Advances In Multidisciplinary & Scientific Research
A Multidisciplinary & Interdisciplinary Journal

Vol. 2, No. 1, March, 2016

Enhanced Security Testbed Threat Model Based On Parametric Equation and Multi-Level Design Approach

J.A. Ojeniyi[1], V.O. Waziri[2], A.M. Aibinu[3] and H.C. Inyiama[4]
[1,2,4]Department of Cyber Security Science
[3]Department of Mechatronics Engineering
Federal University of Technology, Minna, Nigeria
ojeniyija@futminna.edu.ng
+2348073303909

ABSTRACT

The threat model of a security testbed provides quantitative and qualitative conceptual insights into robustness of the developed testbed. Security of any digitally designed system is not guaranteed until an appropriate modelling and assessment of threat is carried out and proper mitigation is iteratively done. Several existing techniques could not handle the complexities, multi-dimensionality and layered nature inherent in threat modelling of multi-layered systems. Most approaches focus on a single level of data communication. This article aims to develop a threat model that considers different levels and dimensions of data communications over a network architecture. Parametric-based equation and data flow diagrams were used in the modelling. The model was iteratively assessed and evaluated to ensure conformity with pre-defined security requirement for the testbed.

Key words: *computer network, testbed, threat model, data flow diagram*

Aims Research Journal Reference Format:
J.A. Ojeniyi, V.O. Waziri, A.M. Aibinu & H.C. Inyiama (2016): Enhanced Security Testbed Threat Model Based On Parametric Equation and Multi-Level Design Approach. Vol 2, No. 1 Pp 87-100.

1. BACKGROUND TO THE STUDY

Security of any digital system or design is not guaranteed until an appropriate threat model is developed. The security of computing systems is not based on assumptions or vendor's claims. Potential threats and vulnerabilities at design-stage and execution time must be put into proper quantitative and qualitative assessment of systems' security requirements. In order to give formal specification of security requirements of computing systems, threat modelling approach is generally used (Myagmar, 2005). The first formal approach to design-level software security modelling was done by the work of (Xu & Nygard, 2005). The properties and inconsistency behaviours between software components were verified in their work. As a result, design-level vulnerabilities were mitigated to a reasonable extent.

The focus of earlier researchers on software-based threat modelling coupled with an increasing rate of storage security breaches necessitated work in other areas. The work of (Hasan, Myagmar, Lee, & Yurcik, 2005) was targeted towards proactive protection of storage systems. Domain-specific modelling approach was used. It is based on two different processes. The first, consideration was given to security principles like confidentiality, integrity, availability and authentication and second, the data lifecycle model was used. In order to take modelling of threats from design level to execution time, Wang, Wong, & Xu (2007) focused on runtime threat modelling. Unified modelling language sequence diagrams were used to show the consistency of threats at design stage and at runtime. This serves as a guide to code implementation and security testing of such code.

Threat analysis and modelling cannot be limited to qualitative description alone. Quantitative description of security models will help to give discrete measures to system threats. The contribution of (Khan & Hussain, 2010) gives various quantification models that can be used for mathematic or statistical analysis of system security issues.

2. STATEMENT OF PROBLEM

Existing researches in the literature worked mainly on software-based threats. Little attention was given to hardware-based threats. In addition, there are other dimensions to threat modelling that have not been fully explored such as asset-centric and attacker-centric threats. Essentially, there is persistent problem of threat relations and hierarchy. This problem has caused dependency threat challenges in which one threat depend on the other while another threat is independent of the other. If a relation functions could be defined within various threats categories at different hierarchies, then threat dependency problems will be solved.

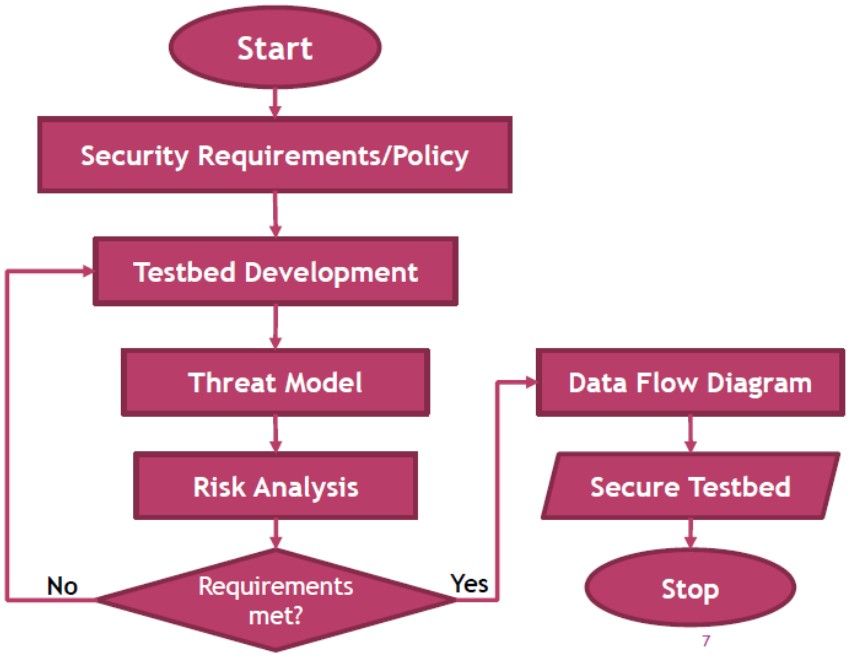
3. OBJECTIVE

In mitigating the dependency threat problem, this research work was carried out in order to develop a layered and hierarchical threat model that will consider various nodes (signifying threat elements) at different layers of data flow. Multi-layered data flow diagram was also used to model the flow of data across several layers of developed system.

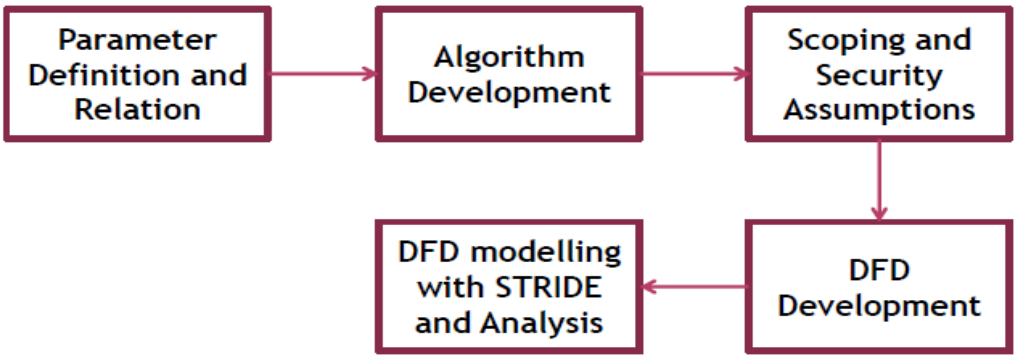
4. METHODOLOGY

4.1 The Research Design

The methods employed in this research work followed the following stages. The security requirements were first specified and policy formulated as will be required of ideal computer network setup. Based on these requirements, the testbed for the simulation of network scenario was designed and developed. Threat model for validating the security of the developed testbed was then formulated. Iteratively, this model was assessed and evaluated to ascertain its conformity with pre-defined security requirements and policy. Then data flow diagram was used to model different layers of abstraction of the pre-defined security conformed testbed. The research design is depicted by the Fig. 1.



(a) flowchart



(b) block diagram

Fig. 1: Flowchart and block diagram for multi-level threat modelling

4.2 Security Requirement and Policy Formulation

Based on the local network area setup for prototypic implementation of the research design, security hardware like hardware firewalls were used to secure the network from external aggression and to shield the data centre from both internal and external attackers. Trust boundaries were also setup to allow free flow of traffic within the organizational users. Layer 2 and layer 3 authentication of frames and packets were respectively provided by the implementation of hardware-based usage of firewalls. At various internal terminals, user-level authentication and authorization was also ensured through the use of password and biometric features (finger print and facial capturing).

4.3 Testbed Design and Development

VMware Esxi was installed directly on the Hp ProLiant ML 110 G7 server so as to make optimal resource utilization available to the guest machines on the VMware Esxi. Guest machines are the virtual operating systems or software installed on the VMware Esxi among which is Windows 8.1. In order to set up the testbed, Graphical Network Simulator 3 (GNS3) and Virtual box were installed on the Windows 8.1.

The GNS3 was used in this research work to set up the testbed. The GNS3 simulated testbed contains the following network devices: a router which was labelled as Edge-Router, two firewalls labelled as Firewall1 and Firewall2, layer 2 switch labelled as SW1 and layer 3 switch labelled as cloud1. The live Internet Operating Systems (IOS) of these devices are contained in the QEMU and dynamips of GNS3. The network terminals are ADMIN_PC, Workstation1, Workstation2, Database Server and Web/portal Server. The live operating systems for the network terminals were safely housed in virtual box.

VMware Esxi cannot be accessed directly on the server but through a remote client system. Four systems were used as remote login systems to the VMware Esxi on the server.

5. RESULTS AND DISCUSSION

5.1 Threat Model definition

In the work of (Ojeniyi, Waziri, Aibinu, & Inyiama, 2016), definitions of multi-layer threat model and the hierarchical threat relations were given. In this research, the work was extended by modelling the relations with parametric equations.

For quantitative analysis of the element histories of software centric threat consists of probability of vulnerability denoted by $P_V(t)$ and probability of threat denoted by $P_T(t)$ where t signifies time t, asset centric threat consists of probability of risk, $P_R(t)$ and asset reliability denoted by by $R_{EL}(t)$, attacker centric threat consists of probability of an attack denoted by $P_A(t)$ and consequence of an attack, $C(t)$. The relations were modelled from equation (1) to (15).

The three categories of threats modelled in this work are software centric, asset centric and attacker centric threats. The threat categories were modelled using equations (1) to (15). Particularly, total software centric threat is shown in (12), total asset centric threat in (13) and total attacker centric threat in (14). The overall total threat (TT) is modelled in (15).

$$P_T(t) = \frac{T(t)}{T} \quad (1)$$

$$P_V(t) = \frac{V(t)}{V} \quad (2)$$

When the occurrence of threats and vulnerabilities are not independent, then the equations (1) and (2) translates into (3) and (4).

$$P_{T/V}(t) = \frac{p_{V \cap T}(t)}{P_V(t)} \quad (3)$$

$$P_{V/T}(t) = \frac{P_{T \cup V}(t)}{P_T(t)} \tag{4}$$

If the reliability of the asset is known, then equation (1) can be computed as in (5).

$$P_T(t) = P_V(t) * R_{EL}(t) \tag{5}$$

$$R_{EL}(t) = e^{-ft} \tag{6}$$

Where *f* is given as the failure rate.

$$P_R(t) = P_V(t) * P_T(t) * C(t) \tag{7}$$

$$P_A(t) = e^{-\lambda t} * \frac{(\lambda t)^n}{n!} \tag{8}$$

Where λ is the inter-arrival rate of attacks and *n* is the number of attacks occurred in time interval *t*.

$$C(t) = \frac{P_A(t)}{P_T(t)} \tag{9}$$

The total threats (TT) across the three threat categories is given as follows:

$$TT = \sum_{1}^{n} T_C \tag{10}$$

$$TT = TT_{SW} + TT_{AS} + TT_{AK} \tag{11}$$

$$\begin{aligned} TT_{SW} &= P_V(t) + P_T(t) \\ &= \frac{V(t)}{V} + \frac{T(t)}{T} \end{aligned} \tag{12}$$

$$\begin{aligned} TT_{AS} &= P_R(t) + R_{EL}(t) \\ &= P_V(t) * P_T(t) * C(t) + e^{-ft} \end{aligned} \tag{13}$$

$$\begin{aligned} TT_{AK} &= P_A(t) + C(t) \\ &= e^{-\lambda t} * \frac{(\lambda t)^n}{n!} + \frac{P_A(t)}{P_T(t)} \end{aligned} \tag{14}$$

$$TT = \frac{V(t)}{V} + \frac{T(t)}{T} + P_V(t) * P_T(t) * C(t) + e^{-ft} + e^{-\lambda t} * \frac{(\lambda t)^n}{n!} + \frac{P_A(t)}{P_T(t)} \tag{15}$$

5.2 Threat assessment algorithms

Threat assessment algorithm is the algorithm that is used to generate the model of the threats or vulnerabilities inherent in your developed testbed based on layers and hierarchy. The algorithm is shown in Table 1.

Table 1: Algorithm for threat assessment

	Input: *nodes, attributes*
	Output: *attack damage*
1:	*Initialize* $n = 1$
2:	*If node* N_n *communicates with* N_{n+1} *then*
3:	*Compute node hierarchy*
4:	*Special equipment consideration*
5:	*If special equipment needed is true then*
6:	*Compute Location attribute module*
7:	*Compute Time attribute module*
8:	*Compute Semantic/Context attribute module*
9:	*Compute Logic attribute module*
10:	*elseif special equipment needed is false then*
11:	*Compute Location attribute module*
12:	*Compute Time attribute module*
13:	*Compute Logic attribute module*
14:	*endif: return attack node attribute*
15:	*Compute attack cost module {*
16:	*Read attack node attribute*
17:	*Attack cost* $= TT'_{SW} + TT'_{AS} + TT'_{AK}$ *}*
18:	*Compute Attack probability*
19:	*If attack is software centric then*
20:	$Attack\ prob = \frac{V(t)}{V} + \frac{T(t)}{T}$
21:	*elseif attack is asset centric then*
22:	$Attack\ prob = \left[\frac{V(t)}{V} * \frac{T(t)}{T} * C(t)\right] + e^{-ft}$
23:	*elseif attack is attacker centric then*
24:	$Attack\ prob = e^{-\lambda t} * \frac{(\lambda t)^n}{n!}\left[1 + \frac{T}{T(t)}\right]$
25:	$Compute\ attack\ damage = \frac{Attack\ prob}{Attack\ \cos t}$
26:	$n = n + 1$
27:	*goto line 2:*
28:	*End.*

5.3 Scoping and security assumptions

This constitutes the basis for threat declaration. Due to the security controls of the simulating testbed for authentication and availability (implemented in the firewalls), the following assumptions are necessitated:

i. Layer 2 Authentication (MAC spoofing)
ii. Layer 3 Authentication (IP spoofing)
iii. Limited Data Store (DoS Possibility)
iv. Database Access Authentication (SQL injection)

Anything against these assumptions constitute a threat otherwise not a threat.

5.4 Threat model data flow diagram for simulating testbed

The data flow diagram (DFD) threat models for simulating testbed particularly showing the processes and threats involved are shown according to level of abstraction from Figure 2 to 6.

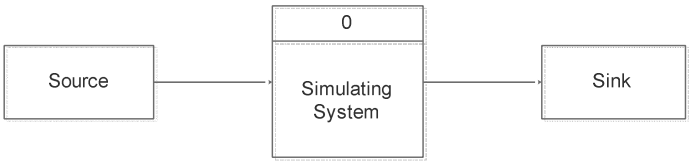


Figure 2: Context Data Flow Diagram for Simulating Testbed

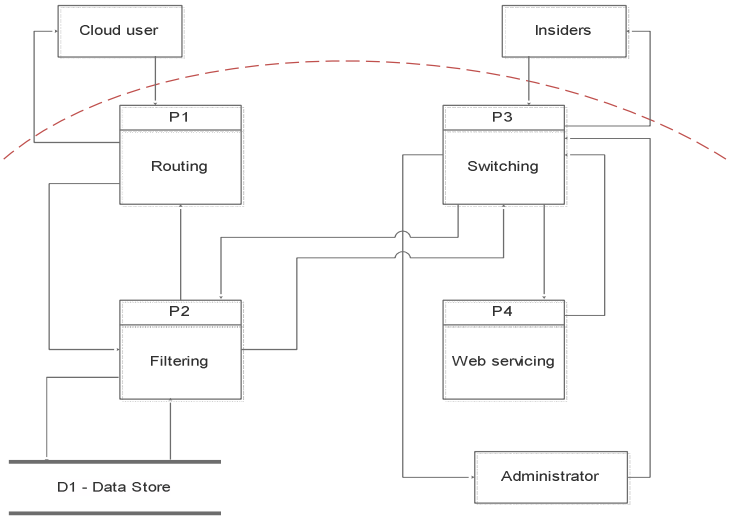


Fig. 3: Level-0 Data Flow Diagram

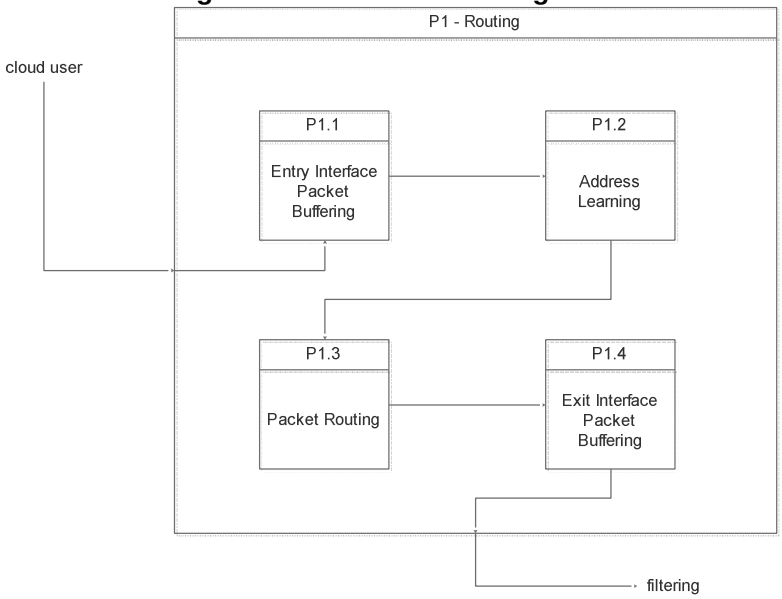


Fig. 4: Level-1 Routing Data Flow Diagrams for sub-processes in P1

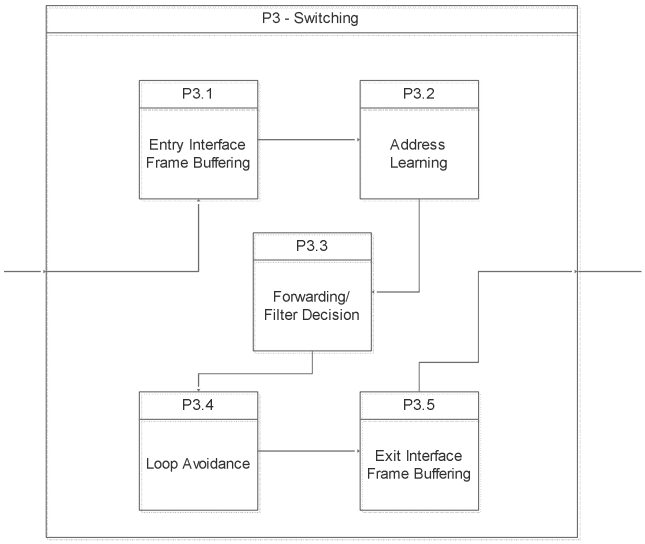


Fig. 5: Level-1 Switching Data Flow Diagrams for sub-processes in P3

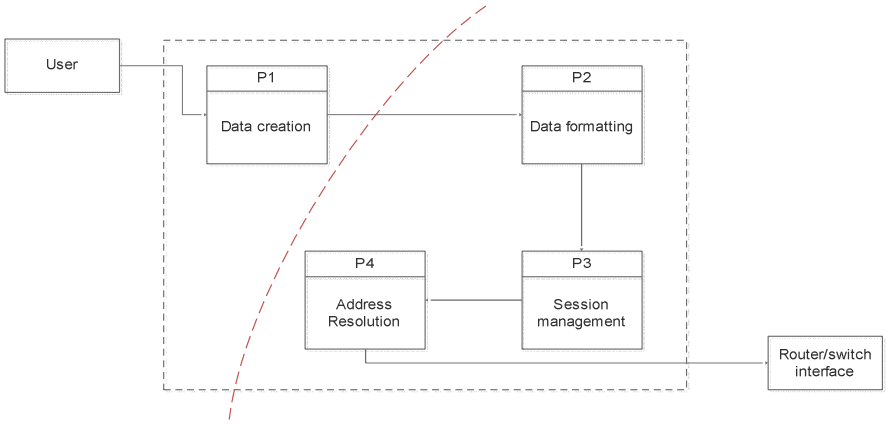


Fig. 6: Levcl-1 cloud/insider user data flow diagram

5.5 DFD Threat Modelling with STRIDES

In order to convert DFD model to quantization model, STRIDE-based approach is used because it has equivalent in the security pillars of data communications as in Table 2. The word 'STRIDE' is an acronym whose meaning is given as follows:

STRIDE means

- **S = spoofing**
- **T = tampering**
- **R = repudiation**
- **I = information disclosure**
- **D = denial of service**
- **E = elevation of privilege**

Table 2: STRIDE mappings to security pillars

Applicable Threat Category	Security Pillars
Spoofing	Authentication
Tampering	Integrity
Repudiation	Auditing
Information disclosure	Confidentiality
Denial of Service	Availability
Elevation of privilege	Authorization

The Procedures STRIDE-based DFD threat modelling are given as follow:

- **DFD elements mapping to applicable threat categories**
- **Threat Analysis**
- **Threat elicitation**
- **Threat documentation**

5.5.1 DFD element mapping to applicable threat categories and analysis

Generic DFD element mapping

Generic DFD element susceptible categories are given in Table 3.

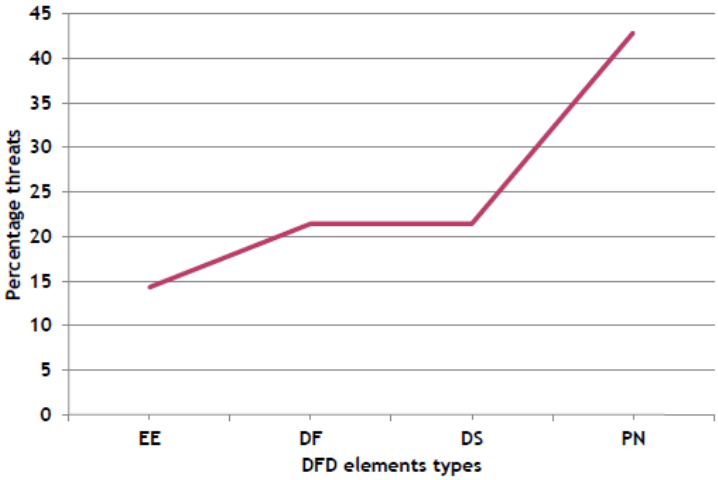
Table 3: Generic DFD susceptible category mapping

DFD element types	Meaning	Applicable threat categories						TOTAL
		S	T	R	I	D	E	
EE	System users	1	0	1	0	0	0	2
DF	Network traffic	0	1	0	1	1	0	3
DS	Database points	0	1	0	1	1	0	3
PN	Active software components	1	1	1	1	1	1	6
TOTAL		2	3	2	3	3	1	

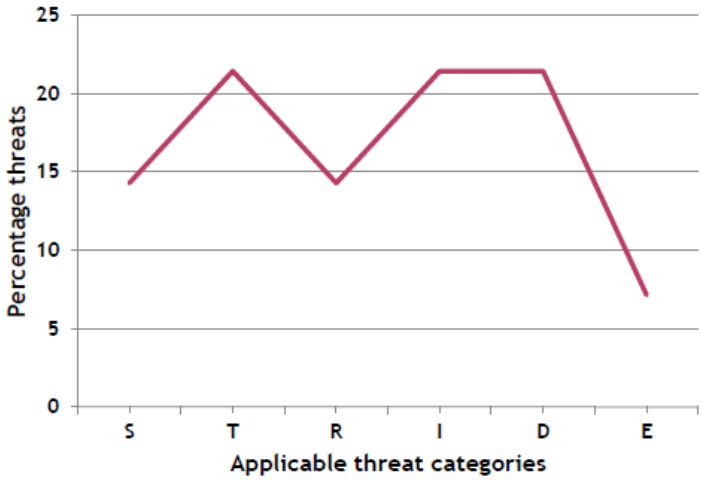
KEY:
EE=external entities DF=data flows DS=data storage points PN=processing nodes
S=spoofing T=tampering R=repudiation I=information disclosure D=denial of service
E=elevation of privilege

NOTE: Applicable threat categories are denoted by "1" while inapplicable threats by "0".

The elements of generic DFD were analysed in Fig. 7(a). It shows highest susceptibility at processing nodes. The analysis of applicable threat categories were shown in Fig. 7(b).



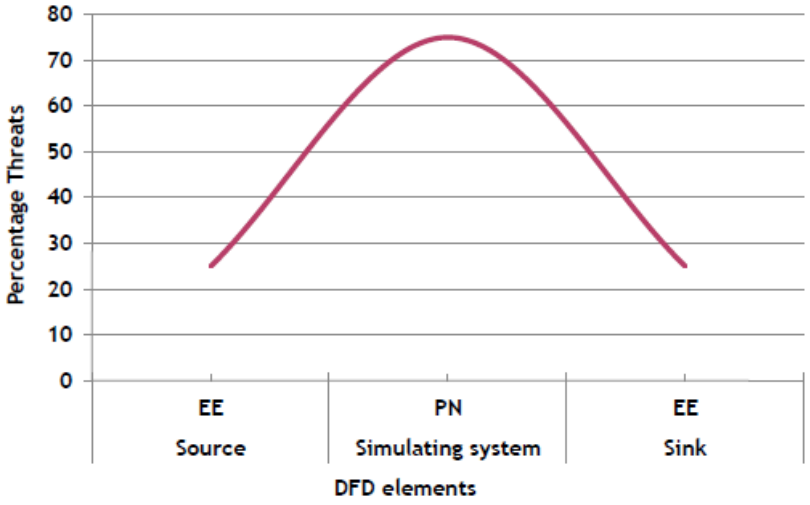
(a)



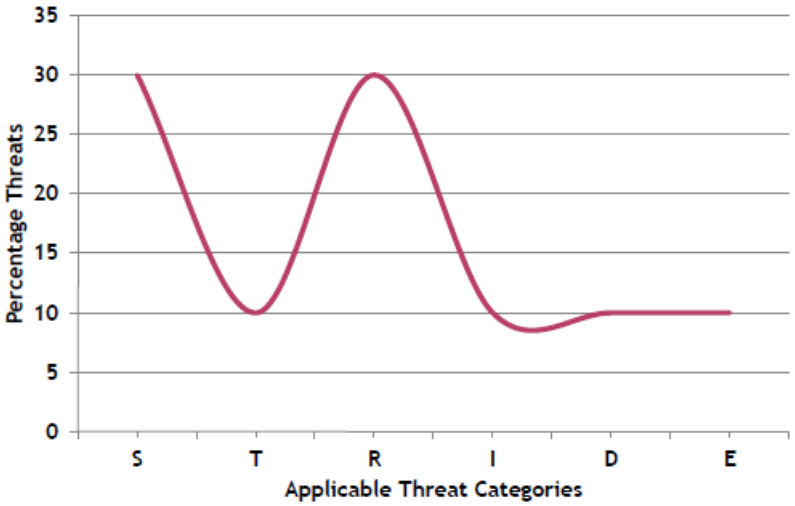
(b)

Figure 7: Generic element types and applicable threat categories

In the analysis of context DFD, the simulating processing nodes show the highest percentage threats as displayed in Fig. 8(a). As seen in Fig. 8(b), spoofing and repudiation threats show highest percentage.



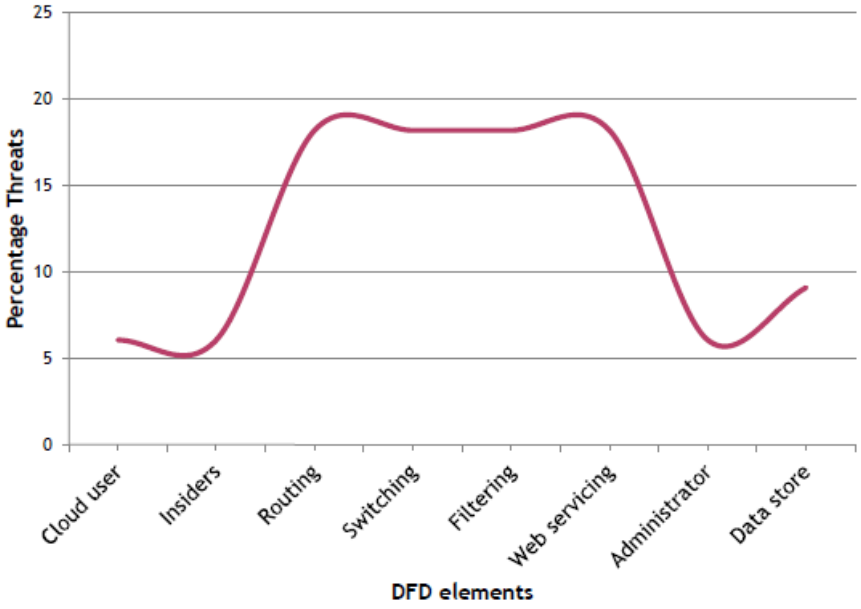
(a)



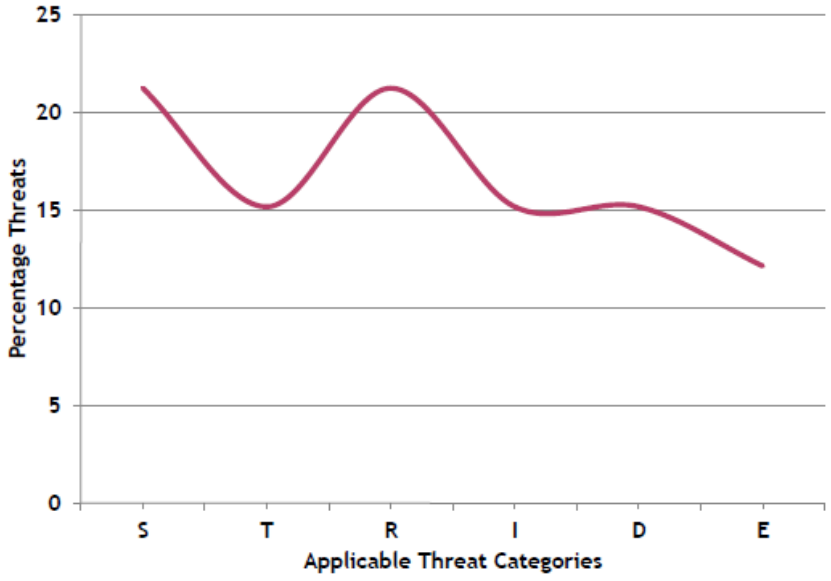
(b)

Fig. 8: Context DFD element types and applicable threat categories

In Fig. 9(a), the processing nodes of routing, switching, filtering and web servicing clearly show higher percentage of threats. In Fig. 9(b), the applicable threat categories with high threat percentage are spoofing and repudiation.



(a)



(b)

Fig. 9: Level-0 DFD element types and applicable threat categories

5.5.2 Threat elicitation

The threats were elaborated in an hierarchical method showing parent-child relations, as shown in Fig. 10.

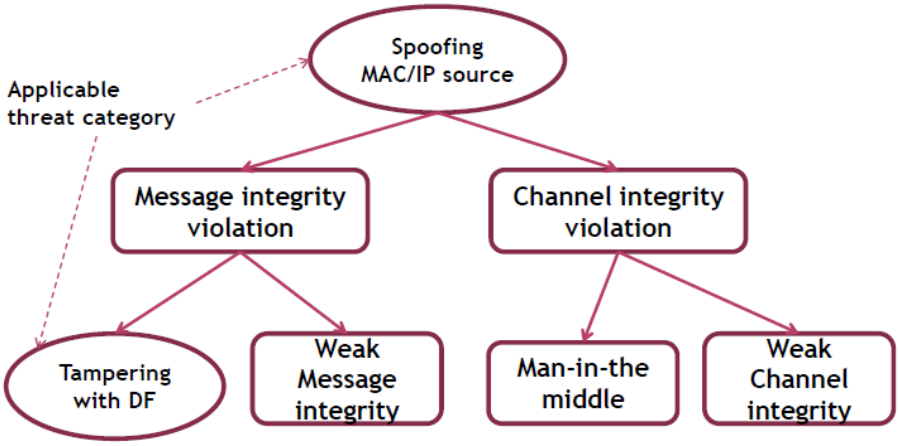


Fig. 10: Elicitation of applicable threats

6. CONCLUSION

This particular technique to threat modelling of computer network has provided layered and hierarchical-based approach which is encompassing. Three major threat categories were taken into consideration. Even though the mathematical-based model and data flow diagram were developed it is recommended that further assessment and validation be carried out on the model and the diagram. The major contribution of this research work is the formulation and development of multi-level threat model for validating the security of layered-based testbed model.

7. FUTURE WORK

In order to address non-parametric aspect of the security threat model, the direction of the future work is the use of adaptive regression of model. Based on the quantification of the flow of data, non-parametric regression model will be developed.

REFERENCES

1. Hasan, R., Myagmar, S., Lee, A. J., & Yurcik, W. (2005). Toward a threat model for storage systems. In V. Atluri (Ed.), *Proceedings of the 2005 ACM workshop on Storage security and survivability - StorageSS '05* (p. 94). Alexandria, VA, USA: ACM New York, NY, USA. http://doi.org/10.1145/1103780.1103795

2. Khan, M. A., & Hussain, M. (2010). Cyber Security Quantification Model. *Bahria University Journal of Information Amp; Communication Technologies VO - 3*, (1), 39. http://doi.org/10.1145/1854099.1854130

3. Myagmar, S. (2005). Threat Modeling as a Basis for Security Requirements. In V. Atluri (Ed.), *In StorageSS '05: Proceedings of the 2005 ACM workshop on Storage security and survivability* (pp. 94–102).

4. Ojeniyi, J. A., Waziri, V. O., Aibinu, A. M., & Inyiama, H. C. (2016). Layered and hierarchical approach for modelling multidimensional design threats 1,2,4. In O. B. Prof. Longe, D. I. M., O. Prof. Adekunle, & R. Dr. Jimoh (Eds.), *iSTEAMS Cross-Border Multidisciplinary Conference on Addressing Human-Centred Challenges Through Multidisciplinary Innovations and Inter-tertiary Collaborations* (pp. 279–288). Accra, Ghana: iSTEAMS.

5. Wang, L., Wong, E., & Xu, D. (2007). A threat model driven approach for security testing. In *Proceedings of the Third International Workshop on Software Engineering for Secure Systems* (p. 10). http://doi.org/10.1109/SESS.2007.2

6. Xu, D., & Nygard, K. (2005). A threat-driven approach to modeling and verifying secure software. In D. Redmiles (Ed.), *Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering - ASE '05* (p. 342). Long Beach, CA, USA: ACM New York, NY, USA. http://doi.org/10.1145/1101908.1101965