Vulnerability Modelling for Hybrid IT Systems

Attiq Ur-Rehman
Internet Commerce Security Lab
Federation University Australia
Mt Helen, Australia
attiqur-rehman@students.federation.edu.au

Iqbal Gondal
Internet Commerce Security Lab
Federation University Australia
Mt Helen, Australia
iqbal.gondal@federation.edu.au

Joarder Kamruzzuman
Internet Commerce Security Lab
Federation University Australia
Mt Helen, Australia
joarder.kamruzzuman@federation.edu.au

Alireza Jolfaei
Internet Commerce Security Lab
Federation University Australia
Mt Helen, Australia
a.jolfaei@federation.edu.au

***Abstract*—Common vulnerability scoring system (CVSS) is an industry standard that can assess the vulnerability of nodes in traditional computer systems. The metrics computed by CVSS would determine critical nodes and attack paths. However, traditional IT security models would not fit IoT embedded networks due to distinct nature and unique characteristics of IoT systems. This paper analyses the application of CVSS for IoT embedded systems and proposes an improved vulnerability scoring system based on CVSS v3 framework. The proposed framework, named $CVSS_{IoT}$, is applied to a realistic IT supply chain system and the results are compared with the actual vulnerabilities from the national vulnerability database. The comparison result validates the proposed model. $CVSS_{IoT}$ is not only effective, simple and capable of vulnerability evaluation for traditional IT system, but also exploits unique characteristics of IoT devices.**

Keywords—CVSS, IoT, vulnerability, supply chain, security.

I. INTRODUCTION

With the rapid emergence of the Internet of Things (IoT) in the recent years, concerns around its security and privacy have surged. The organised and expensive hacking campaigns and their impacts to the society are regular news in today's media. Cyber hacking tools and zero-day vulnerabilities have huge demand in cyber-criminal markets. The security vulnerabilities of IoT devices has attracted hackers to develop sophisticated tools to hack connected systems for financial and political gains. Some governments have even invested huge financial resources and efforts to breach IT security of other governments and individuals to fulfil their specific agendas [1]. In the recent years, researchers have increased efforts to devise techniques and security models to address these concerns.

To this end, various IoT security modellings were developed, such as the common vulnerability scoring system (CVSS) [2], to mathematically represent the probabilities of vulnerabilities. Such models calculate the probability of vulnerabilities in data integrity, confidentiality and availability using various factors, such as network complexity and pre-conditions required for a successful attack. Such a calculation requires a consideration of the characteristics linked to organisation's nature of business and related local factors as well as the historical impact of the vulnerabilities.

CVSS is a free and open industry standard for assessing the severity of computer system security vulnerabilities. CVSS assigns severity scores (ranging from 1 to 10) to vulnerabilities, where 1 is the least and 10 is the most severe. These severity scores allow the administrators to prioritize responses and resources to the vulnerabilities with high scores. Initially, CVSS version 1.0 was proposed in 2005; then, version 2.0 was proposed in 2007. Later, in 2015, the third version of CVSS was released that can provide vulnerability scoring for the following areas [2]:

- *Base metrics* that measures attacker's access to the vulnerable system, the complexity involved in executing the attack and the pre-conditions attached for a successful execution of the attack. It also indicates the scope and scale of the vulnerability, as well as its impact on confidentially, integrity and data availability.
- *Temporal metrics* score the characteristics that evolve over the lifetime of vulnerability.
- *Environmental metrics* evaluate the vulnerabilities based on local and environmental knowledge.

In this paper, we firstly identify the deficiencies of CVSS v3. To address these deficiencies, we propose an improved CVSS framework using CVSS v3, named $CVSS_{IoT}$. This framework fits both traditional IT and IoT networks. To analyse the proposed $CVSS_{IoT}$ framework, a simple but realistic supply chain IT system topology is selected, consisting of database servers, Windows and UNIX setup, identity and access management systems, HR, payments and other related web base systems, network devices, IoT sensors and client gadgets. To examine the security threats, these nodes were assigned real world vulnerabilities defined in national vulnerability database (NVD). The CVSS and $CVSS_{IoT}$ scores of these vulnerabilities were used to calculate link probability between connected nodes. The security of this supply chain topology was analysed using vulnerability security analyser (VSA) tool to identify the critical nodes and calculate the weakest links to target node under CVSS and $CVSS_{IoT}$ calculations. The results have confirmed that $CVSS_{IoT}$ assigns more realistic numeric weights to IoT nodes by analysing the unique characteristics of IoT system.

The remainder of the paper is as follows: Section II describes the related work. Section III outlines the deficiencies of CVSS. Section IV proposes the IoT vulnerability scoring system, named $CVSS_{IoT}$. Section V integrates CVSS v3 with $CVSS_{IoT}$. Section VI describes the tool used in this paper for security vulnerability analysis. Section VII describes the application of $CVSS_{IoT}$ to a case study. Section VIII highlights the results. Finally, Section IX concludes the paper.

II. Related Work

Many analytical cybersecurity vulnerability modellings make use of CVSS framework to assign accurate vulnerability probability to IoT devices. For example, in [3], Gallon conducted a detailed statistical analysis of the CVSS v2 framework and highlighted some diversity issues in the base and environmental vectors of CVSS. Gallon and Bascou [4] further enhanced the previous study [3] and suggested techniques to analyse the vulnerabilities using attack graph theory in conjunction with CVSS v2 framework. CVSS score were used to quantitatively analyse the attack scenarios for a system and suggested mitigation techniques which are relatively more effective. Later, Wang et al. [5] proposed an improved CVSS framework by adding the host server information similar to the server type and operating system to calculate the base matrix score. They have also highlighted some discrepancies in CVSS framework and recommended improvements for the temporal vector. Ibidapo and Zavarsky [6] conducted a detailed mathematical analysis of CVSS v2 environmental vector and highlighted some issues with CVSS v2 calculations, where in some cases it was possible to produce a negative value. This result contradicted CVSS v2 documentations and based on this finding, Ibidapo and Zavarsky argued that some results in CVSS v2 may be misleading and suggested improvements in CVSS v2.

Gao [7] suggested the use of new metrics, namely, 'absolute severity value', 'relative severity value' and 'security severity values' to enhance the calculations in the CVSS framework. This addition to CVSS enabled security administrators to evaluate the security of whole software system, instead of focusing on individual vulnerability of a component. Later, Wang [8] proposed an improvement to CVSS framework by using environmental statistics. In this model, the vulnerability calculations are linked with dynamic environmental information that results in different CVSS values with the change of environmental information. Keramati et al. [9] highlighted a downside of CVSS framework, due to its inability to properly score multistep attack on a system, since CVSS is designed to score the individual vulnerability instead of the system as a whole. Keramati et al. suggested the inclusion of a security matrix to CVSS framework, which is calculated using attack graph theory to establish the relationship in various vulnerabilities in multistep attack on a system.

Although the above proposals provide solutions to overcome CVSS shortcomings, but these have added complexity in CVSS scoring system and have limited its usage to a specific systems or environment. In addition to the added complexity and limitations, network topologies and nodes of attack graph theory need to be considered when calculating the CVSS score. These additions may suit particular topologies and may not be suitable for a range of networks, especially heterogeneous topologies of IoT networks due to differences in layers and nodes of attack graph. Also, it is not practical to pre-calculate all possible topologies in the world. Thus, linking the CVSS calculation with attack graph theory may benefit environmental circumstances, but it will affect the universal usage of CVSS framework.

In [10], Doynikova proposed methods to conduct the risk assessment based on the CVSS v3 score. The methods were able to define the risk levels, measure them and plot to environmental attack graph topologies. Zhang [11] proposed the use of conditional probability when performing the risk assessment based on CVSS v3. Aksu [12] defined the risk metrics based on CVSS v3 to calculate the risk of an IT system, similar to vulnerability score calculated under CVSS v3. The above methodologies of risk assessment were mainly proposed for traditional computer networks. It helps the administrators to better understand the risk. However, such methodologies were not designed targeting IoT systems, and hence will not work due to the unique characteristics of IoT networks [13].

In [14], Ge proposed a unique framework for automating security analysis of the IoT. This framework consists of data processing, security model generation, security visualizations, security analyser and security model updates phases. Using the framework, one can compare the severity of multiple potential attack paths and the effectiveness of specific device-level strategies deployed for different devices. This helps to prioritize which devices should be protected first. In the case study presented in [15], probabilities are assumed and numerical representation of a probability associated with devices is unexplored.

In [15], Ando proposed a theoretical mythology to analyse the risk of connected IoT cars sensors by mapping the vulnerability vector of CVSS v3 into a 5W-tree model (who, what, when, where, why). However, real-world case studies and detailed mathematical and experimental analyses are necessary before adopting such a methodology.

III. Deficiencies of CVSS

As explained in Section II, few works have considered the implementation and analysis of CVSS for IoT networks, because the CVSS framework has initially been designed for traditional IT networks. However, CVSS framework (CVSS v3) alone is not capable of analyzing the security vulnerabilities of emerging IoT systems, due to the following shortcomings.

Attack vector scoring: The description and the calculation method of attack vector score in CVSS may not be useful for an IoT system. For example, the traditional definition of remote may not be valid for an IoT node as remote may means few meters away in the IoT network. IoT sensors are usually in outer layers of a network, so attacking an IoT is consider rather easier as compare to IT nodes, which are usually firewall protected.

Attack complexity scoring: The topology of IoT networks usually differ from traditional networks. The calculation of attack complexities based on the traditional IT systems may not be suitable for IoT networks due to a variety of manufacturing designs and dedicated functionality of IoT devices. To attack an IoT system, one requires specific knowledge and understanding of the design; hence, the attack on IoT systems is considered more complex.

Human safety: The integration of IoT with our everyday life means our decisions are increasingly more dependent on IoT as compared to traditional computer systems. IoT systems are being designed to take independent decisions on behalf of humans. Sometimes wrong decision may cause collateral damaged to human life. CVSS framework was designed based on the traditional IT systems and network models. Therefore, the absence of human safety measures may mislead IoT administrators regarding the criticality of the vulnerability.

IV. PROPOSED IOT VULNERABILITY SCORING SYSTEM–$CVSS_{IoT}$

To address the deficiencies of CVSS framework, we have introduced IoT related context to CVSS metrics. The Attack vector (*AV*) and Attack Complexity vector (*AC*) are enhanced in the base and environmental metrics by redefining the "local (*L*)" and "physical (*P*)" definitions in the IoT context. An additional vector is introduced in the base and environmental metrics, named Human Safety Index (*HI*), to factor in the human safety issues linked with IoT devices. These enhancements are proposed under the name of $CVSS_{IoT}$ and details are as below. The below proposed numeric values are based on lab analysis and past experience.

A. Attack Vector for $CVSS_{IoT}$

The CVSS v3 attack vector consists of Network (*N*), Adjacent Network (*A*), Local (*L*) and Physical (*P*) values. A new Local value (L_i) is proposed for IoT devices to distinguish their scoring from traditional IT system definition. Similarly, a new physical numeric value (P_i) is proposed for the attack vector of $CVSS_{IoT}$, to distinguish the definition of Physical values (*P*) in traditional IT systems. For $CVSS_{IoT}$, the numeric values of attach vector are [$N = 0.85$, $A = 0.62$, $L = 0.55$, $L_i = 0.6$, $P = 0.2$, $P_i = 0.44$]. As it is rather easier to physically or locally access IoT devices, higher numeric weights are assigned to IoT devices values (L_i , P_i) compared to traditional devices values (*L*, *P*).

B. Attack Complexity for $CVSS_{IoT}$

The CVSS v3 attack complexity vector contains Local (*L*), and High (*H*) values. To distinguish the complexity difference between IoT and traditional IT devices, the $CVSS_{IoT}$ complexity range is refined with two additional values of M_i and H_i, where M_i represents medium complexity and H_i symbolizes high complexity to calculate the CVSS scoring of IoT devices. For $CVSS_{IoT}$, the numeric values of attack complexity vector are [$L = 0.77$, $M_i = 0.44$, $H = 0.44$, $H_i = 0.2$]. Attacking an IoT node requires specific knowledge due to its heterogeneous nature; hence, compared to traditional IT devices, lower numeric weights are assigned to IoT devices values (M_i and H_i) signifying high complexity.

C. Human safety index for $CVSS_{IoT}$

We propose the human safety index (*HI*) in $CVSS_{IoT}$ for IoT devices to address the human safety concern. *HI* consists of three values [N_i, L_i, H_i], where $N_i = 0$, $L_i = 0.44$, $H_i = 0.97$ and grouped under base metric vector as mandatory. To integrate *HI* with the existing CVSS, N_i is assigned a numeric value of 0. Therefore, setting *HI* value equates to making N_i zero under $CVSS_{IoT}$, and will ignore this factor hence it will have no impact on calculation. Along with N_i, L_i and H_i are other values of *HI* representing low and high impact. L_i is selected for IoT node when vulnerability may cause indirect threat to human safety, while H_i is selected for direct threat to human safety. $CVSS_{IoT}$ groups are listed in Table 1. Differences between CVSS and $CVSS_{IoT}$ are in bold font.

V. INTEGRATION OF CVSS V3.0 AND $CVSS_{IoT}$

$CVSS_{IoT}$ is further integrated with the existing CVSS v3 to have a single framework that works for both traditional IT and IoT devices. When the integrated formula is applied to the traditional IT systems, the vector values remain unchanged; hence, $CVSS_{IoT}$ yields the same results as CVSS v3. However, when this is applied to IoT devices, IoT defined values (symbolled as subscript $_i$ in Table 1) are assigned to the calculation, yielding IoT specific scores.

TABLE1: $CVSS_{IoT}$ POSSIBLE VALUES

$CVSS_{IoT}$ possible values			
Metric	**Metric Name**	**Possible Values**	**Mandatory**
Base	Attack Vector, *AV* Attack Complexity, *AC* Privileges Required, *PR* User Interaction, *UI* Scope, *S* Confidentiality, *C* Integrity, *I* Availability, *A* **Human Safety Index, *HI***	[*N*, *A*, *L*, ***L*$_i$**, *P*, ***P*$_i$**] [*L*, ***M*$_i$**, *H*, ***H*$_i$**] [*N*, *L*, *H*] [*N*, *R*] [*U*, *C*] [*H*, *L*, *N*] [*H*, *L*, *N*] [*H*, *L*, *N*] **[*N*$_i$, *L*$_i$, *H*$_i$]**	yes
Temporal	Exploit Code Maturity, *E* Remediation Level, *RL* Report Confidence, *RC*	[*X*, *H*, *F*, *P*, *U*] [*X*, *U*, *W*, *T*, *O*] [*X*, *C*, *R*, *U*]	No
Environmental	Confidentiality Req, *CR* Integrity Req, *IR* Availability Req, *AR* Modified Attack Vector, *MAV* Modified Attack Complexity, *MAC* Modified Privileges Required, *MPR* Modified User Interaction, *MUI* **Modified Human Safety Index, *HI*** Modified Scope, *MS* Modified Confidentiality, *MC* Modified Integrity, *MI* Modified Availability, *MA*	[*X*, *H*, *M*, *L*] [*X*, *H*, *M*, *L*] [*X*, *H*, *M*, *L*] [*X*, *N*, *A*, *L*, ***L*$_i$**, *P*, ***P*$_i$**] [*X*, *L*, ***M*$_i$**, *H*, ***H*$_i$**] [*X*, *N*, *L*, *H*] [*X*, *N*, *R*] **[*X*, *N*$_i$, *L*$_i$, *H*$_i$]** [*X*, *U*, *C*] [*X*, *N*, *L*, *H*] [*X*, *N*, *L*, *H*] [*X*, *N*, *L*, *H*]	No

The integrated formula of our $CVSS_{IoT}$ is as follows. The Base score, that is a function of the Impact and Exploitability sub score equations, is defined as [2]

if (Impact sub score <= 0)

Base score = 0,

else

Scope Unchanged:

$Base\ score = (M_i[(Impact + Exploitability), 10]),$

Scope Changed:

$Base\ score = (Mi[1.08 \times (Impact + Exploitability),$

end

and the Impact sub score (ISC) is defined as

Scope Unchanged: $ISC = 6.42 \times ISC_{\text{Base}}$,

Scope Changed: $ISC = 7.52 \times [ISC_{Base} - 0.029] - 3.25 \times [ISC_{Base} - 0.02]$,

Where

$$ISC_{Base} = 1 - [(1 - Impact_{Con}) \times (1 - Impact_{Integ}) \times (1 - Impact_{Avai}) \times (1 - Impact_{\text{HI}})],$$

and the exploitability sub score is

if IoT

$$8.22 \times AttackVector_{\text{IoT}} \times AttackComplexity_{\text{IoT}} \times PrivilegeRequired \times UserInteraction$$

else

$$8.22 \times AttackVector \times AttackComplexity \times PrivilegeRequired \times UserInteraction$$

end (1)

When calculating non-IoT node scoring, $Impact_{\text{HI}}$ would always be 0. Similarly, the non-IoT values are selected for exploitability calculations; hence, it will produce the same results as of CVSS v3 framework. However, for IoT nodes, the IoT related values of $Impact_{\text{HI}}$, $AttackVector_{\text{IoT}}$ and $AttackComplexity_{\text{IoT}}$ computes the IoT specific outcome.

VI. Vulnerability Security Analyser (VSA)

Vulnerability Security Analyser is an in-house designed and built tool to analyse risk for an IT system, which also suggests mitigation strategies. The tool works on the basis of attack graph theory where nodes represent IT assets under risk. Like any typical IT system, nodes are placed in layers. The edge nodes (interactive nodes) are easy to access and have public interaction, whereas the deeper layers are marked as inner layer nodes (Systems, Server and Data nodes). The inner nodes are hard to access and usually accessed via edge nodes. Vulnerability security analyser takes IT assets as inputs along with the probability of vulnerability between two given nodes, called link probability. The link probability is calculated using the fitness model of the graph network theory [17] as follows:

$$L_p = \frac{x_j + x_k + p(x_j, x_k)}{2}, \quad (2)$$

where L_p is the link probability, x_j is the probability of node j, x_k is probability of node k, and $p(x_j,x_k)$ is the fitness factor. Fitness factor is not the average, as average probability of two nodes may be misleading. Suppose that a node has a vulnerability score of 9 and the other node has a vulnerability of score 1. The average link probability of value 5 may misleads system administrators to delay their mitigation strategies. With the use of fitness factor as described in graph network theory [17], the average value will tilt towards the heavy weight node.

The VSA examines the link probability (L_p) of each node link and presents the risks in the form of node graph. Based on the link vulnerabilities, it identifies the most critical node and the easiest path to it from the edge node. Thus, it helps security professionals to determine the weakest link of an organisation's IT infrastructure under listed vulnerabilities. For a system administrator, to defend the system, the VSA tool helps to identify the starting point to minimise the possible impact. Once the critical paths are patched, the VSA tools will identify the next weakest link and hence forth. This helps the administrators to analyse the risk of the whole system instead of an individual server. It is recommended that administrators use their environmental knowledge along with the VSA tool.

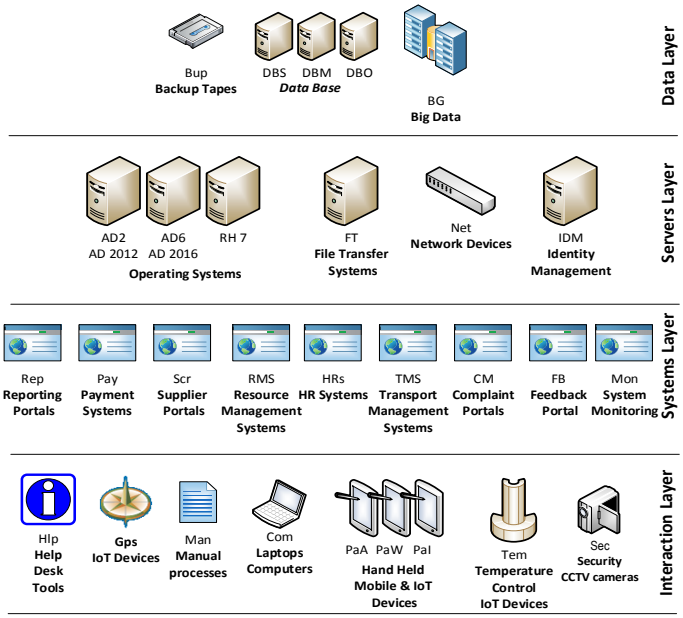


Fig. 1. Layers and nodes of selected supply chain system

VII. Case Study

In this section, we demonstrate the implementation of $CVSS_{\text{IoT}}$ on a supply chain system. In a supply chain system, temperature management is vital for maintaining temperature sensitive assets. Temperature loss can lead to food safety issues and therefore it can lower consumers' satisfaction. It has been a challenge to maintain the temperature of cold assets throughout the supply chain process, particularly during loading, unloading, transportation and in display cabinet. It is estimated that 1.3 billion tons of food is wasted every year within the EU [16]. For our analysis, we assumed a realistic supply chain system as shown in Figure 1. The four layers depicted in Figure 1 are not the network topology layers but logical layers based on physical access and human interactions to the nodes.

Data layer: Database, backup tapes and big data are in the top layer as it deals with organisational data and secrets. This layer is protected with multiple firewalls and is restrictive.

Server layer: IT servers reside in this layer. Usually, the system admin and technical staff have access to these servers via terminals and specific tools. This layer is usually firewall protected. The data of this layer is stored in data layer.

Systems layer: Typical IT solutions are configured on server layers, which communicate with data layer using servers layer. IT system admins have access via system admin tools. These system admin tools are also firewall protected.

Interaction Layer: The end users access and interact with IT solutions using end terminal and devices. The devices in this layer are least restricted and rather considered easy to access and influence.

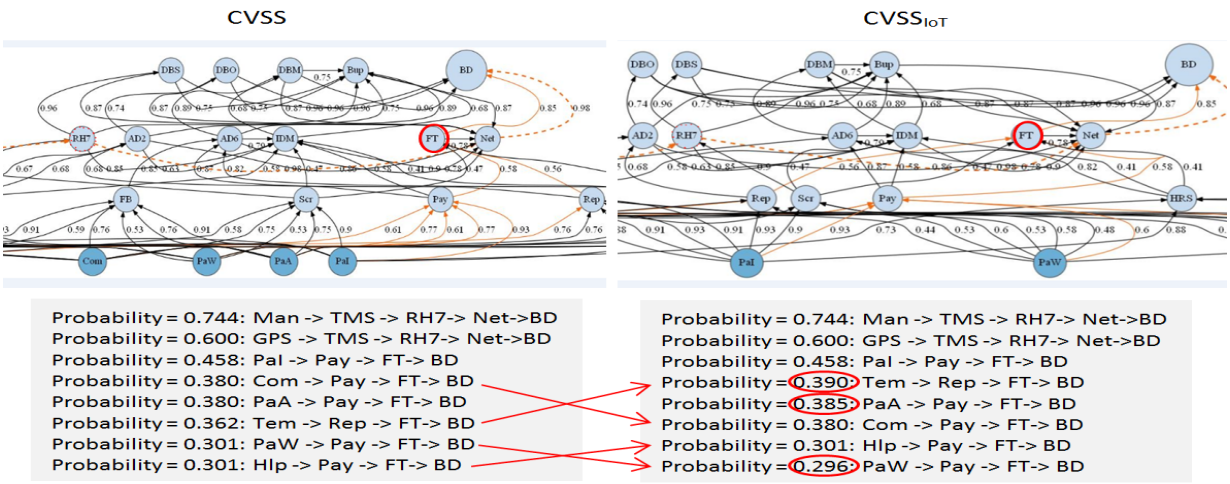


Fig. 2a. VSA analysis of selected supply chain using CVSS

Fig. 2b. VSA analysis of selected supply chain using CVSS$_{IoT}$

Each item in Figure 1 represents a type of nodes. These nodes are assigned with real world vulnerabilities defined in NVD [18]. The CVSS base score indicates the probability of the vulnerability for each node. For example, the temperature sensors (Tem) in Figure 1 are assigned CVE-2017-6798 vulnerability from VND. The Tem node is connected with various nodes in the system layer. The CVSS base score of CVE-2017-6798 is 7.8, and this is used as a node probability of the Tem node. Similarly, the probability of other connected node's vulnerabilities is numerically represented with its CVSS score. The link probability between these nodes is calculated based on these scores. The values are placed in VSA (Section VI) to identify the critical nodes and the weakest paths to the target node.

In our case study, the edge nodes are IoT devices, such as temperature sensors, GPS, mobile devices and smart CCTV cameras. Hence, vulnerability weights of all nodes are recalculated using CVSS$_{IoT}$, following equation 1 in section V. Note that in CVSS$_{IoT}$ framework, equation 1 has no impact on the vulnerability score of non-IoT nodes. Our method only updates the score of IoT nodes. The updated values are listed in Table 2.

TABLE2: REPORTED VULNERABILITY WEIGHTS

Nodes	CVSS v3	CVSS$_{IoT}$	NVD CVE-ID
Temperature sensors	7.8	8.5	CVE-2017-6798
GPS	7.5	7.5	CVE-2017-5239
Android Mobile Devices	7.8	7.9	CVE-2017-0741
Windows Mobile Devices	5.3	4.4	CVE-2017-13080
Apple Mobile Devices	9.8	9.8	CVE-2017-13832
Security CCTV cameras	6.1	8.2	CVE-2017-5367

The base vector of temperature sensor's vulnerability CVE-2017-6798 under CVSS v3 is [*AV*:*L*/*AC*:*L*/*PR*:*N*/*UI*:*R*/*S*:*U*/*C*: *H*/*I*:*H*/*A*:*H*] and scored as 7.8. When this vulnerability is accessed under CVSS$_{IoT}$, IoT specific values of $AC{:}L_i$ are selected and *HI* is assigned with value of H_i, as the temperature sensor has direct impact on human safety. The base vector under CVSS$_{IoT}$ is updated as [$AV{:}L_i$/*AC*:*L*/*PR*:*N*/*UI*:*R*/*S*:*U*/*C*:*H*/*I*:*H*/*A*:*H*/*HI*:H_i] and the computed value becomes 8.5, as shown in Table 2. The CVSS$_{IoT}$ base vectors of other IoT nodes are as below.

CVE-2017-5239: [*AV*:*N*/*AC*:*L*/*PR*:*N*/*UI*:*N*/*S*:*U*/*C*:*H*/*I*:*N*/*A*:*N*/*HI*:*Ni*]
CVE-2017-0741: [*AV*:*Li*/*AC*:*L*/*PR*:*N*/*UI*:*R*/*S*:*U*/*C*:*H*/*I*:*H*/*A*:*H*/*HI*:*Ni*]
CVE-2017-13080: [*AV*:*A*/*AC*:*Hi*/*PR*:*N*/*UI*:*N*/*S*:*U*/*C*:*N*/*I*:*H*/*A*:*N*/*HI*:*Ni*]
CVE-2017-13080: [*AV*:*A*/*AC*:*Hi*/*PR*:*N*/*UI*:*N*/*S*:*U*/*C*:*N*/*I*:*H*/*A*:*N*/*HI*:*Ni*]
CVE-2017-13832: [*AV*:*N*/*AC*:*L*/*PR*:*N*/*UI*:*N*/*S*:*U*/*C*:*H*/*I*:*H*/*A*:*H*/*HI*:*Ni*]
CVE-2017-5367: [*AV*:*N*/*AC*:*L*/*PR*:*N*/*UI*:*R*/*S*:*C*/*C*:*L*/*I*:*L*/*A*:*N*/*HI*:*Li*]

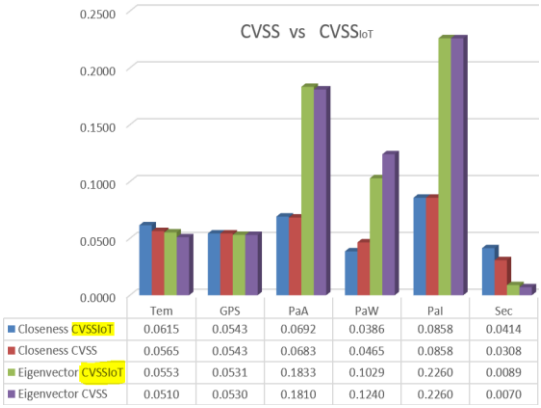
VIII. RESULTS

The link probability between connected nodes of supply chain system is calculated as defined in Equation 2 in section VI. In our experiment x_j and x_k are the CVSS and CVSS$_{IoT}$ scores of NVD vulnerabilities assigned to node *j* and node *k*, respectively. These link probabilities (L_p) are passed to VSA tool. VSA tools analysed the security of whole system by identifying possible paths to a target node, ranked them according to their severity and detected the most vulnerable node of this system.

Figure 2 is the graphical representation of results produced by VSA tool. Figure 2a depicts the results when link probabilities are calculated using CVSS scores, and Figure 2b shows the results when the probabilities are calculated using CVSS$_{IoT}$. The light blue circles (for example, IDM) are middle layer and dark blue circles (for example, PaI, PaW) are edge nodes. An attacker's target is to compromise the BD (Big Data) node. The arrows represent the possible path for an attacker to target node (BD). The shortest possible path to the BD is presented with dotted line. In Figure 2a, the VSA tool has highlighted the FT (File Transfer server, denoted by red circle) as the most critical node of the selected supply chain system. Compromising this node will make it rather easier for attacker to attack BD.

As some of our nodes are IoT devices, in our next set of analysis we used the links probability calculated using CVSS$_{IoT}$ (Figure 2b) and feed it again to VSA tool. It has again identified the FT (File transfer server) as the most critical nodes. But link paths to BD including the easiest path are now updated and re-ranked, due to different probability calculation of IoT nodes. Following Figure 2a, there are eight possible paths to BD node and 6 out of 8 are via FT node. When link probabilities are calculated using CVSS$_{IoT}$, it has re-ranked the attack possibilities for BD node as mentioned in Figure 2b.

Under $CVSS_{IoT}$, temperature sensor node (Tem) is ranked up and windows mobile devices (PaW) are moved down due to more realistic base vector values. So, the use of CVSS framework for a hybrid IT system (that is, mixed with traditional and IoT devices) may mislead system manager to develop inefficient mitigation strategies. The calculation based on proposed $CVSS_{IoT}$ provides much more realistic view of the hybrid IT system.



	Tem	GPS	PaA	PaW	Pal	Sec
Closeness CVSSIoT	0.0615	0.0543	0.0692	0.0386	0.0858	0.0414
Closeness CVSS	0.0565	0.0543	0.0683	0.0465	0.0858	0.0308
Eigenvector CVSSIoT	0.0553	0.0531	0.1833	0.1029	0.2260	0.0089
Eigenvector CVSS	0.0510	0.0530	0.1810	0.1240	0.2260	0.0070

Fig. 3. Graph theory analysis for selected supply chain mode

The supply chain system is further analysed using graph theory to validate our results. We analysed the degree centrality, closeness centrality, and eigenvector centrality of supply chain system. We used the modified sigmoid functions [19] to calculate these values. To transform the calculated values ranging from 0 to 1, the vulnerability score of the nodes is assigned to constant *k* (used in sigmoid functions). First, the values of CVSS score are assigned to the constant. Then, same calculations are repeated using the $CVSS_{IoT}$ score as constant.

$CVSS_{IoT}$ framework produces the same results in terms of above metrics as CVSS is for traditional nodes, but for IoT nodes it assigns more relative values as highlighted in Figure 3. The closeness centrality (which measures the sum of the length of shortest paths between nodes in a graph) and the eigenvector centrality (which measures heavy node connectivity to other highly connected nodes in a connected graph) have slightly high values under $CVSS_{IoT}$ for IoT nodes. In our selected supply chain model, temperature sensors (Tem), windows mobile devices (PaW) and security CCTV cameras (Sec) are IoT nodes and have higher values for closeness and eigenvector centrality when computed under $CVSS_{IoT}$, compared to CVSS v3. The above results confirm the uniqueness of IoT nodes when compared with traditional nodes. Compared to CVSS framework, $CVSS_{IoT}$ produces more accurate results for IoT systems by exploiting unique characteristics of IoT nodes.

IX. Conclusion

In this study, we proposed an extension to CVSS v3 framework to better calculate the vulnerability probabilities for IoT embedded systems. The proposed $CVSS_{IoT}$ can be applied to all network topologies, that is, traditional, hybrid and/or IoT only systems. $CVSS_{IoT}$ is analysed using VSA tools and graph theory, and compared with CVSS v3 framework. The obtained results confirms that $CVSS_{IoT}$ assigns more realistic vulnerability score to IoT nodes by considering the unique characteristic of IoT system. We plan to further expand our $CVSS_{IoT}$ analysis for more complex topologies and design an evolving vulnerability modelling for the identification of potential threats in hybrid IT systems.

X. References

[1] W. R. Marczak and V. Paxson, "Social Engineering Attacks on Government Opponents: Target Perspectives," Proceedings on Privacy Enhancing Technologies, vol. no. 2, Jan. 2017.

[2] "CVSS v3.0 Specification Document," FIRST [Online]. Available: https://www.first.org/cvss/specification-document. [Accessed: 12-Jan-2018].

[3] L. Gallon, "Vulnerability Discrimination Using CVSS Framework," IFIP International Conference on New Technologies, Mobility and Security, 2011.

[4] L. Gallon and J. J. Bascou, "Using CVSS in Attack Graphs," International Conference on Availability, Reliability and Security, 2011.

[5] R. Wang, L. Gao, Q. Sun, and D. Sun, "An Improved CVSS-based Vulnerability Scoring Mechanism," International Conference on Multimedia Information Networking and Security, 2011.

[6] A. O. Ibidapo, P. Zavarsky, D. Lindskog, and R. Ruhl, "An Analysis of CVSS v2 Environmental Scoring," IEEE International Conference on Privacy, Security, Risk and Trust, 2011.

[7] J.-B. Gao, B.-W. Zhang, and X.-H. Chen, "Using CVSS to quantitatively analyze risks to software caused by vulnerabilities," MATEC Web of Conferences, v 31, p. 16004, 2015.

[8] S. Wang, C. Xia, J. Gao, and Q. Jia, "Vulnerability evaluation based on CVSS and environmental information statistics," Intl. Conference on Computer Science and Network Technology (ICCSNT), 2015.

[9] M. Keramati, E. Keramati, "CVSS-based security metrics for quantitative analysis of attack graphs," Iccke 2013.

[10] E. Doynikova and I. Kotenko, "CVSS-based Probabilistic Risk Assessment for Cyber Situational Awareness and Countermeasure Selection," Euromicro International Conference on Parallel (PDP), 2017.

[11] H. Zhang, F. Lou, Y. Fu, and Z. Tian, "A Conditional Probability Computation Method for Vulnerability Exploitation Based on CVSS," IEEE Second International Conference on Data Science in Cyberspace (DSC), 2017.

[12] M. U. Aksu Et al "A quantitative CVSS-based cyber security risk assessment methodology for IT systems," International Carnahan Conference on Security Technology (ICCST), 2017.

[13] H. Lin and N. Bergmann, "IoT Privacy and Security Challenges for Smart Home Environments," Information, vol. 7, no. 3, p. 44, 2016.

[14] M. Ge, J. B. Hong, W. Guttmann, and D. S. Kim, "A framework for automating security analysis of the internet of things," Journal of Network and Computer Applications, vol. 83, 2017.

[15] E. Ando, M. Kayashima, and N. Komoda, "A Proposal of Security Requirements Definition Methodology in Connected Car Systems by CVSS V3," IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), 2016.

[16] J. Gustavsson, C. Cederberg, U. Sonesson, R. van Otterdijk, A. Meybeck Global food losses and food waste: extent, causes and prevention, Food and Agriculture Organisation of the United Nations (FAO), Rome (2011)

[17] G. Bianconi and A.-L. Barabã¡si, "Competition and multiscaling m evolving networks," The Structure and Dynamics of Networks, pp. 54–436, 2011.

[18] "NVD Home," NVD - 800-53. [Online].Available: https://nvd.nist.gov/. [Accessed: 12-Sept-2018].

[19] A. Haider, I. Gondal, and J. Kamruzzaman, "Social-connectivity-aware vertical handover for heterogeneous wireless networks," Journal of Network and Computer Applications, vol. 36, 2013.