Research Article / Review Article FUPRE JOURNAL 7(3): 61-78 (2023)

FUPRE Journal
of
Scientific and Industrial Research

ISSN: 2579-1184(Print) ISSN: 2578-1129 (Online)

http://fupre.edu.ng/journal

Hybrid Integration of Organizational Honeypot to Aid Data Integrity, Protection of Organizational Resources and Dissuade insider Threats

EJEH, P.[1], ADISHI, E.[2], OKORO, E.[3], JISU, A.[4]

[1]*Department of Computer Science, College of Computing and Telecommunications, Novena University, Ogume, Delta State, Nigeria*
[2,3,4]*Department of Intelligence and Security Studies, College of Management and Social Sciences, Novena University, Ogume, Delta State, Nigeria*

ARTICLE INFO

Received: 20/4/2023

Accepted: 07/06/2023

Keywords

Application server, Data integrity, Deception Honeypot, Intrusion Prevention, Intrusion Detection, Organization data

ABSTRACT

The world is rapidly converging and converting a variety of valuable data to its digital equivalent. Its dissemination as eased via the advent of the Internet – has also led to undue attacks/threats due to predictable responses from users. This has evolved the field of social engineering and exploits of user trust-levels. Thus, the use of deception mechanisms now plays a prominent role in enhancing the field of cyber and data security. There exist many techniques to dissuade and/or redirect adversaries via deception mode commonly called honeypots. They seek to detect intrusive activities via services that attract adversaries to it. Honeypots have been successfully used to minimize security breaches. Thus, our study explores deception-based honeypot as an effective means to minimize data exploitation – and implemented via web servers that are now equipped with tracing cum identification capabilities as system learns and defends its user systems against intrusive actions.

1. INTRODUCTION

Our society continues to experience new threat to digital data. Ability to guard and protect data has become both critical and importance to field of informatics (Akazue et al., 2022, 2023). A net administrator must protect the network and data with extreme and diverse measures. One of such measure is the use of honeypots. Honeypots trick adversaries into believing they have accessed the network to a certain degree – and the only way out for an administrator to block them out is to check their logs to see who accessed the network and how they accessed it (Abbasi et al., 2016; Alsowai & Al-Shehari, 2021; Brause et al., 2002; Broadhurst et al., 2018).

Honeypots lure intruders around a network long enough for their identity and other details to be extracted and revealed. Thus, it is a resource that works over a network with fake data that mimic the illusion of real data to an attacker (Algarni et al., 2017; Gokarn & Choudhary, 2021). It is designed to behave as a real host so that it can attract an adversary. Its main thrust is to be savvy to keep an adversary engaged long enough to exploits his/her information. It obtains and monitors the activities or operations of an adversary via a Trojan horse that stores interaction between the adversary and the honeypot. It achieves these via the use of analytical tools that seek to investigate the reason(s) for attack (Barlaud et al., 2019; Benchaji et al., 2021; Ibor et al., 2023; Ojugo, Abere, Eboka, Yerokun, et al., 2013; Ojugo, Oyemade, Yoro, Eboka, et al., 2013; Ojugo, Yoro,

*Corresponding author, e-mail:*kemmyadesanya@gmail.com*
DIO

Oyemade, et al., 2013).

A honeypot can simulate internal/external resources like devices, apps and database servers, firewalls etc (Kumaraguru et al., 2010; Mahajan & Sharma, 2015). Its value as a security resource lies in being attacked, probed, compromised and/or exploited. A honeypot can also act as a detection-response tool (Artikis et al., 2017; Ojugo & Yoro, 2020a). It acts as a server decoy, luring in potential hackers in order to study their tasks and monitor how they intrude into system as well as evade detection or capture.

By design, they mimic systems an intruder would like to access. A honeypot simulates/engages an adversary so that he/she is unaware of being tricked or monitored. It is best installed within a firewall for better control. There is lesser control for an adversary when they are installed outside of the network's firewalls (Ojugo, Abere, Orhionkpaiyo, Yoro, et al., 2013; Ojugo & Yoro, 2021b). A honeypot in a firewall works has its operation reverse-engineered from that of a normal firewall – such that instead of restricting what comes into a network, the new system allows traffic to come in but restricts what the system sends back out.

By luring an intruder into a system, a honeypot serves several purposes to include: (a) allows a network administrator to monitor an intruder exploit of network vulnerabilities, (b) a net-admin learns where the network is most vulnerable and marks such for redesign, (c) a net-admin can reveal the true identify of an intruder via the extracted data, and (d) net-admin can stop intruders from accessing root directory. By studying the activities of an adversary, the network designers can create a more secure system that is potentially invulnerable to future intruders. With data traffic detected in and out of a honeypot, it implies the system has been compromised. Thus, a honeypot waits to monitor and study inbound/outbound traffic logs (Ojugo et al., 2014; Ojugo & Eboka, 2018, 2019b).

1.1 Honeypots based on Architecture

Ezpeleta et al. (2020) developed a low interaction client-side honeypot to detect malicious web servers. It was tested against 84 malicious and 10 benign uniform resource locators with 98% success rate (Ezpeleta et al., 2020). Its data was processed by an external engine averaging 60sec/URL with a 17seconds capture. They conclude that the high/low-interaction honeypot should be integrated to harness the benefits of both schemes in future honeypot design. He noted that for ease in installation – low-interaction honeypot requires external data analyses, while high-interaction honeypot have opposite properties.

Paliwal (2022) used a hybrid low/high-interaction honeypot to achieve lower resource requirements for implementing low- interaction honeypots and to emulate the full responses in high- interaction honeypots. It used a proxy running in each host and responsible for generating virtual hosts and redirecting traffic. Each virtual host emulates a fully functional high-interaction honeypot (Artikis et al., 2017; Paliwal et al., 2022). And on invocations, they seek to minimize resource consumptions as high-interaction are automatically invoked when traffic that requires such high-interactions arrive at a host. His system allowed monitoring of malicious activities by bots, worms and viruses – not letting them leave a honeypot (Rathi & Pareek, 2013; Redondo-Gutierrez et al., 2022). It redirects outgoing traffic via messages that coordinates attacks to other honeypots, prevents malicious attacks to other production servers as well as prevents detection of a honeypot by an intruder (Sahmoud & Mikki, 2022; Sohony et al., 2018). Through this mechanism, an intruder is made to believe that their media still interacts with hosts outside the network; while, they are actually communicating with another honeypot in the same network (Tingfei et al., 2020).

Verma (2020) notes trade-off issues

between system accuracy (to reduce false-positive and false-negative) for high-interaction honeypots in anomaly intrusion detection (Verma et al., 2020). He proposed shadow-honeypots, real production network applications with honeypot embedded therein such that incoming requests to a server executes the shadow honeypot just in same manner as by production server. But, embedded honeypot codes will monitor the behaviours of each request (Ojugo & Yoro, 2021a; Yoro, Aghware, Akazue, et al., 2023; Yoro, Aghware, Malasowe, et al., 2023). If a request is considered and/or confirmed malicious, activities executed by such is rolled back. Thus, incoming request is first processed by an anomaly-detection scheme, configured intentionally with high false-positives. The requests classified malicious are then forwarded to the shadow honeypots; while, those classified as genuine are directed to a production server (Yoro & Ojugo, 2019b, 2019a).

1.2 Honeypots on Malware Detection

Zawislak et al. (2022) proposed the double honeypot to extract signatures for detecting polymorphic worms to achieve their zero-day detections (Zawislak et al., 2022). It consists of inbound honeypots, outbound honeypots, and address translators. Tang (2017) also proposed a new method of data analysis applied to the data collected by the double honeypot called Position Aware Distribution Signatures which counter-utilizes the fact that worms can open outgoing connections (Feng et al., 2013; Tian, 2016; Yan et al., 2018). By monitoring unexpected outgoing connections from an inbound to an outbound honeypot, worms can be identified easily. PADS was designed to increase the chances of detecting polymorphic worms by allowing possible variations in a signature, instead of "all fixed symbols" in the existing signatures. To control" variations" in each position in signatures, PADS uses the byte frequency distribution, which specifies what variations are how likely possible in each position in a signature string (Correia et al., 2015; De Kimpe et al., 2018).

Zhang et al. (2007) proposed Honeybow to automatically detect and capture malware without requiring human experts manually investigating output data from honeypots. It has several parts: MmFetcher, detects modification of files by comparing their initial MD5 hash after it intentionally lets malware modify its files (Okonta et al., 2013; Y. Zhang et al., 2007). If a modification is detected, the process that made such modification is captured as malware, while ***MmFetcher*** restores initial copy of all files. The ***MmWatcher***, monitors system calls that perform file creation and modification to trigger intrusion detection. Finally, the MmHunter monitors code being executed like a debugger to detect malware's suspicious activities (Jayatilaka et al., 2021; Laavanya and Vijayaraghavan, 2019; Sasikala et al., 2022).

Tingfei et al., (2020) used a new architecture, which detects worms by monitoring rate of their outgoing connections. The new architecture slows down worms by throttling the rate of creation of new outgoing connections based on a closed feedback loop control for throttling outgoing connections (Tingfei et al., 2020) and applied the proportional, integral and derivative (PID) model that throttles outgoing connections to a designated set point for reducing spreads of worms (Khaki et al., 2020; Wang et al., 2019). The algorithm significantly slowed down the spread of worms. With a containment to kill infected processes and blacklist hosts using multiple-feedback loops and intelligently queuing the connections, the spread of worms can be stopped with a significantly fewer number of hosts infected (Brofman Epelbaum & Garcia Martinez, 2014; Ojugo et al., 2015; Ojugo & Eboka, 2021; Ojugo & Oyemade, 2021).

1.3. Honeypots on Configuration

A major factor inhibiting the adoption of honeypots is the difficulty in tuning them

as they are simply traps that monitors an adversary's activities without being detected. Thus, like real traps, they must be carefully configured to attract the right targets. A honeypot operator must consider what activities should be monitored, whose activities should be monitored, when such activities should be monitored etc. If incorrectly configured, a honeypot not only fails to draw its prey, but can or may also be hijacked by an adversary. Its blind capture of a huge volume of network activities can also make subsequent data processing extremely difficult. Thus, is extremely important that honeypots are configured to satisfy their specific objective(s) (Hong, 2018; Nivedha and Raja, 2022; Ojo et al., 2021).

With high-interaction honeypots to monitor anomalies and possible abuses, hijack and to automate frequent reinstallations of illegally modified high-interaction honeypot. The solution sought a trade-off between low/high-interaction honeypots. It frequently monitor and examine the current status of each high-interaction honeypot deployed in a network through intrusion detection schemes, and automatically reinstalls system if any anomaly is detected (Filippov et al., 2008; Gao et al., 2021).

Fatahi et al. (2016) a real challenge is how to update clean system image as often, each system can change in legitimate ways – requiring logging at every single activity in each high-interaction honeypot. Frequent reinstallations will increase downtime – which can serve as another target for a DoS attack (Fatahi et al., 2016). We can use games theory to determine optimal distribution of honeypots classifying 4-types of systems: normal production systems, fake normal (honeypots camouflaged production system), honeypot (not camouflaged), and fake honeypots (production systems camouflaged as honeypots) (Goel et al., 2017; Gratian et al., 2018; Halevi et al., 2013). They developed models to achieve equilibrium between attackers and defenders using the four classifications. They used case studies to show how models could be applied to distributions of honeypots. Although the models took a naive view of the defender-attacker relationship, these techniques can be useful in distribution of honeypots, and if combined with a dynamic honeypot distribution technique as (Huang et al., 2021; Ileberi et al., 2022; Q. Li et al., 2017).

Halevi (2013) suggested ways to setup a honeypot specifically for detecting and studying SQL injection attacks. Chen argued that honeypot should be highly interactive for database related activities. He used non-production systems in server to make the honeypot appears as real as possible. It allows an attacker access up to data manipulation. Monitoring and restricting the use of certain procedures were recommended since these could be used to alter the system. He suggested that a proxy between the web and database servers could be used to stop certain SQL commands from reaching the database (Lakshimi and Kavila, 2018; C. Li et al., 2021). Their design used honeynet to simulate a real network. This honeynet would forward all SQL injection attacks to a high interaction honeypot with a database server (Ojugo & Ekurume, 2021a, 2021b). Database should be populated with real-like data called honeytokens and should not be easy to access.

Seleznyov (2002) used Honeyd for dynamic deployments of low interaction honeypots based on the needs detected by network scans. It uses Nmap to scan a network and gather details on the systems on the network, including operating systems and open ports. After the entire network is scanned, the configuration manager deploys and starts low-interaction honeypots based on the configuration files (Ojugo & Eboka, 2019a, 2020b, 2020a). The configuration files are introduced to allow administrators to set open ports and a network address to each honeypot, as well as to specify which server services should be emulated at each honeypot. For high interaction honeypots, we can use passive network scanning to

create a more realistic honeynet, dynamic modification for running the honeypots. This will in turn consequently, improve the overall performance in the system emulation by the honeypots (Ojugo et al., 2012; Parsons et al., 2015).

The challenges in designing and implementing honeypots is ensuring data integrity, confidentiality, availability, and privacy – knowing that an adversary will continually seek to breach a network and evade detection. This study is motivated therefore by these challenges (Ojugo & Nwankwo, 2021a, 2021b, 2021c, 2021d; Yildiz Durak, 2019; Zanin et al., 2018):

1. Previous researches and deployment of security measures geared towards ensuring that adversaries are detected and kept at bay remains an error-prone and continuous task.
2. Adversaries evolve to evade detection via adoption of new techniques. This is now a chronic issue
3. Use of firewalls only in its faulty packet filtering methodology can make adversary evasion more possible
4. Use of application gateways has several demerits such as cost and the bottleneck of its slowing down network speed accounts for performance inefficiency and ineffectiveness
5. The chaotic nature of signature/anomaly-based detection data makes adaptation by an attacker, flexible and robust.

Deception though not often used, plays a critical role, which is fundamental since it differs from conventional security. This is because it aims to manipulate an adversary in a way, beneficial to an administrator. It is an important and effective way that compensates for conventional measures and inherent vulnerabilities therein (Ojugo et al., 2021; Ojugo & Obruche, 2021). Use of firewalls, gateways, and intrusion detection systems, have their benefits and bottlenecks. Concern arises as adversaries continue to change their tactics that has led to faulty detection. To overcome these, we implement a hybrid firewall, IDS and honeypots explicit design. Honeypots are a new technology – powerful and have great potentials as they can detect new attacks even before data is compromised (Ojugo & Otakore, 2018b, 2018a, 2020).

2. MATERIALS AND METHODS

2.1 Basic Experimental Honeypot Set-Up

We set-up our honeypot to specifically detect and study SQL-code injection via non-production systems on a webserver to make the honeypot appears real as possible. This allows adversaries access the network to level of data manipulation. The system is monitored cum restricted via use of certain procedures such as addition of a proxy between the database and webservers. This will stop all SQL commands from reaching network database. The honey-net design simulates to a real network; And, consequently help forward all SQL injection attacks to the honeypot with any server (Ojugo, Yoro, Oyemade, et al., 2013; Ojugo, Yoro, Yerokun, et al., 2013; Ojugo & Yoro, 2020b).

Specifically, if the honeypot is designed to protect the database, the database is then populated with real-like data called honey-tokens (i.e. data that looks real enough and can be traced when it is accessed/used). The honey-net with firewalls, gateways and IDS configuration will ensure the uneasy access to the designated server. The system will achieve: (a) honeypot easily identifies system vulnerabilities as an adversary tries to access the honey-net and used them, (b) honey-net will gather data that seeks to identify methods used by the adversary to capture data, (c) a honey-net will seek alternatives to such attack with various purposes aimed to capture, delete or alter data in the server, (d) knowing that some adversaries access a network via malicious script on a user browser and/or via malware – the honey-net will monitor, redirect any user to other sites via designated URLs and/or detect user source IP, (e) for connection logs recorded, a honey-net will

show which attack was done more frequently on the web application, (f) a honey-net seeks to unveil the source IP of an adversary via trace-back tools to track the source of an attack, (g) studying the connection logs, it tells an administrator the pattern of attacks that were successful and those that failed, (h) the recorded data on the honeypot will also further show tools and techniques are employed by hackers (Xuan et al., 2018; Yeboah-Boateng & Amanor, 2014; Zareapoor & Shamsolmoali, 2015; D. Zhang et al., 2020).

2.2 Network Structure: Feasibility Study

To successfully deploy a honeypot, we must correctly setup its architecture on a firewall so that it performs well. There are no rules on deploying a honeypot. For effectiveness – there are three elements that defines its architecture (Al-Qatf et al., 2018; Altman, 2019; Ojugo & Okobah, 2018; Okobah & Ojugo, 2018):

1. Data capture: The system tries to monitor log activities in the honeypot. To ensure effective data capture, the honeypot system uses several methods as no single layer can capture all the data required

2. Data Control – controls the activities of an attacker. It grants access to an adversary to the network; But, traps them from redirecting their access to other parts of the network. This is achieved by isolating the target systems in honeynet with a layer-two bridge-device so as to control the amount of data the adversary has access to. It controls an adversary's out- bound activities by limiting their capability and permission. For our study, we use embedded honeynet (grants entry but blocks all outbound connections). A well-implemented honeynet successfully blocks all out-bound connections, stops the adversary from harming other systems, and its real value is in its ability learn and track what an adversary does once access is granted or have been obtained.

3. Data Collection: With captured data stored in a central location, the value of a honeynet is embedded in its ability to review captured attacks. A honeynet easily detect and capture attack, including an attacker's keystrokes off a compromised system. This helps a network administrator to study and determined the nature of the adversary, commands executed on remote system and source Address, MAC number, Browser, etc as acquired from the honeypot log. These are reviewed to enhance the new creation of newer honeypots.

2.3. Experimental Framework Design

Honeypots are generated on-demand to meet the needs of the designated network. Thus, with such scheme – an adversary meets with an obscure network that redirects him/her to a newly created honeypot as he tries to connect to the victim host. The machine will remain secure from any of such malicious attacks. Proposed honeypot on-demand will allocate some resources on a network emulating real processes using either low- or high-interaction processes.

For our proposed system, we employ a hybrid low and high interaction request via SQL injection to make the system more dynamic. Such a dynamic honeypot will radically revolutionize the deployment and maintenance of these real-life-like processes. The use of firewall on the hybrid honeypot will then help us monitor and observe the networks in real time. In setting up honeypot specifically for detecting and studying SQL injection – suggests using the non-production systems by the server to make the honeypot appears as real as possible. Its setup allows attackers access up to the point of data manipulation.

Monitoring and restricting use of certain processes is also recommended. This simply disallows an adversary or does not grant such attacker the permission to alter the target machine (honeypot). This is achieved via the use of proxies between the Web and Database Servers to stop use of certain SQL-

injected commands from reaching and accessing the database as in fig 1. Proposed design uses a honeynet to simulate a real network, which forwards all SQL injection attacks to the database server (already populated with real-like data called honey-tokens). These are data that cannot be traced back to any source when it is used. Thus, making the database not be too easy to access.

With hybrid design allows for low-interaction services that attackers typically seek. They are easy to maintain, and less resource intensive. They are also harder to use as a launch point for attacks on other systems. Our high- interaction honeypots execute and runs all other services that a production system runs – including a proper operating system.

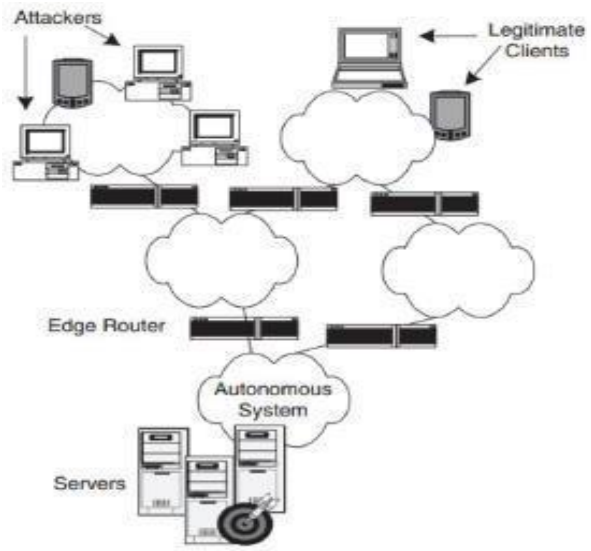

Fig. 1. Web-server with a Honeypot scheme

This hybrid design allows the deploying organization to learn a lot more about the attacker's behaviour and attack methods. Also, detecting that it is a honeypot is much harder as it mimics production systems. But these systems are more resource- intensive and are harder to set up and maintain. These are also loaded with more capabilities. Thus, are more likely to be used to attack other systems. The network administrator thus, must be liable (referred to as downstream liability) for such attacks. Our mid-interaction honeypots emulate component aspects of the application layer – providing interactivity that are unlikely and unsuitable to be used for other attacks.

With these, to an adversary – intrusion into such a designed system (honeypot) appears as a legitimate target machine or system, running processes a production system is expected to as well as containing files that seem important and real enough – even though they are not. Thus, it appears real enough to have the proper sniffing and logging capabilities, alongside dummy files and processes. Since by definition, traffic in and/or out of a honeypot is malicious – it is best practice as implemented by us to place the honeypot inside a Web server. This enhances also the logging and alerting capabilities as well as provides a way to block outgoing traffic, so that it cannot be used as a launch point for attacks on other systems.

2.4. Honeypot Detection Phase

Here, this phase observes the details entered by users into the network – sensing if the user exceeds permission granted him or her. If true, it alerts the network administrator of an intruder as well as requests permissions to launch a capture. We use the SQL-injection technique to achieve this, which displays the data on a web page – as it is common to let web users input their own search values and also have access to their own files. Since SQL statements are text only, it is easy to dynamically change SQL statements to provide the user with selected data as thus in listing as in algorithm 1. Code creates a select statement by adding a variable (txtUserId) to a select string. The variable is fetched from the user input (Request) to the page. With the honeypot knowing that the SQL injection is a technique where malicious users can inject SQL-commands into an SQL statement, via web page input. Injected SQL commands can alter SQL statement and compromise the security of the application. Our detection phase automatically launches the logging

capabilities.

Listing 1: SQL-code Injection technique

```
txtUserID = getRequestString ("UserID")
txtSQL = "Select *from Users WHERE
UserID =" + txtUserID
End
```

The Logging Capabilities grants users ability to enter their login details (sign-in). The honeypot redirects the user if such SQL- injection is observed. Else, user is redirected to next valid page so that user can access their profile. If the intended purpose of the code was to create an SQL statement to select a user with that given UserId and there is nothing to prevent a user from entering" wrong" input – a user can input:
UserID: 105 or 1=1

The system observes this logic statement and automatically redirect such a user (usually an intruder) to honeypot. Thus, it yields the expected result the adversary requested – which is potentially a database. It is thus, worthy of note that there exist boundaries in a network that users cannot exceed. If a user tries to exceed this limit, they are also redirected from their present address to the Honeypot – since the system senses such a user as an attacker trying to get into the potential database.

2.5. Honeypot Capture Phase

An adversary explores the full and potentials assets on access to a network. They also observe the network architecture and capture traffic to better understand the network topology. The honeypot then captures everything used by the intruder and also keeps track of paths and guesses taken by the intruder while he/she explores through the network. These tracks are kept in a log file for later reviews as in listing 2 below:

Listing 2: Honeypot capturing a MAC Address

```
ob_start; /* turn on output buffering */
system('ipconfig /all');  /*execute external
programs to output*/
For k number of attacks
    $mycom=ob_get_contents(); /*capture
    output as variable*/
    ob_clean(); /*clean and erase output
    buffer*/
    $findme = "Physical";
    $pmac = strops($mycom, $findme);
    /*find the position of the physical text*/
    $mac=substr($mycom, ($pmac+36),
    17); /* get physical address*/
end for
echo $mac
```

The listing above captures the MAC Address of the intruder's system once he is inside the honeypot. The system also captures the intruder's Web Browser. It seeks to get the Name of the web browser the intruder is using, also gets the Version of his/her web browser and finally, it gets the Platform of which the intruder is operating on: which is the OS (i.e. Windows, Linux or Mac).

2.6. Honeypot Lure Phase

Attempting mix of exactly the assets and intelligence desired by the attacker. Tokens are designed as bait which further lure the attacker into the traps. Once the Attacker has completed his Reconnaissance, He will move Laterally to map other parts of the network and target the resources they want for Theft and/or Destruction. Assets use powerful emulation allowing traps to imitate any asset, whether it is a workstation, server or specialized devices. This Trap draws the Attacker in and traps the Attackers Malware tools. Now the Intruder is ready to Query the server to get valuable information or ready to compromise the Database for his malicious interest, the honeypot starts by enticing him to his preferred choice of query. Recall that the honeypot already knew what he wants based on his search queries; Data collection feeds the Honeypot with his demands and paths, with this information; the honeypot already knows the kind of files the intruder is looking for and tries to feed him/her with it.

The Intruder is lured to Emails, Multimedia files, Documented, and intruder feels he/she is attacking the real system.

2.7. Honeypot Termination Phase

Here, the network administrator is still keeping watch of the intruder in the network (Honeypot) because high valued targets are compromised for the attacker so that he will engage himself with complex attack procedures that require deep interaction with the traps. The Honeypot continues to extend the illusion to deeper levels. While the intruder is busy exploring the honeypot looking for information to steal or crafting plans to compromise the Database leaving no trace of him behind. The administrator has collected all the information of the intruder including his Web Browser.

3. RESULT FINDINGS AND DISCUSSION

3.1. Findings and Discussion

For the study, honeypots are deployed as a web application on the Windows Servers with all deployments and configuration done on production and virtual machines. The honeypots are designed to be accessed as a web application running on the servers (a replica as will be considered during the actual attacks by an adversary). Thus, the attackers see the web application and then attack it finding it vulnerable as in figure 3; while figure 4 shows the cloud technology and firewall logins of some users on the network system.

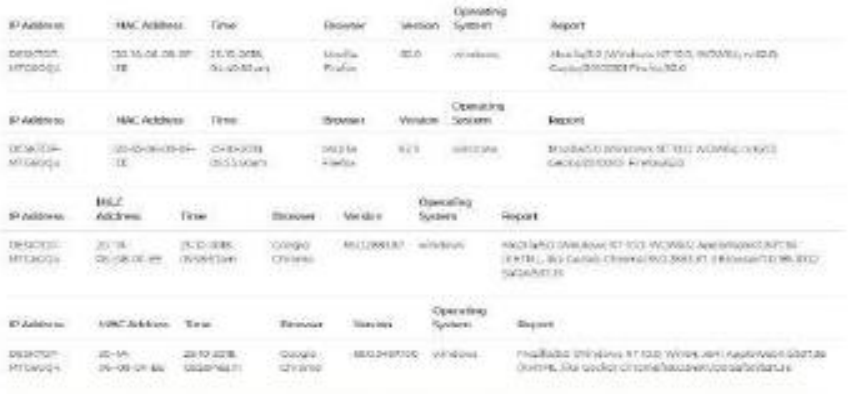

Figure 3. Snapshot of system trapped by the honeypot

id	fname	lname	email	password
1	Andrea	Bocelli	andrea.bocelli@gmail.com	password
2	Sam	Smith	sam.smith@gmail.com	password
3	Troye	Sivan	troye.sivan@gmail.com	password
4	Hayley	Kiyoko	hayley.kiyoko@gmail.com	password
5	Sarah	Brightman	sarah.brightman@gmail.com	password
6	James	Bay	james.bay@gmail.com	password
7	Selena	Gomez	selena.gomez@gmail.com	password
8	Adam	Young	adam.young@gmail.com	password
9	Sarah	Liberman	sarah.liberman@gmail.com	password
10	David	Guetta	david.guetta@gmail.com	password
11	Elisa	Toffoli	elisa.toffoli@gmail.com	password
12	Kendrick	Lamar	kendrick.lamar@gmail.com	password
13	Katie	Herzig	katie.herzig@gmail.com	password
14	Delaney	Jane	delaney.jane@gmail.com	password
15	Nicky	Romero	nicky.romero@gmail.com	password
16	John	Snow	john.snow@gmail.com	password
17	Marcel	Gerald	marcel.gerald@gmail.com	password
18	Clark	Kent	clark.kent@gmail.com	password
19	Tyrion	Lannister	tyrion.lannister@gmail.com	password
20	Jamie	Lannister	jamie.lannister@gmail.com	password

Figure 4: Cloud technologies honeypot logins

Figure 3 shows the various headers of data to be captured namely: IP Address, MAC Address, Time, Browser, Browser version, operating system and Report (i.e. details of adversary). These are only captured when the intruder tries to exceed boundaries allocated to him/her. The system detects there is an invalid log in procedure and alert the Administrator. The network admin is alerted and confirms the data collection from the intruder. With log data collected, the admin chooses to track the intruder, or keep his/her data for future reference (Allenotor & Ojugo, 2017; Ibor et al., 2023; Oyemade & Ojugo, 2021).

Figure 4 shows that the system also takes note of the time the intruder made his/her first attempt escalating privileges and keep monitoring his movement as the intruder goes on. When the intruder is in the network, he/she tries to access the robots.txt, Then the system tricks the intruder by showing the intruder the page below:

Listing 3: SQL-code Injection with honeypot in action

```
robot.txt for our website
User-agent: *
disallow: /backdoor.php
```

The intruder sees the backdoor.php file and tries to load the page (at which the system fully confirms that the intruder has an ulterior motive) – and sends the intruder an error message.

Honeypots are best alongside production web server(s) or in a separate Database protected by firewall(s) and IDPS (intrusion detection and prevention system). The logs generated by Honeypot can help detect adversaries and web servers that have been compromised. Honeypot is a computer technology which is spreading day by day in virtual environment. It's a technology which is not just for a big organization, but this is also beneficial for a single computer system as it provides an additional step in security of a computer system. This technology has lots of benefit but also has some of its disadvantages as well and at present research is on to improve the efficiency of honeypot and trying to overcome its disadvantages. Global communication is getting more important every day. At same time, computer crimes are increasing.

4. CONCLUSION

Deception technology integrates real world military tactics into a new generation of cybersecurity solutions. It enhances the cyber defence landscape and allows real, "while being attacked" intervention capabilities. Instead of taking a passive role, deception technology introduces technical solutions to create engaging assets and services to lure attackers. A significant number of cyber solution pioneers have launched various products to educate the market on this new approach for coping with cyberattacks. Honeypots and deception technology will continue to grow and be employed as a common practice in cyber defence.

Countermeasures are developed to detect attacks - most of these measures are based on known attack patterns. Knowledge of attack strategy helps improve countermeasures and fixes the inherent vulnerabilities on a network. Honeypot comes into play for such purposes. It is a resource, which is intended to be attacked and computerized to gain more information about the attacker, and used tools Honeypots are closely monitored decoys that are employed in a network to study the trail of hackers and to alert network administrators of a possible intrusion. The traditional approach to security has been largely defensive so far, but interest is increasingly being paid to more aggressive forms of defence. One of these forms is decoy-based intrusion protection using honeypots.

References

Abbasi, A., Zahedi, F. M., and Chen, Y. (2016). Phishing susceptibility: The

good, the bad, and the ugly. *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, 169–174. https://doi.org/10.1109/ISI.2016.7745462

Akazue, M. I., Ojugo, A. A., Yoro, R. E., Malasowe, B. O., and Nwankwo, O. (2022). Empirical evidence of phishing menace among undergraduate smartphone users in selected universities in Nigeria. *Indonesian Journal of Electrical Engineering and Computer Science*, *28*(3): 1756–1765. https://doi.org/10.11591/ijeecs.v28.i3.pp1756-1765

Akazue, M. I., Yoro, R. E., Malasowe, B. O., Nwankwo, O., and Ojugo, A. A. (2023). Improved services traceability and management of a food value chain using block-chain network : a case of Nigeria. *Indonesian Journal of Electrical Engineering and Computer Science*, *29*(3): 1623–1633. https://doi.org/10.11591/ijeecs.v29.i3.pp1623-1633

Al-Qatf, M., Lasheng, Y., Al-Habib, M., and Al-Sabahi, K. (2018). Deep Learning Approach Combining Sparse Autoencoder With SVM for Network Intrusion Detection. *IEEE Access*, *6*: 52843–52856. https://doi.org/10.1109/ACCESS.2018.2869577

Algarni, A., Xu, Y., and Chan, T. (2017). An empirical study on the susceptibility to social engineering in social networking sites: the case of Facebook. *European Journal of Information Systems*, *26*(6): 661–687. https://doi.org/10.1057/s41303-017-0057-y

Allenotor, D., and Ojugo, A. A. (2017). A Financial Option Based Price and Risk Management Model for Pricing Electrical Energy in Nigeria. *Advances in Multidisciplinary & Scientific Research Journal*, *3*(2): 79–90.

Alsowai, R. A., & Al-Shehari, T. (2021). A multi-tiered framework for insider threat prevention. *Electronics (Switzerland)*, *10*(9). https://doi.org/10.3390/electronics10091005

Altman, E. R. (2019). *Synthesizing Credit Card Transactions*. http://arxiv.org/abs/1910.03033

Artikis, A., Katzouris, N., Correia, I., Baber, C., Morar, N., Skarbovsky, I., Fournier, F., and Paliouras, G. (2017). A Prototype for Credit Card Fraud Management. *Proceedings of the 11th ACM International Conference on Distributed and Event-Based Systems*, 249–260. https://doi.org/10.1145/3093742.3093912

Barlaud, M., Chambolle, A., and Caillau, J.-B. (2019). *Robust supervised classification and feature selection using a primal-dual method*.

Benchaji, I., Douzi, S., El Ouahidi, B., and Jaafari, J. (2021). Enhanced credit card fraud detection based on attention mechanism and LSTM deep model. *Journal of Big Data*, *8*(1), 151. https://doi.org/10.1186/s40537-021-00541-8

Brause, R., Hamker, F., and Paetz, J. (2002). *Septic Shock Diagnosis by Neural Networks and Rule Based Systems* (pp. 323–356). https://doi.org/10.1007/978-3-7908-1788-1_12

Broadhurst, R., Skinner, K., Sifniotis, N., and Matamoros-Macias, B. (2018). Cybercrime Risks in a University Student Community. *SSRN Electronic Journal*, *May*. https://doi.org/10.2139/ssrn.3176319

Brofman Epelbaum, F. M., and Garcia Martinez, M. (2014). The technological evolution of food traceability systems and their impact on firm sustainable performance: A RBV approach. *International Journal of Production Economics*, *150*, 215–224. https://doi.org/10.1016/j.ijpe.2014.01.007

Correia, I., Fournier, F., and Skarbovsky, I.

(2015). The uncertain case of credit card fraud detection. *Proceedings of the 9th ACM International Conference on Distributed Event-Based Systems*, 181–192. https://doi.org/10.1145/2675743.2771877

De Kimpe, L., Walrave, M., Hardyns, W., Pauwels, L., and Ponnet, K. (2018). You've got mail! Explaining individual differences in becoming a phishing target. *Telematics and Informatics*, *35*(5): 1277–1287. https://doi.org/10.1016/j.tele.2018.02.009

Ezpeleta, E., de Mendizabal, I. V., Gómez Hidalgo, J. M., and Zurutuza, U. (2020). Novel email spam detection method using sentiment analysis and personality recognition. *Logic Journal of the IGPL*, *28*(1): 83–94. https://doi.org/10.1093/jigpal/jzz073

Fatahi, M., Ahmadi, M., Ahmadi, A., Shahsavari, M., and Devienne, P. (2016). Towards an spiking deep belief network for face recognition application. *2016 6th International Conference on Computer and Knowledge Engineering (ICCKE)*, 153–158. https://doi.org/10.1109/ICCKE.2016.7802132

Feng, J., Fu, Z., Wang, Z., Xu, M., and Zhang, X. (2013). Development and evaluation on a RFID-based traceability system for cattle/beef quality safety in China. *Food Control*, *31*(2): 314–325. https://doi.org/10.1016/j.foodcont.2012.10.016

Filippov, V., Mukhanov, L., and Shchukin, B. (2008). Credit card fraud detection system. *2008 7th IEEE International Conference on Cybernetic Intelligent Systems*, 1–6. https://doi.org/10.1109/UKRICIS.2008.4798919

Gao, Y., Zhang, S., Lu, J., Gao, Y., Zhang, S., and Lu, J. (2021). Machine Learning for Credit Card Fraud Detection. *Proceedings of the 2021 International Conference on Control and Intelligent Robotics*, 213–219. https://doi.org/10.1145/3473714.3473749

Goel, S., Williams, K., and Dincelli, E. (2017). Got Phished? Internet Security and Human Vulnerability. *Journal of the Association for Information Systems*, *18*(1): 22–44. https://doi.org/10.17705/1jais.00447

Gokarn, S., and Choudhary, A. (2021). Modeling the key factors influencing the reduction of food loss and waste in fresh produce supply chains. *Journal of Environmental Management*, *294*, 113063. https://doi.org/10.1016/j.jenvman.2021.113063

Gratian, M., Bandi, S., Cukier, M., Dykstra, J., and Ginther, A. (2018). Correlating human traits and cyber security behavior intentions. *Computers & Security*, *73*: 345–358. https://doi.org/10.1016/j.cose.2017.11.015

Halevi, T., Lewis, J., and Memon, N. (2013). A pilot study of cyber security and privacy related behavior and personality traits. *Proceedings of the 22nd International Conference on World Wide Web*, 737–744. https://doi.org/10.1145/2487788.2488034

Hong, I. B. (2018). Social and personal dimensions as predictors of sustainable intention to use facebook in Korea: An empirical analysis. *Sustainability (Switzerland)*, *10*(8). https://doi.org/10.3390/su10082856

Huang, D., Lin, Y., Weng, Z., and Xiong, J. (2021). Decision Analysis and Prediction Based on Credit Card Fraud Data. *The 2nd European Symposium on Computer and Communications*, 20–26. https://doi.org/10.1145/3478301.3478305

Ibor, A. E., Edim, E. B., and Ojugo, A. A.

(2023). Secure Health Information System with Blockchain Technology. *Journal of the Nigerian Society of Physical Sciences*, *5*(992): 1–8. https://doi.org/10.46481/jnsps.2022.992

Ileberi, E., Sun, Y., and Wang, Z. (2022). A machine learning based credit card fraud detection using the GA algorithm for feature selection. *Journal of Big Data*, *9*(1):24. https://doi.org/10.1186/s40537-022-00573-8

Jayatilaka, A., Arachchilage, N. A. G., and Babar, M. A. (2021). Falling for Phishing: An Empirical Investigation into People's Email Response Behaviors. *ArXiv Preprint ArXiv ...*, *Fbi 2020*, 1–17.

Khaki, S., Wang, L., and Archontoulis, S. V. (2020). A CNN-RNN Framework for Crop Yield Prediction. *Frontiers in Plant Science*, *10*. https://doi.org/10.3389/fpls.2019.01750

Kumaraguru, P., Sheng, S., Acquisti, A., Cranor, L. F., and Hong, J. (2010). Teaching Johnny not to fall for phish. *ACM Transactions on Internet Technology*, *10*(2): 1–31. https://doi.org/10.1145/1754393.1754396

Laavanya, M., and Vijayaraghavan, V. (2019). Real Time Fake Currency Note Detection using Deep Learning. *International Journal of Engineering and Advanced Technology*, *9*(1S5), 95–98. https://doi.org/10.35940/ijeat.a1007.1291s52019

Lakshimi, S. V. S., and Kavila, S. D. (2018). Machine Learning for Credit Card Fraud Detection System. *International Journal of Applied Engineering Research*, *15*(24): 16819–16824. https://doi.org/10.1007/978-981-33-6893-4_20

Li, C., Ding, N., Dong, H., and Zhai, Y. (2021). Application of Credit Card Fraud Detection Based on CS-SVM. *International Journal of Machine Learning and Computing*, *11*(1):34–39. https://doi.org/10.18178/ijmlc.2021.11.1.1011

Li, Q., Chen, H., Huang, H., Zhao, X., Cai, Z., Tong, C., Liu, W., and Tian, X. (2017). An Enhanced Grey Wolf Optimization Based Feature Selection Wrapped Kernel Extreme Learning Machine for Medical Diagnosis. *Computational and Mathematical Methods in Medicine*, *2017*, 1–15. https://doi.org/10.1155/2017/9512741

Mahajan, A., and Sharma, S. (2015). The Malicious Insiders Threat in the Cloud. *International Journal of Engineering Research and General Science*, *3*(2): 246–256. www.ijergs.org

Nivedha, M. A., and Raja, S. (2022). Detection of Email Spam using Natural Language Processing Based Random Forest Approach. *International Journal of Computer Science and Mobile Computing*, *11*(2): 7–22. https://doi.org/10.47760/ijcsmc.2022.v11i02.002

Ojo, O. E., Gelbukh, A., Calvo, H., and Adebanji, O. O. (2021). Performance Study of N-grams in the Analysis of Sentiments. *Journal of the Nigerian Society of Physical Sciences*, *3*(4), 477–483. https://doi.org/10.46481/jnsps.2021.201

Ojugo, A. A., Abere, R. A., Eboka, A. O., Yerokun, M. O., Yoro, R. E., Onochie, C. C., and Oyemade, D. A. (2013). Hybrid neural network models for rainfall runoffs: Comparative study. *Advancement in Scientific and Engineering Research*, *1*(2): 22–34.

Ojugo, A. A., Abere, R. A., Orhionkpaiyo, B. C., Yoro, R. E., and Eboka, A. O. (2013). Technical Issues for IP-Based Telephony in Nigeria. *International Journal of Wireless Communications and Mobile Computing*, *1*(2): 58. https://doi.org/10.11648/j.wcmc.20130102.11

Ojugo, A. A., Ben-Iwhiwhu, E., Kekeje, O. D., Yerokun, M. O., and Iyawa, I. J.

(2014). Malware Propagation on Social Time Varying Networks: A Comparative Study of Machine Learning Frameworks. *International Journal of Modern Education and Computer Science*, *6*(8): 25–33. https://doi.org/10.5815/ijmecs.2014.08.04

Ojugo, A. A., and Eboka, A. O. (2018). Comparative Evaluation for High Intelligent Performance Adaptive Model for Spam Phishing Detection. *Digital Technologies*, *3*(1): 9–15. https://doi.org/10.12691/dt-3-1-2

Ojugo, A. A., and Eboka, A. O. (2019a). Extending Campus Network Via Intranet and IP-Telephony For Better Performance and Service Delivery: Meeting Organizational Goals. *Journal of Applied Science, Engineering, Technology, and Education*, *1*(2): 94–104. https://doi.org/10.35877/454ri.asci12100

Ojugo, A. A., and Eboka, A. O. (2019b). Signature-Based Malware Detection Using Approximate Boyer Moore String Matching Algorithm. *International Journal of Mathematical Sciences and Computing*, *5*(3): 49–62. https://doi.org/10.5815/ijmsc.2019.03.05

Ojugo, A. A., and Eboka, A. O. (2020a). An Empirical Evaluation On Comparative Machine Learning Techniques For Detection of The Distributed Denial of Service (DDoS) Attacks. *Journal of Applied Science, Engineering, Technology, and Education*, *2*(1):18–27. https://doi.org/10.35877/454ri.asci2192

Ojugo, A. A., and Eboka, A. O. (2020b). Memetic algorithm for short messaging service spam filter using text normalization and semantic approach. *International Journal of Informatics and Communication Technology (IJ-ICT)*, *9*(1): 9-18. https://doi.org/10.11591/ijict.v9i1.pp9-18

Ojugo, A. A., and Eboka, A. O. (2021). Empirical Bayesian network to improve service delivery and performance dependability on a campus network. *IAES International Journal of Artificial Intelligence (IJ-AI)*, *10*(3): 623. https://doi.org/10.11591/ijai.v10.i3.pp623-635

Ojugo, A. A., Eboka, A. O., Okonta, E. O., Yoro, R. E., and Aghware, F. O. (2012). Genetic Algorithm Rule-Based Intrusion Detection System (GAIDS). *Journal of Emerging Trends In Computing Information Systems*, *3*(8),:1182–1194. http://www.cisjournal.org

Ojugo, A. A., Eboka, A. O., Yoro, R. E., Yerokun, M. O., and Efozia, F. N. (2015). Hybrid model for early diabetes diagnosis. *Mathematics and Computers in Industry*, *50*(3–5): 55–65. https://doi.org/10.1109/MCSI.2015.35

Ojugo, A. A., and Ekurume, E. O. (2021a). Predictive Intelligent Decision Support Model in Forecasting of the Diabetes Pandemic Using a Reinforcement Deep Learning Approach. *International Journal of Education and Management Engineering*, *11*(2): 40–48. https://doi.org/10.5815/ijeme.2021.02.05

Ojugo, A. A., and Ekurume, E. O. (2021b). Deep Learning Network Anomaly-Based Intrusion Detection Ensemble For Predictive Intelligence To Curb Malicious Connections: An Empirical Evidence. *International Journal of Advanced Trends in Computer Science and Engineering*, *10*(3): 2090–2102. https://doi.org/10.30534/ijatcse/2021/851032021

Ojugo, A. A., and Nwankwo, O. (2021a). Spectral-Cluster Solution For Credit-Card Fraud Detection Using A Genetic Algorithm Trained Modular Deep Learning Neural Network. *JINAV: Journal of Information and Visualization*, *2*(1): 15–24.

https://doi.org/10.35877/454RI.jinav274

Ojugo, A. A., and Nwankwo, O. (2021b). Multi-Agent Bayesian Framework For Parametric Selection In The Detection And Diagnosis of Tuberculosis Contagion In Nigeria. *JINAV: Journal of Information and Visualization*, *2*(2): 69–76. https://doi.org/10.35877/454RI.jinav375

Ojugo, A. A., and Nwankwo, O. (2021c). Modeling Mobility Pattern for the Corona-Virus Epidemic Spread Propagation and Death Rate in Nigeria using the Movement-Interaction-Return Model. *International Journal of Emerging Trends in Engineering Research*, *9*(6): 821–826. https://doi.org/10.30534/ijeter/2021/30962021

Ojugo, A. A., and Nwankwo, O. (2021d). Tree-classification Algorithm to Ease User Detection of Predatory Hijacked Journals: Empirical Analysis of Journal Metrics Rankings. *International Journal of Engineering and Manufacturing*, *11*(4): 1–9. https://doi.org/10.5815/ijem.2021.04.01

Ojugo, A. A., and Obruche, C. O. (2021). Empirical Evaluation for Intelligent Predictive Models in Prediction of Potential Cancer Problematic Cases In Nigeria. *ARRUS Journal of Mathematics and Applied Science*, *1*(2):110–120. https://doi.org/10.35877/mathscience614

Ojugo, A. A., Obruche, C. O., and Eboka, A. O. (2021). Quest For Convergence Solution Using Hybrid Genetic Algorithm Trained Neural Network Model For Metamorphic Malware Detection. *ARRUS Journal of Engineering and Technology*, *2*(1): 12–23. https://doi.org/10.35877/jetech613

Ojugo, A. A., and Okobah, I. P. (2018). Prevalence Rate of Hepatitis-B Virus Infection in the Niger Delta Region of Nigeria using a Graph-Diffusion Heuristic Model. *International Journal of Computer Applications*, *179*(39): 975–8887.

Ojugo, A. A., and Otakore, O. D. (2018a). Improved Early Detection of Gestational Diabetes via Intelligent Classification Models: A Case of the Niger Delta Region in Nigeria. *Journal of Computer Sciences and Applications*, *6*(2): 82–90. https://doi.org/10.12691/jcsa-6-2-5

Ojugo, A. A., and Otakore, O. D. (2018b). Seeking Intelligent Convergence for Asymptotic Stability Features of the Prey / Predator Retarded Equation Model Using Supervised Models. *Computing, Information Systems, Development Informatics & Allied Research Journal*, *9*(2), 13–26.

Ojugo, A. A., and Otakore, O. D. (2020). Intelligent cluster connectionist recommender system using implicit graph friendship algorithm for social networks. *IAES International Journal of Artificial Intelligence*, *9*(3): 497~506. https://doi.org/10.11591/ijai.v9.i3.pp497-506

Ojugo, A. A., and Oyemade, D. A. (2021). Boyer moore string-match framework for a hybrid short message service spam filtering technique. *IAES International Journal of Artificial Intelligence*, *10*(3): 519–527. https://doi.org/10.11591/ijai.v10.i3.pp519-527

Ojugo, A. A., Oyemade, D. A., Yoro, R. E., Eboka, A. O., Yerokun, M. O., and Ugboh, E. (2013). A Comparative Evolutionary Models for Solving Sudoku. *Automation, Control and Intelligent Systems*, *1*(5):113. https://doi.org/10.11648/j.acis.20130105.13

Ojugo, A. A., and Yoro, R. E. (2020a). Forging A Smart Dependable Data Integrity And Protection System Through Hybrid-Integration Honeypot

In Web and Database Server. *Technology Report of Kansai University*, *62*(08): 5933–5947.

Ojugo, A. A., and Yoro, R. E. (2020b). Predicting Futures Price And Contract Portfolios Using The ARIMA Model: A Case of Nigeria's Bonny Light and Forcados. *Quantitative Economics and Management Studies*, *1*(4): 237–248. https://doi.org/10.35877/454ri.qems139

Ojugo, A. A., and Yoro, R. E. (2021a). Forging a deep learning neural network intrusion detection framework to curb the distributed denial of service attack. *International Journal of Electrical and Computer Engineering*, *11*(2): 1498–1509. https://doi.org/10.11591/ijece.v11i2.pp1498-1509

Ojugo, A. A., and Yoro, R. E. (2021b). Extending the three-tier constructivist learning model for alternative delivery: ahead the COVID-19 pandemic in Nigeria. *Indonesian Journal of Electrical Engineering and Computer Science*, *21*(3): 1673. https://doi.org/10.11591/ijeecs.v21.i3.pp1673-1682

Ojugo, A. A., Yoro, R. E., Oyemade, D. A., Eboka, A. O., Ugboh, E., and Aghware, F. O. (2013). Robust Cellular Network for Rural Telephony in Southern Nigeria. *American Journal of Networks and Communications*, *2*(5), 125. https://doi.org/10.11648/j.ajnc.20130205.12

Ojugo, A. A., Yoro, R. E., Yerokun, M. O., & Iyawa, I. J. (2013). Implementation Issues of VoIP to Enhance Rural Telephony in Nigeria. *Journal of Emerging Trends in Computing and Information Sciences ©2009-2013*, *4*(2): 172–179. http://www.cisjournal.org

Okobah, I. P., and Ojugo, A. A. (2018). Evolutionary Memetic Models for Malware Intrusion Detection: A Comparative Quest for Computational Solution and Convergence. *International Journal of Computer Applications*, *179*(39), 34–43. https://doi.org/10.5120/ijca2018916586

Okonta, E. O., Ojugo, A. A., Wemembu, U. R., and Ajani, D. (2013). Embedding Quality Function Deployment In Software Development: A Novel Approach. *West African Journal of Industrial & Academic Research*, *6*(1), 50–64.

Oyemade, D. A., and Ojugo, A. A. (2021). An Optimized Input Genetic Algorithm Model for the Financial Market. *International Journal of Innovative Science, Engineering and Technology, 8*(2): 408–419. https://ijiset.com/vol8/v8s2/IJISET_V8_I02_41.pdf

Paliwal, S., Mishra, A. K., Mishra, R. K., Nawaz, N., and Senthilkumar, M. (2022). XGBRS Framework Integrated with Word2Vec Sentiment Analysis for Augmented Drug Recommendation. *Computers, Materials and Continua*, *72*(3): 5345–5362. https://doi.org/10.32604/cmc.2022.025858

Parsons, K., McCormac, A., Pattinson, M., Butavicius, M., and Jerram, C. (2015). The design of phishing studies: Challenges for researchers. *Computers & Security*, *52*: 194–206. https://doi.org/10.1016/j.cose.2015.02.008

Rathi, M., and Pareek, V. (2013). Spam Mail Detection through Data Mining – A Comparative Performance Analysis. *International Journal of Modern Education and Computer Science*, *5*(12): 31–39. https://doi.org/10.5815/ijmecs.2013.12.05

Redondo-Gutierrez, L. Á., Jáñez-Martino, F., Fidalgo, E., Alegre, E., González-Castro, V., and Alaiz-Rodríguez, R. (2022). Detecting malware using text documents extracted from spam email through machine learning. *Proceedings of the 22nd ACM Symposium on*

Document Engineering, 1–4. https://doi.org/10.1145/3558100.3563854

Sahmoud, T., and Mikki, D. M. (2022). *Spam Detection Using BERT*. https://doi.org/10.48550/arXiv.2206.02443

Sasikala, G., Laavanya, M., Sathyasri, B., Supraja, C., Mahalakshmi, V., Mole, S. S. S., Mulerikkal, J., Chidambaranathan, S., Arvind, C., Srihari, K., and Dejene, M. (2022). An Innovative Sensing Machine Learning Technique to Detect Credit Card Frauds in Wireless Communications. *Wireless Communications and Mobile Computing*, *2022*, 1–12. https://doi.org/10.1155/2022/2439205

Sohony, I., Pratap, R., and Nambiar, U. (2018). Ensemble learning for credit card fraud detection. *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, 289–294. https://doi.org/10.1145/3152494.3156815

Tian, F. (2016). An agri-food supply chain traceability system for China based on RFID & blockchain technology. *2016 13th International Conference on Service Systems and Service Management (ICSSSM)*, 1–6. https://doi.org/10.1109/ICSSSM.2016.7538424

Tingfei, H., Guangquan, C., and Kuihua, H. (2020). Using Variational Auto Encoding in Credit Card Fraud Detection. *IEEE Access*, *8*, 149841–149853. https://doi.org/10.1109/ACCESS.2020.3015600

Verma, S., Bhatia, A., Chug, A., and Singh, A. P. (2020). *Recent Advancements in Multimedia Big Data Computing for IoT Applications in Precision Agriculture: Opportunities, Issues, and Challenges* (pp. 391–416). https://doi.org/10.1007/978-981-13-8759-3_15

Wang, D., Chen, B., and Chen, J. (2019). Credit card fraud detection strategies with consumer incentives. *Omega*, *88*: 179–195. https://doi.org/10.1016/j.omega.2018.07.001

Xuan, S., Liu, G., Li, Z., Zheng, L., Wang, S., and Jiang, C. (2018). Random forest for credit card fraud detection. *2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC)*, 1–6. https://doi.org/10.1109/ICNSC.2018.8361343

Yan, C., Huanhuan, F., Ablikim, B., Zheng, G., Xiaoshuan, Z., and Jun, L. (2018). Traceability information modeling and system implementation in Chinese domestic sheep meat supply chains. *Journal of Food Process Engineering*, *41*(7): e12864. https://doi.org/10.1111/jfpe.12864

Yeboah-Boateng, E. O., and Amanor, P. M. (2014). Phishing , SMiShing & Vishing : An Assessment of Threats against Mobile Devices. *Journal of Emerging Trends in Computing and Information Sciences*, *5*(4): 297–307.

Yildiz Durak, H. (2019). Human Factors and Cybersecurity in Online Game Addiction: An Analysis of the Relationship Between High School Students' Online Game Addiction and the State of Providing Personal Cybersecurity and Representing Cyber Human Values in Online Games. *Social Science Quarterly*, *100*(6):1984–1998. https://doi.org/10.1111/ssqu.12693

Yoro, R. E., Aghware, F. O., Akazue, M. I., Ibor, A. E., and Ojugo, A. A. (2023). Evidence of personality traits on phishing attack menace among selected university undergraduates in Nigerian. *International Journal of Electrical and Computer Engineering (IJECE)*, *13*(2): 1943–1953. https://doi.org/10.11591/ijece.v13i2.pp1943-1953

Yoro, R. E., Aghware, F. O., Malasowe, B.

O., Nwankwo, O., and Ojugo, A. A. (2023). Assessing contributor features to phishing susceptibility amongst students of petroleum resources varsity in Nigeria. *International Journal of Electrical and Computer Engineering*, *13*(2): 1922–1931. https://doi.org/10.11591/ijece.v13i2.pp1922-1931

Yoro, R. E., and Ojugo, A. A. (2019a). An Intelligent Model Using Relationship in Weather Conditions to Predict Livestock-Fish Farming Yield and Production in Nigeria. *American Journal of Modeling and Optimization*, *7*(2): 35–41. https://doi.org/10.12691/ajmo-7-2-1

Yoro, R. E., and Ojugo, A. A. (2019b). Quest for Prevalence Rate of Hepatitis-B Virus Infection in the Nigeria: Comparative Study of Supervised Versus Unsupervised Models. *American Journal of Modeling and Optimization*, *7*(2): 42–48. https://doi.org/10.12691/ajmo-7-2-2

Zanin, M., Romance, M., Moral, S., and Criado, R. (2018). Credit Card Fraud Detection through Parenclitic Network Analysis. *Complexity*, *2018*: 1–9. https://doi.org/10.1155/2018/5764370

Zareapoor, M., and Shamsolmoali, P. (2015). Application of Credit Card Fraud Detection: Based on Bagging Ensemble Classifier. *Procedia Computer Science*, *48*: 679–685. https://doi.org/10.1016/j.procs.2015.04.201

Zawislak, P. A., Reichert, F. M., Barbieux, D., Avila, A. M. S., and Pufal, N. (2022). The dynamic chain of innovation: bounded capabilities and complementarity in agribusiness. *Journal of Agribusiness in Developing and Emerging Economies*. https://doi.org/10.1108/JADEE-04-2021-0096

Zhang, D., Bhandari, B., and Black, D. (2020). Credit Card Fraud Detection Using Weighted Support Vector Machine. *Applied Mathematics*, *11*(12): 1275–1291. https://doi.org/10.4236/am.2020.1112087

Zhang, Y., Egelman, S., Cranor, L. F., and Hong, J. (2007). Phinding Phish: Evaluating Anti-Phishing Tools. *In Proceedings of the Network & Distributed System Security Symposium (NDSS 2007)*, *March*, 1–16.