Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 8 (2013) pp. 147-155

An Improved Cost-Sensitive Intrusion Response Model

Aderonke J., Ikuomola[1], Adesina S., Sodiya[2], Adio T. Akinwale[3] and Dada O. Aborisade[4]

[1]Department of Computer Science, Federal University of Agriculture Abeokuta
Ogun State, Nigeria
deronikng@yahoo.com

[2]Department of Computer Science, Federal University of Agriculture Abeokuta
Ogun State, Nigeria
sinaronke@yahoo.co.uk

[3]Department of Computer Science, Federal University of Agriculture Abeokuta
Ogun State, Nigeria
aatakinwale@yahoo.com

[4]Department of Computer Science, Federal University of Agriculture Abeokuta
Ogun State, Nigeria
aborisadeda@funaab.edu.ng

Abstract: **Intrusion Response Systems (IRSs) are being used as counter-measure against detected intrusions in order to guarantee the confidentiality, integrity and availability of information. The goal of cost-sensitive response system is to ensure that response cost does not outweigh the intrusion cost. In order to ensure this, some cost-sensitive response models have been developed. Some of these models do not consider the effectiveness of previous actions and lack standard approach for estimating associated cost. In this work, we present a model COSIRS for assessing cost of responses. The architecture of COSIRS comprises of six components namely; alert filter and correlation module, response manager, database, cost-sensitivity evaluation module, adaptability module and response-deployment module. Principal Component Analysis was employed to reduce the dimension of alerts raised by the intrusion detection system. A Neural Network-based classifier scheme that distinguishes among true positive, false positive and false negative alerts was deployed to enable COSIRS learn from its previous behaviour. COSIRS combines the response efficiency and response cost $(rc; 0 \le rc \le 1)$ in its inference engine for deploying cost-sensitive responses based on the inherent cost parameters (cost of damage, cost of automatic response and operational cost). The performance analysis indicates that in a public web server, the costs for deploying 10,000 responses in COSIRS, CRS and Static were 0.15rc, 0.20rc and 0.24rc while the efficiency rate of COSIRS and CRS were 99% and 90% respectively. This showed that COSIRS design had the lowest processing time requirements and minimal response cost when compared with existing ones.**

Keywords**:** Cost-Sensitivity; Intrusion Cost, Intrusion detection system; Response Cost, Intrusion response system

I. Introduction

The advancement in technology over the past few years has brought about increase in the number of intrusions on computer networks. The constant increase of intrusions against networks and their resources inspires the need to adequately protect these valuable assets [1]. Intrusion detection is a technology for detecting hostile attacks against computer systems [22]. [16] defined intrusion detection as the process of monitoring the events occurring in a computer system or network and analyzing then for signs of intrusion. Intrusion Detection System (IDS) is an important measure to protect computer and network. The final goal for intrusion detection systems is to assist site security officers or system administrators to estimate the state of system and suggest an appropriate response [24]. [9] defined intrusion detection system as the system that detects and logs improper access on the computer system or network. [8] was of the view that intrusion detection systems are software and/or hardware structures that detect malicious behavior in the systems they protect and produce relevant alerts. An intrusion detection system is used to monitor network traffic, check for suspicious activities and notifies the network administrator or the system [18]. However, intrusion detection system is limited to detect intrusion events without prompt, automatic and effective response [12]. Intrusion detection system does not perform any action to prevent intrusion; its main function is to alert the site security officer or system administrator that there is possible security violation [2]. In the process of detecting an attack, it is necessary to take corrective action to tackle the attack and ensure safety of the system. The process of counter-measuring these attacks is referred to as intrusion response [14, 15]. Intrusion Response Systems (IRS) continuously monitor system health based on intrusion detection system alerts so that malicious or unauthorized activities can be handled effectively by applying appropriate

countermeasures to prevent problems from worsening and return the system to a healthy mode [3].

In recent years, the trend toward modeling of Cost-Sensitive response system has become more important. The main goal of cost-sensitive response system is to strike a balance between damages made by the intrusion and the cost of response. However, defining an accurate measurement of these cost factors and ensuring consistent evaluation across various computing environments are common challenges in using a cost-sensitive approach. The problem of intrusion response system is that when the responses are deployed against a detected intrusion, they often alter the state of the system negatively, affecting resources and leading to damage. An intrusion response system needs to be cost-effective in that it should not cost more than the expected level of loss from intrusions. This requires that an intrusion system considers the trade-off among cost factors, which at the minimum level should include: the cost of damage caused by the intrusion, the cost of manual or automatic response to an intrusion and the operational cost. For example, an intrusion which response cost is higher than the damage cost should not be acted upon beyond simple logging action. In this work, an improved cost-sensitive intrusion response model was developed.

This paper is structured as follows. Related work on cost-sensitive intrusion response system was reviewed in Section 2. Section 3 describes the architecture of the proposed cost- sensitive intrusion response system. Section 4 presents the implementation and performance evaluation of COSIRS while Section 5 is the conclusion.

II. Review of Related Work on Cost Sensitive Intrusion Response Systems

The selection of reaction mechanism or countermeasure to attacks in a computer networks has always been a challenging field of work. Many cost-sensitive intrusion response system techniques have been proposed and deployed over the last five year. A general overview on existing work in the area of intrusion response was published by [14, 15].

The cost-sensitive approaches to intrusion response proposed by [20] address the cost of deploying responses. This work introduces a cost benefit measure which incorporates multiple dimensions of cost in the face of an intrusion.

In the models proposed by [19], the costs and benefits of the response actions in association with dependencies between services in the system were considered. Such modeling reveals priorities in response targets and evaluates the impact of different response strategies on dependent services and system. This approach uses the concept of response benefit or effectiveness as a factor related to the response's ability to mitigate the intrusion damage, the operational cost of the response is not included.

[17] extended the idea of representing services and their inter-dependencies in a graph for selecting responses through creating a resource type hierarchy, so that every service type has common response measures associated with it. Response sequences need to be optimal for each service node, i.e every response step needs to produce maximum benefit at minimum costs. In this approach, the process of cost assignment is completely manual and the cost assignment method is only an approximation of the real resource cost.

The approach proposed by [5,6] maps alarms provided by the intrusion detection system to I-Graph nodes and appropriate response actions are deployed targeting identified attack goals. The response actions for the affected nodes in the graph are selected based on the effectiveness of this response to the particular attack in the past, the disruptiveness of the response to legitimate users and the probability measure that a real intrusion is taking place. ADEPTS is specifically designed to reflect the characteristics of the considered system. This significantly limits the applicability of the model to varying system constraint. It also relies on semi-manual development of Intrusion-graph to determine the spread of network attack.

[10, 11] proposed a relatively pragmatic way of defining metrics and characterizing DoS effects on the user of a network. The authors suggested that these metrics can also be used for selecting appropriate response measures, though no specific implementation details are given. However, they present a lot of practical measurement results and also discuss ways of implementing measurement methods for simulation environments. Although the authors does not focused on selecting response measures, but propose to compare the DoS measurement results before and after deployment of a response in order to determine its value.

A method for the evaluation of response cost was proposed by [23]. This method was based on the principle that one should achieve the maximum security goal through a minimal response cost. On this basis, a method for judging the causal relationship between an intrusion and a cooperative intrusion was further suggested. The intrusion response system designed according to the above response strategy was applied to the distributed network environment. Through the cooperation of more than one management domain and a large scale study of relationships among various intrusion response costs, a superior response strategy was deduced.

The framework proposed by [7] and [13] were based on the cost-sensitive assessment of intrusion response. The authors introduced a set of measures which characterized the potential cost associated with the intrusion handling process and proposed a method for evaluating intrusion response with respect to potential intrusion damage, response effectiveness and response cost for a system.

The approach proposed by [21] was based on joint decisions of IDS configuration and alarm investigation capacity under active and passive responses. Both active and passive responses incur costs proportional to the time spent in the system by an arbitrary alarm. The active response has delay costs while the passive response has damage costs caused by the alarms on the system under investigation.

[12] proposed an Automatic Intrusion Response System (AIRS) for responding to attacks as soon as possible in order to avoid unnecessary loss. This model adopts the multidimensional classification model of intrusion events, and applies the clustering model formula in order to reduce the unnecessary loss. This model still has some defeets that need to be dealt with. For instance, the cost analysis method is not accurate enough, the increasing number of sophisticated attacks and their costs cannot be defined primitively.

[14, 15] proposed a cost-sensitive model for preemptive intrusion response system and cost-sensitive assessment of

intrusion response selection. These models compare the costs of deploying response to the costs of damage caused by a non-responded attack. Additionally, a methodology for adapting responses in a changed environment through an evaluation of previously applied response measures was discussed. These models still have some challenges which need to be dealt with. For instance, no comprehensive off-the-shelf solution is available for capturing information related to potential intrusions and available responses. Hence, there is difficulty in ensuring the consistent ranking of responses and also, large amount of manual input is required by the system.

[4] investigated the intrusion detection process, its technical cost implication, and its divergent nature and further proposed a system that is platform independent for an appropriate impact sensitive intrusion response system with an embedded database. The authors did not really discuss on how intrusion response cost was evaluated.

We proposed to assess response impact with respect to resources of the affected system.

III. An Architecture of a Cost-Sensitive Intrusion Response System (COSIRS)

The architecture of a cost-sensitive intrusion response system is presented in figure 1. The main task of IDS is to monitor the events occurring in the network and then analyze them for signs of intrusion. Once an intrusion has been detected, IDS raises alert and then passes the attack specific parameters to the alert filter and correlation module in the response system (COSIRS). COSIRS take over after signs of intrusions are detected and then attempt to actively counter it.

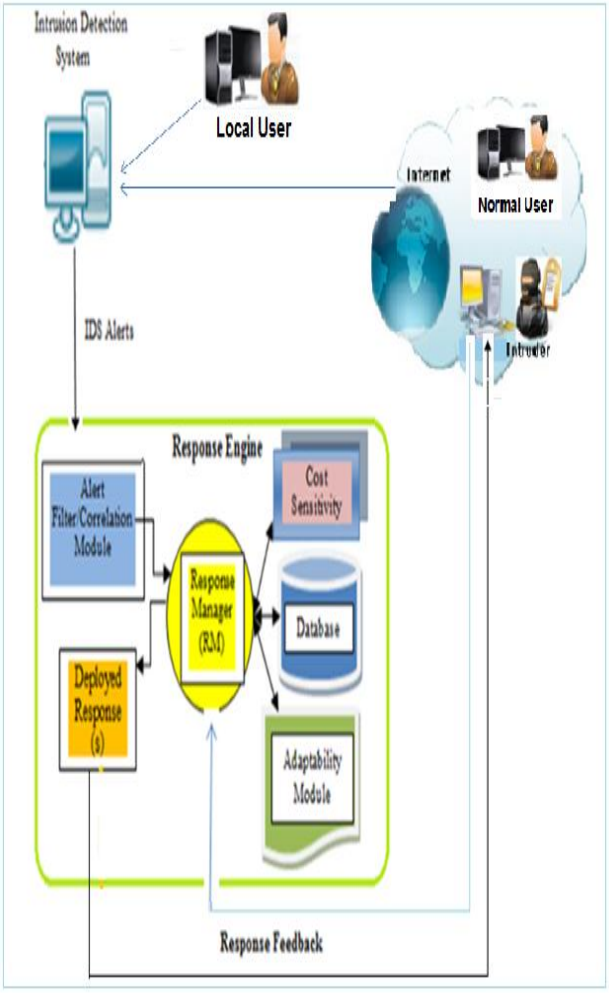

Figure 1. Architecture of a Cost-Sensitive Intrusion Response System (COSIRS)

COSIRS comprises of six basis components namely; alert filter and correlation module, response manager, database (consist of response action, profile and intrusion specification), cost-sensitivity evaluation module, adaptability module and response deployment module.

A. Alert Filter and Correlation Module (AFCM):

The Alert filter and correlation module provide high-level insight to the security state of the network and filter false positives as well as redundant alerts efficiently from the output of network IDS. The alert filter and correlation module is made up of two modules which are the alert filter and the alert correlation modules.

1) Alert Filter

The filter gathers alerts from the sensor in each managed network and eliminates redundancies among those alerts. The features used to eliminate the redundancies are the source and class of the attack. The filter merges the redundant alerts into thread events and forwards them to the correlation module at regular intervals. The filter consists of three sub-modules: an alert collector, an attack_thread maker and an attack_thread sender. The alert collector forms one process and the other two modules behave as multiple threads in a single process. Figure 2 shows the internal architecture and processing flow of the filter.

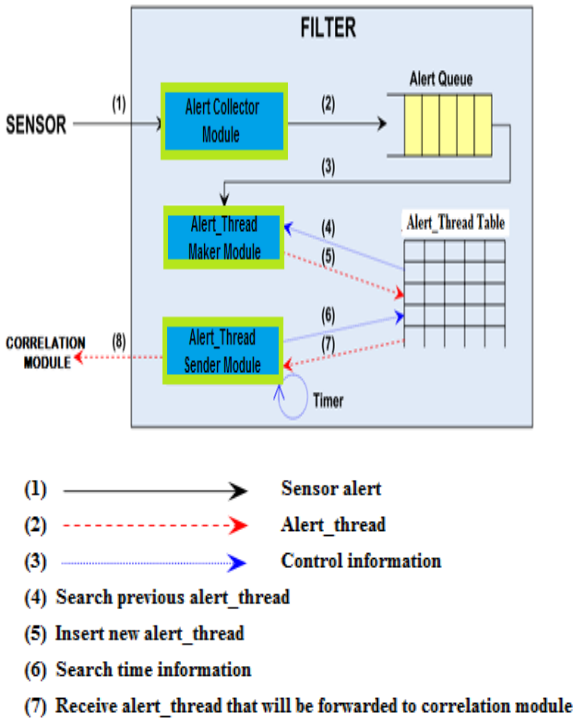

Figure 2. Internal architecture and processing flow of filter

(a) Alert collector/normalisation

The alert collector receives alerts from the sensors in the form of an Alertpacket. Alerts may come from different sensors and security systems. Since they are encoded in different formats, it is necessary to translate each alert or adapt or pre-process the message reported by sensors (IDSs) into a standardized format that is understood by correlation components by using intrusion detection message exchange format (IDMEF) for normalisation. The intrusion detection message exchange format (IDMEF) developed by the intrusion detection working group (IDWG) was used because it expresses relationships between alerts which are actually an essential requirement of alert correlation. IDMEF is object-oriented and is implemented in the extensible markup language (XML). A sample of an alert in IDMEF is illustrated in Figure 3.

```
<IDMEF-Message/>
<?xml version="1.0"?>
<!DOCTYPE IDMEF-Message PUBLIC "-//IETF//DTD
RFC XXXX
IDMEF v1.0//EN" "/usr/local/etc/idmef-message.dtd">
<IDMEF-Message version="1.0">
        <Alert ident="289">
                <Analyzer analyzerid="109"
        model="snort" version="2.0.5">
                <Node>
                        <name>tcpdump_dmz</nam
                e>
                </Node>
        </Analyzer>
        <CreateTime
ntpstamp="0xc36cc187.0xd3aa9b49">2007-11-
                24T17:42:31Z</CreateTime>
                <Source>
                <Node>
                  <Address category="ipv4-addr">

        <address>135.013.216.191</address>
                  </Address>
                </Node>
                    <Service>
                   <port>22</port>
                   <protocol>tcp</protocol>
                </Service>
        </Source>
        <Target>
        <Node>
                <Address category="ipv4-addr">

        <address>172.016.112.149</address>
                </Address>
        </Node>
                <Service>
                   <port>22</port>
                     <protocol>tcp</protocol>
                </Service>
        </Target>
    <Classification origin="vendor-specific">
            <name>msg=(spp_stream4) STEALTH
ACTIVITY
            (NULL scan) detection</name>
                <url>none</url>
    </Classification>
  </Alert>
```

Figure 3. Intrusion detection message exchange format representation of an alert in an XML document

The dimension of the data is then reduced using principal component analysis (PCA) algorithm. The goal of PCA is to ultimately reduce the number of effective variable or feature used for classification while retaining as much as possible of the variation present in the original dataset. The alerts are

then sent to the alert queue where they are saved in the order of arrival.

(b) Alert_thread maker

The alert_thread maker compares the alerts received from the queue with previous alerts. In this process, the filter simply compares the source and attack class. Alerts with same attributes other than time and which differ only by a small amount of time are fused together for the purpose of alert reduction (this is possible since multiple IDS or sensor may be there in the network which produces redundant alerts and same event may causes IDS to trigger hundreds of similar alerts). If exact matches exist between the alerts, alert_thread maker merges the alerts into a matching thread event. Otherwise, if there is no match in the source or class of the attack, a new thread event is generated.

(c) Alert_thread sender

This transfers the thread events to the correlation module at predefined intervals. That is, whenever the alert aggregation interval defined in the timer expires, the alert_thread sender stops updating the alert_ thread event table and transfers thread events to correlation module for further processing. The timer is reconfigurable according to the status of the network.

2) Correlation module

The correlation module is made up of the aggregation and the classfier components.

(a) Alert aggregator

The alert aggregator compares if there is any similarity between the features of the thread events transferred from each filter. If common features exist between two thread events, the aggregator merges them together into one meta_event. The aggregator can merge thread events that may not be merged into a similar thread event in the filter because the aggregator has longer merging interval than the filter.

When new thread events are transferred into the aggregator component, the aggregator then extracts the previous aggregation events generated for certain period of time from the database and then compares them with the newly transferred thread events to determine whether they have common features. If they have, the aggregator updates the previous aggregation event to include the new thread event; otherwise, it generates a new aggregation event. In this case, the source, destination and attack class are the features used for comparison.

(b) Alert classifier

The idea of alert classifier is to distinguish between successful and failed intrusion attempts (both false and irrelevant positives). Identifying failed intrusion attempts allows other components to reduce the influence of these alerts on their decision process.

B. Database (DB)

The database contains information about the intrusion specification, profiles and response actions.

1) Intrusion specifications

It contains information about the attack, that is, the severity and the impact of the attack in terms of confidentiality, integrity and availability and the speed with which the attack (specific types of intrusion) is likely to evolve.

2) Profiles

It contains data about users, systems and attackers, which can provide additional context for response decisions. The target profiles contain information about the characteristics of systems within the organization. After the response manager retrieves the address of the target from the detection engine, it uses its profile to retrieve additional characteristics that are relevant for taking response action.

3) Response actions

It contains the details of available response actions, enabling selection of responses with the most appropriate characteristics. In order for appropriate responses to be selected, the Response Manager needs to know more about the characteristics of the response action themselves which is stored in the database. Examples of response actions are block source IP, restart services, block port, kill services privilege shell, kill process, deny process, reboot server, set directory read only, etc.

C. Cost Sensitivity Evaluation Module (CSEM)

The evaluation of the response action effectiveness is based on the following factors which are: factors associated with the intrusion damage and factors describing the response cost.

1) Computing the Intrusion Cost

The potential system damage caused by a true intrusion i_n is given as

$$DC_{i_n} = IS_i + OC_i \quad (1)$$

DC_{i_n} = the cost of damage caused by an intrusion

IS_i = intrusion impact on the system

OC_i = cost of daily maintenance of various aspect of the detection system

The system damage caused by an attack can be identified using the following three components: the system resources affected by intrusion, intrusion impact on system resource and the operational cost.

(a) Operational Cost Index associated with detecting intrusion

It is the baseline cost present for an intrusion, regardless of system damage caused.

$$\text{Operational Cost}(OC_i) = \frac{\text{Associated Cost}}{\text{Total value of the system}} = \frac{l}{v} \quad (2)$$

Where:

l = labour associated with manually addressing an intrusion + loss of reputation to the organization due to intrusion occurring + the direct costs for contracted services

v = Organization-assigned system value

(b) Intrusion Impact Evaluation

Below is an enhanced formula for the computation of intrusion impact on the system resources:

$$IS_i = \frac{\sum_{sr_i \in SR} E_i\left(sr_i \omega_j\right)}{k * m} \quad (3)$$

$I = (i_1, i_2, ..., i_n)$, the set of considered intrusion, where I_n is associated with the corresponding intrusion signature.

IS_i = Intrusion impact on system resource where $0 \le IS_i \le 1$

SR = $(sr_1, sr_2, \ldots, sr_m)$; the set of resources provided by the system.

m = number of resources which are being affected by intrusion

E_i = severity level of the attack

j = represent the security policy (confidentiality, integrity and availability)

ω_j = weight of each security policy

k = a normalization value to bring the value IS to range between 0 and 1

2) Computing the Response Cost

Response Cost (RC) is the price of deploying a response on a given system. The cost of the response includes evaluating the damage the response will cause to the system (response impact) and determining the operational costs of deploying the response. Given a response r, a response impact on system SI_r and an operation cost value OC_r the response cost RC_r is define as

$$RC_r = IS_r + OC_r \quad (4)$$

(a) Operational Cost associated with deploying response

Operational cost of a response is the cost of daily maintenance of various aspect of the response system.

$$OC_r = \frac{Associated\ Cost}{Total\ value\ of\ the\ system} = \frac{l}{v_s} \quad (5)$$

$0 \le OC_r \le 1$

Where:

l = Human resource + System resources + Direct expenses

v_s = Organization-assigned system value

(b) Response Impact Evaluation

The response impact is computed using the following:

$$IS_r = \frac{\sum\left(1 - \frac{r}{n}\right) Sr_i \omega_j}{k * m} \quad (6)$$

Where

IS_r = response impact on the system resources

r = response deployed

n = total number of available response ranks

m = total number of resources affected

Sr_i = resources affected by the deployed response

ω_j = the weight of each security policy

k = a normalization value to bring the value IS to range between 0 and 1

D. Adaptability Module (AM)

The adaptability of the response is the ability of the system to dynamically adjust response selection to the changing environment during the attack time. The adaptability is based on the effectiveness of the previous response action and feedback received.

The response effectiveness is calculated as

$$R_{eff} = \frac{\eta_P}{\eta_T} \times 100 \quad (7)$$

where

R_{eff} = response effectiveness

η_P = number of positive or correct decision

η_T = total number of decision/response issue

E. Response Manager (RM)

The response manager uses an expert-system to determine an appropriate response system. The response manager is a rule-based reasoning system made up of three components which are the rule-base, database and decision engine/reasoning.

After receiving information about the intrusion and the context in which it has occurred, these information are stored in the database and the response manager now proceed to the decision making phase and then select the appropriate response.

The deployment of response is determined through the following statements which must be fulfilled.

```
if ( response was successful in the past) and
        ( damage cost > response cost) then
                select response
```

Based on the above decision, the response manager can now initiate an approved response automatically. Figure 4 shows the components of a rule-based response manager for response selection

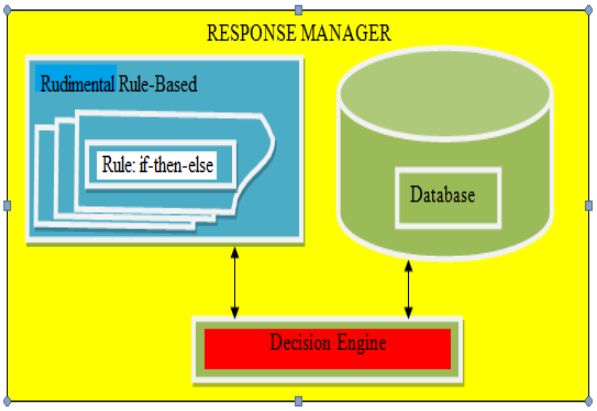

Figure 4. A rule-based response manager for response selection

F. Response Deployment Module

It manages the responses that are available for a particular system and triggers a recommended response by the response manager.

IV. Implementation and Performance Evaluation of COSIRS

To evaluate the effectiveness of COSIRS, the response selection was implemented as a plugin tool for intrusion detection system (IDS) The model is evaluated using the KDD dataset and a series of experiments were performed focusing on the cost effectiveness and its scalability. The experiments were performed using three primary parameters: the system resources, responses available in the system and the suspected intrusion.

A. Structural Comparison of the model used

COSIRS is compared with Static response system (a traditional system) and Cost-Sensitive Response Selection (CRS) that is relevant with the model. The comparison is done using the following criteria (i) Alert Correlation, (ii) Intrusion Cost evaluation (iii) Response Cost evaluation, (v) Feedback and (vi) Intrusion impact. Table 1 shows the structural comparison of the model used.

Table 1. Comparison of models

Criteria	Models		
	Static	**CRS (Stakhanova et. al, 2012)**	**COSIRS (Proposed Approach)**
Alert filter	No	No	Yes
Alert correlation	No	No	Yes
Intrusion cost evaluation	No	Yes	Yes
Response cost evaluation	No	Yes	Yes
Feedback	No	Yes	Yes
Intrusion impact	Yes	Yes	Yes

B. Comparison using response cost

In these experiments, COSIRS was compared with Static and CRS (which also consider response costs for a particular intrusion) and their performances were evaluated. The results of the experiments in comparison with Static and CRS are shown in figures 5, 6 and 7.

Figure 5 shows that in a public web server, the costs for deploying 10,000 responses in COSIRS, CRS and Static approaches are 0.15rc, 0.20rc and 0.24rc while the costs for deploying 20,000 responses are 0.29rc, 0.45rc and 0.47rc respectively.

Figure 6 shows that in a medical data, the costs for deploying 10,000 responses in COSIRS, CRS and Static approaches are 0.012rc, 0.015rc and 0.016rc while the costs for deploying 20,000 responses are 0.020rc, 0.026rc and 0.031rc respectively.

Figure 7 shows that in a central file repository, the costs for deploying 10,000 responses in COSIRS, CRS and Static approaches are 0.014rc, 0.017rc and 0.020rc while the costs for deploying 20,000 responses are 0.025rc, 0.033rc and 0.037rc respectively.

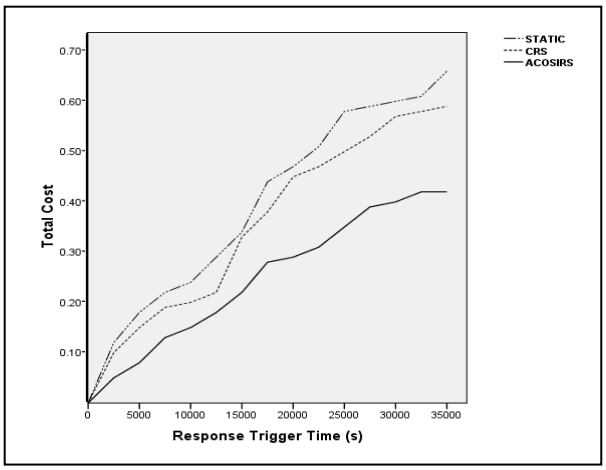

Figure 5. Evaluation of the response cost using public web server

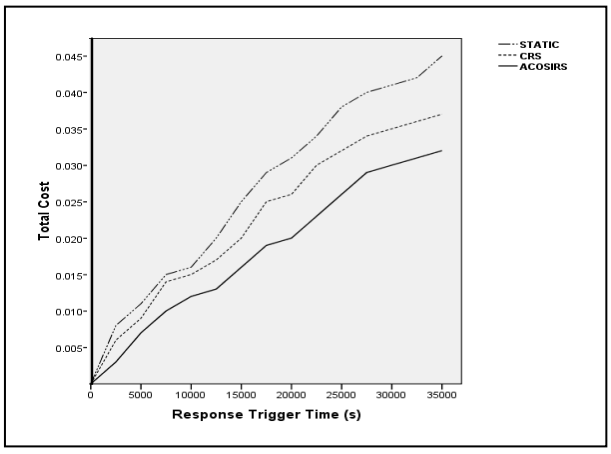

Figure 6. Evaluation of the response cost using medical data

Figure 7. Evaluation of the response cost using central file repository

From figures 5, 6 and 7 above, the difference between the COSIRS, Static and CRS cumulative value is shown by the separation between the lines. COSIRS approach clearly outperforms the Static and CRS approaches. Although, initially the cumulative response value taken by COSIRS, Static and CRS are small, as the number of responses deployed on the system increases, the difference in cost becomes significant.

The total response cost of the Static and CRS tends to increase in the public web server; medical data input and central file repository systems, while the COSIRS attains moderate value throughout the experiment. This indicates that, even in system were the security priority is heavily in favour of both Static and CRS, an COSIRS approach will still outperform both approaches over time because COSIRS has minimal cumulative cost value compared to Static and CRS.

C. Comparison using processing time

The results of the experiments, given in figure 8, show that in Static approach, for 1,000 suspected intrusions at 0.017s average computation time, 0.017s was required to assess the available responses while for 10,000 suspected intrusion at 0.248s, 0.042s was required to assess the available responses. In CRS approach, for 1,000 suspected intrusions at 0.015s average computation time, 0.014s was required to assess the available responses while for 10,000 suspected intrusion at 0.221s, 0.035s was required to assess the available responses. In COSIRS approach, for 1,000 suspected intrusions at 0.013s average computation time, 0.011s was required to assess the available responses while for 10,000 suspected intrusion at 0.170s, 0.026s was required to assess the available responses. These results show that COSIRS approach has reasonable processing time requirements that are considered suitable for the efficient analysis of response selection.

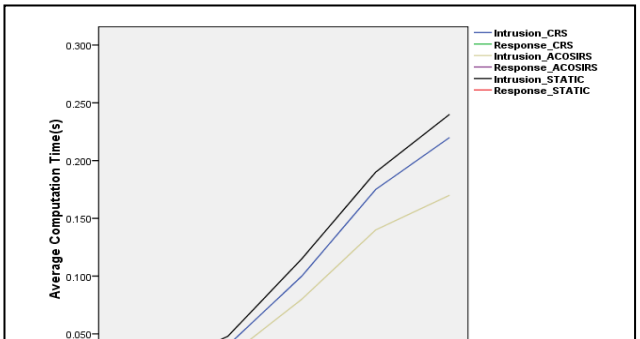

Figure 8. Process Time Evaluation

D. Comparison using response efficiency rate

Figure 9 shows the efficiency rate of the responses. In a public web server, the efficiency rates of COSIRS and CRS for deploying 10,000 responses are 99% and 90% respectively. In a medical data, the efficiency rates of COSIRS and CRS for deploying 10,000 responses are 93% and 87% respectively while in a central file repository; the efficiency rates of COSIRS and CRS for deploying 10,000 responses are 95% and 88% respectively.

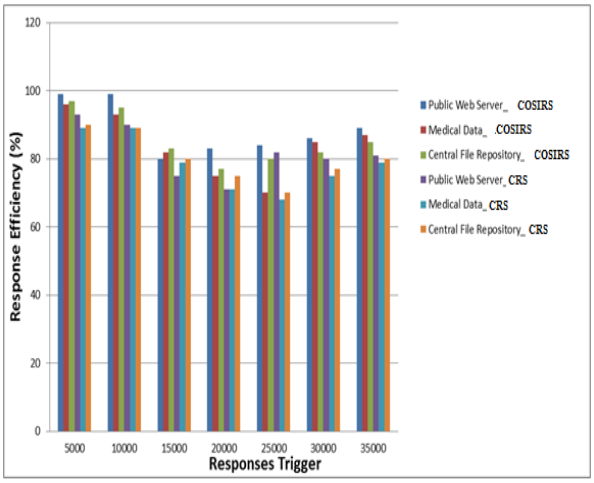

Figure 9. Response efficiency rate

V. Conclusion

In this paper, we proposed a model called COSIRS for evaluating intrusion damage and response cost. COSIRS automatically choose the least costly response in time to minimize the damage caused by an intrusion. The proposed model identifies three main factors that constitute response cost, namely the cost of damage caused by the intrusion, the cost of manual or automatic response to an intrusion and the operational cost. These response metrics provide a consistent basis for assessing response across systems while allowing the response cost to adapt to system environment. The adaptability of the response is based on the effectiveness of the previous response action and feedback received. The experimental results show that the performance of this approach is better than using a traditional based (Static) intrusion response system and Cost Response Selection (CRS). The results of evaluation show that the design, COSIRS has better performance over existing ones. In order to build more

intelligent into the response system, a computational intelligent method can be adopted in future.

References

[1] A. J. Ikuomola and A. S. Sodiya, A credible cost-sensitive model for intrusion response selection**.** 2012 Fourth international conference on computational aspects of social networks (CASoN), pp. 222-227, 2012

[2] A. S. Sodiya, O. Adeniran and A. J. Ikuomola, "An expert system-based site security officer," Journal of computing and information technology - CIT 15, 3, pp. 227–235, 2007

[3] A. Shameli-Sendi, N. Ezzati-Jivan, M. Jabbarifar and M. Dagenais, Intrusion response systems: survey and taxonomy. *ijcsns international journal of computer science and network security, vol. 12* , 2012

[4] A. T. Enikuomehin , A. G. Idowu, C. B. Akerele, A. P. Owate and B. A.. Aina, "Cost minimization model for an adaptive intrusion response system," *Indian journal of computer science and engineering (ijcse),vol. 3,* 2012.

[5] B. Foo, Y. Wu, Y. Mao, S. Bagchi and E. Spafford, "Automated adaptive intrusion containment in systems of interacting services," *Computer Networks, vol. 51(5),1334–1360,* 2007.

[6] B. Foo, Y. S. Wu, Y. C. Mao, S. Bagchi and E H. Spafford, "ADEPTS: Adaptive intrusion response using attack graphs in an e-commerce environment. *In Proceedings of the 2005 International Conference on Dependable Systems and Networks," 508–517,* 2005.

[7] C. R. Strasburg, N. Stakhanova, S. Basu and J. Wong, "A Framework for Cost Sensitive Assessment of Intrusion Response Selection," *33rd Annual IEEE International Computer Software and Applications Conference*, 2009.

[8] G. P. Spathoulas and S. K., Katsikas," Reducing false positives in intrusion detection systems. computer and security, vol. 29(2010), pp. 35 – 44, 2010

[9] I. Ahmad, A. Abdullah, and A. Alghamdi, " Towards the selection of best neural network system for intrusion detection," I*nternational journal of the physical sciences, vol. 5, pp.1830-1839,* 2010.
intrusion response" 2008.

[10] J. Mirkovic, A. Hussain, B. Wilson, S. Fahmy, P. Reiher, , R. Thomas, W. Yao and S. Schwab, Towards user-centric metrics for denial-of-service measurement. *in proceedings of the workshop on experimental computer science,* 2007.

[11] J. Mirkovic, P. Reiher, S. Fahmy, R. Thomas, A. Hussain, S. Schwab and C. Ko, "Measuring denial of service," i*n proceedings of the 2nd acm workshop on quality of protection,* 2006

[12] M. Zhou and G. Yao, "Improved cost-sensitive model of intrusion response system based on clustering," International conference in electrics, communication and automatic control proceedings, 931- 937, 2011.

[13] N. Stakhanova, C. Strasburg,. S. Basu and J. Wong, "Towards cost-sensitive assessment of intrusion response selection," Journal of computer security vol. 20(2012), 169-198, 2012.

[14] N. Stakhanova, S. Basu and J. Wong, "A cost-sensitive model for preemptive intrusion response systems," *in proceedings of the ieee international conference on advanced information networking and applications, niagara falls, Canada ,* 2007a.

[15] N. Stakhanova, S. Basu and J. Wong, "A taxonomy of 5ntrusion response systems," I*nternational journal of information and computer security*, 1, 169–184, 2007b.

[16] S. Lakhina, S. Joseph and B. Verma, "Feature reduction using principal component analysis for effective anomaly-based intrusion detection on nsl-kdd", 2010

[17] S. M. Balepin, J. Rowe, and K. Levitt., "Using specification-based intrusion detection for automated response," in *proceedings of the 6th international symposium on recent advances in intrusion detection,* 2003

[18] T. F Khan, Z. Farooqui. and V. Richhariya, "Identification of intrusions in network for large data base using soft computing approach," International journal of computer sci ence and technology, vol.3(1), 2012.

[19] T. Toth and C. Kruegel,."Evaluating the impact of automated intrusion response mechanisms," in *proceedings of the 18th annual computer security applications conference*, 2002.

[20] W. Lee, W. Fan, M. Millerand, S. Stolfo and E. Zadok, "Toward cost-sensitive modeling for intrusion detection and response. in *journal of computer security",* vol 10, pp. 5–22, 2000.

[21] W. T. Yue and M. Cakanyildririm, "A cost-based analysis of intrusion detection system configuration under active and passive response, 2010.

[22] X. Guan, W. Wang and X. Zhang, "Journal of network and computer applications," vol. 32 (2009), pp. 31– 44, 2009.

[23] Y. Wu, " A cost-sensitive method for distributed intrusion response

[24] Z. Zhang,, P. H. Ho and L. He, "Measuring ids-estimated attack impacts for rational incident response: a decision theoretic approach," C*omputers & security, 2009*.

Author Biographies

Dr. A. J. Ikuomola is presently a lecturer in the Department of Computer Science, Federal University of Agriculture Abeokuta, Nigeria. Her research interest are information and network security, cloud computing, artificial intelligence and HCI. She has published in both local and international journals and refereed conferences.

Dr. A. S. Sodiya is presently a senior lecturer at the Department of Computer Science, Federal University of Agriculture, Abeokuta. His research interests are Information Security, Artificial Intelligent and Software Engineering. He has published in both local and international journals.

Dr. A. T. Akinwale is presently a lecturer at the Department of Computer Science, Federal University of Agriculture, Abeokuta. His research interests are Artificial Intelligence, Database Systems and Discrete Computing. He has published in both local and international journals.

A.D. Olaniyi is a PhD student in the Department of Computer Science, Federal University of Agriculture Abeokuta, Nigeria. He has interest in Cloud Database Security, Digital Forensics and HCI.