ASLR has for instance been available for Windows OSs since Windows Vista. The default state of this defense is TRUE as most modern OSs have it implemented; it is not dependent on the existence of the other defense.

**4.1.5. Non Executable Memory**

Non executable Memory could be a feature that if implemented and functional is intended to stop an application or service from executing code from a non executable memory region. If the OS has non-executable memory implemented and working there should so be a smaller chance of success for a certain variety of exploits (buffer overflow attacks). an example of this defense mechanism is data Execution prevention (DEP), that is offered for Microsoft OS from Windows XP SP2 and onward. The default state of this defense is TRUE as most modern OSs and hardware support it; it's not hooked in to the existence of the other defense.

**4.1.6. Anti Malware Solution**

Anti Malware solution, is software used to prevent, detect and report malicious package (i.e., malware) . several exploits involve injection of some variety of package code; associate opposed Malware resolution includes a likelihood to discover and forestall such code. If associate OS is connected to a Network Zone, that successively is connected to a Zone Management method, and Zone Management method. Managed By Anti-Malware resolution is TRUE, then the default state of anti Malware solution is TRUE; it's FALSE in alternative cases.

**4.1.7. USB auto Run Disabled**

USB auto Run Disabled involves whether or not the autorun practicality (that is enabled per default in most OSs) has been disabled. If associate OS is connected to a Network Zone, that successively is connected to a Zone Management method, then the default state of USB auto Run Disabled is TRUE if Zone Management method. USB Auto Run Disabled In Domain is TRUE; it is FALSE in alternative cases.

## V. CONCLUSION

CySeMoL is a modeling language coupled to an inference engine for analyzing the security of enterprise system architectures. For these attack paths, the inference engine estimates the probability that the attack can be accomplished by a professional penetration tester within one week using publicly available tools.CySeMoL has been implemented in an existing tool and validated on the component and system levels. On the component level, the theory specified in the dependences is drawn from empirical studies in domain security and domain experts. On the system level, a Turing test suggests that the reasonableness of assessments produced by CySeMoL compares with that of a security expert and that both CySeMoL and the experts are more reasonable than security novices. These results suggest that CySeMoL would be useful where no security expert is available.

## VI. ACKNOLEDGEMENT

No report is ever complete without the guidance of those experts who have already traded this past before and hence become master of it and as a result, our leader. So we would like to take this opportunity to thank all those individuals who have helped us in visualizing this report. I take this opportunity to express my deep sense of gratitude to my report guide **Prof. M. S. Burange** for providing timely assistance to my queries and guidance that he gave owing to his experience in this field for past many years. He had indeed been a lighthouse for me in this journey.

## REFERENCES

[1] T. Sommestad, M. Ekstedt, and P. Johnson, "A probabilistic relational model for security risk analysis," Comput. Security, vol. 29, no. 6, pp. 659–679, Mar. 2010.

[2] B. Taskar et al., "Probabilistic relational models," in Introduction to Statistical Relational Learning, L. Getoor and B. Taskar, Eds. Cambridge, MA: MIT Press, 2007, pp. 129–175.

[3] A. J. A. Wang, Information security models and metrics, in: Proceedings of the 43rd annual Southeast regional conference-Volume 2, ACM, 2005, pp. 178–184.

[4] CCRA, Common Criteria for Information Technology Security Evaluation, Available on http://www.commoncriteriaportal.org/, accessed June 24, 2013 (2012).

[5] C. Alberts, A. Dorofee, J. Stevens, C. Woody, Introduction to the octave approach, Pittsburgh, PA, Carnegie Mellon University.

[6] F. den Braber, I. Hogganvik, M. S. Lund, K. Stølen, F. Vraalsen, Model-based security analysis in seven steps-a guided tour to the coras method, BT Technology Journal 25 (1) (2007) 101–117.

[7] R. Breu, F. Innerhofer-Oberperfler, A. Yautsiukhin, Quantitative assessment of enterprise security system, in: Availability, Reliability and Security, 2008. ARES 08. Third International Conference on, IEEE, 2008, pp. 921–928.