ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering and Technology (ICIET'14)

On 21st & 22nd March Organized by

K.L.N. College of Engineering, Madurai, Tamil Nadu, India

Information Flow Analysis Based On Security Metrics

S.Sheela, T.Rajasundari

PG Scholar, Department of Computer Science and Engineering, Mepco Schlenk Engineering College, Sivakasi, India

Asst. Professor, Department of Computer Science and Engineering, Mepco Schlenk Engineering College, Sivakasi, India

***Abstract* --** The numbers of users sharing sensitive information are increasing day by day which is highly vulnerable to various attacks and may be exploited. Analyzing and securing the information flow is a great challenge faced by most of the user in an organization. Intrusion Detection Systems usually generates number of alert messages by the sensing devices, IDSs whenever malicious activities are detected. In this paper, security evaluation framework that handles low-level IDS alerts and system security measure selection mechanism is proposed based on this how crucial they are for the organization. Seclius framework includes three phases as: Alert generation phase, Consequence Tree construction phase and Dependency graph generation phase. In the alert generation, the security requirements are located in the administrator server. If any malicious activity is detected, the seclius framework going to generate an alert based on the security measures of all systems in an organization. Consequence Tree is manually defined for capture the critical assets and organizational security requirements. The Dependency graph provides system learning process and going to free the administrator work.

***Keywords* --** Intrusion Detection Systems, Consequence Tree, Dependency graph, Security Metric

I. INTRODUCTION

Information flow security is the critical part of network security. It refers to a broad set of policies, technologies, and controls deployed to protect data, applications, and the associated infrastructure of information flow. Information flow based system security metric called Seclius is used to overcome these limitations such as unsuccessful buffer flow, unsuccessful reaching goal state and partial damage of the system. Information flow is the transfer of information from one system to another system. Securing the information transferred among systems has been a challenge in the past years. Several methods to limit the information disclosure exist today, such as access control lists, firewalls, and cryptography. The main aim of the paper is to develop a security evaluation frame work system for malicious detection. The security evaluation framework system is going to reduce the human involvement by automatically learning the system characteristics with low performance [1]. Malicious disclosure of system framework uses the security evaluation techniques such as consequence tree that captures the subjective security requirements, IDS employs alert management to evaluate the actual results of attacker behaviors and dependency graph which signifies the system characteristics by collecting information flows between files and process within the system across the network [4].

Security Metric is defined as least number of weakness exploitations needed to get from that state to the goal state in which the intruder gains the privileges necessary to cause his or her final malicious consequence. A security metric implies a system of measurement that is based on scientific measures. A method of measurement used to determine the unit of a quantity could be a measuring instrument, a reference material, or a measuring system. The measurement of an information system for security involves the application of a method of measurement to one or more parts of the system that have an assessable security property in order to obtain a measured value.

The goal of security metric is to enable an organization to evaluate its security objectives.

IDSs frequently generate several hundreds of intrusion alarms that should be manually checked by the administrator. To provide situational alertness exposure systems usually employ (aware, precedence) mappings that are either built in the IDS without concern of the high level mission objectives of the communications or physically defined by administrators through a long task that requires deep system-level expertise. However, IDS alone are not sufficient to allow operators to understand the security state of their organization, because monitoring systems usually report all potentially malicious traffic without regard to the actual network configuration, vulnerabilities, and mission impact. Moreover, given large volumes of network traffic, IDS with even small error rates can overwhelm operators with false alarms. Even when true intrusions are detected, the actual mission threat is often unclear, and operators are unsure as to what actions they should take. In fact, to respond effectively to system compromises, security administrators need to attain efficient approximate summaries concerning the protection status of their mission-critical resources exactly and constantly, based on alerts that occur, in order to prioritize their response and recovery actions [8].

To address those different limitations, this project introduces an information flow-based system security metric called Seclius system. This system works by evaluating IDS alerts acknowledged in immediate to assess how much system and network assets protection has been affected by attackers. This evaluation is performed using two components: 1) a dependency graph, and 2) a consequence tree. These two components are designed to identify the context required around each IDS alert to accurately assess the security state of the different information assets.

II. System Overview

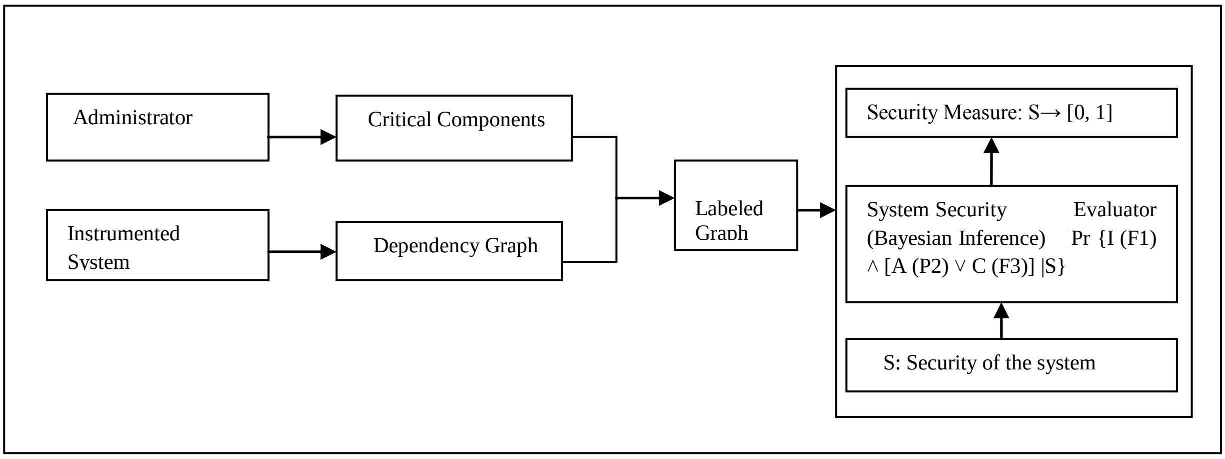

Administrator
Critical Components
Instrumented System
Dependency Graph
Labeled Graph
Security Measure: S→ [0, 1]
System Security Evaluator (Bayesian Inference) Pr {I (F1) ^ [A (P2) ˅ C (F3)] |S}
S: Security of the system

Fig. 1 Architecture of seclius framework

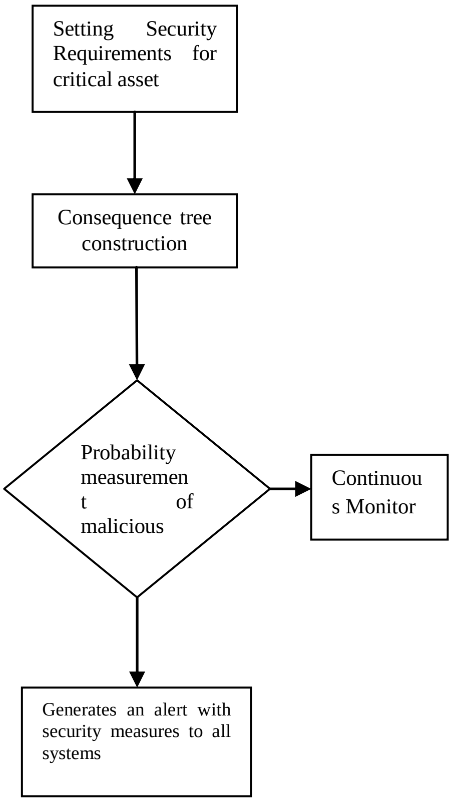
The architecture of the overall system consists of two components: Consequence tree and Dependency graph which is shown in fig 1.These two components are designed to identify the context required around each IDS aware to accurately assess the security state of the different assets. The goal of this architecture has to provide administrators with a framework to compute such measure with two major barriers. First, the important resources are system-specific and should be defined by administrators, a framework that requires too much individual involvement has constrained usability. Administrators can easily to map the important resources. In second method represent with higher framework and quantifies how many of the protection attributes the whole system.

Thus the system works by evaluating IDS alerts to acknowledge the real time system how much it is secured. The dependency graph captures the dependencies between these resources and all the files in the system. All inter resource dependencies and system-level information is captured by the dependency graph that is generated and analyzed mechanically. The small size of the manually constructed consequence trees and the automatic generation of the dependency graphs improve the scalability of system remarkably, as shown in the experiments.

Overall System processes the alerts from IDSs by using the dependency graph and probably determines whether the critical resources are compromised. If vulnerability operation in the customer web server was detected by IDS, Seclius system would inform not only the protection assess of the consequent web server, but also the protection access of the set of systems that depend on the web server. By observing the real-time IDS alerts and the learned inter-asset dependencies, system can precisely measure 1) the privileges gained by the attacker and which protection domains administrator was able to reach, and 2) how the integrity, confidentiality, or availability of the assets has been affected by the exploit directly or indirectly.

The security measurement evaluated by seclius represents the extent to which the system is insecure and consequently the calculated probability measures ranges between 0 and 1.The system security measure is represented as [0, 1] which implies {low, medium, high} levels. The security measure [0]-{low} indicates that the system is secure.

A. Alert Generation


Setting Security Requirements for critical asset
Consequence tree construction
Probability measurement of malicious
Continuous Monitor
Generates an alert with security measures to all systems

.

Fig. 2 Design of alert generation phase

In an alert generation phase, the security requirements are located in the server system. In this phase, the security requirements are set in the consequence tree, the IDS generates alert and gives security measure to all system otherwise it monitors the system continuously for any malicious behavior.

B. Consequence Tree

The goal of the Consequence Tree is to confine important resources and managerial protection requirements which are physically defined. The CT follows hierarchical structure facilitate administrators simply to plot the important resources and without conflicts according to the individual task of the organization. The CT defines consequences as the violations of the CIA criteria (Confidentiality, Integrity, and Availability) applied to important resources in the organization, such as specific files and process.

The Consequence Tree consists of two major types of logical nodes, namely AND and OR gates as shown in Fig 3. AND gate starts with the tree's root node which identifies the main high-level security requirements e.g., Organization is not secure, OR gate represents tree's leaf node as CIA components of critical asset. The rest of the tree recursively defines how different combinations of the more concrete and lower-level consequences can lead to the undesired status described by the tree's root node.

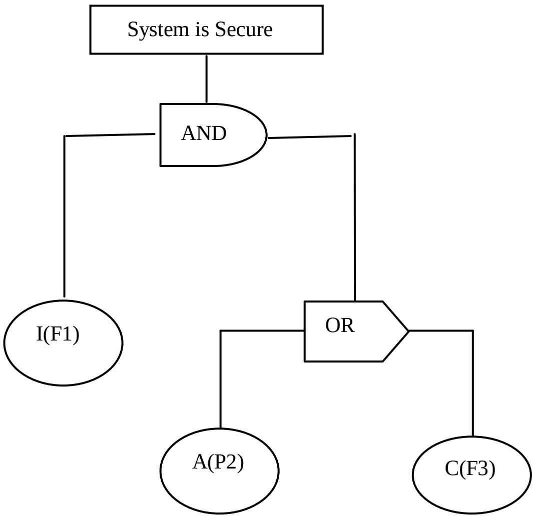


Fig. 3 Logical Tree Structure

The rest of the tree recursively defines how different combinations of the more concrete and lower-level consequences can lead to the undesired status described by the tree's root node. The recursive decomposition procedure stops once a node explicitly refers to a consequence regarding a security criterion of a system asset, e.g., "availability of the Apache server is compromised." These nodes are in fact the CT's leaf consequence nodes, each of which takes on a binary value indicating whether its corresponding consequence has happened (1) or not (0) .

The function notations refer to C, I, and A the CIA criteria of the assets. For instance, C(F2) and I(P6) denote confidentiality of file F2 and integrity of process P6, respectively [6]. The leaves values can be updated by IDS. The CT is derived as a Boolean expression, and the root node's value is consequently updated to indicate whether the organizational security is still being maintained.

C. Dependency Graph

The Dependency graph provides system learning process and going to free the administrator work. It receives the input from information flow and check the system status. The processing of dependency graph relies on various files and processes are represented as F1 and P1 [3].They are the important aspects in dependency tree formation. Administrator collects the data on inter-host dependencies between files and processes, the result would be stored in the dependency graph.

Each vertex in the dependency graph represents an object, namely a file, a process, or a socket, and the direct dependency between two objects is established by any type of information flow between them. More specifically, each DG vertex is modeled as a binary random variable.

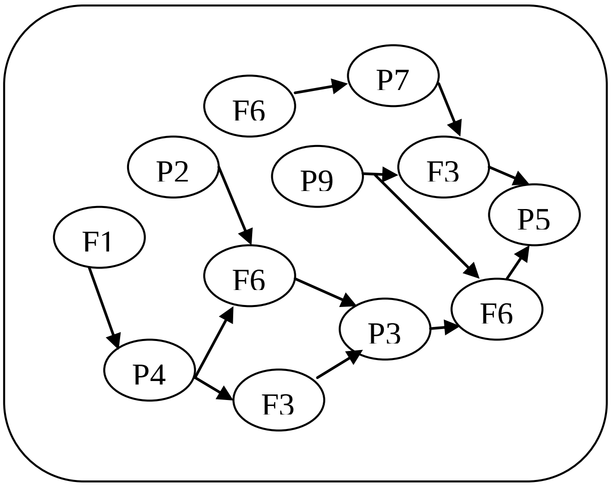


Fig. 4 Dependency Graph

III. Implementation And Results

A. Alert Generation

In the alert generation phase, the security requirements are located in the server system. These security requirements are placed in the critical assets of the organizational network. IDS detect attacks on critical assets and generate an alert. Based on the severity, seclius identifies the malicious activities, intruder detection and provides security measurements to related systems [2].

Module 1**:** Alert Generation

Purpose: User makes an invalid login attempt **Input :** User name and password
Output : Alert message

Pseudo code for alert generation

```
begin
    user enters user name and password for entering a system

    The IDS verifies the user credentials

    if
        the provided user is valid
        then the user can have access to the file server
    else
        IDS allows to enter a system and log activity of unauthorized also generates the alert message to the owner
        alert message indicating the attacker try to hacking the system  end
```

B. Consequence tree Construction

The consequence tree follows a hierarchical structure that facilitates the administrators easily to map the critical assets which is not conflicts according to the organization. Server side Functions are performed by Administration entry for organization in the Consequence Tree Construction:

Step1: Design an organizational Consequence tree, the administrator starts with the tree's root node, which identifies the main high-level security requirements as web server system

Step2: Leaf nodes of the CTs in seclius address security requirements (Confidentiality, Integrity, and Availability) of critical assets

Step3: Administrator gets the root nodes and its corresponding consequence list. The critical assets in the web server system and file server system are considered by the organization network. Server side functions for the organization are performed by administration entry in the Consequence Tree Construction.

TABLE I
Organizational Aware Element in Root to Leaf Node

Root Node	Leaf Node1	Leaf Node 2
www.google.com	www.gmail.com	www.annauni.edu.in
www.youtube.com	www.twitter.com	www.facebook.com

Web server is taken as the main critical assets www.google.com is taken as the main level requirement in the root node. Leaf nodes are the next level of requirements. www.gmail.com, www.tweitter.com is taken as the next level requirements.

TABLE II
List of Consequence Element from Root to Leaf

Root nodes	Consequence list	Consequence list	Consequence list
www.google.com	www.google.com	www.gmail.com	www.facebook.com
www.youtube.com	www.youtube.com	www.twitter.com	www.annauniv.edu.in

Administrator gets the root nodes and its corresponding consequence list is displayed in the format as shown in Table2**.** The root node is www.google.com and its consequence list contains nodes such as www.google.com, www.gmail.com and www.facebook.com.

C. Malicious Detection

The server identifies the user as an intruder and gives the status, illegal entry as shown in the fig 5.

User Gateway

Login:

Enter User Name: fthe

Enter Password : sftd

Incorrect User Name or Password

User Login Status:

Three times login Failed, This is an Illegal entry

Attacker Identified

Fig. 5 Malicious Detection

D. User Resource Access Control

Server side functions are performed by user resource access control.

- Identification and verification of a user by user name and password provides by user who is going to access the system.
- Identification and protection of system resources.
- Resource access control gives access to a computer system only to users who have the authorization to use a requested resource (such as a file, a printer queue, space to run a program, and so forth).
- Resource access control allows an enterprise to manage the security threat.

Information Flow based Control

Access right for users to file:

File Name	*Access Type*
File1	*read*
File2	*read*
File3	*write*
File4	*write*
File5	*read*

Enter the File Name : File1

Enter the Access Type : write

Wrong Access write

User is trying to hack the organization security

Fig. 6 Information Flow based control

The files with different access types are mentioned in fig 6. System process specifies the list of files in the critical assets set that needs to be monitored. Server can have some files with specific read and write permissions for each file. Each client have a user name and password, the client uses correct username and password but tries to write a read-only file, then it is an malicious behavior. In such situations an alert must be generated indicating that a security breach has happened [5].

E. Organization Security Measure

Security measures are detection of malicious behaviour and then calculation of security measure the security level in the file server system.

IV. Performance Evaluation

The analysis is done by comparing the proposed schemes with the existing schemes. The existing schemes may either provide intruder detection or access control. Although the typical behaviour of an intruder differs from the typical behaviour of an authorized user, there is an overlap in these behaviours [7]. Thus a loose interpretation of intruder behaviour, which will catch more intruders, will also lead to a number of “false positives” or authorized users identified as intruders. On the other hand, an attempt to limit false positives by a tight interpretation of intruder behaviour will lead to an increase in false negatives or intruders not identified as intruders.

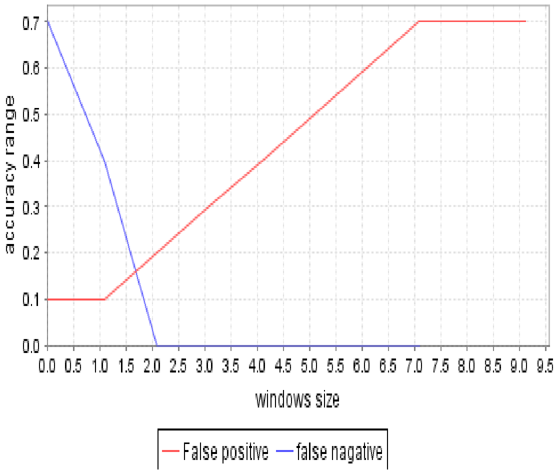


Fig. 7 Performance analysis of Accuracy of Dependency graph

This graph shows the accuracy rate to identify false positives and false negatives.

V. Related Work

In this section, the literature most related to this approach and how Seclius contributes to advance the state-of-the-art is discussed. There are dynamic methods, most of which are based on attack graph analysis [2], [3]. The main idea is to capture potential system vulnerabilities, and then extract all possible attack paths. The generated graph can be used to compute security metrics [4], assess the security strength of a network [5], to identify the most critical assets in the organization [6], or for security visualization [7].

They can also be used predictively to rank IDS alerts. The main issue with attack-graph based techniques is that they require important assumptions about attacker capabilities and vulnerabilities [8]. There have been several efforts to take into account unknown vulnerabilities during the system security analysis. N. Idika and B. Bhargava proposed Extending attack graph-based security metrics and aggregating their application.The attack graph is an abstraction that reveals the ways an attacker can leverage vulnerabilities in a network to violate a Security policy [2]. Xu Chen, Ming Zhang proposed file system processing. Here the description of file formation and forwarding process gives the assessment technique for security evaluation [3]. It minimizes the number of files created during a live analysis because they could overwrite evidence in unallocated space.

VI. Conclusion And Future Enhancement

In this paper, security evaluation framework Seclius that handles low-level IDS alerts and system security measure selection mechanism based on this how crucial they are for the organization. Intrusion Detection System detects attacks on critical assets and generates an alert and the consequence tree captures the security requirements in the organization network. It also provides the learning and detection of local intrusions. By processing the alert, the system gives information about intrusion behavior and details about affected system. Thus the proposed one of the solution dependency greatly reduces the involvement of the administrator in security aspects. This project illustrates the IDS approaches to measure the actual attack consequences of attackers and returns the security measure.

This paper can be further enhanced by analyzing the protection administrators with an absolute situational alertness result and performing proper countermeasures for the attacks. The detection accuracy can be improved by using enhanced and effective Information flow systems.

Acknowledgment

The authors wish to express their heartfelt thanks and gratitude to the Department of Computer Science and Engineering of Mepco Schlenk Engineering College, Sivakasi for providing good support and encouragement for this work. The authors also thank their principal and management for providing the necessary facilities to carry out this work.

References

[1] Saman A. Zonouz, Robin Berthier, Himanshu Khurana, William H. Sanders, and Tim Yardley, Seclius: An Information Flow-based, Consequence-centric Security Metric, *IEEE Transactions on Parallel and Distributed Systems (TPDS*)

[2] N. Idika and B. Bhargava. Extending attack graph-based security metrics and aggregating their application. Dependable and Secure Computing, *IEEE Transactions on, 9(1):75 –85, jan.-feb. 2012.*

[3] Brian Carrier. File System Forensic Analysis. *Addison-Wesley Professional, 2005.*

[4] Steven Noel and Sushil Jajodia. Optimal ids sensor placement and alert prioritization using attack graphs. *Journal of Network and Systems Management, 16(3):259–275, 2008*.

[5] Marcus J. Ranum, Kent Land field, Michael T. Stolarchuk, Mark Sienkiewicz, Andrew Lambeth, and Eric Wall. Implementing a generalized tool for network monitoring. *In Proceedings of the 11th Conference on Systems Administration (LISA '97), pages 1–8, Berkeley, CA, USA, 2004.USENIX Association*.

[6] Brian Wotring, Bruce Potter, Marcus Ranum, and Rainer Wichmann. Host Integrity Monitoring Using Osiris and Samhain. *Syngress Publishing, 2005.*

[7] I. Kotenko and M. Stepashkin. Attack graph based evaluation of network security. In Communications and Multimedia Security, *pages 216–227.Springer, 2006.*

[8] Steven Noel and Sushil Jajodia. Optimal ids sensor placement and alert prioritization using attack graphs. *J. Netw. Syst. Manage., 16:259–275, September 2008.*