Analysis of Complex Networks for Security Issues using Attack Graph

Tanvirali Musa, Kheng Cher Yeo, Sami Azam,
Bharanidharan Shanmugam, Asif Karim,
Friso De Boer
College of Engineering, IT and Environment
Charles Darwin University, NT, Australia
musatanvir@gmail.com,

Fernaz Narin Nur, Fahad Faisal
Department of Computer Science and Engineering
Daffodil International University, Bangladesh
narin@daffodilvarsity.edu.bd, fahad.cse@diu.edu.au

Abstract- Organizations perform security analysis for assessing network health and safe-guarding their growing networks through Vulnerability Assessments (AKA VA Scans). The output of VA scans is reports on individual hosts and its vulnerabilities, which, are of little use as the origin of the attack can't be located from these. Attack Graphs, generated without an in-depth analysis of the VA reports, are used to fill in these gaps, but only provide cursory information. This study presents an effective model of depicting the devices and the data flow that efficiently identifies the weakest nodes along with the concerned vulnerability's origin. The complexity of the attach graph using MulVal has been greatly reduced using the proposed approach of using the risk and CVSS base score as evaluation criteria. This makes it easier for the user to interpret the attack graphs and thus reduce the time taken needed to identify the attack paths and where the attack originates from.

Keywords- Network Vulnerabilities, Vulnerability Assessment, Attack Graph, Attack Graph Generation Tools.

I. INTRODUCTION

Today's information is stored and processed in different electronic forms through fleet of computing devices and its networks. On the rise of information exchange through these resources, current establishments face massive challenges in securing the information. Firewalls have been deployed widely and extensively to block unauthorized access to systems from all but a few, well defined access ports. However, these devices are unable to uproot the diverse kinds of security threats that are being seen now days, nor detect attacks when they transpire [1].

As technology advances, the security challenges are increasing exponentially, which means there is a significant increase in information usage and also reveals that computing networks and its resources will be considerably attacked to compromise the security of the information stored. With trending technologies, offering wide variety of services in helping an individual or an organization to store and process their information, relying heavily on the computing environment, only marks the importance of network security to grow and strengthen [2]. A great example is the proliferation of Cloud Computing or delivery of services through Clouds. This delivery is done through vast interconnection of computer networks where ARP (Address Resolution Protocol) spoofing has become a significant threat against this emerging technology [3], even though strong encryption techniques, suggested by W. Diffie and M. Hellman, is in place [4], and varieties of encryption techniques are available [24].

Though, Hunt & Zeadally [5] believe that many security controls and tools are employed, from the perimeter level to endpoint level of the organization to address the security problems, still the networked infrastructures routinely come under attacks which are often sophisticated enough to combine multiple vulnerabilities to bypass the security controls.

Furthermore, what is often seen is that organizations find it difficult to quantify the risks posed from internal network. When it is about analysing the enterprise security, one must think of multi-staging and multi host attacks. Based on Collin's [6] statement the situation warrants such an approach which at the very beginning, will analyse the network configuration and identify the security weaknesses; so, the network graphs are to be denoted with the attack paths by simulating multi stage and multi host attack processes [6].

The intent of this research is to quantitatively asses the attacks performed on the computing networks. The network attack graphs will prove handy in visualizing the attack pattern of multi-stage network/hosts attacks in the form of graphs. Also, this research discusses existing systems and some of trends hackers follow to compromise networks.

Furthermore, the research will move ahead deriving the proposed system mined from vulnerability trends. In parallel, the effectiveness of this approach is measured with projected increase in attacks and how this model can defy hackers with dynamically emerging system.

II. LITERATURE REVIEW

A. Introduction to attack graphs

Along with hosts on the network, vulnerabilities are also increasing proportionately; it is evident that the process of evaluating vulnerabilities needs to be automated. While evaluating the security of computer networks, consideration need to be given to the identified isolated vulnerabilities [7]. When it comes to large-scale networks, it contains numerous platforms and multiple software packages employed with several modes of connectivity. Inevitably, these networks have vulnerabilities which cannot even be noticed by the system administrator [8]. Automatic generation of Attack Graphs through symbolic model checking algorithms have also been proposed to make the task easier [9]. Attack graph systems employ sophisticated techniques concentrating on the individual exploits which has the potential to be part of the attack path [10].

A probabilistic approach to explore attack graphs can also be used to find out the intention of the attack and the probable attack paths [25]. Applying the mechanisms of the attack graphs one can answer the questions on "How an attacker can break into the network, is there any detectable path?" [11].

B. Network attack graphs

A network attack graph represents a collection of probable exploitable scenarios on a given computer networks. Each scenario shows the steps followed by an attacker to achieve his goal which can range from an administrative access, database access, disruption of services to even spying. In a professionally constructed network model, an attack graph can produce an eagle-eye view for every scenario which can lead to a security breach [12].

C. Attack graph tools

This section briefly discusses some of the common attack graph generation tools:

1. Mulval

An open source logic-based tool used to generate attack graphs. MulVAL stands for *Multi stage Multi host Vulnerability Analysis,* authored by Xinming Ou. Basically, the generated attack graph has attack-step nodes. Nodes are of three types, represented in oval (attack state), diamond (privilege nodes) and rectangular (configuration nodes) shapes [13] and it is a command line interface with O (n^2) ~O (n^3) complexity. Input files, which are submitted to this tool, are of (.P) format and there are adapters to generate this file. These adapters help in creating (.P) files of the reports generated from the VA sources like Nessus and OpenVAS. The output of these scanners is of .Nessus/Oval/XML format. The VA report will be altered to (.P) format with the help of adapters present in MulVAL tool; and finally the attack graph gets generated according to the logic present in logic-execution-engine [14]. MulVAL's framework is an integration of five parts which includes rules of interaction, logical-execution-engine, security policies, database (analytical), attack path and unauthorized access.

Rules of interactions are the points which refer to statements from Data Log. The configuration information submitted to database and the rules in the database can simulate the behavior of attacker on the network.

2. Topological Vulnerability Analysis (TVA)

TVA is another tool to generate the attack graphs. This tool has capacity to analyze network vulnerabilities automatically and dig out weaknesses to generate the attack graph. A state transition diagram is established according to the attack conditions and procedures, providing network vulnerability analysis scalable to any size of the network.

3. Net SPA

Net SPA stands for Network Security Planning Architecture, the attack graphs are used to model the adversaries and the impact by providing counter measures. The attack graph generated is termed as Multiple Prerequisite Graph (MP Graph). It delivers a network model devised through firewall rules and network vulnerability scans. It has the capacity to find out the most effective attack path on the given network topology which directly helps in providing an effective solution to long term threats. The software uses a host, running services and given network information to model an attack graph which can show an attacker's view on infiltrating the network.Net SPA can generate analytical suggestions on the attack graph on how to remediate the most severe vulnerabilities in the network [15].

Net SPA also helps in identifying the critical hosts where the vulnerability of that particular host becomes a key node to be under stack compared to other hosts (node). Thus, Net SPA greatly aids administrators in identifying the critical host first and patch it up immediately before any causalities caused by the attacker. Net SPA's limitation is that the graph has many loops which make it harder for network administrators to understand and manage things effectively.

D. Tools selection

With reference to the comparison matrix of both vulnerability scanning tools and attack graph tools, the following section justifies the selecting of tools for the test environment:

1. Attacker's activity

This paper deals with probabilistic ways adopted by an attacker to fulfill their intent in breaking into the systems by

compromising the security. The normal procedure adopted by any attackers will be carried out in phases as shown below in figure 1. Attacker will pass through different stages and the success rate of the attack can only be high when data gathered at each phase is precise and accurate.

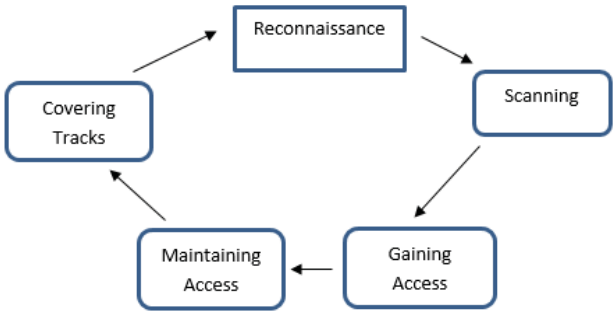

Fig. 1. Attack Phases [13, 16]

E. Findings about attack graph generation tools

1. Comparison matrix – attack graph tools

Table 1: Comparison Matrix - Attack graph tools

Attack graph/ Tools	**Type of Attack graph**	**OS/ Platform**	**Complexity**	**Input supported**	**Software type**
MulVAL	Dependency graph	Linux/ Unix	$O(n^2)$–$O(n^3)$	Nessus, NeXpose, OVAL,xml	Open source
Attack graph toolkit	State enumeration graph	Linux/Unix	Exponential	Nessus	Open source
TVA/ CAULDRON	Dependency graph	Windows	$O(n^4)$ or $O(n^6)$	Nessus, Found scan	Commercial
Net SPA	Dependency graph	Windows	O(n log n)	Nessus	No Information

Reconnaissance is the stage where intruder would gain as much information about the network. Details of target network are learnt at this phase and the IP address and its network connectivity is understood.

In the next phase, the attacker now tries to understand the weakness of the entire system and of individual nodes. The attacker uses a vulnerability scanner and looks for open ports, open services, application exploits, and loopholes in data transit. In the following phase (Gaining Access) the host is compromised to either extract information of value from that node, or to use that node\host to further launch attacks on other targets. The technique is applied following the probable path-remote exploitation followed by execution of code to exploit the weakness in the host.

Once the attacker gains access, the next phase kicks in, that is to maintain this access. Attackers may decide how deep they want to get in, but this phase can increase the attacker's vulnerability to detection with every passing minute [16].

The final phase of covering tracks simply means the attacker completes all steps necessary to eradicate all semblance of detection. This phase will not be consulted in detail in this work as this paper deals with predicting attacker's intent by analyzing the vulnerabilities identified with respect to the hosts of the network.

In a real-time scenario referring to the experimental network topology on how an attacker plans attack. The attacker initiates by performing a reconnaissance over the network, by using most widely used open-source tool like *Nmap* to identify the hosts that are part of the network along with the type of services running on those hosts. Next the second phase works on the information acquired by the first phase. Every identified vulnerability has an assigned reference number which will be unique and gets listed into National Vulnerability Database (NVD) as well as in MITRE system [16]. The database contains complete information about the identified vulnerability besides loads of other useful information.

The following vulnerabilities have been reported in the vulnerability scan performed on the example. The below vulnerabilities are well-known and are easily exploitable. Summary of the vulnerabilities are given below:

- *CVE-2008-4835* is known to affect SMB (Server Message Block) service through memory corruption vulnerability which may allow attacker to execute malicious code or attacker can also carry out denial of services against the remote hosts.
- *CVE-2008-4250* is identified with triggering buffer overrun issues in the "server" service which lets an attacker to execute arbitrary code in the remote host with the system privileges [17].
- *CVE-2014-6321* is related to weakness in processing the packets over a secure channel (SChannel) allowing attacker to craft the packets on their way to the server.
- *CVE-2012-0152* is known to show up when there is a RDP service enabled on the vulnerable system and unauthenticated user can leverage this vulnerability to execute malicious code by sending RDP packets.

Considering the above weak link, it is clear there are two types of vulnerabilities which can be exploited at different levels. First is the remotely exploited vulnerability which works over a network and exploit the machine without any prior access to the vulnerable machine; the other is a local exploit requiring access to the machine prior taking over that very host. Post exploitation will escalate the privilege levels to administrator.

2. Target environment and vulnerability correlation

It is vital for any prediction attack graphs to have collected information about the systems and associated vulnerabilities. Since the graph is purely dependent on the number of hosts

and its vulnerabilities it is vital to perform a comprehensive scan of every host. If anyone of the prime host's information is missing or incomplete, the created graph would not be effective. Other important feature is runtime configuration as it requires HACL (host access control list), which has the information of attacker whereabouts, running network services and user accounts [16]. Another characteristic is attacker's logs which get generated during the process. Every attacker uses their own tactics and every step of the approach will be clearly documented.

F. Methods on reducing attack graphs complexity

The complexity of the attack graph is basically determined by two important factors. Number of Hosts N(h) and Number of identified vulnerabilities present in the NVD database number of vulnerability N(v). For instance, consider a network with **n** number of hosts and after scanning an attack, action have been performed on the **n** hosts and is approximately represented as:

$$F(N(h), N(v)) = N(h)N(v)F(N(h)-1,N(v)) = N(h)N(v)(N(h)-1)N(v)F(N(h)-2,N(v))= N(v)^{N(h)}N(h)!$$ [21].

This shows that the approach faced a combinational explosion with respect to complexity. Therefore it is more suitable to smaller networks, but not applicable unless there is a modification for large networks [19].

In 2009, researchers [19] described about Model Checking which was in use to enumerate the attack chains to link initial access points to the goal of the attacker. Due to explicit enumeration of attacker's state, these families of approaches are always growing exponentially in proportion to the size of the network. Monotonic logic helped attack graph's complexity subside to polynomial from exponential. The complexity was further reduced while having quadratic number of hosts. It is also possible to bring the complexity down by grouping of networks into single domain where connectivity among the hosts is not restricted and this domain has tight security protection rules already in place. With this kind of topology, complexity will reduce to linear considering single domain; generally, the complexity swells to quadratic depending on the count of the protected domains (as the count will be domain number but not host hence it will be a lot lesser than expected). Such graphs can be produced from a mere hundred to tens of thousands of hosts within minutes but not with visualizations. Attempts were also made to measure the network security risk in combination of individual vulnerabilities and its relevant metrics. Converting the attack graphs and vulnerability score to Bayesian networks for better computation of cumulative probability has been proposed by Frigault et al. [20] in his paper, explained a better approach on recognizing the cycles which are existing in attack graphs..

Singhal's work was quite meaningful research in improving visualizations of network security architecture. For any environment the preliminary point is to quantify the attack surface and its impacts because it is the factor to control risk posed to the computer networks described in his research [21].

Due to explicit enumeration of attack states, attack graphs become considerably convoluted. With the work in monotonic graph generation, complexity for the same graph reduces to polynomial from exponential [22]. Alhomidi and Reed [23], proposed a methodology to explore the graph using genetic algorithm where each attack path is an attack scenario from its source to attacker's goal. This evolves to be a natural way to generate maximum number of possible attack paths which again makes the graph a lot more complex gradually.

III. PROPOSED APPROACH – (RISK AND CVSS BASE SCORE AS EVALUATION CRITERIA)

Vulnerability scans were performed using Nessus and the output of the scan can be extracted in multiple formats like Nessus dB, csv, html, pdf, .Nessus. MulVAL and Nessus complement each other, MulVAL have utilities which supports and converts the Nessus file formats to MulVAL readable files. Then these readable files are processed for graph generations. The process has been depicted in figure 2.

Hence before processing the vulnerability report from Nessus scanner to MulVAL's framework, it is possible to analyze and figure out for any false positives, vulnerabilities which are outdated and also to identify those vulnerabilities where CVE-IDs have been registered but do not demonstrate any effectiveness. Thus considering all these factors the current research works in a direction where the output of vulnerability assessments are thoroughly evaluated before generating the attack graphs leading to improvement of complexity. MulVAL identifies the vulnerabilities based on the CVE-IDs but the output of the scan is not precisely evaluated, hence there is a need to verify the output generated considering Risk and CVSS score as the factors.

Nessus vulnerability scanner provides with the facility to export a filtered vulnerability report based on user requirement. Normally in this approach evaluation of Risk along with Common Vulnerability Scoring System (CVSS) base score have been used. Vulnerability with CVSS base score of 10 -7 were considered the first vulnerabilities on the network to be addressed as these vulnerabilities will have exploits available.

The reason for selecting this range is that the vulnerabilities within this range can be remotely exploited, meaning these vulnerabilities provide attacker a gateway for successful exploitation.

So once these gets identified and dealt with, and assigned the highest priority levels, the gap for the attacker gets closed. Depending on the Risk Factor and associated CVSS Score, vulnerability is defined.

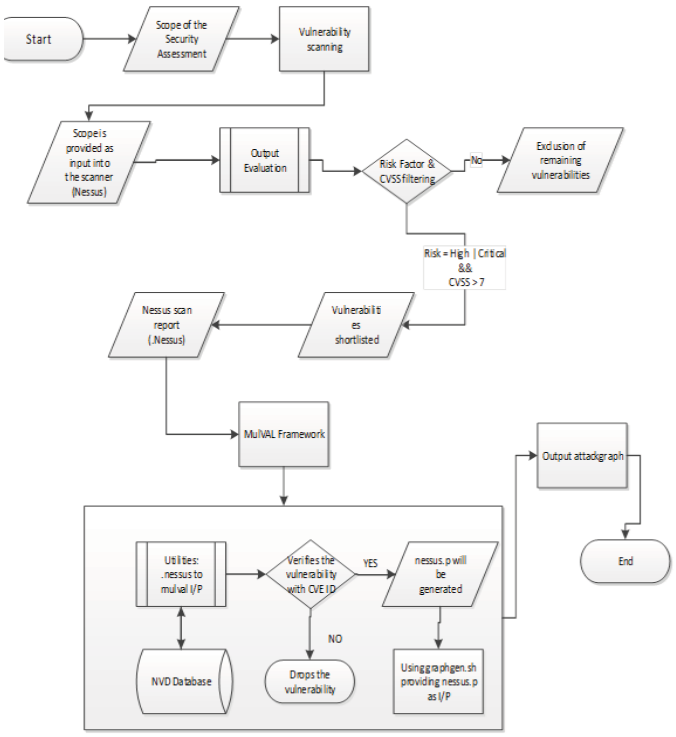

Fig. 2. A Flowchart for the Proposed Approach

1) Risk Factor - This helps while segregation of vulnerabilities according to its risk level (critical, high, medium, low, informational).

2) CVSS - Common Vulnerability Scoring System (CVSS), it is an open framework. Any software/hardware/firmware vulnerabilities can be a threat to the entire organization and it is quite difficult to mitigate. CVSS provides a way to capture the characteristics of the vulnerability and assign a score which is a numerical value indicating its severity. This numerical score is translated into a qualitative representation which is as follows.
   - Critical (Risk) Vulnerability – 10.0 (CVSS Base Score)
   - High (Risk) Vulnerability – 9.0 -7.0 (CVSS Base score)
   - Medium (Risk) Vulnerability – 6.9 – 4.0 (CVSS Base Score)

Rest is not suitable to the current proposed approach.

IV. RESULTS AND DISCUSSIONS

The attack graph generated was quite comprehensive in providing the information of the attacks. The graph is better than the previously generated attack graphs based on the system generated report. Every tool/software result contains some degree of false positives, reports are to be thoroughly analyzed based on vulnerability exploitable factors and then an attack graph generation would be something which has real effect. Hence the graph seems to be simple compared to the first attack graph. It is also observed that the attack loops drastically dwindle. There are few other factors which are helping the cause, like, those which have a CVE-ID but cannot be exploited.

It is mandatory for one to understand though there is a vulnerability registered with CVE-ID, risk posed by the vulnerability should also be considered along with availability of exploits. If any of the mentioned factors associated to a vulnerability is missing, it cannot be exploited, which means the attack path generated in supporting these vulnerabilities considered to be “true negative”- because vulnerability was detected but still couldn’t be exploited. Hence instances like these add to the complexity of the graph and complicate it, rendering it hardly readable. With respect to the above generated graph, the presented paths are the potential paths of the attack for breaking into the network.

It is also understood from the analysis that vulnerabilities which can be remotely exploitable are the prime concern to any networks. Since they can be accessed across the network and in case of being hosted through internet, the risk of threat is top notch. The final attack graph is one such graph with special concentration on remotely exploitable vulnerability family. If these vulnerabilities are rectified, it will then basically end the road for the attacker to sneak into the network, leading to zero attack paths.

A. Findings

It is clearly depicted in Table II the major difference in the count of the vulnerabilities alongside the number of nodes. Hopefully now due to this approach the graph has only those attack paths which have potential to become attacker’s probable paths because this is after adopting the new approach of only processing precise data which can generate better reliable graph. The Nessus report is thoroughly evaluated based on vulnerability risk and exploitable features which were not done earlier. This evaluation helps identifying the false positives and those vulnerabilities which cannot create any impact.

In previous approach Nessus was used to generate information and that same data were processed into MulVAL framework. There was no evaluation of the data which were being generated by Nessus hence the graph was misleading, imperfect and above all too complex to be understood properly.

Hence there was a necessity to evaluate the Nessus output which has possible number of vulnerabilities that cannot be served to be potential enough to create an impact. Those vulnerabilities have been identified and excluded which

directly helped in improving the complexity of the attack graph.

Table 2: Comparison table between two Attack graphs

Complexity Factors	**Attack Graph**	**Attack Graph (Post Evaluation Approach)**
No. of hosts	6	4
Total no. of vulnerabilities	10	5 (6th one is an outlier)
No. of nodes	100	53
No. of attack hoops	38	9
No. of vulnerabilities with "Risk"=None	2	0
No. of vulnerabilities without exploits available	5	0

V. CONCLUSIONS AND FUTURE WORK

The aim of this research is to analyze the security of networks using attack graph concepts and reduce the complexity of attack graph. However, even though this tool is slightly complex but in general provides good foundation for research work with respect to attack graphs, despite the issue of spending some extra hours filtering out the false positives. Future research could be working with other open source vulnerability scanners and incorporating the attack graph into open source scanners like NMAP. Other ways of reducing the complexity of the attack graphs can also be explored. This would enable the network security administrators to have clear idea of the attack and where it originates from.

REFERENCES

[1]. M. Bennet, S.S., M. Deepika, N. Nanthini, S. Bhuvaneshwari & M. Priyanka, "A Memory Efficient Hardware Based Pattern Matching And Protein Alignment Schemes For Highly Complex Databases," International Journal on Smart Sensing and Intelligent Systems, 2017. **10(4)**: p. 101-122.

[2]. I.Kotenko, A.M.S., "Attack Graph Based Evaluation of Network Security," Communications and Multimedia Security, New York, USA: Springer-Verlag, October 2006. **216-227**.

[3]. V.D.S. Vijayarangam, "Detecting Ip Based Attack On Cloud Server Using Passive Ip Traceback," International Journal on Smart Sensing and Intelligent Systems, 2017. **10(4)**: p. 136-146.

[4]. X. Elvis, Kheng Cher Yeo, A. Sami, S. Bharanidharan, "Performance analysis of various encryption techniques in communication network," Asian Journal of Information Technology, 2017. 16**(1)**: p. 125130.

[5]. R. Hunt, A.S.Z., "Network Forensics: An Analysis of Techniques, Tools, and Trends. Computer," December 2012. **45**(12): p. 36-43.

[6]. M.P. Collins, "Graph-based analysis in network security," p. 1333-1337.

[7]. V.N.L. Franqueira, "Finding multi-step attacks in computer networks using heuristic search and mobile ambients," University of Twente, 2009.

[8]. E. Cole, "Network security bible." John Wiley & Sons, 2011. 768.

[9]. C. Wang, N.D., and H. Yang, "Generation and Analysis of Attack Graphs. Procedia Engineering," 2012. **29**: p. 4053-4057.

[10]. C. Phillips, A.L.P.S., "A graph-based system for network-vulnerability analysis," Digests in Proceedings of the 1998 workshop on New security paradigms, 1998: p. 71-79.

[11]. S. Yi, Y.P., Q. Xiong, T. Wang, Z. Dai, H. Gao, J. Xu, J. Wang and L. Xu, "Overview on attack graph generation and visualization technology," International Conference on Anti-Counterfeiting, Security and Identification (ASID), Shanghai, 2013: p. 1-6.

[12]. S. Jajodia, S.N., and B. O'Berry, "Topological analysis of network attack vulnerability," Managing Cyber Threats, Springer, 2005: p. 247-266.

[13]. X. Ou, S.G., and A. W. Appel, "MulVAL: A Logic-based Network Security Analyzer," USENIX security, 2005.

[14]. X. Ou, W.F.B. & M. A. McQueen, "A scalable approach to attack graph generation," Digests 13th ACM conference on Computer and communications security, 2006: p. 336-345.

[15]. M.T.A.A.N.Z. Heywood, "VEA-bility security metric: A network security analysis tool," Digests Third International Conference on Availability, Reliability and Security, 2008: p. 950-957.

[16]. R. Baloch, "Ethical Hacking and Penetration Testing Guide," CRC Press, 2014.

[17]. O.M. Sheyner, "Scenario graphs and attack graphs," Air Force Research Laboratory, 2004.

[18]. L.-H. Hsu and C.-K. Lin, "Graph theory and interconnection networks," 2008: CRC press.

[19]. S. Noel, A.S.J., "Managing attack graph complexity through visual hierarchical aggregation," [Digests in Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security, October 2004: p. 109-118.

[20]. M. Frigault, L.W., A.Singhal, and S. Jajodia, "Measuring network security using dynamic bayesian network," Digests 4th ACM workshop on Quality of protection, October 2008: p. 23-30.

[21]. I.Singhal, A.X.O., "Techniques for enterprise network security metrics," Digests 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies, April 2009. **ACM**: p. 25.

[22]. P. Ammann, D.W., and S. Kaushik, "Scalable, graph-based network vulnerability analysis," Digests 9th ACM Conference on Computer and Communications Security, 2002: p. 217-224.

[23]. M. Alhomidi, A.M.R., "Attack Graph-Based Risk Assessment And Optimisation Approach," School of Computer Science and Electronic Engineering University of Essex, Colchester, UK International Journal of Network Security & Its Applications (IJNSA), 2014. **6**(3).

[24]. A.VMota, S. Azam, B. Shanmugam, K. C. Yeo & K. Kannoorpatti, "Comparative analysis of different techniques of encryption for secured data transmission", IEEE International Conference on Power, Control, Signals and Instrumentation Engineering, ICPCSI 2017.

[25]. Gao, N., He, Y. & Ling, B. Wuhan Univ. J. Nat. Sci. (2018) 23: 171. https://doi.org/10.1007/s11859-018-1307-0 "Exploring Attack Graphs for Security Risk Assessment: A Probabilistic Approach.