Evaluating Damage Potential in Security Risk Scoring Models

Eli Weintraub
Department of Industrial Engineering and Management
Afeka Tel Aviv Academic College of Engineering
Tel Aviv, Israel

***Abstract*—A Continuous Monitoring System (CMS) model is presented, having new improved capabilities. The system is based on the actual real-time configuration of the system. Existing risk scoring models assume damage potential is estimated by systems' owner, thus rejecting the information relying in the technological configuration. The assumption underlying this research is based on users' ability to estimate business impacts relating to systems' external interfaces which they use regularly in their business activities, but are unable to assess business impacts relating to internal technological components. According to the proposed model systems' damage potential is calculated using technical information on systems' components using a directed graph. The graph is incorporated into the Common Vulnerability Scoring Systems' (CVSS) algorithm to produce risk scoring measures. Framework presentation includes system design, damage potential scoring algorithm design and an illustration of scoring computations.**

Keywords*—CVSS; security; risk management; configuration; Continuous Monitoring; vulnerability; damage potential; risk scoring

I. INTRODUCTION

Cyber-attackers cause damage to organizations and personal computers by stealing their business or private data and by making changes in their software and hardware [1]. The damages are usually categorized by security experts to three kinds: loss of confidentiality, integrity or availability. Vulnerabilities are software weaknesses or exposures. An attack is performed by exploiting software vulnerabilities in the target system. Attackers make use of vulnerabilities stemming from bugs that are potential causes to security failures. Exploits are planned to attack certain components having specific vulnerabilities. Users' computers might be damaged by exploited vulnerabilities. Defending computers depends on the amount of knowledge an organization has of it's computing systems' vulnerabilities. This work focuses on gaining accurate knowledge of computers' configuration, thus enabling improved organizational risk mitigation activities, to defend computers from threats caused by attackers. Accurate knowledge of computers' risks assists security managers to adopt security measures effectively. Reference [2] states that Stuxnet worm included a process of checking hardware models and configuration details before launching an attack. Both, attackers and security managers are interested in gaining accurate and detailed information of the system. Risk managers make decisions on activities actions they have to perform in order to limit their exposure to risks according to the amount of potential damage and vulnerability characteristics [3].

Risk has many definitions in research publications. In this research we use the definition of [4]: "An event where the outcome is uncertain". According to this definition, this work is aimed at lessening risk uncertainty. The proposed model focuses on an improved collateral damage potential evaluation process which is based on the real-time information on systems' configuration components, and on system interfaces with users.

Several software products are used to defend computers from cyber attackers. Antivirus software, antispyware and firewalls are examples to some of these tools based on periodic assessment of the target computer by comparing computers' software to the known published vulnerabilities. Those tools are effective only against known threats and not against new unpublished threats. CMS monitor computer systems in a near real time process aimed at detecting vulnerabilities and notifying security managers. Contemporary systems use vulnerabilities databases which are continually updated as new vulnerabilities are detected and a scoring algorithm which predicts potential business damages. This work focuses on the impacts of the components incorporating the configuration about potential damages. The CMS evaluates damage potential relating to the actual configuration. Each time changes are performed to components damage potential is evaluated and updated. CMS's are useful tools for limiting the time-frames organizations are exposed to risks.

Computers are at risk to known threats until the time a patch is prepared for defending the vulnerable software, an activity that may last weeks or months. Even after a patch is prepared by the software vendor a computer might still be at risk until the moment the new patch is loaded to the system. Loading patches to computer systems is usually performed as a periodical process, not continuously to avoid too many interrupts required for uploading the patch on organizations' computers. Other software tools are based on heuristic algorithms which are planned to detect irregular suspicious activities of the software running on the computers. In today's environment of zero-day exploits, conventional systems updating for security mitigation activities has become a cumbersome process. There is an urgent need for a solution that can rapidly evaluate system vulnerabilities' potential damages for immediate risk mitigation [5].

Security Continuous Monitoring (SCM) tools use techniques for monitoring, detecting and notifying of security threats in real time. After identifying these risks, the tools evaluate the potential impacts on the organization. Reference [6] states that SCM systems which are running on computers continuously try to detect systems' vulnerabilities, are aimed at closing the gap between the zero-day of identifying the vulnerability, until the moment the computer is loaded by a patch. The time gap may be considerably long.

This paper describes the mechanisms of a new SCM framework of a system that will produce better risks scoring than current known systems. The framework bases processes on two grounds: 1) knowledge concerning real computers' configuration of the target system, and 2) a prediction algorithm which runs continuously and computes damage potential estimates for use of risk scoring models.

The rest of the paper is organized as follows: In section 2 a description of current known existing solutions. In section 3 a presentation of the proposed framework including systems architecture. In section 4 a description of the risk scoring algorithm which computes risk scores. In section 5 presentation of the results. In section 6 conclusions and future research directions.

II. Existing Solutions

SCM systems are using external vulnerabilities databases for evaluation of the target computers' risk. There are several owners of vulnerability databases [5] for example The Sans Internet Storm Center services and The National Vulnerability Database (NVD). Vulnerability Identification Systems (VIS) aimed to identify vulnerabilities according to three categories: code, design, and architecture. Examples for VIS systems are The Common Vulnerabilities and Exposures (CVE), and The Common Weakness Enumeration (CWE).

This work uses NVD vulnerabilities database as an illustration of the proposed model.

Risk evaluation uses scoring systems which enable parameters estimation for estimating vulnerabilities' impacts on the organization. The Common Vulnerability Scoring System (CVSS) is a framework that enables user organizations receive IT vulnerabilities characteristics [1].

CVSS uses three groups of parameters to score potential risks: basic parameters, temporal parameters and environmental parameters. Each group is represented by score compound parameters ordered as a vector which is used to compute the score. Basic parameters represent the intrinsic specifications of the vulnerability. Temporal parameters represent the specifications of a vulnerability that might change over time due to technical changes. Environmental parameters represent the specifications of vulnerabilities derived from the local IT specific environment used by users' organization. CVSS enables omitting the environmental metrics from score calculations in cases that users' environment has no effect on the score and in cases the users do not specify the detailed description of environment and it's components.

CVSS is a common framework for characterizing vulnerabilities and predicting risks, used by IT managers, risk managers, researchers and IT vendors. It uses an open framework which enables managers to deal with organizations' risks based on facts rather than evaluations. Organizations adopting CVSS framework may gain the following benefits:

- A standard scale for characterizing vulnerabilities and scoring risks.
- Normalizing vulnerabilities according to specific IT platforms. The computed scores enable users make decisions according to vulnerability risks.
- CVSS uses an open framework. Organizations can see the characteristics of vulnerabilities and the logical process of scoring evaluation.
- Environmental scores. Organizations using the environmental parameters benefit by considering changes in its IT environment according to predicted risk scores.

There are few other vulnerability scoring systems besides CVSS differing by the parameters' specifications and scoring scales. CERT/CC emphasizes internet infrastructure risks. SANS vulnerability system considers users' IT configuration. Microsoft emphasizes attack vectors and vulnerabilities' impacts.

Using CVSS scoring system, basic and temporal parameters are specified and published by products' vendors who have the best knowledge of their product. Environmental parameters are specified by the users who have the best knowledge of their environments and business impacts.

This paper focuses mainly on environmental metrics.

Business damages caused by a vulnerability are influenced by the IT exploited component. CVSS environmental parameters specify the characteristics of a vulnerability that is associated with user's IT configurations' components. Environmental parameters are of three groups:

1) Collateral Damage Potential (CDP).

A group of parameters which measure the economic potential damage caused by a vulnerability.

2) Target Distribution (TD).

Parameters indicating the percentage of vulnerable components in users' environment.

3) Security Requirements (CR, IR, AR).

Parameters indicating security importance measures in users' organization. Those parameters are subdivided to parameters indicating the confidentiality (CR), integrity (IR), and availability (AR). Higher security requirements may cause higher security damages on the organization.

Categorization of IT components according to security requirement measures should encompass all assets to raise the possibility of predicting organizational damages. Federal

Information Processing Standards (FIPS) requirements demands implementation of a categorization [6], but does not require using any particular scale, thus risk comparison of users' systems is difficult.

III. The Proposed Framework

Federal organizations are moving from periodic to continuous monitoring implementing SCM's which will improve national cyber security posture [7]. The proposed framework includes two capabilities which are not found in current practices. First, the environmental parameters are based on the components of the system as updated in the systems' Configuration Management Data Base (CMDB) [8]. This capability enables basing the scoring models to predict organizational damage potential relating to actual IT configuration rather than relying on user's estimates. According to [9] it is impossible for organizations to make precise estimates of the economic damages caused by an attack without having full knowledge of users' IT environment. Reference [10] [11] states that network configuration should be monitored continually and available vulnerabilities must be analyzed in order to provide the necessary security level.

Several researchers tried to simulate IT configuration processes using graphs. Researchers studied the impacts of component dependency graphs [12] [13] [14]. [15] Claims that CVSS does not take into consideration component dependencies, which impacts dramatically the exploitability of a vulnerability. [15] States that current CVSS do not reveal the fact that vulnerabilities on highly depended packages usually bring larger attack surfaces compared to those detected on a client application, even when they have the same CVSS scores. [15] Studied the impacts of components dependencies which refer to a code reuse by a component from the library packages that it relies upon. [16] Presents a risk estimation model that makes use of CVSS to produce security risk levels implemented as a Bayesian Belief Network (BBN) topology.

This research models the configuration using a visual directed graph to represent network structure describing network components and component' messages relationships. Visualization tools are used to model network structure or attack paths. Modeling attack paths enables analyzing network security to predict future attacks. Attack graphs can represent potential attack paths which an attacker can take to reach the system. According to [17] attack graphs act as a tool in finding critical paths in large networks based on the threats and vulnerabilities identified. The layout of an attack graph can be adjusted to represent the real enterprise network. [18] Proposes an attack graph-based probabilistic metric for network security. According to the model this research proposes, knowledge concerning the environmental components is represented as a directed graph which includes information on systems' components, the links which represent data reads/writes between components and systems' impacts on external users such as an error caused to a users' interface or errors in transactions routed to other interfacing systems. Each link is assigned a probability which resembles the occurrence probability of the specific link between the two components. Occurrence probabilities are computed regularly by monitoring the daily system' processes at production activities capturing all message passing among components. In the past such automated systems were not advanced, but according to [19], there are currently automated tools to generate visualization maps of systems activities (specifically for attack vectors monitoring), with the inputs from the system and its environment. Components' collateral damage potential scores are computed by activation of a rollback algorithm using the directed graph. Graph design represents all software activities processed by the system. The activities are initiated by external inputs which belong to the attack surface. Those external input components pass messages to internal components of the system which pass further messages to other components, ending at generation of external interfaces. This research assumption is that users are capable of estimation business damages relating to external interfaces only. They have no capability of estimating business damages relating to internal software components or to external systems' inputs. The focus of this work is in evaluating the potential damages to all input and internal components. The damage caused by an exploit to a vulnerable surface input component is computed by evaluating all message passing and impacts on neighboring components to all internal components according to their occurrence probabilities, ending at the generation of a wrong output, which is delivered to a certain user. The user is capable of estimating business damages caused by wrong information written on users' interfaces.

The proposed CMS model examines a database of published asset vulnerabilities, compares in real time computers' assets for existing exposures and calculates computers' potential damages, based on the directed graph. Risk scoring is performed by considering vulnerabilities even before patches are prepared and loaded on the computers' system. The CMS proposed architecture presented in Fig. 1. Following, a description of systems' components and processes.

- Continuous Monitoring System (CMS)

The system runs continuously and starts computing potential damages in two cases: first is whenever a new vulnerability is publishes and indicated in the NVD, second is whenever a change is made in a systems' component or in systems' interfacing component. Following a description of systems' modules.

- Vulnerabilities database (NVD).

Vulnerabilities database includes all known vulnerabilities and their specification as published by database owners. Examples of vulnerability specifications used by NVD are: vulnerability category, vendor name, product name, published start and end dates, vulnerability update dates, vulnerability severity, access vector, and access complexity [6].

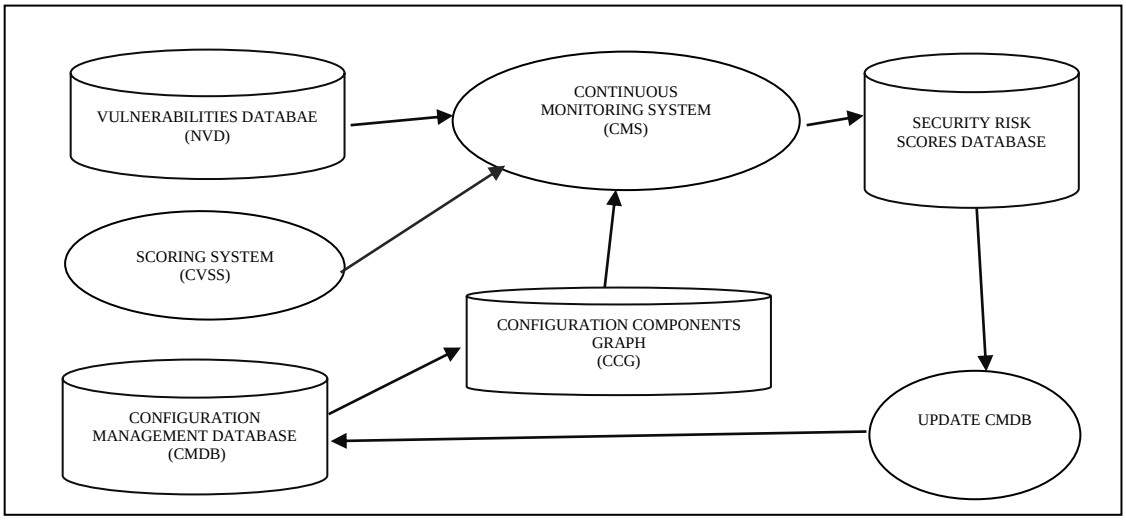

Fig. 1. Continuous Monitoring System architecture

- Scoring system (CVSS)

Scoring system (CVSS) is the algorithm this research uses for illustration of the proposed model. CVSS computes security risk scores according to parameter groups: basic, temporal and environmental. There are also other known scoring algorithms, some of them for public use other commercial.

- Configuration Management Database (CMDB).

CMDB is a database which includes all hardware and software components of the target system. According the proposed model the CMDB includes detailed information of the hardware and software. The CMDB contains detailed information of each module, systems' components and relationships among the components. Software is dealt in the resolution of programs, services and parameters. Data is handled in the resolution of databases, tables and data items. Input/output interfaces are handled using screen-names, reports and messages. The target system might be one computer or a group of organizations' computers. The CMDB includes all components in computers' configuration, components which interface with the target system directly or indirectly up to external and end-users' interfaces. The CMDB includes also the security requirements (CR, IR, AR) of each component in the resolution of data items' security requirements. Security requirements are specified by systems' owners according to business potential damages. CMDB includes also all interfaces among components. For each interface an indication of the direction of messages passing between the components and the probability of messages passing occurrence.

- Configuration Components Graph (CCG).

CCG is a directed graph including all the components in the CMDB organized as a directed graph which enables operating a rollback process which is aimed to compute components' collateral damages. The Rollback process starts at output external components, which potential damages has been assigned by users, continues backward to connected internal components, ending in input surface components [see Fig 2]. At the end of the rollback process all collateral damage potential values of systems' components are calculated. The graph includes three kinds of nodes: external inputs (IN), External output (OUT), and internal components (INTERNAL). Arcs between nodes represent message passing from one component to other components. Each arc is assigned a real number between [0, 1] representing the occurrence probability of the link between the connected nodes. External inputs represent all kinds of inputs to the system such as user interfaces, e-mails etc'. A subgroup of the external input components are surface attack components which might be of the following types [20].

1) Services available in the firewall which handles incoming messages

2) Systems code that processes incoming data, email, XML, office documents, industry-specific custom data exchange formats (EDI)

3) Interfaces, SQL, web forms

4) Employees accessing sensitive information

Messages are forwarded from the external inputs to internal components following to external outputs. Damage potential score evaluation goes in the backward direction: from external output scores (which are estimated by the users), back to their corresponding input components, and finally back to the corresponding external input components.

- Security Risk Scores Database.

The database includes all computed damage potential scores as computed by the CMS. The scores are then updated in the CMDB by the UPDATE CMDB process. The CMDB is used for retrieval purposes by business managers and analysts. Regular requests for damage potential score will be supplied by the CMDB. In cases of updates to systems' components or to NVD records, CMS will initiate an activation of the rollback process using the CCG. CMDB scores represent the damage potential evaluated updated scores for all systems' components including internal and external input components. This update process is needed to prevent unnecessary risk

score heavy computations which were already evaluated and has been written in the past in the CMDB.

IV. The Risk Scoring Algorithm

CVSS's framework is based on three kinds of parameters: basic and temporal parameters are specified and published by products' vendors who have the best knowledge of their product. Environmental parameters are specified by the users who have the best knowledge of their environments and vulnerability business impacts. This work deals with the environmental parameters. According to [6], in many organizations IT resources are labeled with criticality ratings based on network location, business function, and potential for loss of revenue or life. For example, the U.S. government assigns every unclassified IT asset to a grouping of assets called a system. Every system must be assigned three "potential impact" ratings to show the potential impact on the organization if the system is compromised according to three security objectives: confidentiality, integrity, and availability. Thus, every unclassified IT asset in the U.S. government has a potential impact rating of low, moderate, or high with respect to the security objectives of confidentiality, integrity, and availability. This rating system is described within Federal Information Processing Standards (FIPS) 199.5 [21]. CVSS follows this general model of FIPS 199, but does not require organizations to use any particular system for assigning the low, medium, and high impact ratings. Reference [22] states that organizations should define the specifications of security risks of their specific environment, but does not define the ways organizations have to specify that information. The Department of State (State) has implemented an application called iPost and a risk scoring program that is intended to provide continuous monitoring capabilities of information security risk to elements of its information technology (IT) infrastructure. According to [23] the iPOST scoring model does not refine the base scores of CVSS to reflect the unique characteristics of its environment. Instead, it applied a mathematical formula to the base scores to provide greater separation between the scores for higher-risk vulnerabilities and the scores for lower-risk vulnerabilities. This work is targeted to fill-in this vacuum.

The CMDB defined in this work handles configurations' information of the system including the following entities: database tables, software components, system components such as operating system, database management systems, utility programs, development components, UI screens, etc. Each component is describes including knowledge relating to security requirements needed for operation of the risk scoring algorithm. The CMDB includes also all relationships among components, for example message passing to/from two components and function calls. The CMDB manages five kinds of environmental information for every system component. Table I includes information concerning the characteristics assigned to systems' components. Characteristic values are based on [21] definitions. The information is categorized according to its security type which is defined as a specific category of information (e.g., privacy, medical, proprietary, financial, investigative, contractor sensitive, security management). Reference [21] states that the potential impact is low if the loss of confidentiality, integrity, or availability could be expected to have a limited adverse effect on organizational operations, organizational assets, or individuals. The potential impact is moderate if the loss of confidentiality, integrity, or availability could be expected to have a serious adverse effect on organizational operations, organizational assets, or individuals. The potential impact is high if the loss of confidentiality, integrity, or availability could be expected to have a severe or catastrophic adverse effect on organizational operations, organizational assets, or individuals.

TABLE I. CMDB – Components Table

Column ID	Column Name	Column Description	Values (*)
COMPONENT ID	Software or Hardware	Value is equal to component ID in NVD	unique
COMPONENT TYPE	According to external or internal entities	I = Input external O = Output external INT = Internal	I, O, INT
CDP	Collateral Damage Potential	This metric measures the potential for loss of life or physical assets through damage or theft of property. The metric may also measure economic loss of productivity or revenue.	N, L, M, MH, H
TD	Target Distribution	This metric measures the proportion of vulnerable systems.	N,L,M,H
CR	Confidentiality Requirement	The importance of the affected IT asset to a user's organization, measured in terms of confidentiality.	L,M,H
IR	Integrity Requirement	Guarding against improper information modification or destruction.	L,M,H
AR	Availability Requirement	"Ensuring timely and reliable access to and use of information…".	L,M,H

(*) N=none, L=low, LM=low medium, M=medium, MH=medium high, H=high

Table II describes the relationships among couples of components which were described in Table I. A relationship between two components represents certain activities performed between the components for example read from an external input component, write to an external output component, function calls from one to another component. This table is used for generation of the directed graph. Each component will be represented by a node in the graph, and each relationship will be represented as an arc. This table includes description of all the relationships among systems' components. Each row represents one link between two components. Each link between two components is assigned a

link probability which represents the occurrence probability of the specific activity, meaning the statistical probability that the input component will activate the output component relating to all the activities generated by that specific input component.

TABLE II. CMDB – LINKS TABLE

Column ID	Column Name	Column Description	Values
COMPONENT ID	Value is equal to component ID in NVD and CMDB – Components table. This in the input of the link	ID of the component which performs a certain activity on the component on the second end component of the link	I, INT (cannot be an output external component)
COMPONENT ID	Value is equal to component ID in NVD and CMDB – Components table. This in the output of the link	ID of the component which is impacted by certain activity of the components which is the input of the link	O, INT (cannot be an input external component)
LINK PROBABILITY		Probabilities distribution of all links from one input component to other components. The probability is calculated by monitoring the operational system.	A real number between [0.1] The sum of all the probabilities outgoing from one component us equal to 1.

The components directed graph is outlined using Tables I and II as described in Fig. 2.

Two external input nodes are component no' 1 and 2 which belong to the attack surface. Components 3,4,5,6 are internal. Components 7, 8 are external outputs. Arrows represent links among components. Following the graph structure formalism.

Let i be a component which activates components j and k. (The presented algorithm enables a varying number of linked components). Each link from i is assigned a value which indicates occurrence probability of the event that component i activates components j and k. For example the probability that component 2 activates component 3 is equal 0.2, while the probability of i activating component 4 is 0.8.

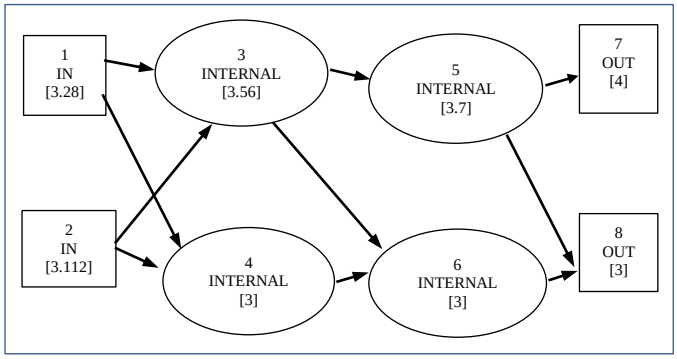

Fig. 2. Configuration Components Graph

In order to illustrate the scoring algorithm we use the graph of Fig 2 which is generated from the CMDB defined in Tables I and II. The contents of the CMDB follows in Tables III and IV.

TABLE III. CMDB – COMPONNETS TABLE EXAMPLE

Component ID	Component Type (*)	CDP	TD	CR	IR	AR
1	I	3.28	L	L	L	L
2	I	3.112	M	M	M	M
3	INT	3.56	M	M	M	M
4	INT	3	H	H	H	H
5	INT	3.7	L	L	L	L
6	INT	3	H	H	H	H
7	O	4	L	L	L	L
8	O	3	M	M	M	M

(*) I = INPUT, O= OUTPUT, INT=INTERNAL

TABLE IV. CMDB – LINKS TABLE EXAMPLE

Component ID	Component ID	Link Probability
1	3	0.5
1	4	0.5
2	3	0.2
2	4	0.8
3	5	0.8
3	6	0.2
4	6	1
5	7	0.7
5	8	0.3
6	8	1

Following an illustration of the rollback scoring process. The underlying assumption is that systems' owner is capable of estimating CDP values to external outputs since only those components have direct impact on business users. Business users are unable to assign CDP to internal component, nor to input components since they have no knowledge of the technical relationships between an internal software component on other components, nor impacts on his business. This research underlying assumption is contrary to [24] assumption who state that according to their scoring model the temporal and environmental metrics including CDP need to be specified by users with no differentiation between user interfaces and internal components. The rollback algorithm presented in this work calculates the CDP of all systems' component, based on two types of information: the CDP of the external components, and occurrence probabilities of links among all systems 'components. The rollback algorithm is performed according to the following formalization outlined in Fig. 3:

The algorithm activates a function running on graph nodes computing CDP's for all system components. The algorithm starts by computing the CDP's of the components which generate the external outputs and continues backwards until ending by computing the CDP's of the internal and external inputs. Following are the notations used by the algorithm and algorithms' logic.

Graph G represents all system components. Graph nodes represent components, arcs represent occurrence probabilities of links between couples of components.

Ci indicates component number i.

CDP (Ci) is the CDP of component Ci. CDP's are represented as real number between [0, 5] according to CVSS definitions. CDP value 0 indicates no potential damage, CDP value 5 indicates the maximal damage potential.

n indicates the number of components in graph G.

Input: A directed graph G including all Pr (i, j) assigned for all the nodes in G.
Input: CDP values assigned by system users to all external outputs. Internal and input CDP's are assigned to null.
Output: A set of computed CDP C(i) assigned to all nodes of graph G.

Method:

1. For all components of graph G. i running from n to 1.
2. While all CDP C(j) connected to C(i) as output are not equal null

$$\text{Compute CDP(Ci)} = \sum_{j=1}^{m(i)} \text{CDP(Cj)} * \text{Pr(i, j)}$$

3. Return the set of computed CDP C(i) of graph G.

Fig. 3. Algorithm for computing CDP values for graph nodes

m (i) indicates the number of components linked to component i as output nodes. Each component i might be linked to a varying number of output components.

Pr (i, j) indicates the occurrence probability of activities performed by component i to component j. The sum of all the probabilities of activities performed by component i to all output connected components equals 1.

For illustration, following the computation of CDP's of all systems components according to the algorithm. The computations according to the iterations are shown in Table V. The computed CDP's are also presented inside brackets in each component of Fig 2.

TABLE V. CDP COMPONENTS COMPUTATIONS

Step Number	Component ID	CDP	Remarks
1	7	4	Estimated by systems' owner
2	8	3	Estimated by systems' owner
3	5	3.7	4 * 0.7 + 3 * 0.3
4	6	3	3 * 1.0
5	3	3.56	3.7 * 0.8 + 3 * 0.2
6	4	3	3 * 1.0
7	1	3.28	3 * 0.5 + 3.56 * 0.5
8	2	3.112	3.56 * 0.2 + 3 * 0.8

V. RESULTS

AS presented in Table V the algorithm calculated all CDP's starting from the external output users' estimates, continuing to all internal CDP's, ending with external input components. The user estimates only two CDP's of the external outputs 7 and 8. In steps 3 and 4 the algorithm calculates the CDP's of components 5 and 6. THE calculated CDP of component 5 is 3.7 higher than component 6 which is 3. The rational is that component 6 impacts are less harmful to external component number 8, which is CDP 3, whereas component 5 has higher impacts due to impacting on component 7 which has a higher CDP of 4 according to users' estimates.

Second example of the rational implemented by the algorithm is the CDP's calculated for input components 1 and 2. Component number 1 is an external input with calculated CDP of 3.28 while component number 2 with a calculated CDP of only 3.112. The rational is that input component 2 has less impacts on external outputs since it impacts more on external output component 8 than on component 7, together with the fact that component 8 CDP is less harmful to the user than output component 7 having a higher CDP of 4.

It was illustrated that all CDP's are calculated basing on users' CDP's estimates of external outputs only, while all internal and input components CDP's are calculated by the algorithm. This illustrates the advantage of the algorithm compared to other algorithms which are based on user's

estimates, making to use of the technological characteristics of the specific environment and all relationships among components.

Questions remaining for future improvements include adding more information to the CCG. Present information includes occurrence probabilities of links, but it is reasonable to assume that varying external inputs causing varying probabilities. In such cases it might be logical to define a graph in which link probabilities depend on several external inputs, instead of relying on their average. Such a solution may be more accurate.

The algorithm computes CDP using the expected CDP's according to their corresponding occurrence probabilities. It should be said that with a minor change in the logic, CDP could be calculated according to the maximal damage potential instead of the expected potential. Such decision should be taken by business risk manager. Incorporating the CDP computed values in CVSS scoring model needs a minor modification to CVSS algorithm: using the calculated CDP's instead of the estimated CDP's for all systems' components. Using CVSS model needs no other modifications.

VI. Conclusions

This work presents a new framework of a Security Continuous Monitoring System, structure and mechanisms. The CMS uses CVSS scoring model for risk scoring operating in real time. According to the proposed model CVSS uses CDP's environmental parameters which are evaluated by the suggested algorithm, based on the technological configuration of the system, instead of CDP figures which are currently estimated using users' personal knowledge. A structure of a directed graph and scoring algorithm described and illustrated.

The model helps risk managers in estimating the organizational damages related to security risks, basing their estimates on the specific technological structure by using the algorithm. Using this model will bring more accurate estimates to vulnerability risks, thus enabling efficient risk mitigation plans and improved defense to organizations.

Further research of the model is incorporating more information in the scoring model such as detailed specifications of the configuration such as certain types of components (operation system, browsers, application, development languages etc') and modeling several relationship types among components such as varying kinds of links indicating varying relationships such as read activities, write activities and function calls. It might be reasonable to research the impacts of the varying component types on the evaluated CDP's and eventually on business risk scores.

More research is needed in supplying quantitative measures to the CVSS model. In our view CVSS model uses too many qualitative measures. At present most measures are based on users' estimates. Parameters such as TD – Target distribution may use the technological aspects of the configuration instead of users' rough estimates. Other environmental parameters such as confidentiality, integrity and availability requirements might also be based on models relating to quantifiable business damages to technological components.

References

[1] P. Mell, K. Scarfone, and S. Romanosky, “CVSS – A complete guide to the common vulnerability scoring system, version 2.0”, 2007.

[2] L. Langer, "Stuxnet: dissecting a cyber warfare weapon, security and privacy", IEEE, Volume 9 Issue 3, pages 49-51, NJ, USA, 2011.

[3] S. Tom and D. Berrett, "Recommended practice for patch management of control systems", DHS National Cyber Security Division Control Systems Security Program, 2008.

[4] A. Terje and R. Ortwin, "On risk defined as an event where the outcome is uncertain", Journal of Risk Research Vol. 12, 2009.

[5] Y. F. Nũez, "Maximizing an organizations' security posture by distributedly assessing and remeding system vulnerabilities", IEEE – International Conference on Networking, Sensing and Control, China, April 6-8, 2008.

[6] K. Dempsey, N. S. Chawia, A. Johnson, R. Johnson, A. C. Jones, A. Orebaugh, M. Scholl and K. Stine, "Information security continuous monitoring (ISCM) for federal information systems and organizations", NIST, 2011.

[7] M. G. Hardy, "Beyond continuous monitoring: threat modeling for real-time response", SANS Institute, 2012.

[8] A. Keller and S. Subramanianm, "Best practices for deploying a CMDB in large-scale environments", Proceedings of the IFIP/IEEE International conference and Symposium on Integrated Network Management, pages 732-745, NJ, IEEE Press Piscataway, 2009.

[9] M. R. Grimalia, L. W. Fortson and J. L. Sutton, "Design considerations for a cyber incident mission impact assessment process", Proceedings of the Intrnational Conference on Security and Management (SAM09), Las Vegas, 2009.

[10] I. Kotenko and A. Chechulin, "Fast network attack modeling and security evaluation based on attack graphs", Journal of Cyber Security and Mobility Vol. 3 No. 1 pp 27-46, 2014.

[11] Weintraub E., "Security Risk Scoring Incorporating Computers' Environment", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 4, 2016.

[12] I. Chowdhury and M. Zulkernine, "Can complexity, coupling, and cohesion metrics be used as early indicators of vulnerabilities?", In Proceedings of the 2010 ACM Symposium on Applied Computing, ACM, 2010.

[13] R.J. Ellison, J.B. Goodenough, C.B. Weinstock, and C. Woody, "Evaluating and mitigating software supply chain security risks", Technical report, DTIC Document, 2010.

[14] V.H. Nguyen and L.M.S. Tran, "Predicting Vulnerable Software Components with Dependency Graphs", Proceedings of the 6th International Workshop on Security Measurements and Metrics, NY, USA, 2010.

[15] S. Zhang, X. Zhang, X. Ou, L. Chen, N. Edwards, and J. Jin, "Assessing Attack Surface with Component-based Package Dependency", 9TH International Conference on network and system security, 2015, USA.

[16] S.H. Houmb, V.N.L. Franqueira, and E.A. Engum, "Quantifying security risk level from CVSS estimates of frequency and impact", The Journal of Systems and Software 83 (2010).

[17] V. Viduto, W. Huang, and C. Maple, "Toward optimal multi-objective models of network security", Survey. In: 17th International Conference on Automation and Computing, 10th, September 2011, University of Huddersfield, Huddersfield, United Kingdom, http://eprints.hud.ac.uk/22831/ retrieved April, 16, 2016.

[18] L. Wang, T. Islam, T. Long, A. Singhal, and S. Jajodia, "An Attack graph-Based Probabilistic Security Metric", IFIP International Federation for Information Processing 2008.

[19] V. Shandilya, and C. B. Simmons, and S. Shiva, "Use of Attack Graphs in Security Systems", Journal of Computer Networks and Communications, Vol. 2014.

[20] S. Northcutt, "The Attack Surface Problem", SANS Technology Institute-Security Laboratory – Defense in Depth Series, http://www.sans.edu/research/security-laboratory/article/did-attack-surface, retrieved April, 03, 2016.

[21] FIPS Publication 199 - Federal Information proccessing standards publication, "Standards for security categorization of federal information

and information systems", Department of Commerce, USA, February, 2004.

[22] E. Weintraub and Y. Cohen, "Continuous monitoring system based on systems' environment", ADFSL - Conference on Digital Forensics, Security and Law, May 19, 2015, Florida, USA.

[23] GAO – United States Government Accountability Office Report to Congressional Request, "Information security – state has taken steps to implement a continuous monitoring application but key challenges remain", July, 2011.

[24] J. A. Wang, M. Guo, H. Wang, M. Xia, & L. Zhou, "Ontology-based Security Assessment for Software Products", CSIIRW '09 Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies, 2009.