

# An Intelligent Incidence Reporting System For Online Attacks

Monsuru O. Abdullahi, Adesina S. Sodiya, Adebukola S. Onashoga, C. O. Tinubu

Department of Computer Science, Federal University of Agriculture, Abeokuta

[abdullahi.monsur.olalekan@gmail.com](mailto:abdullahi.monsur.olalekan@gmail.com), [sodiyaas@funaab.edu.ng](mailto:sodiyaas@funaab.edu.ng), [onashogasa@funaab.edu.ng](mailto:onashogasa@funaab.edu.ng), [scientistore@yahoo.com](mailto:scientistore@yahoo.com)

**Abstract**— *Cyberattacks occur every second around the world and many attacks use the same patterns to exploit systems. Most of these patterns remain effective as a result of inadequate awareness or lack of information security culture among cyber users. Thwarting these attack techniques therefore becomes a cause for concerns, and consequently this work focuses on the development of an Intelligent Incidence Reporting System (IIRS) for reporting online attacks. A Model-View-Controller (MVC) system architecture was adopted. Model represents the state of the system where domain data is encapsulated. The user interface (UI) is contained in the View while the system logic is handled by the Controller. The Stanford Natural Language Processing (NLP) API is included in the controller to facilitate real-time online processing of reported details with focus on language detection, sentiment analysis, and named entity recognition. Wu and Palmer algorithm was adopted to handle the semantic similarity of extracted entities, and this forms the basis on which each reported incidence is classified. The proposed system was implemented in ASP.Net Core with C# being the programming language of choice. The system was tested on a local server and evaluated by twenty students of the affiliated department, and each reported incidence was categorized appropriately as either phishing, ransomware, cyberwarfare or other most suited attack technique. Other attributes which include the ease of use of the system, consistency and stability were also evaluated and the system received an average rating of 94% with respect to these parameters. In particular, the system was rated 95% over its contribution to cybersecurity awareness raising, as considered by our respondents.*

## KEYWORDS

**Cyberspace, cyberattack, incident reporting, semantic similarity, natural language processing, information security culture, vulnerabilities.**

## I INTRODUCTION

Computer security, also known as cyber security or IT security is the protection of computer systems from the theft or damage to their hardware, software or information, and as well from disruption or misdirection of the services

they provide, as defined by Morrie, 1988 in. Cyber security plays an important role in the development of information technology, as well as Internet services. Therefore, enhancing cyber security and protecting critical information infrastructures are essential to each nation's security and economic well-being.

A cyberattack is any type of offensive manoeuvre employed by nation-states, individuals, groups, or organizations that targets computer information systems, infrastructures, computer networks, and/or personal computer devices by various means of malicious acts usually originating from an anonymous source that either steals, alters, or destroys a specified target by hacking into a susceptible system. These can be labelled as either a cyber-campaign, cyberwarfare or cyberterrorism in different context.

The cyberspace has gained countless followers and society has become dependent on cyber systems across the full range of human activities, including commerce, finance, health care, agriculture, energy, entertainment, communications, and national defence. As a result, the nature, complexity and severity of cyber threats are increasing. On a daily basis across the world, hackers are taking control of networks, locking away files and demanding sizable ransoms to return data to the rightful owner. From phishing attacks to ransomware and advanced persistent threats attacks. More painfully, many of these attacks use the same pattern to exploit systems and most of these patterns remain effective because most cyber users are unaware of their existence or have had no opportunity to learn from the history of attacks in which they've been employed. Therefore, developing proper security measures requires a thorough understanding of such attacks through incident reporting and awareness raising.

Thus, this study aimed at focusing on the development of an intelligent incident reporting system for online attacks. The intelligence of the system is defined by its natural language processing capability, which facilitates real-time online processing of reported details with focus on language detection, named entities extraction, semantic similarities and sentiment analysis. This capability enables cyber victims to submit details of cyberattack incidence, even when they have no idea of the kind of attack

techniques employed, thereby encourage reporting and as well achieve a better classification of reported incidences based on identified entities. As a result, the availability of cybercrime statistics is promoted and trends of different attack techniques can be determined. The system proceeds further to make meaning from the captured data by presenting them with descriptive statistics data visualization tools such as bar chart, pie chart, etc. in a way that support human understanding and as well improve cybersecurity awareness.

The rest of this paper is organized as follows: review of related work is presented in section II. The proposed IIRS architecture is presented in section III and the system was implemented in section IV. Performance evaluation and results is provided in section V, and section VI covers the recommendations and conclusion.

## II RELATED WORK

Several related works have addressed the issue of cyber-attack incidents reporting from organisational levels. However, this study provides a general platform for individuals, as well as organisations who experiences cyber victimisation to freely submit reports of such occurrences.

Kuypers and Elisabeth (2015) in [7] identified some frameworks for recording and reporting cyber security incidents. These frameworks include Vocabulary for Event Recording and Incident Sharing (VERIS) and Common Attack Pattern Enumeration and Classification (CAPEC). For many years, the US CERT Federal Incident Reporting Guidelines mandated what information was collected and how it was reported for certain organisations. Some of the general information to be reported include the incident date and time, attacker/ victim IP, port/ protocol information, operating system information, detection information, impact, resolution, and incident category. VERIS structures information for incident tracking, victim demographics, incident description, discovery and response, and an impact assessment. One of the drawbacks to VERIS is the large number of fields, which can create data fatigue for investigators repeatedly entering the same information for low-impact incidents.

Johnson (2015) in [5] identified some architectures of cyber-security incidence reporting in safety-critical system to include the Internal Reporting Architecture, the Gatekeeper Architecture, Active External Monitoring Architecture and then Joint Public-Private Architecture for cyber safety. These architectures provide templates for reporting security incidence on networks or applications related to critical safety systems. The Internal Incident Reporting Architecture happens to be the simplest among others because it assumes that all incidents will be analysed to the same level of detail. The Gatekeeper architecture is relatively more elaborate and it was designed to extend the simplified architecture by considering the reporting chain for adverse events. The Active External Monitoring Architecture assumes a relatively active role for external

agencies in the investigation of adverse events. And the fourth model highlighted was the Joint Public-Private Architecture for cyber-safety, which focuses on a joint public-private approach based on cooperation between industries and government with implicit mechanisms for cost sharing. This model builds on the previous architectures and also assumes the creation of a Joint Monitoring Group cyber-security incidents. Conclusively, Johnson work shows that irrespective of the architecture that is being used, the most important thing is not to delay any action in integrating reporting mechanisms for cyber-attacks into safety-critical, national infrastructures.

In an approach to analyse the social engineering vulnerabilities of organisations, Edwards (2017) in [3] demonstrated a passive automated approach for analysing online information of organisation's social media followers, so as to determine who among them are actually its employees and the extent of the organisation's vulnerability to social engineering attacks based on these employees' information available online. Two critical challenges were addressed. The research identified the accounts of employees within the online footprint of the organisation and also retrieved extended information on these employees by linking the information obtained in this footprint to other social networks. Employees information were sourced using the social network accounts such as twitter account or Facebook page of each targeted organisation, and some information were also obtained from a link to a "roster page", which lists the names and sometimes positions of these employees. The research consulted some expert social engineers, who evaluated each individual attack method against some criteria such as the frequency of use, the effectiveness and efficiency, and the data captured from this consultation were used as basis for ranking some potential real world attacks. In addition, the use of Open Source Intelligence (OSINT) data for such attacks was considered, and as a result, detailed OSINT items were obtained and as well categorised to be either essential or non-essential to the attack process. Moreover, OSINT data items were preliminarily separated into two key information types: bootstrap (data which facilitate the attack) and accentuator (data which are used to enhance the effectiveness of such attack). Edwards also highlighted some mitigation techniques such as security awareness, revised security policies and practices, network restrictions, review of company website and the need for further social engineering penetration tests.

The use of intelligent sensors in incident reporting has also gained significance, as being demonstrated by the Symantec Global Intelligence Network, who has established the largest threat collection network in the world by tracking over 700,000 global adversaries, and recording events from 98 million attack sensors worldwide. This network monitors threat activities in over 157 countries and territories through a combination of Symantec products, technologies, and services, including

Symantec Endpoint Protection™, Symantec DeepSight™ Intelligence, Symantec Managed Security Services™, Norton™ consumer products, and other third-party data sources, generating more than nine trillion rows of security data, as reported in Internet Security Threat Report (ISTR), volume 22, 2017 by Symantec [14]

Cisco Annual Cybersecurity Report (2017) [2] also presents research, insights, and perspectives from Cisco Security Research. It highlights the relentless push-and-pull dynamic between adversaries trying to gain more time to operate and defenders working to close the windows of opportunity that attackers try to exploit. It examines data compiled by Cisco threat researchers and other experts. This research and insights are intended to help

organizations respond effectively to today’s rapidly evolving and sophisticated threats.

### III THE PROPOSED SYSTEM ARCHITECTURE

The proposed model includes View, Controller, Model and domain Database The system architecture in Figure 1 illustrates the loose coupling of these different components. The architectural design was based on a Model-View-Controller approach, while a Natural Language Processing API is incorporated into the controller to facilitate processing of the unstructured incident descriptions.

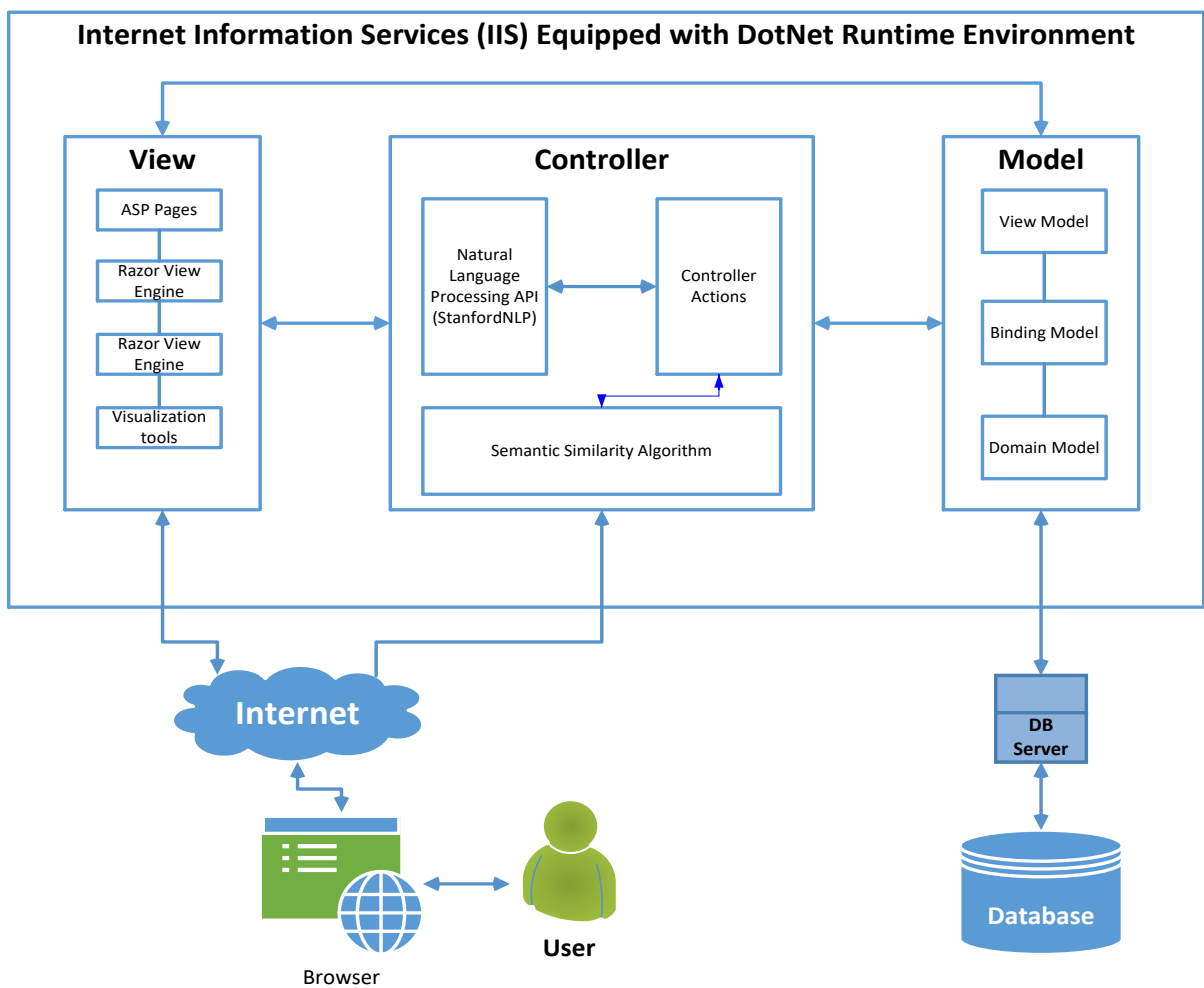


Figure 1:Proposed Architecture

### A. The View

The view serves as the presentation layer through which users can interact with the system. The view is rendered along with the view-model, which is responsible for passing data from the controller to the user interface in response to user's request. It also facilitates incident reporting and as well provide access to result visualisation.

### B. The Controller

The controller represents the system logic. It handles the binding of the Http-request values encapsulated in the binding model to the controller's action parameters, and as well returns the view-model to the user interface in response to the user's request. It also facilitates retrieval of data, as well as update the domain database through the domain model. More importantly, it encapsulates the system's actions which are callable as an Http endpoint, and as well communicate with the Natural Language Processing API to perform tokenisation, stemming, stop-word removal, language detection, entity extraction and sentiment analysis on the incident description provided.

### C. The Model

The model is the state of the system where the domain data is encapsulated. It facilitates the transfer of data from the domain database to the view in form of view-model and from the view to the domain database via the controller as domain model. The binding model binds the value contained in the user's request to the controller actions parameters.

### D. Incidence Details Processing Algorithm

Algorithm:	Incident	Details
Processing		
<i>Input: V → Incident Report Details</i>		
<i>Output: Named Entity</i>		
<i>Recognition (NER), Sentiment</i>		
<i>Analysis (between 0 &amp; 1), Detect</i>		
<i>Language</i>		
<i>Process:</i>		
1. $V \leftarrow$ <i>Incident_Description</i>		
2. <i>if lang(V) == 'en'</i>		
3. <i>then report_analysis(V);</i>		
4. <i>else</i>		
5. <i>return 'Express yourself in</i>		
6. <i>end if</i>		
7. <i>function report_analysis(V)</i>		
8. <i>begin:</i>		
9. $V^1 \leftarrow$ <i>tokenize (V);</i>		
10. <i>foreach w ∈ V<sup>1</sup></i>		
11. <i>stem(w);</i>		
12. <i>end foreach</i>		
13. <i>foreach w<sup>1</sup> ∈ V<sup>1</sup></i>		
14. <i>if w<sup>1</sup> == stopword</i>		
15. <i>then remove (w<sup>1</sup>)</i>		
16. <i>end if</i>		
17. <i>end foreach</i>		
18. $S \leftarrow$ <i>sentiment(V<sup>1</sup>);</i>		
19. <i>list entities ← ner (V<sup>1</sup>);</i>		
20. <i>foreach entity ∈ entities</i>		
21. <i>entity_matching(entity);</i>		
22. <i>end foreach</i>		
23. $C \leftarrow$ <i>incident_category;</i>		
24. <i>Populate database;</i>		
25. <i>Update View;</i>		
26. <i>Stop</i>		

### E. StanfordNLP Annotators

**tokenize:** Tokenizes the report description into a sequence of tokens. The English component provides a PTB-style tokenizer, extended to reasonably handle noisy and web text. The tokenizer saves the character offsets of each token in the input text.

**ssplit:** Splits a sequence of tokens into sentences.

**truecase:** Determines the likely true case of tokens in text (that is, their likely case in well-edited text), where this information was lost, e.g., for all upper-case text. This is implemented with a discriminative model using a CRF sequence tagger (Finkel et al., 2005) [15].

**pos:** Labels tokens with their part-of-speech (POS) tag, using a maximum entropy POS tagger (Toutanova et al., 2003) [16].

**lemma** Generates the lemmas (base forms) for all tokens in the annotation.

**ner:** Recognizes named (PERSON, LOCATION, ORGANIZATION, MISC) and numerical (MONEY, NUMBER, DATE, TIME, DURATION, SET) entities. With the default annotators, named entities are recognized using a combination of CRF sequence taggers trained on various corpora (Finkel et al., 2005), while numerical entities are recognized using two rule-based systems, one for money and numbers, and a separate state-of-the-art system for processing temporal expressions (Chang and Manning, 2012) [17].

**parse:** Provides full syntactic analysis, including both constituent and dependency representation, based on a probabilistic parser (Klein and Manning, 2003 [19]; de Marneffe et al., 2006) [18].

**sentiment:** Sentiment analysis with a compositional model over trees using deep learning (Socher et al., 2013) [20]. Nodes of a binarized tree of each sentence, including, in particular, the root node of each sentence, are given a sentiment score.

Most of these annotators have various options which can be controlled by properties. These can either be added to the Properties object when creating an annotation pipeline via the API.

#### F. StanfordNLP Named Entity Recognition Algorithm

```

INPUT
{
  "document": String
}
OUTPUT
{
  "sentences": List [
    {
      "detectedEntities": List [
        {
          "word": String,
          "entity": String
        }
      ]
    }
  ]
}

```

- document – (required) an arbitrary length text document describing the incidence reported.

#### G. StanfordNLP Sentiment Analysis

```

INPUT
{
  "document": String
}
OUTPUT
{
  "sentiment": a value between
0 and 4
}

```

#### H. Entity Semantic Similarity Matching

The approach adopted for the similarity matching of extracted entities with pre-defined cyber-attack-specific keywords is based on the Wu and Palmer algorithm. This algorithm uses WordNet and measures semantic relatedness between two entities or concepts by considering the depths of the two synsets in the WordNet taxonomies, along with the depth of the Lowest Common Sub-sumer (LCS). The LCS of two concepts  $c1$  and  $c2$  denoted as  $LCS(c1, c2)$  is defined as the lowest node in the WordNet hierarchy that subsumes (is a hypernym of) both  $c1$  and  $c2$ . In this comparison, the similarity is twice the depth of the LCS of the two concepts divided by the sum of the depth of individual concept, as shown in equation 1.

#### From Wu & Palmer Formula

$$sim_{wup} = \frac{2 * depth(lcs(c1, c2))}{depth(c1) + depth(c2)} \quad (1)$$

Where:

$$0 < sim_{wup} \leq 1$$

The  $sim_{wup}$  can never be zero because the depth of the LCS is never zero. The  $sim_{wup}$  of the root of the taxonomy is one and the  $sim_{wup}$  is one if the two concepts are the same.

I. IIRS Data Flow

Figure 2 presents the flow of data between the user and the different processes contained in the system. It shows how data migrate from the user to the controller actions, then to the database and vice versa.

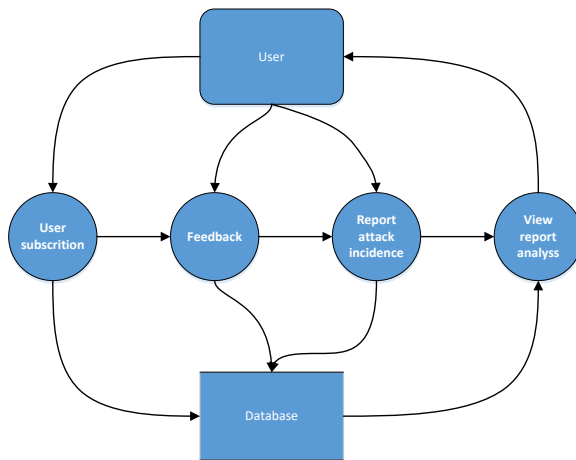


Figure 2: The Data Flow Diagram

IV SYSTEM IMPLEMENTATION

A. The View

The Razor View Engine was adopted for the implementation of the front-end. Razor Pages is a new feature of ASP.NET Core MVC that makes coding page-focused scenarios easier and more productive. To use the Razor View Engine, it has to be enabled in the *Startup.cs*, file, by adding *services.AddMvc()* directive to the *ConfigureServices* method of the Startup class. Afterward, the *@page* directive is hard coded on every view to make the file into an MVC action. *@page* directive must be the first Razor directive on a page. Razor Pages features are designed to make common patterns used with web browsers easy. Tag Helpers, HTML helpers and Model binding, all just work with the properties defined in a Razor Page class.

The View includes a Reporting platform as indicated in Figure 3, where the user can submit report of any attack incidence. Part of the view is also a page dedicated for raising cyber-security awareness, as presented in Figure 4, where users can learn about the most occurring cyber-attacks. Figure 5 provides an interface for users to get in touch or send a feedback to the system administrator. Also, included are layouts containing cylindrical chart (as indicated in Figure 7), and progress bar and general rating table presented in Figure 6. And part of these visualization tools was also used to present the percentage distribution of attacks based on the different forms of identities exposed or based on the top high level sectors affected.

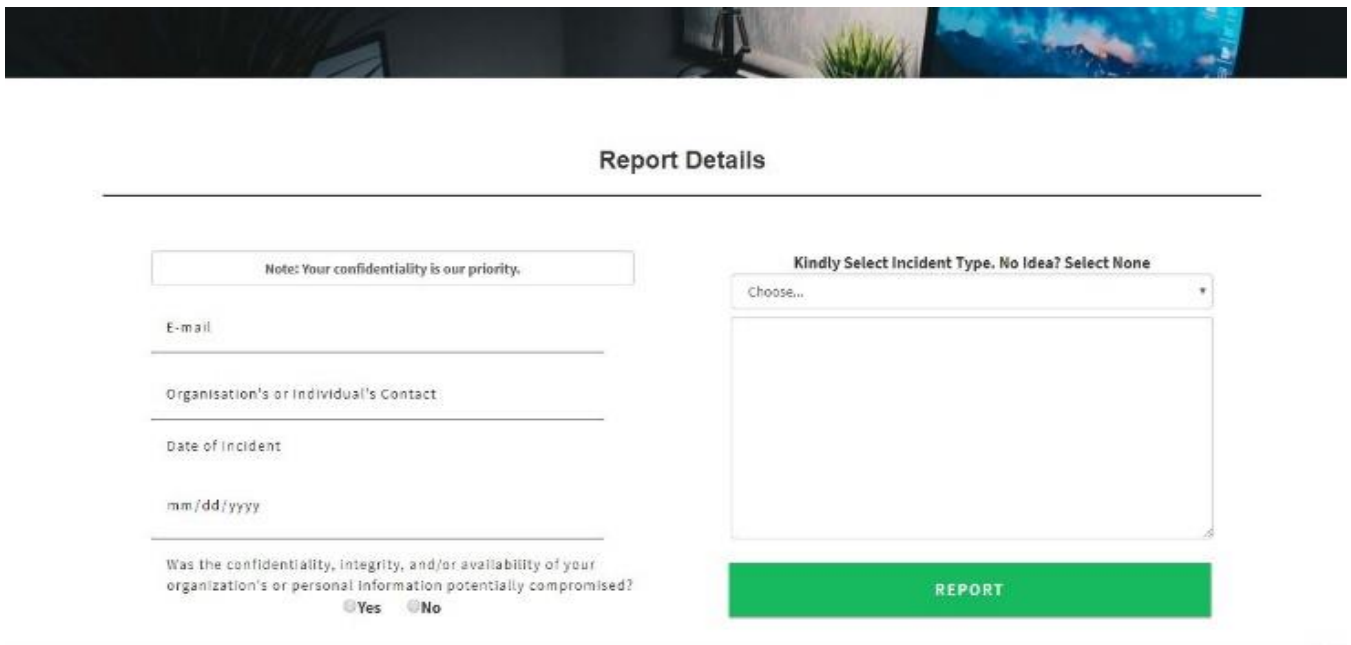


Figure 3: Reporting Interface

## Get To Know

	<h3>Backdoor Attack</h3> <p>Any secret method of bypassing normal authentication or security controls. They may exist for a number of reasons, including by original design or from poor configuration.</p> <p><a href="#">Read More &gt;</a></p>		<h3>Denial-of-service</h3> <p>Attackers can deny service to individual victims, such as by deliberately entering a wrong password enough consecutive times to cause the victim account to be locked, or they may overload the capabilities of a machine or network and block all users at.</p>
<h3>Phishing</h3> <p>Phishing is the attempt to acquire sensitive information such as usernames, passwords, and credit card details directly from users. Phishing is typically carried out by email spoofing or instant messaging.</p>		<h3>Spamming</h3> <p>Email spam, also known as junk email, is a type of electronic spam where unsolicited messages are sent by email. Many email spam messages are commercial in nature but may also contain disguised links hosting malware.</p>	

Figure 4: Cybersecurity Awareness Page

## Send A Feedback

Home > Feedback

## Get In Touch





<h3>Get In Touch</h3> <p>Full Name <input type="text"/></p> <p>Your Mail <input type="text"/></p> <p>Mobile Number <input type="text"/></p> <p>Message <input style="height: 80px;" type="text"/></p> <p style="text-align: center;"><a href="#" style="background-color: green; color: white; padding: 5px 10px;">SEND</a></p>	 <p> <span style="display: inline-block; vertical-align: middle; margin-right: 10px;">  Address                      PMB 2240, FUNAAB                      Abeokuta, Ogun State.                 </span> <span style="display: inline-block; vertical-align: middle; margin-right: 10px;">  Phone                      +2348131167172                 </span> <span style="display: inline-block; vertical-align: middle;">  Email                      excellent1491@gmail.com                 </span> </p>
---	---

Figure 5: Feedback Page



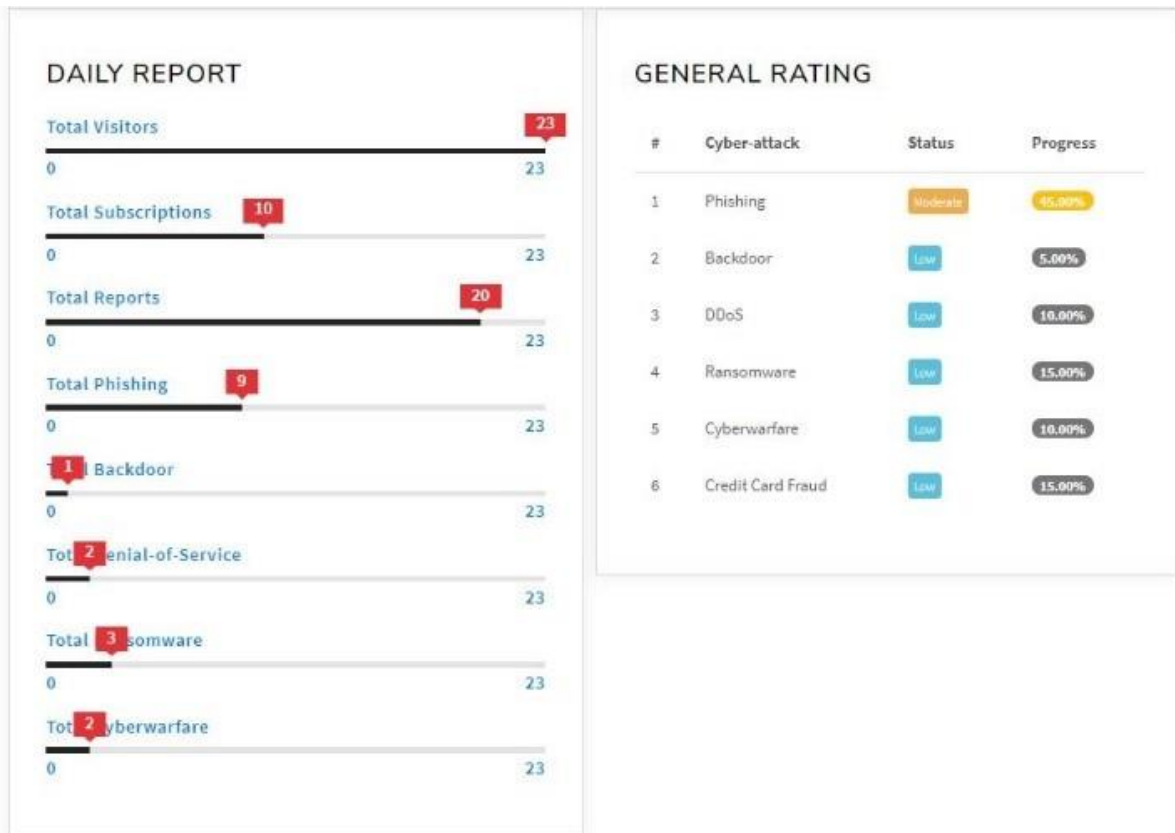


Figure 6: Daily Report History

The cylinder chart presented below shows the overall distribution of reported incidences based on their corresponding categories.

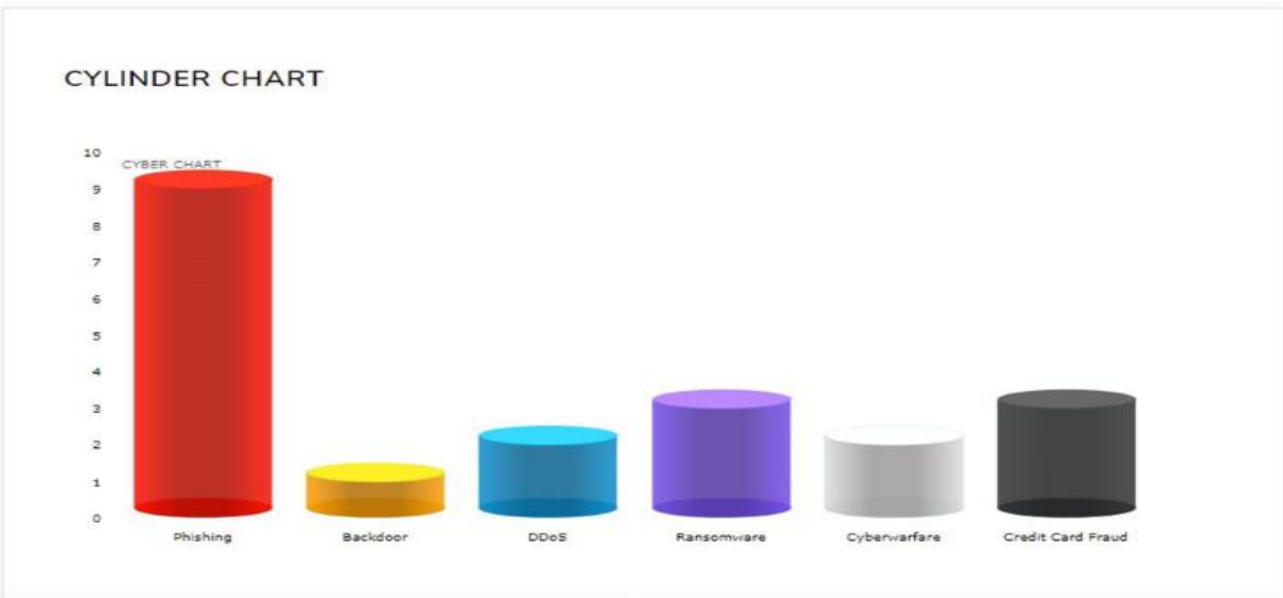


Figure 7: Cyberattack Statistics



## B. The Model

C# classes are used to represent each entity of interest which include the user's profile, report details and the type of attack incident being reported, among others. The getter and setter methods are set for each property and appropriate data annotations are used to achieve the required validation and authorization. The *DataAnnotation* Class also contain formatting attribute that are applied declaratively to a class or property, like *DataType* which help with formatting and not validation. The *DisplayName* attribute is set where required and other constraints such as the *MinimumLength*, *MaximumLength*, *RegularExpressions* are all taken into considerations.

These model classes are used with Entity Framework (EF) to work with the database. EF Core is an object relational mapping (ORM) framework that simplifies the data access code that is required to manage the database.

## C. The Controller

The system logic and controls are handled by the controller. Every public method in the controller is callable as an HTTP endpoint. An HTTP endpoint is a targetable URL in the web application, such as *http://localhost:1234/report*, and combines the protocol used: HTTP, the network location of the web server (including the TCP port):localhost:1234 and the target URI `report`. MVC invokes controller classes (and the action methods within them) depending on the incoming URL. The default URL routing logic used by MVC uses a format like this to determine what code to invoke: */{Controller}/{ActionName}/{Parameters}*.

## V EVALUATION AND RESULTS

### A Semantic Matching Result

A sample result of some extracted entities in a reported detail is presented in table 1. The score between each pair represents the similarity score and it is based on the Wu & Palmer formula in equation 1. A score of 0.90 reveals a very high degree of relatedness between the entity "credit" and "debit" or 0.89 between "payment" and "transaction". While a score of 0.22 shows a low relatedness between "email" and "invalid". In general, as the similarity score ranges from 0 to 1, the degree of relatedness can be described as either low, moderate, or high.

**Table 1: Extracted Entities Similarity Matching**

extracted entities	email/NN	credit/NN	payment/NN	account/NN	password/NN	fraud/NN
transaction/NN	0.67	0.71	0.89	0.71	0.40	0.63
card/NN	0.38	0.67	0.44	0.71	0.89	0.70
message/NN	0.40	0.83	0.46	0.83	0.77	0.38
debit/NN	0.30	0.90	0.33	0.47	0.44	0.29
code/NN	0.38	0.62	0.43	0.63	0.57	0.35
invalid/NN	0.22	0.27	0.25	0.27	0.25	0.70

### B Performance Evaluation

Twenty students of the Department of Computer Science, Federal University of Agriculture, Abeokuta were invited to test the system on a local server, taken some selected attributes into consideration. The result of this evaluation is presented in Table 2.

**Table 2: System Evaluation**

Attribute	Exceptional (%)	Exceeds expectation (%)	Meets Expectation (%)	Below Expectation (%)	Needs Expectation (%)
Ease of use	95.0	5.00	0.00	0.00	0.00
Functionality	90.0	10.00	0.00	0.00	0.00
Stability	95.0	5.00	0.00	0.00	0.00
Consistency	95.0	0.00	5.00	0.00	0.00
Awareness	95.0	5.00	0.00	0.00	0.00
Contribution	94.0	5.00	1.00	0.00	0.00
Average rating	94.0	5.00	1.00	0.00	0.00

On average, the system was rated 94% exceptional with respect to the selected attributes, while only 5% rated the system to exceed their expectation, and it was able to meet up the expectation of 1% of the selected audience. This result shows an excellent level of acceptance and as well emphasizes the importance of such system in the quest to improve cyber-security culture in the ever-growing cyberspace.

## VI CONCLUSION AND RECOMMENDATIONS

The proposed solution was designed and implemented under the ASP.Net Core (C#) environment and the result of the evaluation as shown in table 2 reveals the degree of effectiveness and acceptability of the system. Among the twenty attack incidences reported during the evaluation period, nine were identified as phishing attacks, one as backdoor attack, two each as distributed denial of service (DDoS) and cyberwarfare attacks, three attacks each was as well classified as ransomware and credit card fraud, as indicated in figure 7, and the classification was only based on the extracted named entities identified in the reported details for each incidence. With this achievement, the system demonstrated automatic classification of reported incidences for all categories of users, both knowledgeable or ignorant of the kind of attack techniques they've suffered.

95% of the respondents rated the security awareness facts provided as being exceptional with respect to the contribution to knowledge achieved by the awareness details provided. And 95% rated the cybercrime statistics has been beneficial and capable of providing cybersecurity alertness for users.

The semantic similarity analysis illustrated in table 1 reveals the ability of the system to relate entities extracted from reported details even when the named entities do not have similar base words. The semantic similarity scores form the basis of classification of reported incidences.

This study has achieved a framework for integrating natural language processing capability into the core concept of cyber-security incident reporting, to facilitate online real-time analysis of reported attack incidences. And as well accomplished a real-time availability of cybercrime statistics using visualization tools such as cylindrical chart, progress bar, etc. as indicated in figure 5 and 7.

Convincingly, as cyber threats are increasing on a daily basis and protection measures such as firewalls and anti-virus software have proven insufficient, the adoption of this system would be useful in thwarting cyber-attacks through cyber-security awareness raising, encouraging incident reporting, promoting cyber-security culture, and providing real-time statistics to users, as well as security experts who can help in deploying systems to manage vulnerabilities and prevent future reoccurrences.

## REFERENCES

- [1] R. Sharma (2016). *Study of Latest Emerging Trends on Cyber Security and its challenges to Society*. International Journal of Scientific & Engineering Research.
- [2] Cisco. (2017). *Cisco Annual Cybersecurity Report*. Journal of Cybersecurity.
- [3] M. Edwards (2017). Panning for gold: Automatically analysing online social engineering attack surfaces. *Computers and Security*.
- [4] F. O. Sveen, J. M. Sarriegi, E. Rich, J. J. Gonzalez, (2007) "Toward viable information security reporting systems", *Information Management & Computer Security*, Vol. 15 Issue: 5, pp.408-419, <https://doi.org/10.1108/09685220710831143>
- [5] C. W. Johnson (2015). Architectures for Cyber-Security Incident Reporting in Safety-Critical Systems.
- [6] T. Koontorm (2018, February). *Tan, Koontorm Center. "Phishing and Spamming via IM (SPIM)*.
- [7] M. Kuypers, & P.-C. Elisabeth (2015). *Working Paper: Documenting Cyber Security Incidents*, Stanford University.
- [8] J. Lim & S. A. Maynard, A. C. Shanton. (2015). Information Security Culture: Towards an Instrument for Assessing Security Management Practices. *International Journal of Cyber Warfare and Terrorism (IJCWT)*. 5. 10.4018/IJCWT.2015040103.
- [9] T. C. Lin (2017). "The New Market Manipulation". *Emory Law Journal*. 66: 1253. SSRN 2996896.
- [10] V. D. Merwe, A. J. Looock, M. Dabrowski (2010) *Characteristics and Responsibilities involved in a Phishing Attack, Winter International Symposium on Information and Communication Technologies*. Cape town.
- [11] G. Morrie. (1988). *Building a Secure Computer System*. Van Nostrand Reinhold.

- [12] Z. Ramzan (2010). Phishing attacks and countermeasures. In P. Mark & Stavroulakis, *Handbook of Information and Communication Security. ISBN 978-3-642-04117-4*. Springer.
- [13] A. K. Verma. A. K. Sharma (2015). *State of Cybersecurity and Cyber Crime*. International Journal of Advanced Computer Research.
- [14] Symantec. (2017). *Internet Security Threat Report*. Symantec Cybersecurity Report.
- [15] J. R. Finkel, T. Grenager, and C. Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In ACL 43, pages 363–370.
- [16] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In NAACL 3, pages 252–259.
- [17] Angel X. Chang and Christopher D. Manning. 2012. SUTIME: A library for recognizing and normalizing time expressions. In LREC 2012.
- [18] Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In LREC 2006, pages 449–454.
- [19] Dan Klein and Christopher D. Manning. 2003. Fast exact inference with a factored model for natural language parsing. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15, pages 3–10. MIT Press.
- [20] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In EMNLP 2013, pages 1631–1642.