IJICTR
International Journal of Information & Communication Technology Research

Volume 13- Number 3 – 2021 (24 -38)

Security Enhancement in Open-Source Healthcare Developer Network using Graph Theory

Mehdi Fasanghari*
Iran Telecommunication Research Center, Tehran, Iran
Corresponding Author:
fasanghari@itrc.ac.ir

Hamideh Sadat Cheraghchi
Iran Health Insurance Organization
Tehran, Iran
h.cheraghchi@ihio.gov.ir

Farzaneh Abazari
Iran Telecommunication Research Center, Tehran, Iran
f.abazari@itrc.ac.ir

Farhad Pouladi
ICT Faculty, Tehran, Iran
pouladi@ictfaculty.ac.ir

Received: 2 March 2021 - Accepted: 9 May 2021

Abstract—**Self-organizing software generation teams are increasing due to the more innovation and convivence these networks provide. This article focuses on the security evaluation of the open-source software development (OSSD) team. A newly generated formula for assessing the network security of a collaborative network is proposed and evaluated. For this purpose, we take advantage of graph theory criteria. Using this measure, if the developer's network is vulnerable in some parts, the guideline can be used for attracting/strengthening the ties in OSSD.**

Keywords: Graph Theory; Open-source developer network; Security.

Article type: Research Article

[DOI: 10.52547/ijict.13.3.24]

I. INTRODUCTION

With recent advancements in telecommunication and a pervasive digital networked environment worldwide, collaboration over the Internet among different developers to produce goods and services is very common. A virtual community known as an open-source software network/community brings together software developers from all over the world to cooperate on developing new software. For developers and end-users, this structure offers a variety of advantages as well as problems [1]. It is desired that security knowledge and practice be leveraged in these networks to avoid vulnerabilities. "Source Forge" is the most prominent example of these OSDD communities organized for technological and innovation purposes. Furthermore, "Openemr" is the most widely used open-source electronic health record and practise management system [2].

In OSDD, the source code is fully accessible by the user trying to use the software. Programming experts can edit programs based on what they need. Additionally, participating in these open source communities is unpaid

* Corresponding Author

employment outside of management hierarchies. [4]. Due to the volunteer nature of the work and the geographical spread of workers, trust has become a critical problem within the open-source community.

These collaborative networks forming virtual communities are subject to enormous research under the social network analysis area. This branch of research takes excessive advantages of the findings in graph theory in computer science [3].

Security aspects of collaborative OSSD networks are considered an essential issue by different communities. Especially when the software produced is intended to be used for commercial purposes. Despite the security community's emphasis on the importance of establishing secure open-source software (OSS), research on the vulnerability of these networks remains insufficient. On the other hand, new vulnerabilities found in OSS and the inherent role of the people developing and using those applications produce a different kinds of vulnerable behaviors. These issues demand more research in software security studies. Open-source software developer networks must be evaluated from a security point of view for generating software from new perspectives. It is desired that network components must maintain the desired security level in all of the security principles. If high security levels are established in all of the roles in the OSSD network, the probability of unauthorized access to information becomes less. In fact, understanding these structures from a security perspective has many benefits, such as vulnerability detection, monitoring, and prediction [1]. Furthermore, it gives an idea of human factors' relationship on security vulnerabilities of these networks.

Due to the critical nature of evaluating security in OSSD networks, we use a collaborative social network model to investigate the open-source software development network. In fact, the many components of an open-source project, including the publicly accessible code archive, programmers, testers, release management, and bug databases, are reviewed in terms of security requirements and attributed to them the security level.

This article makes two contributions:

- First, we evaluate the safety of a given open-source software developer's network by applying quality of security service and network design theory. For this purpose, we model the OSSD network as a weighted and directed graph, and ten sophisticated chosen social network criteria are proposed to evaluate the security aspect.
- Second, we recommend network administration to improve network security according to the assessment.

To the best of our knowledge, no previous work has used graph security measures for the purpose of devising security principles including integrity, confidentiality, avalablity, authentication and authorization. Using the social network security measures to find whole architectural vulnerability is a new feature of this paper.

The following sections are listed below. The second section details what other research has been done in this area. Section III describes three specific sub-sections in the framework's design (in Section IV), which rely on the background knowledge presented in the following three sub-sections: the security principles, protocols, and graph theory contained in the following section. Section V presents the steps taken for achieving a security framework. Section VI examines the findings through the lens of a case study. This paper concludes with Section VII.

II. Related Work

The use of open-source software in different fields are growing. Health care software are not an exception due to benefits offered by open-source solutions. The related literature of open source software solution in healthcare area specially health records is studied in [2]. However, achieving secure software is a very challenging process and is a major problem in software communities. This aggravates in healthcare systems specially those including health records [3]. This is due to critical information of patients and medical groups involved in their data. In fact, security of these software needs to be investigated in different areas. To be exact, software should be conceived, built, evaluated, and tested using methods, tools, and techniques that instil sufficient confidence in its reliability for its intended purpose [4]. Open-source software systems are susceptible to attack in a variety of ways and serval works are proposed in this domain [5-8]. In an ideal world, security should be kept throughout the Software Development Life Cycle (SDLC), including requirement creation, architecture and design, programming, debugging, certification, and support. This results in Secure SDLC (SSDLC), which is equivalent to [9]:

- Security Requirement
- Security Architecture and Design
- Secure Coding
- Security Verification and Testing
- Release and Operation

The security vulnerabilities in open-source software (OSS) has grown by 371 percent during 2014, according to the Open-Source Software Survey.

Structured open-source projects have a central repository of code versions with changes made to them. Although the whole public should have access to information, access to writing should be confined to a select group of people. As a result, one of the most crucial aspects of open-source software security is developer identification and authentication. Two key security problems arise when utilizing open-source software: code correctness and availability. Due to the project's testers and defect reporting mechanism, the inclusion of a Trojan in open-source software code is less likely. Before placing a code on the code warehouse, a series of tests should include penetration test, code monitoring, and statistical tests.

In this section, different methods for network security assessment and network security in OSS are presented. The model to review the related work is based on OWASP Software Assurance Maturity Model (SAMM) [10]. In this model core business function of software development is presented and then security practice and works related to each work is presented.

[DOI: 10.52547/ijict.13.3.24]

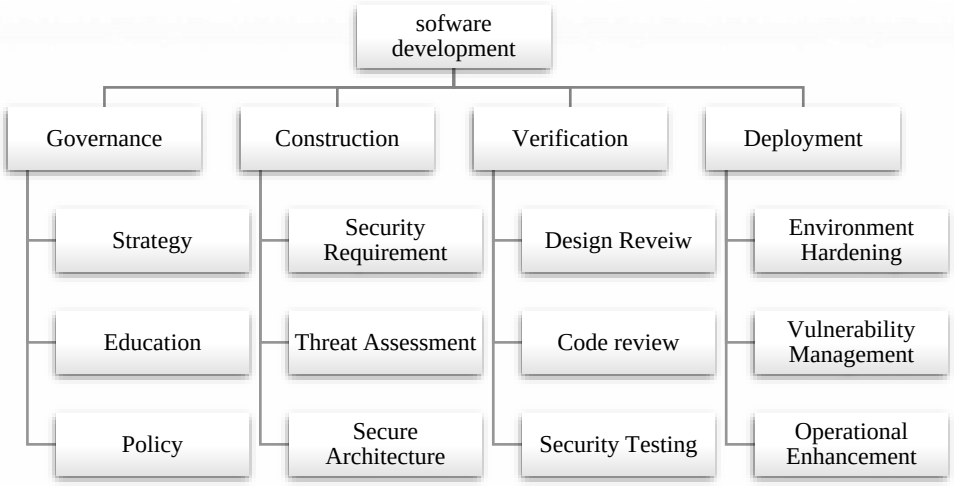


Fig 1. Software Assurance Maturity Model: layer two represent business function and layer three security proccesses [10].

Certain studies in open-source software concentrate on the underlying network structure that forms developer networks.

A network of individuals engaged in the production of open-source software establishes a social network that determines project assignment [11]. Therefore, these networks can have the characteristics of real networks. The rapid growth of open-source software has made it very important to examine community formation in the developer network. The success of an open-source project can depend on its social structure [12].

The papers discussed the security of open-source software using the SAAM approach. [13, 14] can be categorized in four important category. Studies related to "Verification" is the most cited category. This is due to lack of formal methods in OSS. This amounts to almost half of the studies in this area. Further, if formal methods are used, vulnerabilities in design stage will be revealed before the code review stage.

Use of formal method has been previously used as a security practice. A series of formal methods are performed to analysis security. In [15] access rules are defined formally. When the rules are being established, access to the desired resource are secured.

TABLE I. OSS SECURITY DOMAINS STUDIES [16]

Category	Subcategory
Verification	Design Review [17], Code Review [18], Security Testing [19]
Construction	Threat Assessment [20], Security Requirement [21], Secure Architecture[22]
Deployment	Vulnerable Management [23], Operational Enhancement[24]
Governance	Policy [25], Strategy and metrics [26]

On the second rank is the studies related to securing OSS at "Construction" stage. In this category finding the secure architecture is the focus of research. Studies of this topic include how to design a secure system, security requirement of OSS and tools used in these systems. Threat assessment articles are also in this category [20].

Some researches in "Construction" category use graph theory to detect critical node to cease publishing worms and viruses in general networks. Simulation of worm propagation in large networks and network protection against virus has been done by vertex cover algorithm in [27]. That paper shows topological routing affects worm propagation. The idea is finding a graph with minimum overlapping node which the nodes are routing server and the edges are relation between them. In [28] a tool for finding vulnerable points in network by means of graph theory is presented. The tool is applicable in those networks which users share files. By modeling relation with graphs, points with high overlap degree with graph clusters are considered as a vulnerable point. One of the important issues which is proposed in network is network coding. Network coding is applied to increase the information flow in network. Coding theory is based on graph theory. Different methods are presented for coding security in [29] and a method is mentioned to deal with terrorist attacks by detecting critical nodes in the graph in [30]. In all works done for network security there is a factor to measure importance of graph nodes.

The fuzzy rough set is used in [31] to assess network security. The FCM was used to assess network security by Mingji, Z. [32]. Donnet, B., et al. use triangles inequality to evaluate network security. If an eavesdropper is placed between the three sides a, b, and c, d (a, b) + d (b, c) ≥ d (a, c) does not hold true. Ming-zhong, M. [34] uses a combination of neural networks and graph theory to analyse network security.

The last two category relates to "Deployment" and "Governance" areas where fewer number of papers exists. This is primary due to the nature of OSS networks which lack any central administration and governance. However, the operation of the network can be enhanced if management and monitoring is considered.

Policy management in other networks is common and even intelligent methods are deployed in this domain. Some articles use intelligent methods as a solution. Using knowledge management and normal behaviour description, a multi-agent system is utilised to monitor network security and detect assaults [35].

Each function associated with the creation of software applications will require a specific level of security.

[DOI: 10.52547/ijict.13.3.24]

Generally, security evaluations made in graphs use clustering criteria to determine the importance of components, which alone is not sufficient.

The work done in this paper falls in both “Construction” and “Governance” category. Different measures are used to assess the security of OSS network based on different roles and component of the network. According to this measure restructuring is applied to secure the network.

III. BACKGROUND

A. Security Principles

Data confidentiality, integrity, and availability are the cornerstones of information security, and software security is rapidly adopting these aspects. Authentication and permission are also critical components of information security. The definitions obtained for each are listed below:

1) *Confidentiality*: Measures taken to prevent the disclosure of information and unauthorized access to information. Encryption preserves confidentiality.
2) *Integrity*: means no unrecognizable change of information.
3) *Availability*: This is about availability of information when needed.
4) *Authentication*: The parties to a relationship must verify the identity of the other party.
5) *Authorization*:. People's access to resources must be controlled. This requirement is met using access control.

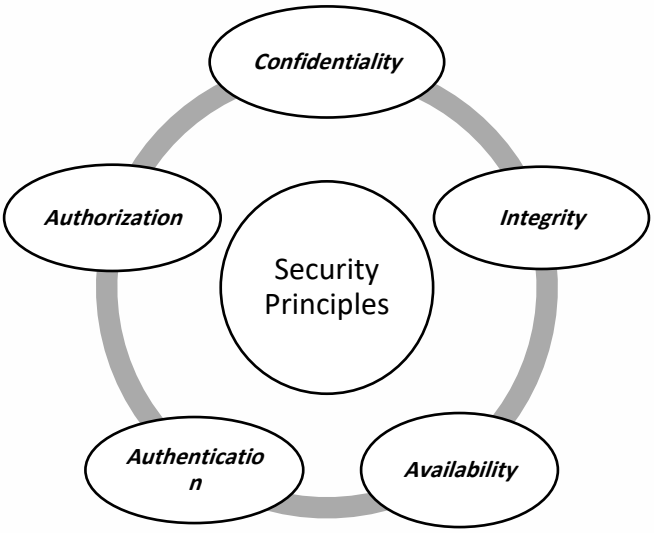


Fig 2. Fundemntal dimensions of security[36]

B.Security Protocols in TCP/IP Architecture

Each layer in the TCP / IP architecture has security mechanisms and protocols dedicated to itself. Following are some of the common security features associated with each of the TCP / IP protocol layers [37] (as illustrated in Fig. 3).

Link layer: Equivalent to the first and second layers of the OSI reference model [38]. Some security feature in this layer include:

- Packet Filtering The router's ACL (which is derived from the Access Control List) is an example of a packet filter in operation.
- NAT (short for Network Address Translation) is an address translation method. To safeguard users' security, the aforementioned technology conceals their internal IP address from external networks.
- CHAP (short for Challenge Handshake Authentication Mechanism) is a "verification" protocol that is used in lieu of unencrypted username and password submissions.

Network layer: As in the OSI reference model, equivalent to the third layer [39]

- PPTP (Point to Point Tunneling Protocol) is used by Microsoft and 3com to encapsulate data and is be employed by many companies.
- L2TP, which is based on the PPTP and L2F protocols, has been implemented for security.
- IPsec is used to encrypt IP packets and protect networks from assault.

Transport layer: Equivalent to the OSI reference model's fourth and fifth layers [40]:

- SSL (also known as the Secure Sockets Layer) is a protocol that is used to provide assurance to people that they are able to exchange information in a secure manner (such as the Internet).
- TLS (short for Transport Layer Protection) is a comparable technology to SSL that employs a layered approach to data security. TLS is subdivided into various more protocols.

Application layer: Several of the fifth layer's functions are equivalent to those of the sixth and seventh layers in the OSI reference model [41]:

- RADIUS (Remote Authentication Dial-In User Service) is a widely used protocol s for authenticating dialup users. .
- TACACS (Terminal Access Controller Access Control System) is an archival "authentication" protocol that was used on Unix-based networking networks to enable a distant server to input a user-supplied password.
- MIT implements Kerberos as a robust authentication scheme.. Kerberos provides the essential encryption, data integrity, and confidentiality capabilities.
- S-MIME (Secure / Multipurpose Internet Mail Extensions) ensures electronic security through the use of encryption and digital signatures.

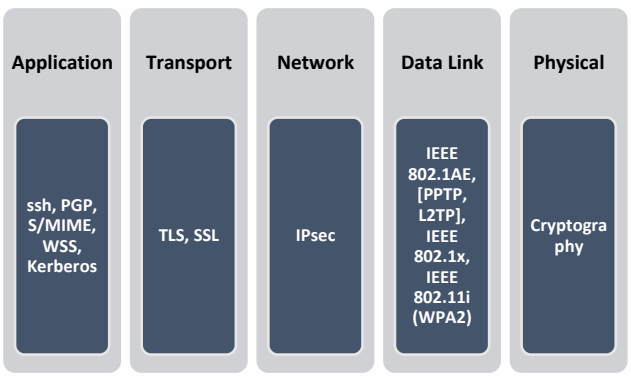


Fig 3. Security protocols for OSI layers [41]

[DOI: 10.52547/ijict.13.3.24]

C.Social Network Models

As previously stated, the network of open-source software developers is regarded a self-organizing social network.. This section provides an insight of these networks and how graph theory applies to them[۴۲] .

A social network is a subset of an information network, with nodes denoting individuals or entities and edges denoting their relationships. A graph can be used to depict the structure of a social network. A static and unweighted graph G is composed of a set of nodes V and a set of edges E: G = (V, E). The letters N and E signify the sizes of V and E. Different types of networks are as follows:

Unweighted/Weighted Social Network: In a weighted graph, relationships between nodes have a magnitude, which is significant for the relationship under examination. An unweighted graph may be used when a magnitude relationship does not exist or is trivial. Let e_{ij} be the edge between nodes i and j in a weighted graph G. The 'neighboring nodes' or 'incidental nodes' of edge e_{ij} will be referred to these two nodes. Assume that w_{ij} is the weight on the edge e_{ij}. The weight of w_i, is defined as the sum of the weights for its incident edges [43].

- **Undirected/ Directed Social Network**: A directed graph is a special kind of graph that has directional edges (sometimes shortened to digraph). Every edge in a directed graph represents a unidirectional relationship: an arrow from one node to another, but never the reverse. All edges in an undirected graph are two-way.. [43].

As depicted in Fig. 4, there are 4 primary categories of graphs [44]:

- Undirected & Unweighted: Relationships are bidirectional and do not have a magnitude associated with them.
- Undirected & Weighted: Relationships have a measurable magnitude and are bidirectional in their nature.
- Directed & Unweighted: Relationships have no significance and are only one-way in nature.
- Directed & Weighted: Relationships have a certain magnitude and are one-directional.

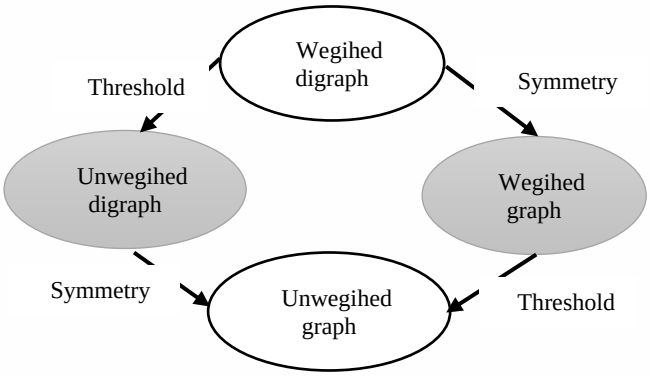
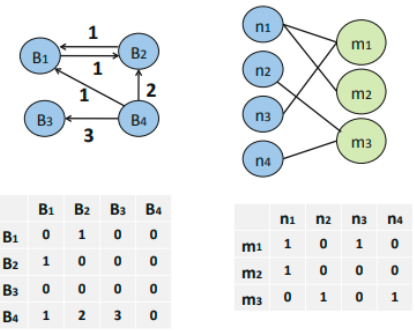


Fig 4. Four Main Type of Social Networks [44]

Additionally, we can visualize a graph or express it numerically using an adjacency matrix A, in which the matrix contains integers to denote the presence of edges, and nodes are arranged in rows and columns. Elements in unweighted graphs are either 0 or 1, while weighted graphs store their values in the adjacency matrix. Fig. 5 illustrates a graph and its adjacency matrices. OSDD are a type of directed and weighted graph.



	B1	B2	B3	B4
B1	0	1	0	0
B2	1	0	0	0
B3	0	0	0	0
B4	1	2	3	0

	n1	n2	n3	n4
m1	1	0	1	0
m2	1	0	0	0
m3	0	1	0	1

Fig 5. Illustrations of many types of graphs. On the left is a unipartite weighted graph with its adjacency matrix. On the right is a bipartite undirected graph and associated matrices.

D.Social Network Graph Measures

In this section, we review the appropriate criteria for evaluating the importance of edges and nodes [44]. These criteria assign a value to nodes based on their position in the graph.

Degree: k_i is the degree of a vertex i, and presents the number of edges connected to this vertex:

$$k_i = \sum_j a_{ij}. \quad (eq.1)$$

For all vertices in a network, K is defined as the average of k_i for all vertices when the average degree is computed for the network:

$$\langle k \rangle = \frac{1}{N}\sum_i k_i = \frac{1}{N}\sum_{ij} a_{ij}. \quad (eq.2)$$

For directed graphs, the out-degree and in-degree of each node are computed.

$$k_i^{out} = \sum_j a_{ij} \quad , \qquad k_i^{in} = \sum_j a_{ji}, \ (eq.3)$$

and the average degree in and out degree for graph is equal:

$$\langle k^{out} \rangle = \langle k^{in} \rangle = \frac{1}{N}\sum_{ij} a_{ij}. \quad (eq.4)$$

For weighted graph, degree of each node can be computed using above formula but using the definition of the strength of each node i, denoted as s_i:

$$s_i^{out} = \sum_j w_{ij} \quad , \qquad s_i^{in} = \sum_j w_{ji}. \ (eq.5)$$

Nodes with more edges are known as having a higher node degree. As a result, the cost of implementing these nodes in the network increases. However, if we want to have a security view of a node, the node with a greater degree requires more security, as it is coupled with more nodes, increasing the likelihood of infection in the network.

- **Degree Centrality**: Degree centrality is a metric that expresses a node's degree of connectivity. Indeed, it demonstrates how easily a node can communicate with other nodes. The amount of

[DOI: 10.52547/ijict.13.3.24]

edges that a node has is used to estimate its relative centrality[۴۵] :

$$Degree\ Centrality(v_i) = \frac{\deg(v_i)}{N-1}. \quad (eq.6)$$

- **Clustering Coefficient**: A node's local clustering coefficient indicates how close its neighbors are to forming a clique (complete network), and is written as follows [44]:

$$C_i = \frac{1}{k_i(k_i-1)} \sum_{j,k} A_{ij} A_{jk} A_{ki} \quad (eq.7),$$

Where $k_i = \sum_j A_{ij}$.

- **Average Distance & Global Efficiency**: The length of a path is equal to the number of edges between vertices i and j. The shortest path between these two nodes is the one with the shortest length, denoted by d_{ij}. This definition is for unweighted graph and for weighted graph the weight should be included. The average distance of graph can be computed using this measure as follows [44]:

$$l = \frac{1}{N(N-1)} \sum_{i \neq j} d_{ij}. \quad (eq.8)$$

Global Efficiency: When the number of unconnected pairs is large, the average distance metric returns a tiny value. As a result, another metric called global efficiency is created to quantify the network's efficiency in transmitting information between nodes that is proportional to their distance. Indeed, overall efficiency or performance is a metric used to describe the capacity of a network's traffic. Nodes that are not connected to the network have no effect on the graph's performance in heterogeneous graphs, as their distance from the other graph nodes is limitless. The closer E is near 1, the more efficient the network connectivity. This is referred to as the global efficiency metric, which is specified in the following [44]:

$$E_{\text{Global}} = \frac{1}{N(N-1)} \sum_{i \neq j} \frac{1}{d_{ij}}. \quad (eq.9)$$

Local Efficiency: This criterion is based on neighboring subnets of edges and is connected to the idea of clustering coefficient. If the subgraph Gi is used to represent the neighbors of node I and E(Gi) is used to represent the graph's overall performance, the local efficiency value is obtained using Equations 10 and 11 [44].

$$E_{loc} = \frac{1}{N} \sum_i E(G_i), \quad (eq.10)$$

$$E(G_i) = \frac{1}{N_i(N_i-1)} \sum_{k,j \in G_i} \frac{1}{d_{kj}}. \quad (eq.11)$$

Vulnerability: The section of the graph (node or edge) that has the greatest effect on boosting the graph's efficiency is more significant in the graph and has a higher permeability. Efficiency is a metric used to quantify the capacity of a network's traffic. Equations 12 and 13 yield the vulnerability value of a graph for node or edge I [44]:

$$V_i = \frac{E - E_i}{E}, \quad (eq.12)$$

$$E_i = \frac{1}{N(N-1)} \sum_{i,j,i \neq j} \frac{1}{d_{ij}}. \quad (eq.13)$$

Where E_i is Global Efficiency of the graph after deleting node or edge i.

If the network is attacked and the high-permeability node is removed from the network, the network efficiency will be greatly reduced. Therefore, in graph reinforcement, nodes with higher permeability are resisted so that in case of attack or failure, the efficiency of the graph does not decrease much.

Closeness Centrality: The term "closeness centrality" describes the degree to which a node is connected to all other nodes in the network. It is determined as the average of the network's shortest paths. Closeness centrality has the advantage of indicating that nodes are more central if they are closer to the majority of the nodes in the graph. This value is calculated for each vertex and graph using the following formula[۴۵] :

$$Closeness\ Centrality(v_i) = \frac{N-1}{\sum_j d_{ij}}, \quad (eq.14)$$

$$Closeness\ Centrality(G) = \frac{\sum Clossness(v_i)}{N}. \quad (eq.15)$$

The problem of disconnected graphs is solved by a variant of centrality called harmonic centrality.

- **Eigenvalue centrality or prestige score**: is a metric for a node's influence in a network. Each node in the network is awarded a relative score based on the concept that connecting to high-scoring nodes influence more than connecting to low-scoring nodes. A high eigenvector score shows that a node is related to a significant number of other high-scoring nodes. When eigenvalues are generalized, two attributes emerge: Authority and Hubness. The term "authority" refers to the amount of knowledge, information, and so forth that a node on a particular topic possesses. Hubness informs about how well a node 'knows' where to find information on a given topic. The best hubs direct you to the best authorities. It is a component of the HITS algorithm. Moreover, the term "prestige" is used to refer to directed networks. In this case, we can distinguish two distinct types of prestige: one for outgoing arcs (influence measures), and another for incoming arcs (measures of support). For measuring prestige, the same formula as for calculating relative degree centrality for directed graphs can be used. Prestige increases when an actor becomes the subject of additional ties, but not always when the actor initiates the ties [45]:

$$Prestige(v_i) = \frac{k_i^{in}(v_i)}{N-1}. \quad (eq.16)$$

where k_i denotes the number of edges connecting vertex I in-degree and N is used to indicate the total number of nodes.

- **Betweenness Centrality**: The concept of betweenness centrality states that a node is central if it is connected to multiple other nodes via the shortest paths. Each node has a betweenness value equal to the number of pathways that cross through it [45]:

$$B\ Centrality\ (G) = \sum \frac{\#\ d_{kj}\ through\ i}{\#\ d_{kj}}, (eq.17)$$

where d_{kj} denotes the shortest path between nodes k and j. Betweenness can be normalized by dividing by the number of vertex pairs not comprising i which is (n-1) (n-2) for graphs that are directed and (n-1) (n-2)/2 for

[DOI: 10.52547/ijict.13.3.24]

graphs that are not directed. It counts the number of shortest paths that pass through each node..

- **Eccentricity**: One of the needs of social network users is to receive a prompt response to a request. If one of the nodes on the network sends a request, the request is routed to the neighboring nodes based on the destination. If the target node is a neighbor of the origin node, The request takes a shorter route than the standard route, which results in the response arriving at the origin faster. This criterion can be modeled by the graph's eccentricity criterion. Equation 3 is used to get the value of this relationship:

$$Ecc(v_i) = max_{\forall j \in V}(d_{ij}), \quad (eq.18)$$

$$Ecc(G) = \sum_i \sum_j max(d_{ij}). \quad (eq.19)$$

- **Tie strength**: The strength of the connection between two nodes is determined by the extent to which their neighbors overlap. When the tie strength between two nodes is high, it indicates that they share many neighbor [۴۶]:

$$S(A,B) = \frac{|n_A \cap n_B|}{|n_A \cup n_B|}. \quad (eq.20)$$

- **Bridge**: Another criterion for evaluating a graph is the number of edges that are bridges. A bridge edge is an edge that divides a graph into two sections when it is removed. If one of these bridges is attacked, the network is rendered inoperable and incapable of serving its users. Security-wise, the presence of bridge edges in a network renders it vulnerable. As a result, bridge edges should employ more stringent security measures [47].

IV. Proposed OSDD Security Foundation

This Section discusses the many components required to create our OSSD security framework.

The security evaluation OSS network was done using two perspectives. In the first view, graph criteria are used and in the second view, the level of security protocols implemented in graph components is used.

The model's result is a score for each security principle in the network. Finally, the network administrator can comprehend the network's security flaws and take action to address them.

This section includes the following parts: First, we derive our security criteria to determine OSSD network security from different perspective and make it more secure. Then, in part B, we propose different level security for nodes and edge according to their importance. In part C, we link the proposed security level and security principles. In part D, we draw upon all previous parts, i.e., a graph security measure for each security principle is designed and is related to protocol security level to determine what practice should be used to make to network more secure based on the vulnerability observed for each security principles.

A. Desiging Graph Security Critiera

Those individuals who are actively involved in the production of open-source software are represented by nodes in the corresponding network. Individuals' edges represent their data interchange, which is described as directional and weighted. As previously said, this graph is modeled after a social network. We determine the significance of each edge and node by assigning values to the appropriate criteria. Indeed, we take the following two steps. [20], [21]:

Step 1: Determine the most acceptable criterion for determining the significance of each node and edge.

Step 2: Define the level of security for edges and nodes.

For each security service, we design a graph security measure (GSM_i) as follows.

- *Confidentiality Measure*: In OSSD users should be able to keep some data confidential and share the necessary information. To share files, the repository is used as one of the graph nodes that users who need to share information can use to share information. As centrality shows easily a node can be reached by other node, we take degree centrality as a measure of confidentiality. Nodes must be secured in such a way that users' confidential information is protected. These nodes have higher concentration criteria. Therefore, they must have high security:

$$GSM_{confidentiality} = 1 - Deree\ Centrality. \ (eq.21)$$

- *Integrity Measure*: unintended changes should be avoided to have high integrity. When graph's betweenness is high it is more probable to bear less integrity and split during the attacks. Any node with a greater value of this value must adhere to stronger security procedures, as infection spreads more quickly along the graph if that node becomes infected. So, we use negative of this measure. Further as high authority nodes in graph have more information about the other nodes, integrity of the graph with high authority nodes will be higher:

$$GSM_{integrity} = Avg[(1 - \text{Betweenness}) + \text{Authority})]. \quad (eq.22)$$

- *Availability Measure*: An important need of users is to minimize the down time of the main components. To achieve this goal, a number of organizations use backup servers. Sometimes one of the communication edges may be out of reach due to overload capacity. If the nodes can find an alternative route to access the destination node, the network efficiency is high. Further, to get a quick response after making a request is very important. On the network, the request goes to the neighboring nodes if it originates with a node and is sent to a target location. Requests to other neighbors in the origin node's neighborhood arrive faster, shortening the time required to arrive at the origin. So, we use distance related measures for this security principle which contribute to the following equation:

$$GSM_{availabiltiy} = Avg(Closeness + Eccentricity). \quad (eq.23)$$

- *Authentication Measure*: authentication process is about the verification of nodes. When a node is validated by more nodes, it is supposed to have higher valid security. This attribute is manifested in Hubness and clustering coefficient as they are about having access to more neighbors and cliques in their neighbors:

$$GSM_{authentication} = Avg(Hubness + ClusteringCoefficient). \quad (eq.24)$$

- *Authorization Measure*: From security perspective, if one of the network nodes has a large number of neighboring nodes from which it receives data, that

[DOI: 10.52547/ijict.13.3.24]

node needs higher security. In the graph, this criterion is inversely correlated with prestige and hubness measure. Since the hubness gives more information about the neighbor nodes, we use this criterion for authorization:

$$GSM_{authorization} = 1 - hubness. (eq. 25)$$

In the second step, we use the measure selected criteria to measure the level of network security and obtain a general criterion for graph security. Some criteria in the graph should be low to increase the security of the graph and some should be higher. For example, the criterion of vulnerability in graph nodes should be low for the graph to be more secure. Therefore, for some criteria, the phrase (1-criterion) has been used to have an adverse effect on the overall security assessment of the graph:

$$Graph\ Security = \sum_{i=1}^{5} GSM_i. (eq. 26)$$

B.Assigning Security Levels For Security Measures

This part aims to attribute a level of security to the graph components given the importance of each component in the social network structure. Users may have expectations of the security services they get, such as requirements for functionality and assurance.

It is more graceful for the underlying system to adapt to changes in resource availability during task execution when users or network tasks are presented with varying levels of security services and requirements. This ensures that the requested or required levels of service are maintained across all of its dimensions.

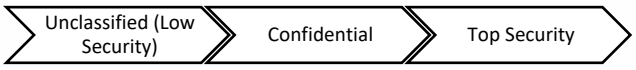


Fig 6. Security levels of OSSD.

As a result, specific structures in the OSSD do not require security, for example, the code that is accessible to the majority of parties. Some components in OSSD have more security concerns and are classified as confidential, and some parts have top security. The purpose of applying these levels to nodes and their connections is to prevent information from being leaked as much as possible. This is done by determining the mechanisms for implementing security levels for network nodes and edges using appropriate security services.

Security Level 3 Implementation

For implementing security level 3, "sharing information as needed" is implemented and multiple layers of security on hardware, software, and memory platforms are required. To implement these policies, we need to implement a set of protocols and infrastructures in network components, including nodes and links when a user can access content and applications that previously has been authenticated and allowed to access them. Abnormal activities are prevented by using user behavior analysis tools. The user is blocked to prevent the future damages. The security services required for Top Secret levels are as follows [22-24]:

- Use of physical protections,
- Access control for IT systems,
- Identification and authentication using technologies:
  - Secure passwords
  - Token Password authentication plan
  - PPP authentication protocol
  - TACACS+, RADIUS, Kerberos protocols
  - Smart cards
  - Biometric tools
- Procedure for determining which permissions each user has (to ensure authenticity, use digital signatures),
- Use encryption to protect information against unauthorized access,
- IT management tools for maintenance and management of IT equipment,
- Firewall and VPN,
- Reporting and auditing to investigate events and detect unauthorized activities.

Security Level 2 Implementation

For security level 2, users should be able to keep some data confidential and share the necessary information. To implement, a repository is placed as one of the graph nodes that users who need to share information can use to share information. Nodes must be secured in such a way that users' confidential information is protected.

C. Linking the Security protocols and Security measures

This section introduces a set of evaluation criteria for each of the security principles stated in Section III, part A. They are confidentiality, integrity, availability, authentication and authorization. Given the network infrastructure examined in this paper modeled as a social collaborative network and five security principles/service to be covered in this model, a taxonomy for assessing the level of security of graph components is introduced by the idea in [48]. In this model, each security service in the graph should be implemented in two area of nodes and edge denoted as:

- IN (Internal nodes),
- NC (Network connections).

Further, the security mechanism and protocols to secure the network is detailed in Security model based on the security service needed and related protocols.

As already discussed, the security levels created in the implementation of security protocols are divided into three levels. At each of these levels, a set of protocols is considered. At security level 1, nodes and network connections did not use the security protocols to establish security principles. At security level 2, only internal nodes use security services to maintain security principles, and at security level 3, in addition to internal nodes, network communications are secure. In short, the following rules applies:

- Security level 1: no usage of protocols,
- Security level 2: usage of security protocols for internal nodes (IN),
- Security level 3: usage of security protocols for both nodes and edges (IN & NC).

TABLE II. SECURITY MODEL BASED ON THE SECURITY SERVICE NEEDED AND RELATED PROTOCOLS

[DOI: 10.52547/ijict.13.3.24]

Security Service	Service Area	Example Security Mechanism	Protocols and Services
CONFIDENTIALITY	IN	OS Access control, Cryptographic credential	SSL/TLS IPsec
	NC	Encryption using 40-bit DES and 128-bit Blowfish, communication over a Virtual Private Network (VPN) comprised of packets that have been encapsulated, and an Active Network Node that analyzes traffic and responds to specified triggering circumstances by injecting dummy packets.	
INTEGRITY	IN	Access controls on the operating system, cryptographic credentials	The X.509 Standard, IPsec
	NC	Numbers corresponding to the integrity sequence , and digital signatures, and Chaining Cryptoghraphic	
AUTHENTICATION	IN	Internode authentication via digital signatures is supported in an active network.	The TACACS+ Protocol The RADIUS Protocol SSL/TLS, CHAP The X.509 Standard
	NC	Standard; reliance on a reputable certificate authority Authentication of the source of data like digital signatures, mechanism for OS identification and authentication, and IP address	
AVAILABILITY	IN	Priority-based application traffic scheduling, Bandwidth is reserved on network nodes that are active for network administration traffic.	
	NC	Scheduler based on the FIFO principle with preemptive interruptions, Protocol for bandwidth reservation	
AUTHORIZATION	IN	Access Control Group/Role based Approach,	
	NC	-	

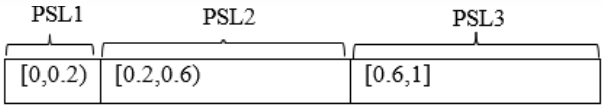


Fig 7. Values for protocol security levels.

The weight of security principles varies depending on the network on which the evaluation takes place and the level of security of the information exchanged. For example, it can be said that due to the open-source nature of the project, there is no need to maintain confidentiality in all communications. In this case, the weight given to privacy and authorization on network connections must be less. These weights can be obtained from project managers insights.

To incorporate these levels into the proposed framework, an interval value is suggested for each security level. Based on the security protocols used in the network and the metrics for which the protocols are most effective, experts assign a numeric value to the network's protocol security level in the range of zero to one, as illustrated in Fig. 7.

TABLE III. PROPOSED SECURITY MODEL FOR OSSD

Weight (W)	Security Service (SS)	Graph Service Measure (GSM) (Value: [0,1])	Protocol Level Security (PSL) (Value: [0,1]) Expert opinion *based on protocol security in* Security model based on the security service needed and related protocols	Security Amount= (Wi*(GSM+PSL))/2	Score(S)
W_1	SS_1 : CONFIDENTIALITY	GSM_1: 1(-),	PSL_1	$(W_1*(GSM_1+PSL_1))/2$	$S_{CONFIDENTIALITY}$
W_2	SS_2 : INTEGRITY	GSM_2: 7, 8, 5	PSL_2	$(W_2*(GSM_2+PSL_2))/2$	$S_{INTEGRITY}$
W_3	SS_3 : AVAILABILITY	GSM_3: 2,3,6(-),9	PSL_3	$(W_3*(GSM_3+PSL_3))/2$	$S_{AVAILABILITY}$
W_4	SS_4 :AUTHENTICATION	GSM_4: 4(-)	PSL_4	$(W_4*(GSM_4+PSL_4))/2$	$S_{AUTHENTICATION}$
W_5	SS_5 : AUTHORIZATION	GSM_5 :4(-)	PSL_5	$(W_5*(GSM_5+PSL_5))/2$	$S_{AUTHORIZATION}$
TOTAL SCORE	AVG ($S_{CONFIDENTIALITY}$+ $S_{INTEGRITY}$+ $S_{AVAILABILITY}$+ $S_{AUTHENTICATION}$+ $S_{AUTHORIZATION}$)				

[DOI: 10.52547/ijict.13.3.24]

D.Proposed Security Framework: Linking All Together

In this section, we draw upon the foundations mentioned earlier and define a formula for each of five security measure to design a robust OSSD network. For this purpose, we use weighted average of Graph Security Measure and Protocol Service Level to find the score of each security service. This is presented in Proposed Security Model for OSSD.

V. Graph Security Assessment Framework

The steps performed to evaluate the security of graph are shown in the flowchart presented in Flowchart of OSS security framework. The input that this method evaluates is the graphic proximity matrix, which consists of network components including nodes and edges. After performing the seven output steps, the security level method is required for each of the network components and an overall evaluation of the network security performance.

The required level of security is determined for both nodes and edges. Establishing security in open-source projects is of particular importance for the success and development of open-source software. If an important component of the network is attacked, it is possible that the final code will be damaged and will not be accessible to the public. Documentation of any open-source project is also a valuable asset because developers cannot understand the work done and change the source code to access more features if they do not have access to the documentation. Therefore, the database that is used to store documents is one of the important components in the organization that requires high security.

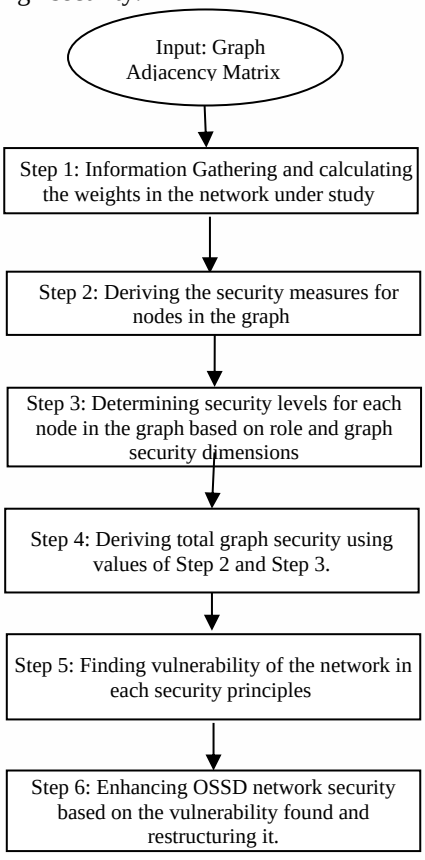


Fig 8. Flowchart of OSS security framework.

VI. Case Study

This section examines a case study and the proposed model in an open-source software. The resulting graph is given as input to the program implemented in MATLAB, and security criterion is obtained for each node and the entire graph. According to the criteria obtained for ninety edges, security services are specified.

Inputs: Graph neighborhood matrix

To produce open-source software, we require tools and infrastructure that enable us to undertake open-source projects. This section discusses infrastructure [19]. These infrastructures include the following items:

- **Public code archive (PCA):** As a primary condition for an open-source project, the source code must be publicly accessible. At any point in time, any developer, within or external to the business, should be able to acquire the most recent version of the code. A developer responsible for module maintenance should have immediate access to the module's source code.
- **Project Documentation:** Along with the normal end-user documentation required of any software product, an open-source project must have acceptable internal development documentation. They must make navigation of the source code as straightforward as possible for new developers.
- **Bug Database:** Bugs occur in software. It is vital to keep track of outstanding bugs. Certain developers prefer an email-based bug tracking system in which they get problem reports and may respond via email. Other developers desire a web-based bug database.
- **Open Mailing Lists or Newsgroups:** All talks about an open-source project must take place in the open. Users and developers should exchange information via a public mailing list or newsgroup. These lectures cover a variety of subjects, including announcements, bug reporting, problems and answers, design challenges, and future work recommendations.

Further, there are afew roles in OSSD which are important in our case study. The main roles are listed as follows:

1) *software developers (Developers),*
2) *The software quality controller (testers):* Actually, each person who use software is also tester of the software,
3) *Release managers:* A new release is created every time a change is made to the source code by someone else. In the context of open-source projects, this is what it means to have a continuous release cycle. It is possible to include a module.

Using the infrastructures and roles defined, the OSDD network under study is illustrated in OSSD network with roles. This graph is the result of people's interactions in the network. There is one node for each of the individual maps in the open-source software development project. The relationships between .the maps are shown in the graph with directional edges. The direction of the edges is determined by the information flow between the maps. Each node and edge in the graph is detailed in detail below.

[DOI: 10.52547/ijict.13.3.24]

- PCA: Storage database for the latest version of the code
- Developer: Developers
- Tester: People who test software to find bugs.
- BugDB: Bug database
- ML: Public News Group
- Rel.Man.: Some people have the role of managing released versions of software.
- Doc: For each project there is a database to store its documentation.

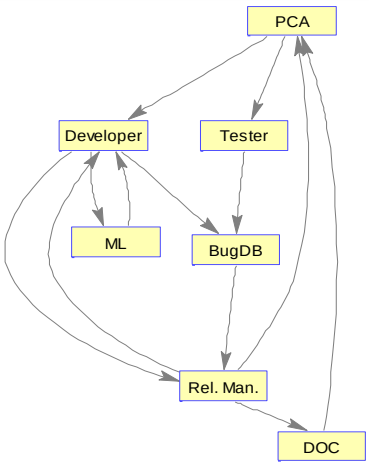


Fig 9. OSSD network with roles

For convenience, we have assigned to each of the nodes and edges of the numerical network shown in OSSD network with numerical roles values. The graph has 7 nodes and 12 edges.

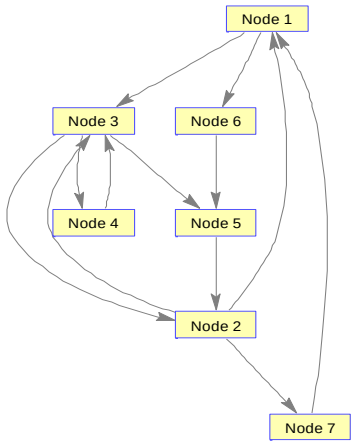


Fig 10. OSSD network with numerical roles values.

Step 1*: Collect information from the evaluated network and gain weight on security principles*: Given the importance of each of the security principles, we attribute weight to them. These weights are as follows with experimental studies for the network of open-source software developers. For gathering the expert opinion, we have interviewed wih 15 experts including 5 security experts, 5 infrastructure and 5 programming professional. We performed two series of interviews. In the first round, opinions are gathered regarding the level of security prtocols for each security principle of availability, confidentiality, integrity, authentication and authorization of an open source software network. The second run of interviews are also performed to derive the final result. The obtained results are as follows:

TABLE IV. NUMERICAL VALUES OF SECURITY MEASURES FOR OSSD NETWORK UNDER STUDY BASED ON EXPERT OPINION

Attribute	Weight
Confidentiality	0.1
Integrity	0.3
Availability	0.2
Authentication	0.25
Authorization	0.15
Sum	1

As can be seen, the sum of the weights is 1. Open-source software does not need to maintain the confidentiality of code generated by developers. Further, it is necessary that the code published on the general code archive is accurate and does not contain trojans or backdoors. So, integrity measure has the highest value. The final version of the code must always be available.

Step 2*: Measure the graph security criteria for nodes*: To calculate the,GSM_i criteria for each node, we first obtain the desired criterion value for that node. Then, based on the criterion's maximum and minimum values, we assign it a value between 0 and 1. The values obtained for the graph are shown in Simulated Data for Different Mixture Volume Ratios

Table 1.

According to the number assigned to the node in each of the columns, the importance of that node in that criterion can be understood. For example, nodes 2 and 3 have a higher closeness than the size of the nodes, and as shown in the figure, these nodes are closer to the center of the graph. To normalize the columns and place the numbers in the range 0 and 1, we divide each of them by the maximum value of that column.

Different number of criteria are important in each of the security principles. Accordingly, GSM_i values are calculated according to formula mentioned in Section IV, part A. The value obtained are inserted in the third column of

Security measures results in OSSD network Under Study.

TABLE V. SIMULATED DATA FOR DIFFERENT MIXTURE VOLUME RATIOS

Table 1. Evaluation of Graph Nodes

Node Label	Out-degree	In-Degree	Authority	Closeness	Eccentricity	Clustering Coefficient	Normalized Betweenness	Hubness	Prestige
1	2	2	0.47	0.54	3	0.33	0.33	0.47	0.33
2	3	2	0	0.66	2	0.16	0.2	0.76	0.33
3	3	3	0.76	0.6	3	0.16	0.31	0	0.5
4	1	1	0	0.4	4	0	0.1	0.36	0.16
5	1	2	0	0.4	3	0	0.2	0	0.3

[DOI: 10.52547/ijict.13.3.24]

				6					3
6	1	1	0.22	0.37	4	0.16	0.15	0	0.16
7	1	1	0.36	0.42	3	0.25	0.33	0.22	0.16
Graph Level	0.57	2	0.26	0.46	0.52	0.21	0.23	0.26	0.28

TABLE VI. SECURITY MEASURES RESULTS IN OSSD NETWORK UNDER STUDY

Security Measure (SM)	Weight (W)	Graph Security Measure (GSM)	Protocol Security Level (PSL)	Score(S) = W/2*(GSM+PSL)
Confidentiality	0.1	0.7	0.2	**0.04**
Integrity	0.3	0.43	0.5	**0.13**
Availability	0.2	0.49	0.4	**0.08**
Authentication	0.25	0.23	0.6	**0.10**
Authorization	0.15	0.73	0.5	**0.09**
Total Score	**0.44**			

Step 3: *Obtain the level of security protocols implemented in graph nodes*: Here, we assign a security protocol number to each node and network communications based on choosing proper security protocols. These protocols are selected according to graph security measures determined for the graph and the nodes' role in OSSD network and security dimension.

Fig 11. The required security level for nodes in the OSSD network is recommended in Suggested security levels for OSSD network roles. Based on the level proposed and the security margins designed (Security levels of OSSD.

and Values for protocol security levels) security levels are determined for the nodes. Then, the average of the assigned protocol level security is determined. This is given in the fourth column of Security measures results in OSSD network Under Study.

TABLE VII. SUGGESTED SECURITY LEVELS FOR OSSD NETWORK ROLES

OSSD Graph Roles	Required Security Level
1(PCA)	Secret
2 (Rel.Man)	Top Secret
3(Core Developer)	Top Secret
4(ML)	Confidential
5(BugDB)	Secret
6(Tester)	Unclassified
7(DOC)	Secret

The result of the the evaluation of graph nodes shows that the Release Manager and the Core Developer should have the highest level of security in the graph. The network hosts need to implement higher security services for the TopSecret level. BugDB, Doc and PCA maps have Secret level services and need moderate security level to implement. ML and Tester nodes require a lower level of security than other nodes.

Each edge of the graph is measured by the two criteria Betweenness and Bridge. If the edge has a high level of security, more security protocols need to be used to implement the edges.

Step 4: *Obtain the general security criteria*: it is obtained from the graph measurements (Step 2) and the level of the implemented protocols (Step 3). This number is given in the fifth column of Security measures results in OSSD network Under Study.

Step 5: *Get the network vulnerabilities in each of the security principles.* The output generated by the model is the score of each security principle in the network. The network administrator can finally understand the security vulnerabilities of the network and improve the security dimension.

Step 6: *Improve network security according to security vulnerabilities at the level of implemented protocols or graph infrastructure:*

BY LOOKING AT THE THIRD COLUMN OF

Security measures results in OSSD network Under Study the network security vulnerabilities are identified. In this network, the level of Integrity is low and it is necessary for the network manager to think of measures to upgrade the network. Adding digital signature and access control services will help improve network security in security protocol level. Further, in terms of graph restructuring for increasing integrity, some adjustment in linkage among nodes is performed.

Fig 12. The effects of these amendments in graph security measures are reported in Security measures results in restructured OSSD network Under Study. Enhanced graph security structure in the term of integrity by some link adjustment in the graph as shown in OSSD network with enhanced graph security structure for increasing integrity

.

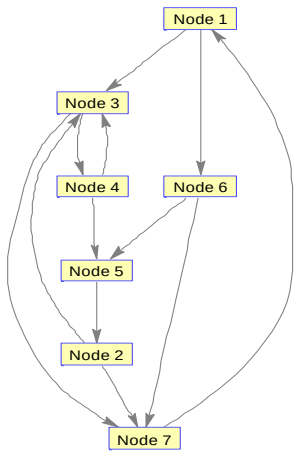


Fig 13. OSSD network with enhanced graph security structure for increasing integrity

[DOI: 10.52547/ijict.13.3.24]

TABLE VIII. SECURITY MEASURES RESULTS IN RESTRUCTURED OSSD NETWORK UNDER STUDY

Security Measure (SM)	Weight (W)	Graph Security Measure (GSM)	Protocol Security Level (PSL)	Score(S) = W/2*(GSM+PSL)
Confidentiality	0.1	0.43	0.2	**0.04**
Integrity	0.3	0.51	0.5	**0.15**
Availability	0.2	0.42	0.4	**0.08**
Authentication	0.25	0.23	0.6	**0.10**
Authorization	0.15	0.69	0.5	**0.08**
Total Score	**0.45**			

VII. CONCLUSION AND FUTURE WORKS

Given that software development is a social activity rather than a technical one, utilizing social findings to improve organizational relationships and links within OSS is quite beneficial. Due to the availability of open health datasets in Iran, particularly the various health statistical data from the Ministry of Health and the Iran Health Organization System (IHIO), which contribute the most to the availability of open health data [34], scientists are more encouraged to leverage these data and develop open-source projects. However, the security concerns of all concerned parties from various vantage points should always be considered. Using graph theory measures, this article proposes various network security principles and evaluates open-source software projects based on their positions in these OSS networks. The network's weaknesses can be recognized and addressed using the security levels and measures obtained.

Monitoring security of OSDD should be executed in intervals to avoid vulnerabilities to this network which cannot be compensated in some cases. Hence, taking snapshots of graph in different intervals and using techniques in dynamic network can be studied in future. Further, investigation of security measures from nodes perspectives and not just roles, according to the structural position are under study.

REFERENCES

[1] Kirkebø, E., *Security-Related Benefits and Challenges of an Open Source Extension*. 2020.

[2] Kiah, M.L.M., et al., *Open source EMR software: Profiling, insights and hands-on analysis.* Computer methods and programs in biomedicine, 2014. **117**(2): p. 360-38.٢

[3] Marwan, M., A. Kartit, and H. Ouahmane, *Security enhancement in healthcare cloud using machine learning.* Procedia Computer Science, 2018. **127**: p. 388-397.

[4] Schaeffer, R., *National information assurance (ia) glossary.* CNSS Secretariat, NSA, Ft. Meade, 2010.

[5] Wang, H., et al., *Combining graph-based learning with automated data collection for code vulnerability detection.* IEEE Transactions on Information Forensics and Security, 2020. **16**: p. 1943-1958.

[6] Cope, R., *Strong security starts with software development.* Network Security, 2020. **2020**(7): p. 6-9.

[7] Ghaffarian, S.M. and H.R. Shahriari, *Neural software vulnerability analysis using rich intermediate graph representations of programs.* Information Sciences, 2021. **553**: p. 189-207.

[8] Mackey, T ,.*Building open source security into agile application builds.* Network Security, 2018. **2018**(4): p. 5-8.

[9] Wen, S.-F., *A Multi-Discipline Approach for Enhancing Developer Learning in Software Security*. 2020.

[10] Chandra, P., *Software assurance maturity model.* A guide to building security into software development v1. 0, OWASP Project, 2008.

[11] Madey, G., V. Freeh, and R. Tynan. *The open source software development phenomenon: An analysis based on social network theory*. in *Eighth Americas Conference on Information Systems*. 2002.

[12] Christley, J.X.Y.G.S. and G. Madey. *A Topological Analysis of The Open Source Software Development Community*. in *Proceedings of the 38th Hawaii International Conference on System Sciences*. 2005.

[13] Hansen, M., K. Köhntopp, and A. Pfitzmann, *The Open Source approach—opportunities and limitations with respect to security and privacy.* Computers & Security, 2002. **21**(5): p. 461-471.

[14] Mansfield-Devine, S., *Open source: does transparency lead to security?* Computer Fraud & Security :(٩)٢٠٠٨ .٢٠٠٨ ,p. 11-13.

[15] Matousek, P., et al. *A formal model for network-wide security analysis*. 2008. IEEE.

[16] Wen, S.-F. *Software security in open source development: A systematic literature review*. in *2017 21st Conference of Open Innovations Association (FRUCT)*. 2017. IEEE.

[17] Feng, Q., et al. *Towards an architecture-centric approach to security analysis*. in *2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA)*. 2016. IEEE.

[18] Bosu, A. *Characteristics of the vulnerable code changes identified through peer code review*. in *Companion Proceedings of the 36th International Conference on Software Engineering*. 2014.

[19] Kim, B., et al., *Design of Exploitable Automatic Verification System for Secure Open Source Software*, in *Advances in Computer Science and Ubiquitous Computing*. 2015, Springer. p. 275-281.

[DOI: 10.52547/ijict.13.3.24]

[20] Chehrazi, G., I. Heimbach, and O. Hinz, *The impact of security by design on the success of open source software.* 2016.
[21] Tan, L., et al., *Bug characteristics in open source software*. Empirical software engineering, 2014. **19**(6): p. 1665-1705.
[22] Bosu, A., et al. *When are OSS developers more likely to introduce vulnerable code changes? A case study*. in *IFIP International Conference on Open Source Systems*. 2014. Springer.
[23] Ransbotham, S. *An Empirical Analysis of Exploitation Attempts Based on Vulnerabilities in Open Source Software*. in *Weis*. 2010.
[24] Anbalagan, P. and M. Vouk. *Towards a unifying approach in understanding security problems*. in *2009 20th International Symposium on Software Reliability Engineering*. 2009. IEEE.
[25] Xiong, M., et al., *Perspectives on the Security of Open Source Software*. eBook, 2004.
[26] Group, F.s.S.R., *Open Source Security Study: How Are Open Source development communities embracing Security Best practices?* 2008.
[27] Burgess, M., G. Canright, and K. Monsen, *A graph-theoretical model of computer security.* International Journal of Information Security, 2004. **3**(2): p. 70-85.
[28] Stang, T., et al. *Archipelago: A network security analysis tool*. 2003.
[29] Shaikh, M.I.N.a.M.S., *Secure Network Coding Schemes: Comparisons and Broader Perspective.* Sindh University Research Journal (Science Series), 2011. **43**: p. 85-90.
[30] Latora, V. and M. Marchiori, *How the science of complex networks can help developing strategies against terrorism.* Chaos, Solitons & Fractals, 2004. **20**(1): p. 69-75.
[31] Qi, Y. and H. An, *The Evaluation Model of Network Security Based on Fuzzy Rough Sets.* Advances in Wireless Networks and Information Systems, 2010: p. 517-525.
[32] Mingji ,Z., *The Application of Factor-Criteria-Metric Model in Network Security Evaluation.* Software Engineering and Knowledge Engineering: Theory and Practice, 2012: p. 887-894.
[33] Donnet, B., B. Gueye, and M.A. Kaafar, *A survey on network coordinates systems ,design, and security.* Communications Surveys & Tutorials, IEEE, 2010. **12**(4): p. 488-503.
[34] Ming-zhong, M., *Network Security Analysis Based on Graph Theory Model with Neutral Network.* Future Communication, Computing, Control and Management, 2012: p. 55.١-٥٥٧
[35] Prusiewicz, A. *A multi-agent system for computer network security monitoring*. 2008. Springer-Verlag.
[36] Giraldo, J., et al., *Security and privacy in cyber-physical systems: A survey of surveys.* IEEE Design & Test, 2017. **34**(4): p. 7-17.
[37] Fung, K.T., *Network security technologies*. 2004: Auerbach Publications.
[38] Kaeo, M., *Designing network security*. 2004: Cisco Press.
[39] de Fuentes, J.M., L. Hernandez-Encinas, and A. Ribagorda, *Security Protocols for Networks and Internet: A Global Vision* ,in *Computer and Network Security Essentials*. 2018, Springer. p. 135-151.
[40] Alwazzeh, M., S. Karaman, and M.N. Shamma, *Man in The Middle Attacks Against SSL/TLS: Mitigation and Defeat.* Journal of Cyber Security and Mobility, 2020: p. 449–468-449–468.
[41] Alkhudhayr, F., et al. *Information security: A review of information security issues and techniques*. in *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*. 2019. IEEE.
[42] Guzman, J.D., et al., *An analytical comparison of social network measures.* IEEE Transactions on Computational Social Systems, 2014. **1**(1): p. 35-45.
[43] Marin, A. and B. Wellman, *Social network analysis: An introduction.* The SAGE handbook of social network analysis, 2011. **11**: p. 25.
[44] Costa, L.F ,.et al., *Characterization of complex networks: A survey of measurements.* Advances in Physics, 2007. **56**(1): p. 167-242.
[45] Musiał, K., P. Kazienko, and P. Brodka. *User position measures in social networks*. in *Proceedings of the 3rd workshop on social network mining and analysis*. 2009.
[46] Mattie, H., et al., *Understanding tie strength in social networks using a local "bow tie" framework.* Scientific reports, 2018. **8**(1): p. 1-9.
[47] Musiał, K. and K. Juszczyszyn. *Properties of bridge nodes in social networks*. in *International Conference on Computational Collective Intelligence*. 2009. Springer.
[48] Irvine, C. and T. Levin. *Toward a taxonomy and costing method for security services*. 1999. IEEE.

Mehdi Fasanghari received his Ph.D. in Industrial Engineering at the University of Tehran. He received his B.Sc. degree from Tarbiat Modarres University. His current research interests are National Broadband Network, 5th generation of Mobile Network, Large Scale System Development, Soft Computing, and Advanced Multi-Criteria Decision Analysis.

Hamideh Sadat Cheraghchi received her Ph.D. in the Department of Computer Engineering at Shahid Beheshti University (2018). She received the B.Sc. degree from Azad University, South Tehran branch (2006); M.Sc. from Azad University, Qazvin branch (2009). Her current research interests focus on Data Mining in Social Networks, Soft Methods, and Health Care Systems.

[DOI: 10.52547/ijict.13.3.24]

Farzaneh Abazari is currently a Postdoctoral Fellow at the University of Saskatchewan in Canada. She received her Ph.D. in the Department of Computer Engineering at Iran University of Science and Technology. She was a visiting scholar in the Department of Computer Science and Engineering of the University of North Texas, Denton, TX, USA. She received her B.Sc. (2008) and M.Sc. (2011) from Amirkabir University of Technology (Tehran Polytechnic) in Software engineering and Information Security. Her research interests include Cloud Computing Security and Malware Propagation.

Farhad Pouladi was born in Shiraz, Iran, in 1972. He received his B.Sc. and M.Sc. degrees in Electrical Engineering from the Sharif University of Technology (SUT) in 1995 and 1998 and his Ph.D. degree in Electrical Engineering from the Isfahan University of Technology (IUT) in 2018 respectively. Since 1991, he has been a faculty member in ICT Faculty, Tehran, Iran. His research interests include the Internet of Things (IoT), Embedded Systems, and Power Electronics.

[DOI: 10.52547/ijict.13.3.24]