Towards a Secure Fog-Computing Cyber Space: A Bayesian Game-Theoretic Risk Management Framework

David Akinwumi[1, *], Arome Junior Gabriel [2], Raphael Olufemi Akinyede[3], Samuel Adebayo Oluwadare[4], Boniface Kayode Alese [2]

[1] Computer Science Department, Adekunle Ajasin University, Akungba-Akoko, Nigeria.
[2] Cyber Security Department, Federal University of Technology, PMB 704, Akure, Nigeria.
[3] Information Systems Department, Federal University of Technology, Akure, Nigeria.
[4] Computer Science Departm ent, Federal University of Technology, Akure, Nigeria.

ajgabriel@futa.edu.ng

Abstract. Cyber security is one of the most difficult and fast-growing concerns to-day's enterprises are focusing on. The practice of reducing potentially dam-aging and unknown events that pose a danger to cyber security is known as cyber security risk management. The Game Theoretic Approach is a popular cyber security risk or threat management strategy (GTA). This study provides a paradigm for cybersecurity risk or threat handling based on a game-theoretic approach to Fog computing, which will encourage proactive cyber risk management and improve cyber-operational efficiency/effectiveness. The method is written in such a way that the PyQt4 framework acts as a shield for the Fog server, performing inline packet inspection and logging any malicious packets to the console and a database on the server using Snort. The study proposes a Bayesian game model for risk management in the cyber domain.

Keywords: cyber security, cloud computing, fog computing, risk management, game theory.

1. Introduction

Cyberspace presents enormous opportunities for socioeconomic development across the globe [1]. Cyber security is a very difficult issue of concern that has posed a great challenge to both corporate organizations and governments of nations. The task is made more difficult by the complex and diffuses nature of the threats and the inability to frame an adequate response in the absence of tangible perpetrators of cyber-crimes [2]. The fast growth of Information Technology and the relative ease with which applications can be customized has increased the use of cyberspace. The implication of this increase is that probability of disruption through threats and vulnerabilities has also grown with the rise in the number of users, which calls for initiation of measures to improve the cyber security. The growing threat of cyber-attacks is also on the increase. The damage could be immense and many countries are taking proactive steps to develop capabilities and build capacities for defending themselves [3]. Many countries are also taking offensive actions in cyber-space because attacks on critical infrastructure such as shutting down of power systems; water supply and so on are of serious concern, especially with respect to national security [4].

Game theoretic approaches have been introduced as a useful tool to handle tricky net-work and cyber-attacks [5]. However, game theoretic approach has not been fully applied in a Fog-based environment. Fog is the architecture that extends the service offered by Cloud to edge devices. A typical architecture of Fog is shown in Figure 1.

The cloud-layer, the fog unit, and the client segment are the three components that make up this system.

The cloud is extended by the Fog layer. This consists of servers that are closer to the clients and are localized (Applications). These servers utilize a proactive strategy to predict the demand for information from mobile clients and store the most desired material. In-between the cloud and the application layer, the Fog layer acts as a bridge. Using IP core technology, numerous Fog servers can indeed be connected to one another in the Fog layer [6].

In this current paper, a Fog-based game-theoretic model for cyber security risk management system is presented. The system uses Bayesian game technique to model the interaction between the attackers and the defenders.

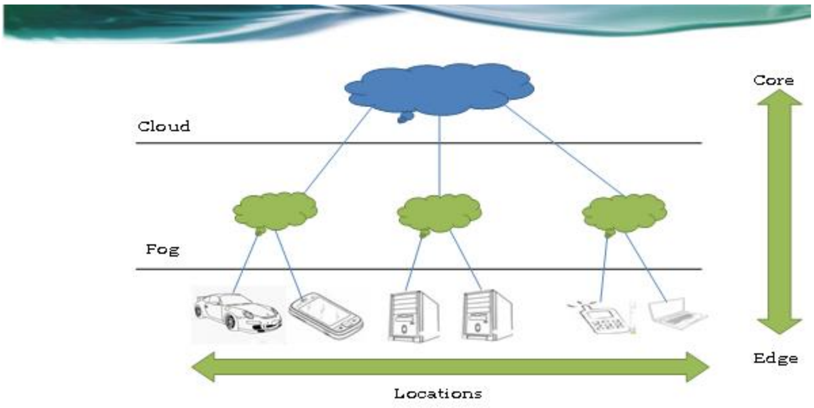


Figure 1. Cloud evolution [7].

2. Related works

The following researchers that have applied GTA method to cyber security risk maagement were reviewed so as to develop a cyber-security risk management technique that is able to reduce or eliminate some of the challenging problems of existing techniques using a competent game theoretic approach. Also, the limitations of the concluded works of other authors were addressed. The salient points of these works are documented as follows;

[8] presents an incentive-based model and inference of Attacker Intent, Objectives, and Strategies (AIOS). The development of a game-theoretic AIOS formalization that can capture the inherent interdependency between AIOS and defender aims and approaches in such a way that AIOS can be deduced automatically. However, this research did not address model-level inference accuracy analysis and sensitivity analysis, which can model and anticipate the impact of inadequate data, and uncertainty.

The authors of [9] suggested a technique that uses four distinct dimensions to provide a comprehensive classification of network and computer assaults. The technique aids in the improvement of computer and network security, as well as the uniformity of attack description language. The assault vector, which is utilized to classify the attack, is the first dimension. The attack's target is classified in the second dimension. The vulnerability categorization number is the third dimension. The effects associated are highlighted in the fourth dimension. Various layers of data are provided inside each dimension to offer assault data. This study was unable to detect a blended threat, which is another approach shortcoming. Besides, the lack of vulnerability information, which prevents information from being captured to help protect a system from attacks is another shortcoming.

A stochastic game for network security with interconnected nodes is provided in [10]. The absence of effective frameworks on the conduct of a rational hacker, as well as intrusion detection, prompted this research. The researchers investigated a stochastic game theoretic method for combating security issues, modeling an attacker and a defender in a two-player game over a network of nodes with associated security assets and vulnerabilities. Weighted directed graphs were used to model the network, with the edges indicating the influence between the nodes. However, because nodes are interconnected, if one is compromised, the remaining nodes' important security assets and weaknesses would not be the same, resulting in complicated system dynamics.

A Game-Theoretic Assessment of Defense and Attack in CPNI was presented by the authors in [11]. Game theory has been applied in the study of dynamic relations involving attackers and defenders in ensuring security, but not in sophisticated cyber-physical networks. The authors offer some fresh insights on CPNI's survivability and optimum resource allocation under a variety of costs and goal valuations. However, the investigation of the interdependent coupling impact for both the virtual and real constituent parts in the CPNI is not included in this approach.

The authors of [12] suggested a Deterministic Stochastic Game-theoretic Modelling (DSGM) approach for analyzing computer network security as a non-zero-sum stochastic game. The research was inspired by the difficulty of existing methodologies to give analytical tools and algorithms whose results may be used to make decisions and forecast attacker behavior. This research cannot forecast or analyze how attackers may exploit vulnerabilities.

The authors in [13] created a framework for modeling attacker-defender interaction as a zero-sum stochastic game to present a quantitative method for studying network security. This study was motivated by the inability of existing techniques to forecast attackers' set of plays and feasible countermeasures. The model couldn't really foretell how attackers leverage loopholes or analyze attacker behavior, which is a drawback of the research.

[14] presents an Attack Tree-Based Integrated Model for the Threat and Security Appraisal of VANET Using Game Theory and Fuzzy Logic Concepts. Since the conventional threat and security evaluation method of VANET fails to perform properly because it is entirely based on ideological views and does not reflect any reality conditions, the Vehicular Ad-hoc Network (VANET) confronts a lot of research hurdles in terms of security. In order to analyze the assault and defense equilibrium, the authors used game theory and fuzzy logic. Their approach has a flaw in that it did not examine the assets in order to do a risk comparison study.

[15] provided a GTA for sharing cyber-threats information among multiple businesses in order to increase the rate of vulnerability discovery while keeping costs low. Despite its benefits, sharing cyber threat information comes at a cost and comes with risks. However, the research is limited to only two users, and no consideration is given to heterogeneous vulnerabilities, heterogeneous players, or inadequate data. Furthermore, the game model's theoretical predictions were not compared to actual statistics on cyber-threat information exchange.

The author of [16] proposed a game theoretic and trust model to assess the risk of shifting important IT assets to the public cloud, using a user viewpoint to model costs and benefits functions for both the user and the attacker. However, because their study focused on a limited number of items, the model may be expanded to include additional assets, actions, and players.

Another study suggested a game theoretic attack-defense tree structure for evaluating risk priority of SSL SYN attacks using VANETs in [17]. Only the SSL SYN attack is considered in this study, and this is a restriction of their research. Other sorts of threats in VANETs were not explored, resulting in security concerns in other areas.

In [18], the authors developed a novel approach for assessing the danger of coordinated cyber-physical attacks on power systems. Between the attacker and the defense, a two-player zero-sum probabilistic game was devised in which each player strives to maximize their benefits on the opponent's optimum strategy. The optimum shedding technology is created to calculate the least cost of shed load in order to evaluate their rewards. However, the research focused solely on coordinated cyber-physical attacks on power infrastructures. Unorganized attacks can have serious implications in real-life settings.

In [19], the authors developed a Monte Carlo simulation technique for ensuring the control of organizational network access. The simulation involves series of iterations that involves four fundamental stages of selection, expansion, rollout as well as updating. Granting or not granting access to a given network therefore depends on results obtained from the most rewarding node during iterations. However, the research focused solely on coordinated cyber-physical attacks on power infrastructures. Unorganized attacks can have serious implications in real-life settings.

In [20], the authors developed a GTA strategy to managing cyber security risk. The authors used a software-based method known as the Cyber Security Game (CSG) to detect and reduce a system's cyber risk, as well as to discover the best cost-effective protection methods for protecting an IT system. The risk score is produced by integrating the likelihood of successful attacks with the results of a mission model approach that calculates the implications of cyber disasters. CSG's performance, on the other hand, is limited by the models which it has to operate with. If they're wrong, the output will be wrong as well.

In view of the benefits and contribution of GTA to research in cyber security risk management, this work studies Bayesian games and how the game can be used to better model cyber security risk management in a more accurate way so as to reduce or eliminate the reported limitations of existing approaches.

3. The proposed fog-based cyber security risk management system

Figure 2 depicts a conceptual diagram of a cyber security threats management system. The system is divided into two phases: risk appraisal and risk reporting. Following the prioritization of risks, the third phase is the allocation of resources to minimize the risk or the risk's consequence. If a risk is less than an acceptable level, it is left untreated and accept-ed as is. Otherwise, risks which do not fit into the accept bracket can be managed by implementing countermeasures to decrease them to a tolerable level, or by rejecting them and employing remedies to avoid the detected issues. Furthermore, the risks may be transferred to third parties via insurance. After the risk has been mitigated, the fourth stage is to assess the success of the risk mitigation methods.

3.1. Risk assessment

Identification of assets to be safeguarded, risks to those assets, and possibility of threat occurrence are all part of risk assessment. It also reveals vulnerabilities that can be abused, losses that could occur as a result of an attack, and security measures that are in place or could be implemented. The risk assessment process is split into two parts: risk analysis and risk evaluation.

3.2. Risk analysis

The probability of dangers and the size of their implications are determined at this stage. Threat and vulnerability identification are the two parts of risk analysis.

3.2.1. Threat identification. The threats are identified by using a systematic approach to identify the systems’ exposure to threats through attack tree method and the tool used in other to generate the attack trees is Snort. List of threats relevant to the Fog application is then created to determine and access the possible sequence of events that would lead to the occurrence of attacks. The logical representation of the tree is given in Equation (1) and Equation (2). The equations have been adapted from the works of [10, 21].

$$t_0 = S_1 \vee S_2 \vee \ldots \vee S_N \tag{1}$$

where t_0 is the root of the tree and $S_i \mathcal{E} \{1, \ldots, \mathrm{N}\}$, is the i^{th} attack scenario corresponding to t_0 and having the structure given in Equation (2).

$$S_i = t_1^{S_i} \wedge t_2^{S_i} \wedge \ \ldots \ \wedge t_{N_i^S}^{S_i} \tag{2}$$

where $\left(t_j^{S_i}\right) j \,\mathcal{E}\, \{\, 1, \dots, \mathrm{N}_{\mathrm{S_i}}\}$ are the elementary threats belonging to S_i.

The security attributes of the attack scenarios, such as probability of occurrence or outcome is computed, as in [21]. Then instantiation principle as in [22], is applied to determine which attack pattern has the same goal as the identified attack.

3.2.2. Vulnerability identification. Vulnerabilities are weaknesses, in a system which can be exploited by attackers to gain access to the system. In this work, the tool used for vulnerability identification is Network Mapper (Nmap). Vulnerability information is obtained from the Computer Emergency Response Team (CERT) Advisories (Https://www.us-cert.gov) and National Vulnerability Database (NVD) [23-24].

3.2.3. Risk estimation. In this step, the risks levels estimated is compared against a risk acceptance criterion, which is a threshold that determines the acceptability of risks. Risk evaluation comprises of likelihood estimation, negative impact estimation and risk estimation.

3.2.4. Likelihood estimation. Using a scale of 0.7, 0.5, and 0.2, the likelihood is classified as High, Medium, or Low. The scale for evaluating the likelihood of risk is as follows;

High: If the threat source is powerful and security measures to preventing the weakness from being exploited are poor, the risk is significant or high.

Medium: If the threat source is very powerful, the risk is medium, however security precautions are taken and sufficient enough to hinder successful exploit of the vulnerabilities.

Low: If the threat source is weak while security procedures are implemented to preventing or at least considerably impede the use of vulnerabilities, the risk is classified low.

The likelihood (L) of an adverse event is computed as follows;

$$L = P(A)P(B) \tag{3}$$

where P denotes likelihood/probability, A is the exploitable weakness present in the system and B is the exploited weakness.

3.2.5. Impact estimation. The impact of a security event can be estimated by either forfeiture or degradation of one or a blend of some of the security objectives (Confidentiality, Integrity, as well as, Availability (CIA)). The impact can be represented using illustrative scales such as extreme, major, moderate, minor, or incidental on 5 to 1 rating. The impact of an adversative occurrence represented by , is estimated using Equation (4),

$$Impact(a) = w_c C(a) + w_i I(a) + w_a A(a) \tag{4}$$

where w_c,w_i, and w_a are weights of CIA respectively and $C(a)$, $I(a)$ and $A(a)$ are the impact of action a on CIA respectively. Empirical evidence and historical data are used to quantify the variables in this equation.

3.2.6. Risk estimation. The impact of a security event can be estimated by either forfeiture or degradation of one or a blend of some of the security objectives (Confidentiality, Integrity, as well as, Availability (CIA)). The impact can be represented using illustrative scales such as extreme, major, moderate, minor, or incidental on 5 to 1 rating. The impact of an adversative occurrence represented by $Impact(a)$, is estimated using Equation (4),

$$Impact(a) = w_c C(a) + w_i I(a) + w_a A(a) \tag{4}$$

where w_c,w_i, and w_a are weights of CIA respectively and $C(a)$, $I(a)$ and $A(a)$ are the impact of action a on CIA respectively. Empirical evidence and historical data are used to quantify the variables in this equation.

3.2.7. Risk estimation. In risk estimation phase, this work adopts the probability function and set theory in [2], to model the class of possible threats *T*, as shown in equation 5;

$$T = \{t_1, t_2, t_3, \dots, t_n\} \tag{5}$$

where t_i denotes instances of threats.

The class of assets A, is as shown in equation 6;

$$A = \{a_1, a_2, a_3, \dots, a_n\} \tag{6}$$

where a_i, represents assets.

The probability that a threat will occur is given as;

$$P(t_i) = \frac{x}{y} \tag{7}$$

where *x* stands for the number of probable threats, while *y* stands for the entire number of assets.

Then, the risk (R) on the assets is calculated as;

$$R = P(t_i) * V(a_i) \tag{8}$$

where $P(t_i)$ is the threat probability and $V(a_i)$ is the asset value with range [1 ∶ 10].

This same risk function is used to determine the category of the risk involved, and it is calculated as follows:

$$f(R) = \begin{cases} 10 \geq R \geq 8 & Very\ High \\ 7 \geq R \geq 6 & High \\ 5 \geq R \geq 4 & Medium \\ 3 \geq R \geq 2 & Low \\ R \leq 1 & Very\ Low \end{cases} \tag{9}$$

From equation (9), the risk is very high if its value ranges from 8 to 10, it is high if it ranges from 6 to 7, it is medium if it ranges from 4 to 5, it is low if it ranges from 2 to 3 and very low if the risk level is less than 2.

3.2.8. Risk reporting. The objective of risk reporting is to bring risk analysis and evaluation back to the business core. The information is afterwards used for rethinking high-level objectives according to the constantly improved knowledge around the risks.

3.2.9. Risk treatment. A risk acceptance condition is a measure of how much risk a person is willing to take. If the risk associated with a specific threat/vulnerability pair is less than this threshold, the threat/vulnerability pair is left untreated but completely accepted. Risks that don't fit into the accept class can be dealt with in one of the following ways: (1) limit risks by implementing countermeasures to decrease risks to a bearable level, (2) reject risks and utilize workarounds to avoid identified difficulties, and (3) transfer risks to third parties via insurance.

3.2.10. Risk monitoring. Risk Monitoring is the last part of the cyber risk management process. Monitoring and review of risks must be done to see whether the measures implemented have reduced risks effectively and whether more needs to be done.

4. Design of The fog-based game-theoretic security model

In this design, Reflected Distributed-Denial-of-Service (RDDoS) attacks on a Fog application are modeled in form of a Bayesian game between the defender (Fog application) and the attacker (conceptualized in Figure 3). There are two types of RDDoS attacks considered to be launched on the Fog application. These are RDDoS attacks that exploit vulnerabilities in the protocol being used by a web service and RDDoS attacks that are designed to use up the network bandwidth of the web service.

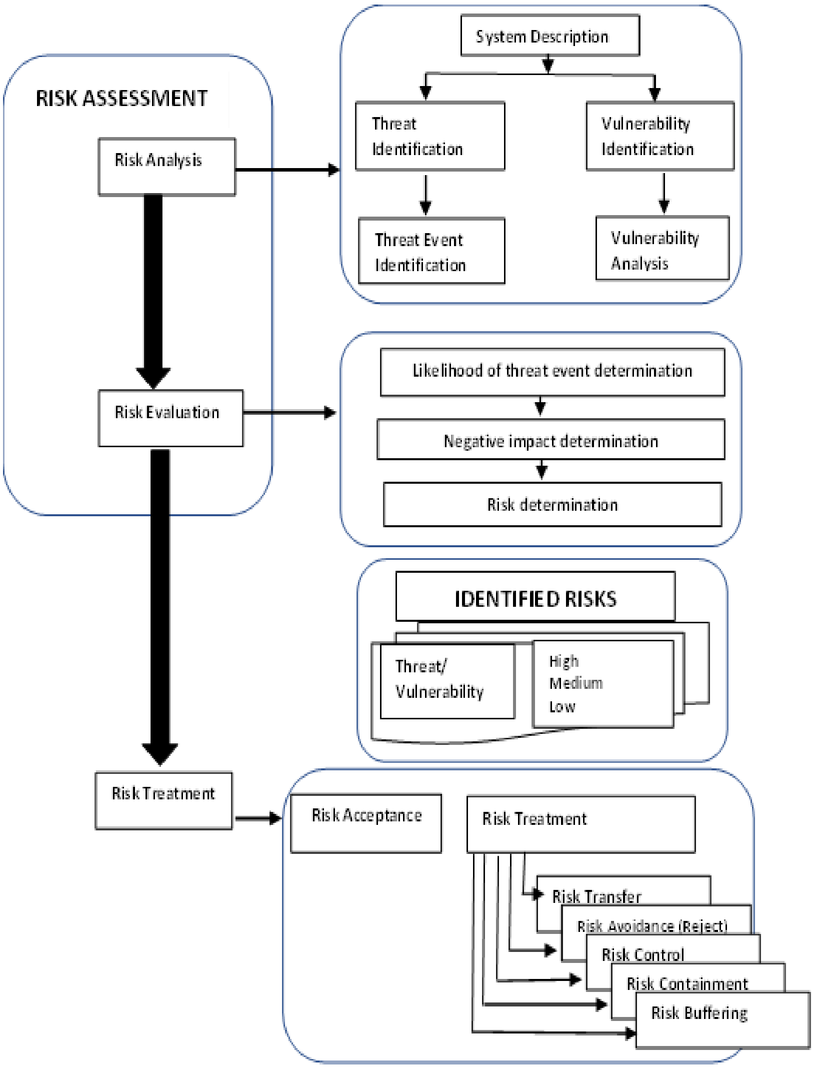


Figure 2. Proposed system architecture.

In our game, each of these attacks represents an attacker. Each attacker has two strategies, that is, “Attack” and “No Attack”. Adopting a few strategies by the defender can provide some security to Fog networks. In this regard, the defender (Fog application) has three strategies, namely;

i. random dropping of packets by the firewall, the firewall helps to protect the Fog network from any unauthorized access by blocking or allowing the pass to the Fog network.

ii. disabling Dynamic Host Configuration Protocol (DHCP), then manually giving IP addresses to clients. This prevents some specified MAC addresses from accessing the router.

iii. provision of excess bandwidth for the Fog network. Over-provision of network bandwidth is necessary to cater for situations whereby the attacker has successfully hacked into the network and uses the opportunity to launch different kind of attacks.

The Bayesian game for the RDDoS attacks on the Fog network is the tuple (N, A, Θ, F, U), where;

N = {1 ,…, n} is the set of attacker and defender A_i= Action Sets, where A = {A_a, A_d} is the set of attacker/defender actions for the attacker and defender.

A_a corresponds to the action set of the attacker and A_d corresponds to the action set of the defender. A set of all actions is given by Action Set, A_i which is defined as the union of both action sets, i.e., $A_i = A_a \cup A_d$.

A_a = **{**Attack, Not Attack**}** is the action set of the attacker.

A_d = {Firewall drop rate, disable DHCP and assign IP addresses manually to the clients and provision of excess bandwidth for the Fog network}.

$\Theta_{i,}$ denotes Type Sets, where $\Theta = \{\Theta_a, \Theta_d\}$ is the set of types for each attacker and defender. $\theta_i \varepsilon \Theta_i$ represents a specific type of each attacker and defender.

Θ_a = {RDDoS attacks that exploit Protocol Vulnerabilities, RDDoS attacks that use up the network bandwidth}

Θ_d = {defender}

F: $\Theta \rightarrow [0, 1]$ is the joint probability distribution function based on the type of attacker and defender.

p (Θ_a = RDDoS Attacks that exploit Protocol Vulnerabilities) = μ

p (Θ_a = RDDoS attacks that use up the network bandwidth) = $1 - \mu$.

$U = \{U_a, U_d\}$

where U_i: $A \times \Theta \rightarrow R$ is the utility function for the player *i*. This implies that, the payoff for a player *i* depends on the action and type of that player. The game assumes that the defender uses only one defense mechanism at a time.

4.1. The RDDoS attacks game

The RDDoS attack game is conceptualized in Figure 4. In this game, the attacker attacks the Fog application and accesses confidential communication without authorization. The RDDoS attack prevents the normal use of Fog facilities. The attack causes a disruption of an entire network by overloading it with messages so as to degrade its performance. The attacker is represented by the attacking system, and the environment is limited to the category of good accesses granted by normal users. The assaulting system is split into two parts: service and defense. The protection part comprises all the elements that provide computer services to users, like the hardware/software devices that route packets, whereas the service part only contains the components which offers computing services to users. Instead of passive monitoring, detection, and reaction to attacks, the relationship between both the attacker and the system is portrayed as a game spanning temporal dimension enabling effective defensive operations. As a proof of concept, the next section employs the Bayesian game model to describe the interactions between attacker and defender.

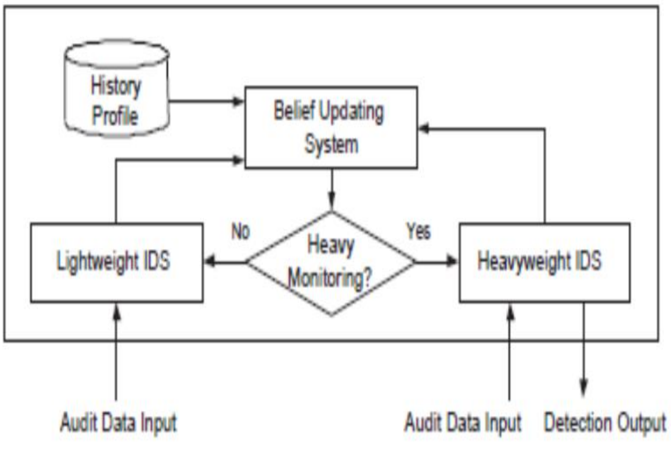


Figure 3. The bayesian detection framework.

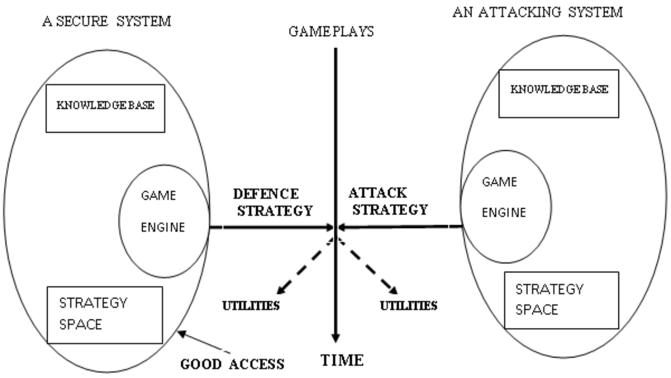
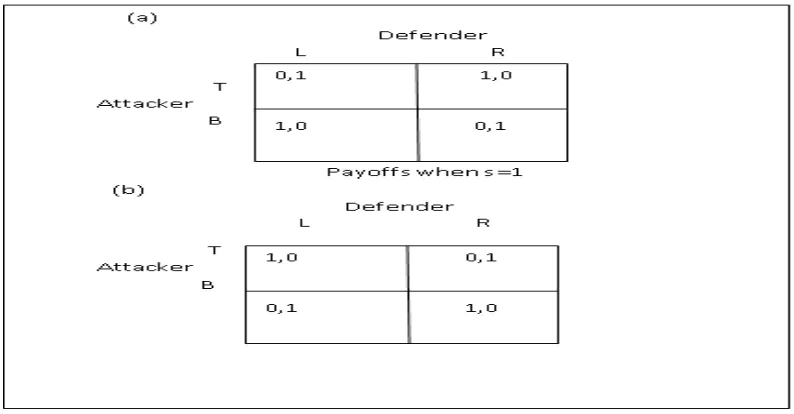


Figure 4. The DDoS attack game.

4.2. The bayesian game representation

When using the Bayesian game technique, a player only has a limited understanding of the game's data. In an interactive scenario, this is insufficient because other participants' decisions, as well as their beliefs, are important because they influence their decisions. As a result, a player must have views about other players' beliefs. The Attacker and Defender play a 2 x 2 game in which the payoffs are determined by the data from the game. are the activities of the attacker. The actions of the Defender are, and the payoffs are shown in the matrices in Figure 5.



(a)

Defender

		L	R
Attacker	T	0,1	1,0
	B	1,0	0,1

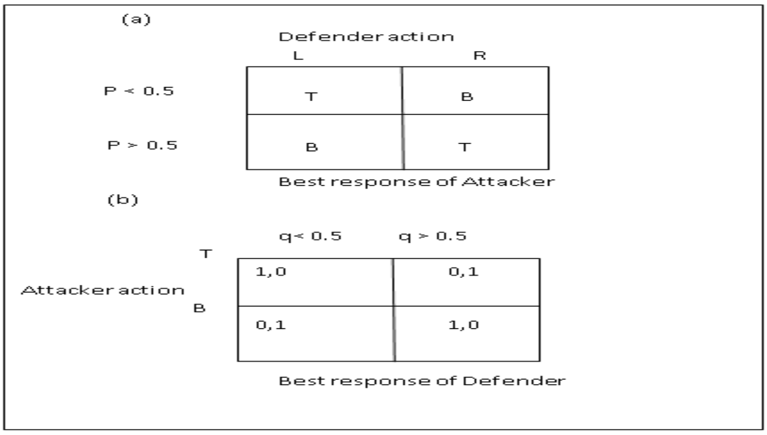
Payoffs when s =1

(b)

Defender

		L	R
Attacker	T	1,0	0,1
	B	0,1	1,0

Figure 5. Illustration of the RDDoS attack defender/attacker 2 x 2 game.



(a)

Defender action

	L	R
P < 0.5	T	B
P > 0.5	B	T

Best response of Attacker

(b)

		q< 0.5	q > 0.5
Attacker action	T	1,0	0,1
	B	0,1	1,0

Best response of Defender

Figure 6. RDDoS attack defender/attacker best action.

4.3. The game's bayesian nash-equilibrium

A game's Nash Equilibrium is a point where/when none of the players may unilaterally change their strategy to increase their payoffs [9]. The Nash equilibrium strategy set is made up of the players' mutual best replies. The solution theory of Bayesian Nash equilibrium can be derived from the Nash equilibrium theory based on the notion of best response. The Bayesian Nash equilibrium describes a behavior for each player that is the perfect response to what he perceives the other players' behavior is, that is, the best response to the other players' strategies given his type. Equation (10) gives the game's Bayesian Nash equilibrium in the experiment.

If for all i in $I$and for all t_i in T_i

$$u_{t_i}(\partial) \geq u_{t_i}(\partial_{-i};\sigma_i),\ \ \forall\sigma_i \epsilon \Sigma_i \tag{10}$$

where, $\partial_{-i} = (\partial_j)_{j\neq i}$ represents the vector of strategies of players other than i.

With defender updates based on new observations, the significance of Nash Equilibria in a static environment is investigated.

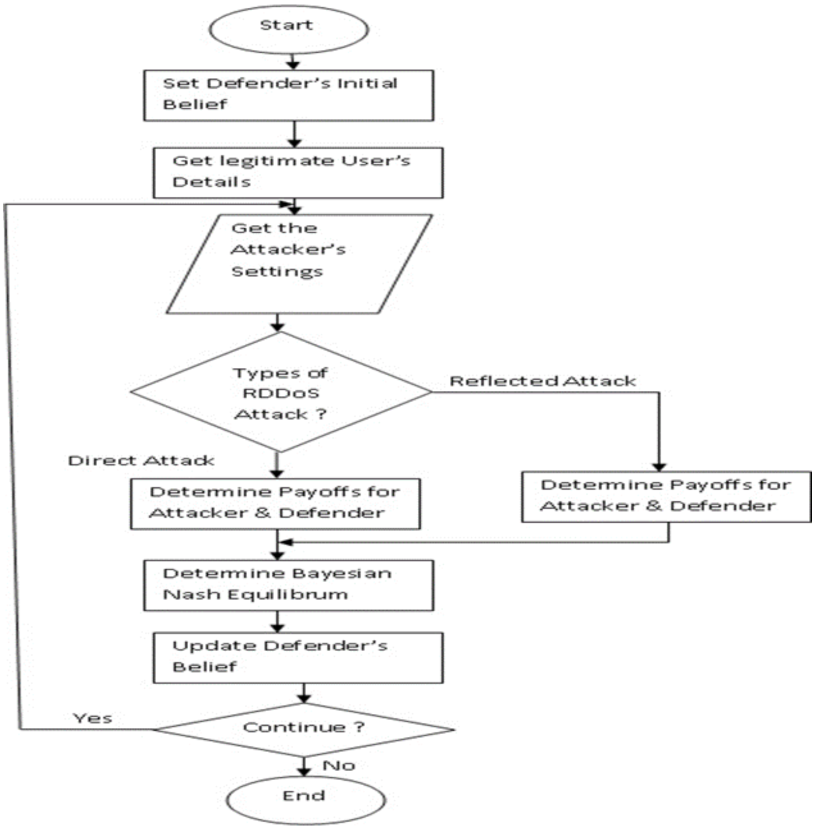
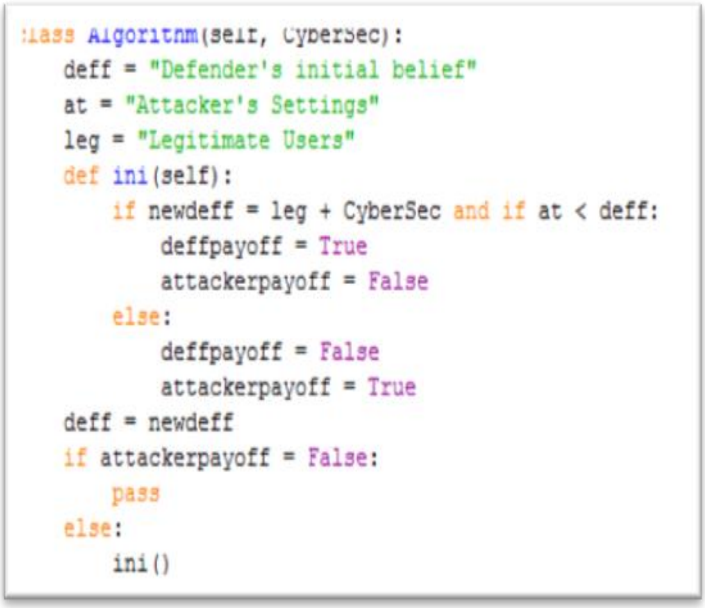

Start
Set Defender's Initial Belief
Get legitimate User's Details
Get the Attacker's Settings
Types of RDDoS Attack ?
Reflected Attack
Direct Attack
Detemine Payoffs for Attacker & Defender
Detemine Payoffs for Attacker & Defender
Detemine Bayesian Nash Equilibrum
Update Defender's Belief
Yes
Continue ?
No
End

Figure 7. Flowchart showing logical design of the game.

4.4. Logical design of the application

This section describes the logic used for the computer program of the RDDoS attacks on the Fog servers as a dynamically evolving attacker-defender game. Figure 7 is a flowchart to visually display the logic of the program.



```
class Algorithm(self, CyberSec):
    deff = "Defender's initial belief"
    at = "Attacker's Settings"
    leg = "Legitimate Users"
    def ini(self):
        if newdeff = leg + CyberSec and if at < deff:
            deffpayoff = True
            attackerpayoff = False
        else:
            deffpayoff = False
            attackerpayoff = True
    deff = newdeff
    if attackerpayoff = False:
        pass
    else:
        ini()
```

Figure 8. Algorithm for the fog-based gt cyber risk management model.

5. Setup for the system implementation

The suggested system's hardware components include a Spear HP Spear Laptop with an Intel Core i3 processor as well as, 4GHz of RAM capacity. Other features of the test environment include a Linux operating system to stand as the Fog server, which handles Snort in NIDS mode. The computer system designated as the attacking machine has Kali Linux, 2016. The choice of Kali was made based on its attacking programs that come preloaded in Kali OS. Python 2.7 programming language is used in conjunction with the following Python libraries: System (sys), Process Utility (psutil), Matplotlib, OpenCV, as well as, PyQt4 as the framework.

The primary application was designated as "Squid3" by third-party software utilized in the studies. It is a caching application that intercepts and sends Hyper Text Transfer Protocol (HTTP) requests on behalf of the user. It saves the response data in its memory so that it can service the next request without having to go through the hops to the Internet again to retrieve the required data. The Squid application is installed on the Ubuntu computer. Similarly, an Intrusion Detection System (IDS) called “Snort” was tweaked and utilized as a Network-Based IDS (NIDS). Snort is used as a network intrusion detection system (NIDS) in the experiment, sniffing the whole network for malicious traffic such as “Ping Sweep” or “Port Scan”. On the Ubuntu system, Snort was installed.

The vulnerability assessment program Network Mapper (Nmap) was preinstalled on the Kali computer.

5.1. Topographical anatomy of the network and setup of the test bed

The test bed for the research was Adekunle Ajasin University's campus network in Akungba-Akoko (AAUA), Nigeria. Three PCs together with the equipment available at AAUA were connected. The application server (defense) is run on the first computer, while the client application is run on the second computer, and the Kali Linux OS is run on the third machine. Two of the PCs are connected via a local area network (LAN), while the third PC serves as the attacker's tool. The entire network used is shown in Figure 9. The university's network, data center, and the demilitarized zone are all part of the entire mesh topology (DMZ).

The University's Internet connection has a bandwidth of 40 MegaBytes per second (Mbps). There will be a speed test as well as a fog test (both of which are designed to determine the time and hops taken to get to the server). In relation to Domain Name Services, aauafog.net was assigned to the test Fog server.

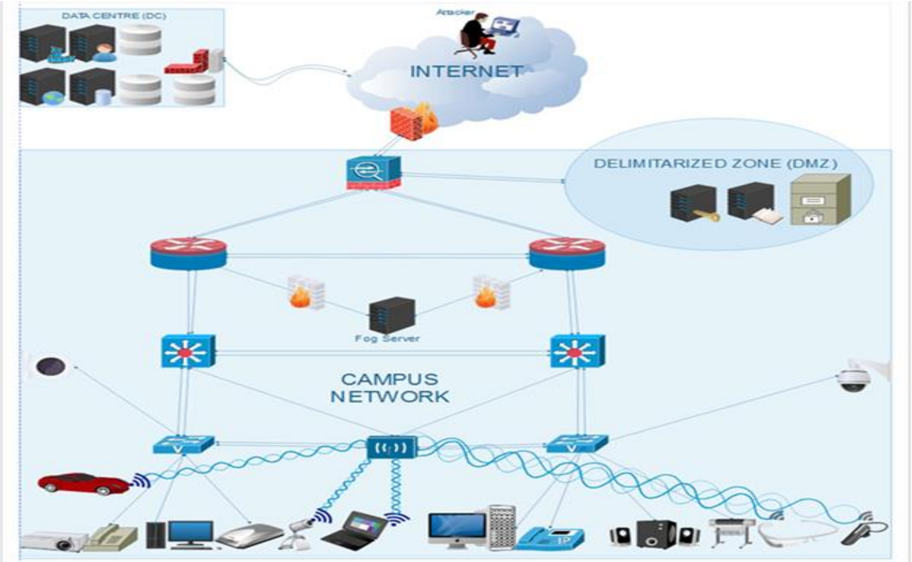


Figure 9. Topographical anatomy of the network for the experiment.

6. Conclusion

The design of a framework for GTA-based cyber-security risk or threat management in Fog computing is the topic of this study. In the Fog cloud infrastructure, it proposes a Bayesian game model for risk management. When implemented, the proposed approach will significantly lower the cost of controlling cyber-risk in the cyber world and will give real-time data that can help understand the threats that a competent attacker faces when attempting to infiltrate a Fog network. The model will also encourage proactive cyber threat management and improve cyber operations' efficacy and efficiency. The model's implementation and system evaluation will be investigated further.

References

[1] Gabriel A. J. (2020) Appliance Scheduling towards Energy Management in IoT Networks using Bacteria Foraging Optimization (BFO) Algorithm. In: A.E. Hassanien et al. (eds.), Artificial Intelligence for Sustainable Development: Theory, Practice and Future Applications, Studies in Computational Intelligence 912, pp. 290-310. Springer, Nature Switzerland. https://doi.org/10.1007/978-3-030-51920-9_15.

[2] Alese, B.K., Gabriel A. J., Olukayode O. and Daramola O.A. (2014); Modelling of Risk Management Procedures for Cybercrime Control Systems; The 2014 International Conference of Information Security and Internet Engineering; World Congress on Engineering, ISBN 978-988-19252-7-7; 505-509.

[3] Alese B. K., Gabriel A. J., Ayodele T. and Akinsowon O. A. (2016) “Cost-Benefit Analysis of Cyber-Security Systems”. Proceedings of the World Congress on Engineering and Computer Science 2016. Vol I, WCECS 2016, October 19-21, 2016, San Francisco

[4] Thompson, A., Abayomi, A., Gabriel, A.J. (2022). Multifactor IoT Authentication System for Smart Homes Using Visual Cryptography, Digital Memory, and Blockchain Technologies. In: Misra, S., Kumar Tyagi, A. (eds) Blockchain Applications in the Smart Era. EAI/Springer Innovations in Communication and Computing. Springer, Cham. https://doi.org/10.1007/978-3-030-89546-4_14

[5] X.G. Shan, J. Zhuang (2020). A game-theoretic approach to modelling attacks and defences of smart grids at three levels, Reliability Engineering & System Safety. Vol. 195. DOI: 10.1016/j.ress.2019.106683.

[6] E. Bagtug, M. Bennis, and M. Debbah, (2014). Living on the Edge: The Role of Proactive Caching in 5G Wireless Networks. *IEEE Commun. Mag.*, *52*, 82–89.

[7] Stojmenovic I. (2014) "The Fog Computing Paradigm : Scenarios and Security Issues," vol. 2, pp. 18.

[8] P. Liu, W. Zang, and M. Yu, (2005). Incentive-based modeling and inference of attacker intent, objectives and strategies. *ACM Transactions on Information and System Security*, *8*(1), 78–118.

[9] S. Hansman, and R. Hunt, (2005). A taxonomy of network and computer attacks. *Computers and Security, February 2005.*, *24*, 31–43.

[10] K. C. Nguyen, T. Alpcan, and T. Basar, (2009). Stochastic games for security in networks with interdependent nodes. *Proc. Of Intl. Conf. on Game Theory for Networks (GameNets)*

[11] F. He, J. Zhuang, and N. S. V. Rao, (2012). Game-Theoretic Analysis of Attack and Defence in Cyber-Physical Network Infrastructures. In *Proceedings of the 2012 Industrial and Systems Engineering Research Conference G. Lim and J.W. Herrmann, eds.*

[12] B. K. Alese, G. B. Iwasokun, and D. I. Haruna, (2013). DGM Approach to Network Attacker and Defender Strategies. In *'Information Security' A Conference Proceedings on International Conference for Internet World Congress on Internet Security Technologies and Secured Transactions ICITST.*

[13] E. O. Ibidunmoye, B. K. Alese, and O. S. Ogundele, (2013). Modeling Attacker-Defender Interaction as a Zero- Sum Stochastic Game. *Journal of Computer Sciences and Applications*, *1*(2), 27–32.

[14] S. Garg, and G. S. Aujla, (2014). An Attack Tree Based Comprehensive Framework for the Risk and Security Assessment of VANET using the Concepts of Game Theory and Fuzzy Logic. *Journal Of Emerging Technologies In Web Intelligence*, *6*(2).

[15] C. Kamhoua, A. Martin, D. K. Tosh, K. A. Kwiat, C. Heitzenrater, and S. Sengupta, (2015). Cyber-threats Information Sharing in Cloud Computing : A game Theoretic Approach, 382–389. http://doi.org/10.1109/CSCloud.2015.8.

[16] L. Maghrabi, (2015). Moving Assets to the Cloud : A Game Theoretic Approach Based on Trust.

[17] S. Garg, and G. S. Aujla, (2016). Accessing Risk Priority of SSL SYN Attack using Game Theoretic Attack Defense Tree Model for VANETs, 729–734.

[18] L. Wei, A. Sarwat, and W. Saad. (2016). Risk Assessment of Coordinated Cyber-Physical Attacks Against Power Grids : A Stochastic Game Approach, 1–7.

[19] P. Y. Matthew-Omole, A. J. Gabriel, A. F. Thompson, B. K. Alese, (2021). Monte Carlo Simulation Approach to Network Access Control. Journal of Internet Technology and Secured Transactions (JITST) 9(1):726-729. DOI:10.20533/jitst.2046.3723.2021.0088.

[20] S. Musman, and A. Turner, (2017). A game theoretic approach to cyber security risk management. *Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, (Special). http://doi.org/10.1177/1548512917699724.

[21] T. Tidwell, R. Larson, K. Fitch, and J. Hale, (2001). Modeling Internet Attacks. *Proceedings of the 2001 IEEE Workshop on Information Assurance and Security United States Military Academy, West Point, NY, 5-6 June, 2001*, *1*, 5–6.

[22] H. Mohamed, (2005). Theoretical Aspects of Computer Network Risk Management. The Communication Network and Security (CN&S) research Laboratory at the Communication School of Engineering University, Carthage, Tunisia.

[23] https://www.us-cert.gov/. (2017). US -CERT. *United States Computer Emergency Readiness Team, Department of Homeland Security.*

[24] https://nvd.nist.gov/. (2017). Computer Security Resource Centre, National Vulnerability Database. *National Institute of Standards and Technology U.S. Department of Commerce*.