ijesir NOVUS Publishers

e-ISSN (2785-5740)
https://www.ejatcs.novuspublishers.org/

2022
Vol.1 Issue.1

Research article

DNS Cache Poisoning / Honeypot Analysis Based on Data Exfiltration Using Stochastic Petri Nets Method to Enhance Cyber Security Hygiene

Akhigbe-mudu Thursday Ehis

African Institute of Science Administration and Commercial Studies Lome- Togo

Email: akhigbe-mudut@iaec-university.tg

Abstract

Experts suspect that companies spend millions of dollars on firewalls, encryption and secure access devices against threats, and all in vain as none of these solutions address the weakest link in the security chain. People who work on secure networks with authorization are suspected of having a weak link in the security chain. Threat detection methods created by researchers can detect many false alarms during detection processes, and these false alarms are responsible for shutting down the system. The nature of traffic between communication systems is unpredictable; therefore, it is common to develop a stochastic model to represent such a system. This study used SPN to create models; statistical models have been conventionally used to analyze networks with security chains. The new stochastic Petri net formalism offers the enhancement of model fidelity by allowing a combination of real-time and continuous events, as well as non-Markovian behavior to be formalized. This allowed us to see special structures within the stochastic process produced by SPN models. We have applied this principle by proposing an effective simulation method that supports deadlock detection andeasy-to-compute point estimates and confidence intervals. The method is novel because it can automatically detect hidden regenerative structures that do not conform to different simple conditions, and can be easily determined by analytical methods.

Keywords: Domain Name System, Deadlock, Exfiltration, Security Chain, Stochastic Petri nets.

Introduction

Before we get into Domain Name System (DNS) poisoning, let's take a look at what DNS is. DNS is the internet version of the Yellow pages. Back in the days of old, when you needed to find a business address, you look it up in the yellow pages. DNS is just like that; except you don't actually have to look anything up, your internet connected to your computer does that for you. For two computers to communicate on an Internet Protocol (IP) network, Protocol dictates that they need an IP address [**16**]. Think of an IP address like a street address- for one computer to locate another, they need to know the other computer's number. Since most humans are better at remembering names- (www.various.com), than numbers- (104.196.44.111), they needed a program for computers to translate names into IP addresses. The program to translate names into numbers and vice versa is called "DNS" and computers that runs DNS are called "DNS" servers.Without the DNS, we would have to remember the IP address of any server we wanted to connect to- no fun.

A DNS converts a human-readable name (for example, www.geeksforgeeks.org) to a numeric IP address. The DNS system responses to one or more IP-addresses by which your computer connects to a website (such as geeksforgeeks.org). The

domain name is resolved by a series of DNS servers. [5:19] Claimed that DNS uses cache to work efficiently so that it can quickly refer to DNS lookups already performed rather than performing a DNS lookup over and over again. DNS cache increases the speed of the domain name resolution processes. Attackers use DNS weaknesses to gain control of the system and redirect users to a malicious website. A recursive server's principal task is to create and maintain a large cache of DNS responses. Cache poisoning aims to corrupt the answers stored in the cache, resulting in the corrupted answer being returned to any subsequent lookups from other clients. In Figure 1 Cache Poisoning is discussed.

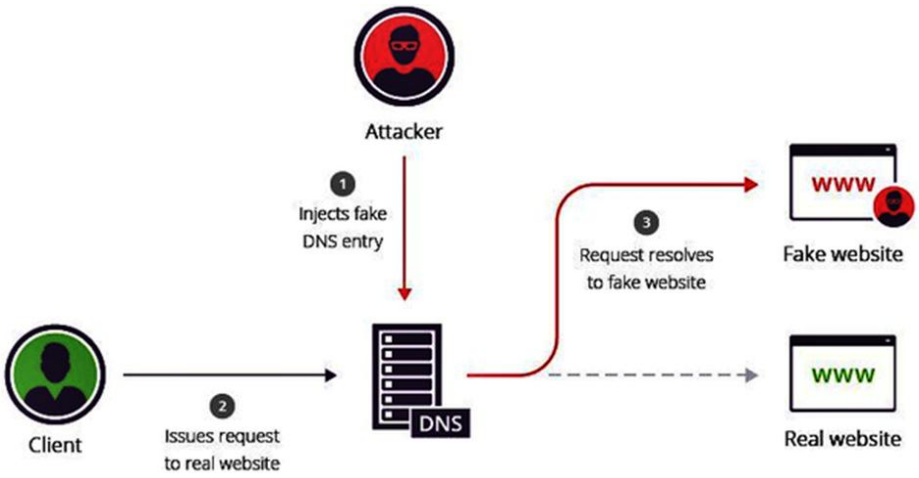


Figure 1: DNS Cache Poisoning

What Is DNS Tunneling?

It's been dubbed the most dangerous DNS hack by experts. DNS requests are sent over UDP (User Datagram Protocol) on Port 53, which is almost always open on systems, firewalls, and clients. There is no security built into this protocol. All cybercriminals are aware that DNS is widely used and trusted. Because DNS isn't designed for data transfer, many firms don't check their DNS traffic for malicious behavior, as an Information Technology (IT) specialist would know. As a result, a variety of DNS-based attacks can be successful when launched against corporate networks. One such assault is DNS tunneling. DNS tunneling attacks take advantage of this protocol to get malicious traffic past your firewall. An attacker can utilize DNS to avoid your network defenses and accomplish data exfiltration by leveraging malicious domains and DNS servers. Attackers can use DNS to set up covert channels to hide communication or get around policies set up by your network administrators [11:18]. Attackers take advantage of this fact by exploiting DNS requests to set up a command and control (C&C) channel for malware in DNS tunneling [14]. Outbound DNS traffic can exfiltrate your sensitive data or offer responses to the malware operator's queries, while inbound DNS traffic can deliver commands to the malware. Assume the DNS server is under hacker control, then they can take data from your database, such as your clients' social security numbers, name, address, phone, mobile, and email (see figure 2).

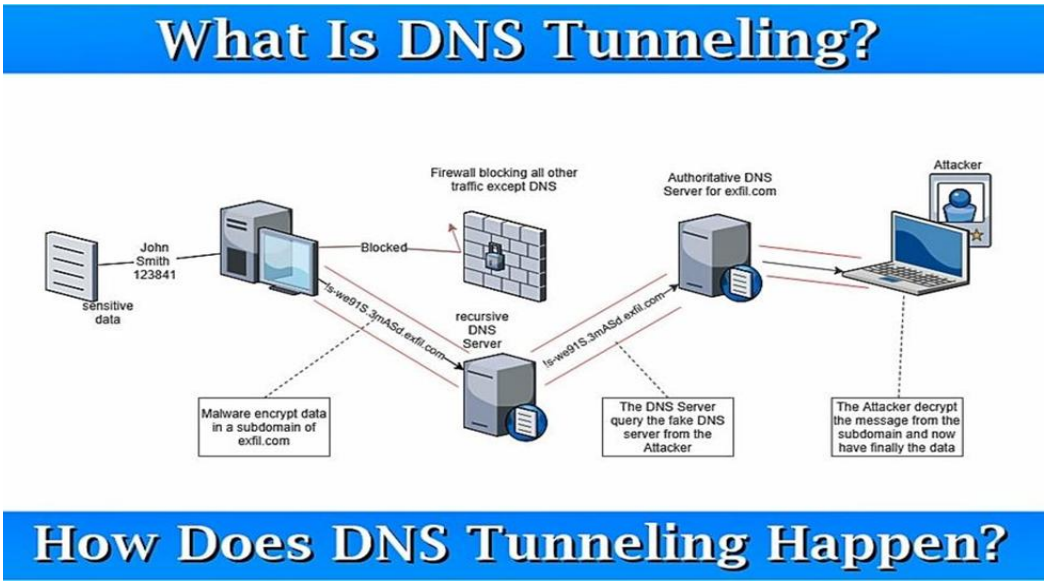


Figure 2: Showing DNS Tunneling and The Probability of it happening

How Does DNS Tunneling Happen?

A DNS Tunnel attack's general strategy would be as follows: The attackers buy a domain name like firstlook.com. The name server for the domain points to the attackers' server, which contains tunneling malware software. The attackers would next use malware to infect machines. These PCs would most likely be protected by the company's firewall. Infected machines are able to send a query to the DNS resolver since DNS requests are always allowed to pass through the firewall. The DNS resolver is a server that forwards IP address requests to the root server and top-level domain servers. The DNS resolver sends the request to the attackers' command-and-control (C&C) server, which contains the tunneling application. Through the DNS resolver, a connection is created between the victim machine and the attacker [**28**]. This tunnel can be used to exfiltrate data or for other nefarious purposes with ease. It is more difficult to trace the attacker's computer because there is no direct connection between the attacker and the victim PC.

What is a Honeypot?

Honeypot is a type of activity that allows you to collect all relevant information about an attacker's activities [**2**]. A honeypot is a one-of-a-kind security resource that is part of your organization's overall security strategy. The objective is that all attackers should only engage with your honeypots and not with real-world systems. Honeypots are useless if the attacker does not interact with them. Honeypot traps entice bad guys to attack these fictitious networks, servers, or other devices by containing applications and data that are identical to those found on real targets. When an attacker falls into this trap, the honeypot provides administrators with important information about the sort of attacker, the activity he was attempting, and, in some situations, even the attacker's identity. They can be employed as early warning systems, slowing down automated attacks and catching new exploits in order to gather intelligence on emerging threats before they harm your networks or vital assets. Honeypots have the advantage of being able to function as a real computer, a virtual machine, a complete (dummy) network, or even an application. Honeypots don't even have to be computer-based. Credit card numbers, Excel spreadsheets, and login and passwords are just a few examples (known as honey tokens). Honeypots come in a variety

of shapes and sizes (See figure 3). The honey pot is made up of a real-time window or linux operating system that may be installed on a physical or virtual machine [15]. The data stream analyzer is configured to receive logs from the honeypot. In the Test Bed, a honey pot sensor is put in the network's active directory, where all systems connected to the network use active directory for system communication across the network, allowing for easy logging of all system activity. The malicious system calls are routed to a read data stream analyzer, which runs on a different machine and visually represents data streams from various sensors based on danger level. The data streams contain a system log which helps to clearly identify which malicious activity is going on the system.

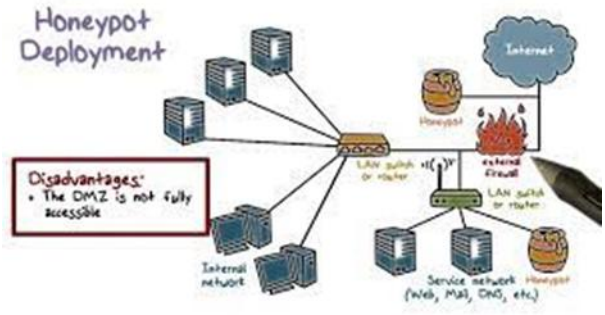


Figure 3: Honeypot deployment in a System

How do attackers poison DNS caches?

Attackers can poison DNS caches by impersonating DNS name servers, making a request to a DNS resolver, and then forging the reply when the DNS resolver queries a name server. The diagram in figure 4 depicts the process in detail. This is possible because DNS servers use User Datagram Protocol (UDP) instead of TCP, and because currently there is no verification for DNS information [17]. The graphic here in figure 4, clearly shows how it happens.

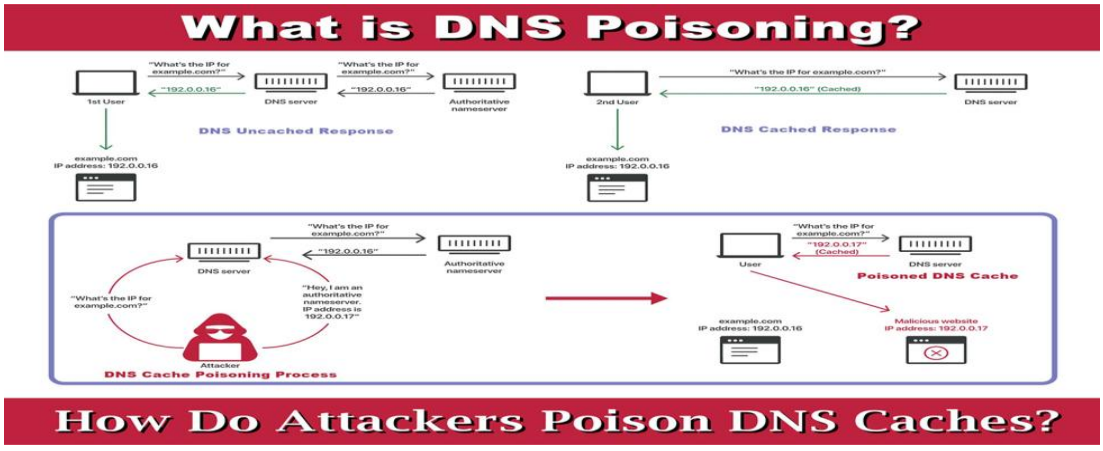


Figure 4: Shows the How Attackers Poison DNS Caches.

UDP is the main offender in this case.DNS requests and responses employ UDP instead of TCP, which requires both communicating parties to execute a 'handshake' to commence communication and verify the identification of the devices. There's no way to know if a connection is open, if the recipient is ready to receive, or if the sender is who they claim they are with UDP. Because UDP is prone to forging, an attacker can send a message over UDP and forge the header data to

make it appear to be a response from a valid server. Because there is no way to check if the information is true and comes from a legitimate source, a DNS resolver accepts and caches fraudulent responses without question.

The Great Firewall of China Spreads to the US IN 2010

This isn't just a theoretical issue; it occurred on a massive scale in the real world. Blocking at the DNS level is one of the ways China's Great Firewall operates [29]. In 2010, an Internet service provider based outside of China made the error of configuring its DNS servers to retrieve data from Chinese DNS servers. The erroneous DNS records were obtained from China and cached on its own DNS servers. Other Internet service providers used that Internet service provider's DNS information on their DNS servers. The poisoned DNS records propagated until certain consumers in the United States were unable to access Twitter, Facebook, or YouTube through their Internet service providers in the United States. The Great Firewall of China had “leaked” outside of its national borders, preventing people from elsewhere in the world from accessing these websites. This essentially functioned as a large-scale DNS poisoning attack [10].

Literature Review

[30] Developed a reliable encryption scheme based on fractional-order chaotic systems. The substitution boxes and other blocks required for generating the encryption were provided with the speech encryption system. Several security studies were conducted, taking into account sensitivity, statistical analysis, and a variety of other factors. According to [31], audio file steganography can be protected using Advanced Encryption Standard (AES) techniques and hashing functions such as the Message Digest Algorithm (MD5), and the AES algorithm was used to encrypt the data and MD5 was used to scramble passwords. The MP3 file was also encoded. Decoding is done on the other end by extracting the sensitive private message and decrypting it to restore the original data. The limitation of this technique is that it can only be used on mp3 files with a homogeneous frame. While [34] used a Cuckoo search followed by other optimization techniques to encrypt mp3 data within an image, the steganographic method used a Cuckoo search followed by other optimization techniques. Their system's result was quite impressive, as the system as a whole achieved good accuracy. Despite the usage of many graphics, the steganographic technique was limited to solely audio files. There have been at least two complementary techniques to evaluating honeypot-captured cyber-attack data. Visualizing cyber-attack data, such as utilizing neural projection techniques to show the ports identified in honeypot data [37] is one option. The limitation of this technique is that it can only be used on mp3 files with a homogeneous frame.

While [32] used a Cuckoo search followed by other optimization techniques to encrypt mp3 data within an image, the steganographic method used a Cuckoo search followed by other optimization techniques. Their system's result was quite impressive, as the system as a whole achieved good accuracy. Despite the usage of many graphics, the steganographic technique was limited to solely audio files. There have been at least two complementary techniques to evaluating honeypot-captured cyber-attack data. Visualizing cyber-attack data, such as utilizing neural projection techniques to show the ports identified in honeypot data [12] is one option. Statistical analysis, on the other hand, is a commonly utilized method. Statistical analysis, on the other hand, is a commonly utilized method. In particular, our study of predicting cyberbullying (in terms of attack rate) should be an important step towards the ultimate goal of understanding / predicting cyber bullying. With regard to the use of honey jars for self-defense, we note that honeypots have been used to help detect various attacks

including DoS (denial-service) [26], worms [13:10], botnets [39: 40: 41], Internet messaging threats [42], production of attack signatures [36: 48], and targeted targeting [8]. These lessons are important, but they are orthogonal to the focus of the current paper.

[22] Made a proposal for a security framework called Japonica, which has a honeypot as a viable business to detect and counteract unknown aggressive attacks. [21] Describe the model framework proposed by the Colored Petri Nets (CPN) to determine the format of exchange messages with similar messages between organizations. The results showed that the framework could work effectively. Apart from that, there are other activities related to Petrol nets. By combining standard Petri nets with ambiguous rules, a software system model called Intelligent Petri Net (PN) was proposed [23], in which the operating time area and system behavior could be modeled. I-PN incorporates the ability to adapt. In [4], Petri nets were used for a physiochemical model, which was associated with Siphon and proteins involved in the targeted detection of therapeutic drugs that have many components in showing pathways. [6] Petri nets have been translated to describe the corresponding control system. The prototyping method of the system was proposed, ruling during the polynomial period. However, the functions mentioned above are closely related to login access, web service firewall, Spoofing Address resolution Protocol (ARP) and program structure in network security. In this paper, we focus on the combination of Petri nets [27] and honey pot. Only in [15], a honeypot was used as part of the proposed framework, and CPN was involved, different from the Stochastic Petri Nets (SPN). Therefore, to our knowledge, there is no work related to both honeypot and SPN. Such a plan is proposed in this paper to analyze the performance of the honeypot.

[40] Also deal with data theft on wireless networks; although their focus is on mobile ad networks deployed for military operations. They use the integration method to find the confusing, using IP and information of the transport topic as features. These methods focus on monitoring data that crosses the organization's network boundary and preventing it from attack like SQL injection. Their initial validation of this method in simulation works, but the flexibility requirements of the wiring network do not appear to be fully represented in the test environment. Identifying strange traffic on a rapidly changing ad-hoc network would seem to be a difficult problem for any partitioning system [33]. [41] Introduced a method for detecting password guessing attempts and DoS attacks by analyzing network log files using data mining techniques. The log file is processed and the data is recorded and stored in a file [43]. The clustering algorithm is designed and used to detect that certain connections occur multiple times. The connection that appears most often indicates suspicious activity. The main limitation of the proposed method is the lack of ability to detect attacks during operation, making the system less efficient. The authors do not provide job evaluation tests because a few associated factors such as acquisition rate and performance impacts remain unclear.

Research Methodology

Sensor Data Parser

This module processes incoming data streams from the honeypot sensors in readable format. Data generated from the honeypot sensor is in the form of strings The module divides the strings into form of source IP, area IP, event name, username, Date, Time and many other important features are represented in readable form. After collecting all relevant

event data, the data is transferred to the threat analyst to identify potential Insider Threat. The threat analyzer is a separate module and is installed on the same server.

Threat Analysis

Threat Analyst analyzes data at three different levels. Each level is designed to filter out any false alarm opportunity. It only sends a notification when it can detect event activity in any of the level details specified in the algorithm below:

Algorithm 1: False Positive Insider Threat Detection

Step 1: If the user event function is normal there is no alarm

Step 2: Alternatively check out a program that uses software and events,

Step 3: When user activity is associated with software and app events there is no alarm

Step 4: Further check system hardware changing events

Step 5: if the hardware changes are enabled on the user then No alarm

Step 6: Further check user rights

Step 7: if the user edits the files according to each user's rights then No alarm

Step 8: Alternatively raise the alarm.

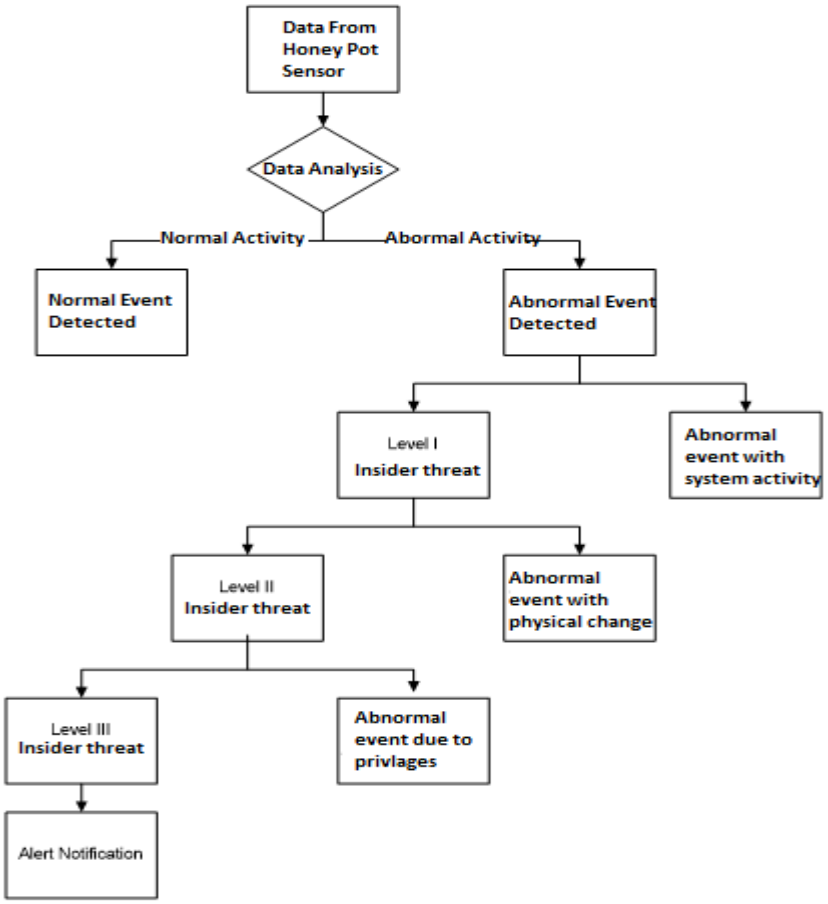


Fig. 6 -Three level classification of threat analyzer algorithm

The algorithm shown in Figure 6 is simple enough to use and works well enough to detect any false alarm generated by a user's system function. The division of the three levels of user system function first determines whether the event is software or a system-produced system previously identified in the profile training section and ignores the event. Next if any hardware changes occur in the system that generated the suspicious events should be analyzed and if the computer replacement events fall into the normal events generated by the user's behavior and become events. Finally evaluates changes to user-generated data. If user-adjusted data complies with given user rights then ignores those events. Any suspicious events generated by random sequences, read, write, network penetration, data transfer, computer hardware changes, and right changes are detected in real time to identify Insider Threat. The three-stage division of the sequence of the proposed threat analysis algorithm is shown in Figure 7. The detected Insider Threat is displayed on a web-based easy-to-use GUI accessible only to system administrators (shown in Figure 7). According to the diagram, SOAP specifications are made up of three parts of the concept: protocol concepts, encapsulation concepts, and network concepts. As a result, the data transmission protocol is structured in the context of web service usage. TCP Dump is an application for analyzing a network data packet with a visible command line. Allows the user to see TCP / IP and other export packets. In this example, the optimizer components are used to analyze the query within the website. Because every question has to be considered at least once [], it is used in this regard for decision making.

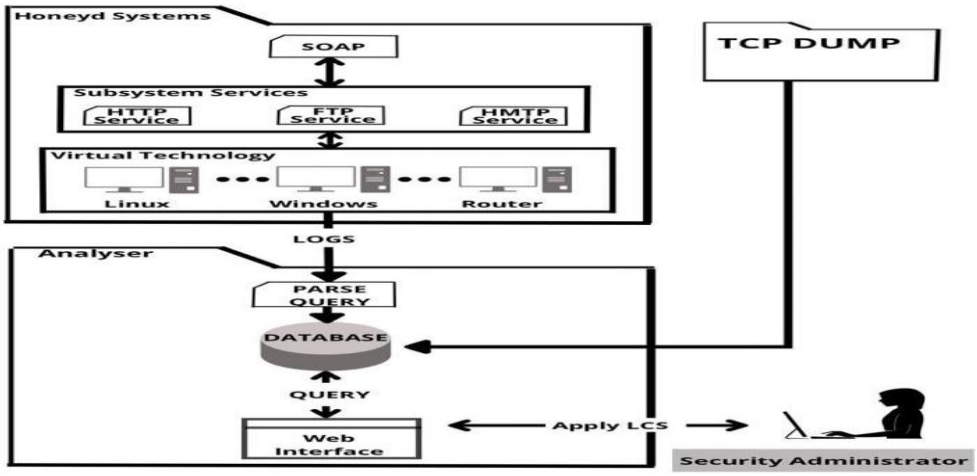


Figure 7: User Friendly Web Based GUI threat analyzer algorithm

Results and Discussion

The honeypot contains a window operating system that runs on a virtual machine. The honeypot is designed to transfer logs to the data distribution analysis. In Test Bed, the honeypot sensor is installed in the active network directory where all systems are connected to the network and use the active communication interface of their system. With network, all activity in the system can be easily accessed. Sensor Data Parser is a module for analyzing incoming data transmission from honeypot sensors in readable format. The honeypot generated data is divided by the module into IP source threads, local IP,

event name, username, date, time and many other important features represented in a readable manner. After collecting all relevant event data, the data is transferred to the threat analyst to identify potential Insider Threat. The threat analyzer is a separate module and is installed on the same server.

At the beginning of the trial, participants were directed to create a normal behavioral profile. Prior to the creation of standard user profiles, the researcher created 40 MB random references within user programs, and created 32 references in the system. After creating a standard profile, each participant created a different version of the modified, deleted, and modified files on drivers that are often inaccessible. Unusual process work can be a prelude to the fact that malicious application, malicious code, or any other malicious program is present in the program. As mentioned earlier, an internal attack on an organization can be extremely dangerous and dangerous if left unchecked. The researcher has developed a separate experiment that can be really helpful in finding unusual process functions such as deadlock. The following tests are performed after the profile section.

Deadlock

Deadlock is a situation that involves the interaction of more than one resource and processes [46]. We can visualize the occurrence of a deadlock as a situation where there are two people on the stairs (figure 8). One goes up the stairs while the other goes down. Stairs are narrow enough to accommodate one person at a time. As a result, one has to step back while the others move on and use the stairs. Once that person has finished, someone else can use those stairs. But here, none of the people are willing to go back and wait for someone else to step back. None of them can use the stairs. The people here are the process and the stairs are the source. If the process asks for an app hosted by another process that requires another application to continue, but is held by the first process, then it is called deadlock.

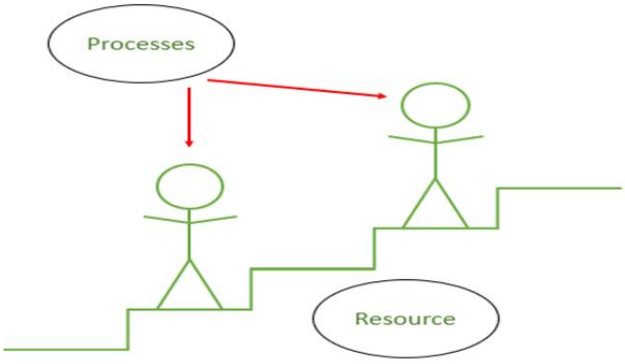


Figure 8: Deadlock Condition

Deadlock can occur if the following four instances hold simultaneously (Required Conditions): 1. Shared Release: Two or more resources cannot be shared (Only one process can be used at a time)

2. Hold and Wait: Process holds at least one app and waits for resources.

3. No Release: The app cannot be downloaded from the system unless the system uninstalls the app.

4. Round Waiting: A set of processes is waiting for one round.

Deadlock discovery:

In this case of Deadlock discovery, we can use an algorithm to test the cycle in the App Distribution Graph. The presence of a cycle on the graph is a sufficient condition for the deadlock.

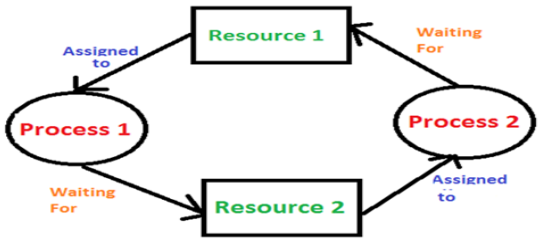


Figure 9: Deadlock discovery

2. In the diagram above, resource 1 and resource 2 have the same shape. There is a cycle

$R_1 \rightarrow P_1 \rightarrow R_2 \rightarrow P_2$ Therefore, Deadlock is verified.

3. If there are multiple service conditions -

Cycle detection is required but there is not enough condition for deadlock detection, in this case, the system may be deadlock or may not depending on different circumstances [42].

Safety Algorithm / Deadlock Detection Algorithm

The algorithm for detecting safe state condition in a system can be defined as follows:

Algorithm steps:

Assume Task and Finish to be vectors of length 'm' and 'n' respectively.

Get Started Task = Available for $i = 0,1,..., n-1$

if Request $i = 0$, then Finish [i] = true; otherwise, Finish [i] = false.

Get an index i such that both:

a) $Finish[i] == false$

b) $\text{Re}\, quest.i <= Task$

If not then go to step 4.

Task = Task + Allocation

Finish [i] = true

Go to step 2.

If Finish [i] == is a false for some $i, 0 < i < n$,it means the system is in a deadlock state.

In addition, if Finish [i] == false, the process p_i is deadlocked.

For example

	Allocation			Request			Available		
	A	B	C	A	B	C	A	B	C
P0	0	1	0	0	0	0	0	0	0
P1	2	0	0	2	0	2			
P2	3	0	3	0	0	0			
P3	2	1	1	1	0	0			
P4	0	0	2	0	0	2			

1. In this case, Function = [0, 0, 0] & Finish = [false, false, false, false, false]
2. $i = 0$ is selected as both Finish [0] = false and [0, 0, 0] <= [0, 0, 0].
3. Function = [0, 0, 0] + [0, 1, 0] => [0, 1, 0] & Finish = [true, false, false, false, false].
4. $i = 2$ is selected as both Finish [2] = false and [0, 0, 0] <= [0, 1, 0].
5. Function = [0, 1, 0] + [3, 0, 3] => [3, 1, 3] & Finish = [true, false, true, false, false].
6. $i = 1$ is selected as both Finish [1] = false and [2, 0, 2] <= [3, 1, 3].
7. Function = [3, 1, 3] + [2, 0, 0] => [5, 1, 3] & Finish = [true, true, true, false, false].
8. $i = 3$ is selected as both Finish [3] = false and [1, 0, 0] <= [5, 1, 3].
9. Function = [5, 1, 3] + [2, 1, 1] => [7, 2, 4] & Finish = [true, true, true, true, false].
10. $i = 4$ is selected as both Finish [4] = false and [0, 0, 2] <= [7, 2, 4].
11. Function = [7, 2, 4] + [0, 0, 2] => [7, 2, 6] & Finish = [true, true, true, true, true]
12. Since Finish is the vector of all that is true it means that there is no deadlock in this example.

Deadlock Solution

Over the decades, Petri nets have become one of the most well-known and fully-fledged mathematical tools to deal with deadlock problems due to their natural properties. According to [47] Petri net process, liveness is an important aspect of system safeness, which implies the absence of global and local deadlock conditions in the Default System (DS). Livenesstesting can be done by verifying the satisfaction of certain pronouns in the siphons, a well-known structural element in the Petri net networks. Therefore, siphons have received a lot of attention in order to analyze and control systems measured by Petri nets. In particular, the theory of early siphon plays an important role in the development of Petri executives who force life easily, leading to a variety of deadlock control methods [44]. There are three key factors in assessing the performance of a life-force manager to control the system: compliance behavior, structural complexity, and computer complexity. Deciding how to design a highly qualified Petri manager is always a big challenge. Because the access graph can fully reflect the behavior of the Petri net system, accessibility graph-based policies can always find a life-force guide with greater or greater permissions.

Petri Net Definition

Petri net is a type of two-dimensional graphs consisting of three types of objects. These are the places, transitions, and directed arcs [45]. Directed arcs connect places to transitions or transitions to places. In its simplest form, the Petri net can be represented by a transition as well as an input and output place. This basic net can be used to represent various aspects of modeling systems. Tokens are an old concept of Petri nets over places and transitions. The presence or absence of a token in a location (place) may indicate whether the status associated with this location is true or false,

For example

A Petri net is formally defined as a 5-tuple N = (P, T, I, O, M_0), where

(1) $P = \{p_1, p_2, ..., p_n\}$ is a finite set of places;

(2) $T = \{t_1, t_2, ..., t_n\}$ is a finite set of transitions, P ∪ T ≠ ∅, and P ∩ T = ∅;

(3) I: P × T → N is an input function that defines directed arcs from places to transitions, where N is a set of nonnegative integers;

(4) O: T × P → N is an output function that defines directed arcs from transitions to places; and

(5) M_0 : P → N is the initial marking.

Tagging on the Petriet net is the assignment of tokens in the Petri net area. Tokens are located in the places of Petri's net [7]. The number and status of tokens may change during the creation of the Petri net. Tokens are used to describe the processing of Petri net. The Petri net graph is the structure of the Petri net as a directed multigraph for bipartite. Consistent with the definition of Petri nets, the Petri net graph has two types of nodes. The circle represents the location (Place); bar or box represents transition. Targeted (directed) arcs (arrows) link places and transitions, with other arcs directed from places to transitions and other arcs directed from transition to locations. An arc directed from a location p_i to the transition t_i defines p_i the input location of t_i identified as $I(t_i, p_j) = 1$. The arc directed from the transition t_i to the place p_j defines p_j the output place of t_i, which is defined as $O(t_i, p_j) = 1$. If $I(t_i, p_j) = k$ or $O(t_i, p_j) = k$ then there are k-directed (parallel) arcs that link the location pj to the transition t_i (or link the transition t_i to the location p_j). represented by a single pointed arc labeled by its quantity, or weight k. A circle contains a dot representing a place containing a token [35].

Example 1: A simple Petri net.

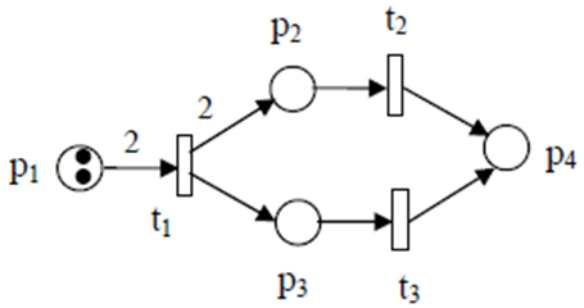


Figure 10: A simple Petri net.

Figure 10 shows a simple Petri net. In this Petri net, we have

$P = \{p_1, p_2 p_3, p_4\};$

$T = \{t_1, t_2, t_3\};$

$I(t_1, p_1) = 2, I(t_1, p_i) = 0, \ for, i = 2,3,4;$

$I(t_2, p_2) = 1,..I(t_2, p_i) = 0..for..i = 1,3,4;$

$I(t_3, p_3) = 1,..I(t_3, p_i) = 0...for...i = 1,2,4;$

$O(t_1, p_2) = 2,..O(t_1, p_3) = 1,...O(t_1, p_i) = 0...for...i = 1,4;$

$O(t_2, p_4) = 1,...O(t_2, p_i) = 0...for...i = 1,2,3;$

$O(t_3, p_4) = 1,...O(t_3, p_i) = 0...for..i = 1,2,3;$

$M_o = (2,0,0,0)^T$

Standard Petri Net. "A Standard Petri Net [37] is defined by 5 − tuple (P, T, F, W, M_o) where;

• P is a finite set of places $(P_1, P_2, ..., P_n)$

T is a finite set of transitions ($t_1, t_2, ..., t_m$)

• $F \subseteq (PxT) \cup (TxP)...is.a.set.of.arcs$

• Wis a weight function of arcs

• M_o: is the initial marking $P \to (0,1,2,...)$ *where* $P \cap T = \Phi...and...p \cup T \neq \Phi$

Stochastic Petri Net. "A Stochastic Petri Net (SPN) [39] is defined by 6 − tuple (P,T,F,W,M_o,Ω) where (P,T,F,W,M_o) are similar as defined in the definition of standard petri net and Ω represents the functions $\Omega: T \to R$ which assigns rate to the transition $t \in T$ according to the negative exponential distribution function". The emergence of Stochastic PetriNet as defined by Continuous Time Markov Chain (CTMC) and a state of a CTMC representsa single Petri Net tag. In other words, CTMC represents the Petri Net accessibility graph [38].The example in Figure 10,shows the definition of Stochastic Petri Net. There are a few behaviors of Petri Nets [37, 38, 40] and some of them are described below:

• **Reachability**: This property is used to study dynamic properties of the system. A marking Mk is reachable from an initial marking M0 if there exists a firing sequence from M0 to Mk.

• **Liveness**: A live Petri Net is a deadlock free Petri Net and from any marking, there exists a firing sequence which contain all transitions.

• **Reversibility**: This property ensures that there will always be a way back to the initial marking M_o from all reachable markings commencing from M_o.

(Properties Boundedness, and Safeness).

A marked PN (Ψ, M_o) is said to be (marking) k-bounded iff each of its places is k-bounded. A 1-bounded net is called safe. A marked PN (Ψ, M_o) is bounded if there exists $(K \in N)$ such that (Ψ, M_o) is k-bounded. A net Ψ is structurally bounded iff M_o the marked PN (Ψ, M_o) are k-bounded for some k $\in$ N.

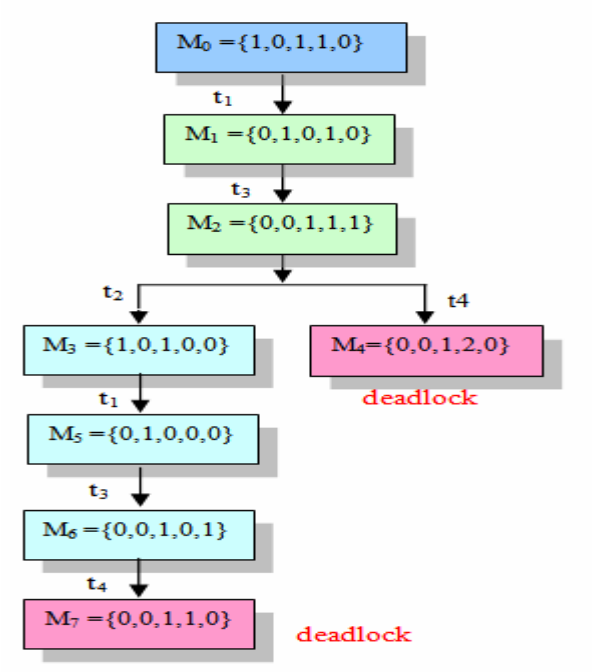
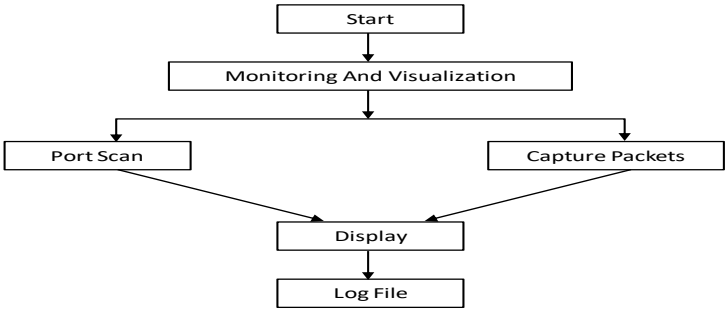


Figure 11: The Reachability tree of Figure 10

Performance Evaluation

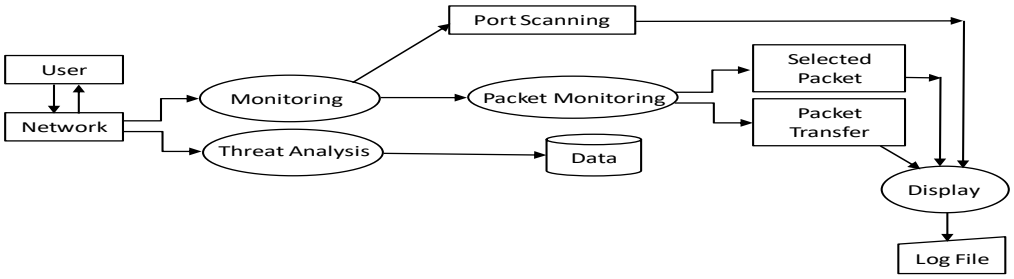
The theoretical aspect of Petri nets allows for accurate modeling and behavioral analysis, while graphical representation of Petri networks enables the realization of structural changes in the system. This combination is the main reason for the selection of performance tests for this study. Simulation contains applications and tasks with the required skill, depending on the package routes. Passive monitoring which is the subject of performance testing is a way to track the performance and behavior of a broadcast package by measuring user traffic without creating new traffic or modifying existing traffic. This is used by integrating additional intelligence into network devices so that they can detect blocked processes, record features and the number of packets flowing through them. The IP header of the passive monitoring package contains the destination address of the receiving node and the source address of the sending node. The reception area sets the total (aggregate) total number of packets received and the current time in the corresponding header fields. The Augmented counting Algorithm (fig. 11) looks at the sender on the receiving package and seeks to identify any blocking. As soon as the work starts, the

machine is set to (not idle) and stores information. To better understand these behaviors, fig 13 shows a simplified flowchart from process perspective to focus on tasks and applications.



Flow chart for monitoring And visualization

Context Diagram



Data Flow Diagram

Figure 12: Flowchart for monitoring and Visualization

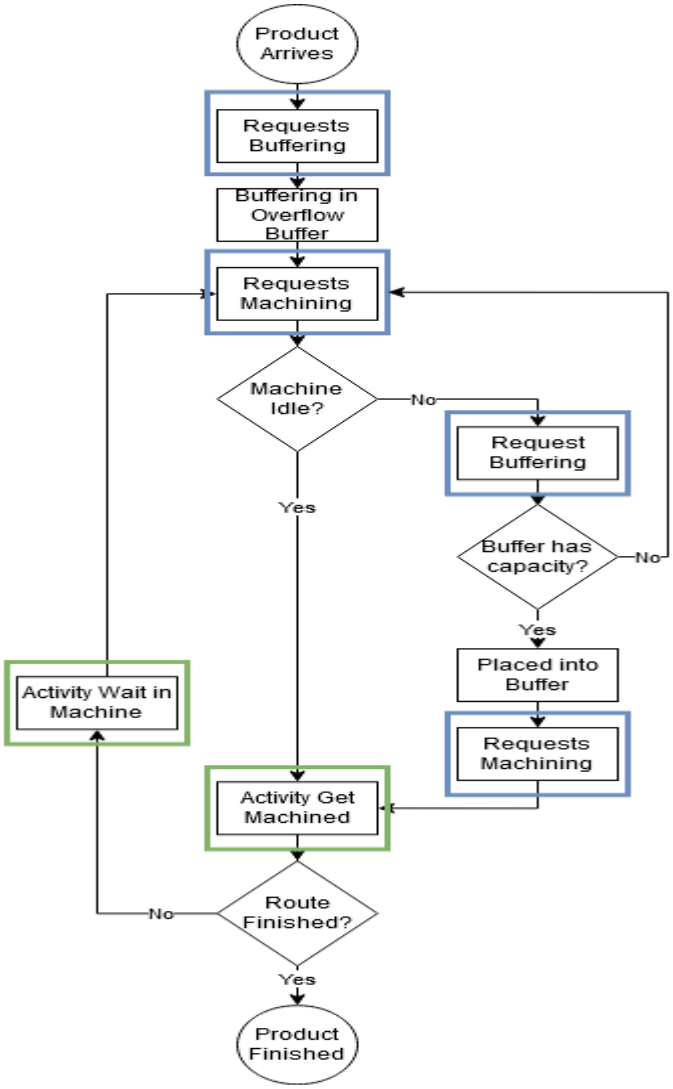


Figure 13: Flowchart of Requests and Activities

Packet Monitoring Algorithms

(A) Algorithms for Port Scanning

Step 1: Variable Declaration

Declare variables for storing IP Address and host name and set them to null

Step 2: Input

2.1 Enter value of Host name (or IP Address)

Step3: Scanning

3.1 Declare variable port = 0

3.2 Declare initial port = value.

3.3 Declare final port = value.

3.4 Check if the port is available between initial port and final port.

3.5 increment port by 1

3.6 Repeat step 3.4 up to final port.

Step4: Display

4.1 Display all the active ports in GUI format

(B) Algorithms for packet capturing

Step 1: Obtaining the list of network interfaces

1.1 Create a variable array of devices

1.2 Detect network interfaces present in user

1.3 Store the above list in devices variable.

Step 2: Displaying the list of network interfaces

2.1 Declare loop counter integer variable i and initialize to 0

2.2 While the value of i is less than the length of the array of devices, do Step 1

2.3 Print out the name and description of the captured Network Interface.

Step 3: Open the network interface.

3.1 Declare integer variable J and initialize to zero (J=0)

3.2 While J < length of array of devices, Goto Step 3.3 else Goto

Step 3.7

3.3 Check if the network interface at Jth index number in devices array is selected. If yes goto

Step 3.6 else goto

Step 3.4

3.4 J=J+1

3.5 Goto

Step 3.2

3.6 Open the selected network interface i.e. Network Interface at Jth index, then Goto

Step 4

3.7 Display that the network interface has not yet been selected by the user.

Goto Step 8

Step 4: Capture packets from the network interface

4.1 Is the menu button of stop capture packet selected? If yes goto

Step 3.8 else goto 4.2

4.2 Capture the upcoming single packet from the network

4.3 Display the captured packet by going to Step 5

Step 5: Display the captured packet to the user in proper GUI format.

5.1 Detect user' menu choice of the format in which captured packet's to be displayed

5.2 Analyze the packet. Display in Hexadecimal format

5.3 Goto Step 6 to save the packets to a temporary file

5.4 Go back to Step 4.1

Step 6: Save captured packets into a file

6.1 Create a temporary file say

6.2 Save captured packets into the opened file

6.3 Go back to Step 5.4

Step 7: Close all the open network interface

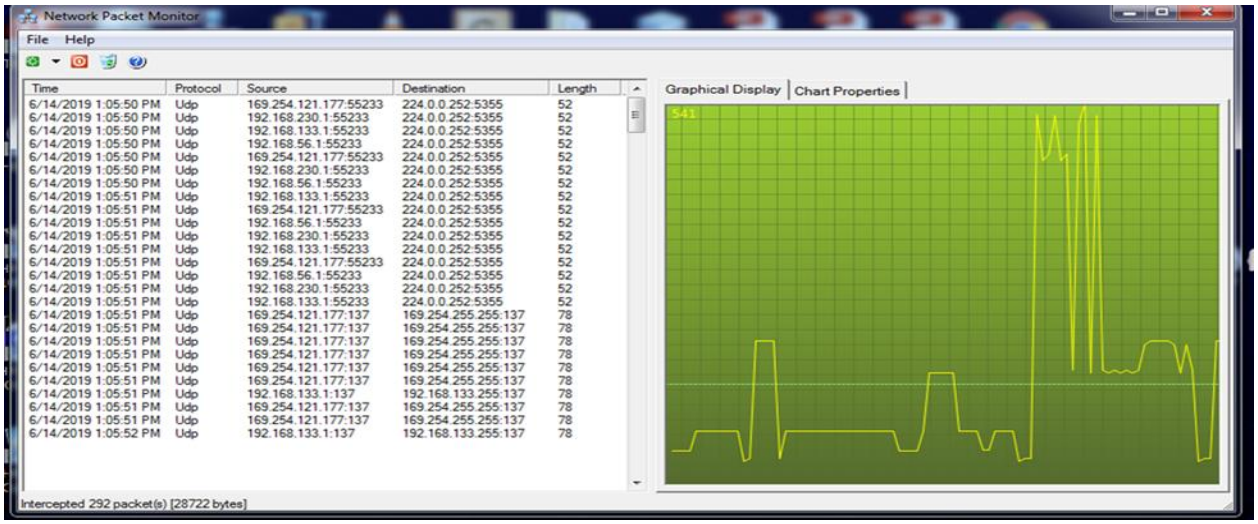
7.1 Delete the temporary file.

7.2 Close the network interface.

Step 8: End

This paper shows the effective siphon structure for PN analysis. This tool is very useful for finding a siphon and setting up an easily accessible tree, deadlock detection, and liveness of petri nets. The Petri net model as a tool helps us to take a deeper look at ethical and structural investigations. Table 3 shows the image representation of that collection. On the left side, from top to bottom is the main simulation time and at the top is the transfer protocol used (UDP), Source (IP address), Location (Internet address) and data size. Simulation as mentioned earlier includes Applications and Tasks. This behavior is used to identify the state of the system when the process is idle, separating the blocked machine. However another required request machine is filed and operations are performed successfully and this may result in increased packet transfer. This is clearly shown in the table on 06/14/2019 at 1.05.50 pm and 1.05.51 pm with data lengths of 52 packets and 78 packets respectively. The outlined steps conclude with the basic function of the structures that PN is a powerful and widely used simulation analysis strategy in which deadlock control mechanisms will be developed.

Table 1: Shows a Graphical Representation



Conclusion

The petri nets method described here has allowed us to make system modeling easier and faster compared to other analytical methods. We have concluded from this study that honeypot techniques are very effective in reducing the risk of deadlock

and, in fact, have a significant effect on controlling systems against hackers. What has been presented in this article is a brief review of rich information in the field of petri nets. It is not possible to discuss all aspects of the field on a single page. Therefore, emphasis is placed on the area known as transformation zones / programs, as well as the theory of petri nets used. Stochastic petri nets, high quality nets and their application models deserve more space, as there is a growing interest in these areas. This field is new and there is still a lot of work to be done. We hope that this paper will help to mimic further research and development in the emerging field of petri nets.

Acknowledgement

This paper and the research behind it would not have been possible without the exceptional support of my supervisor Professor Akinwale A.T. of Federal University of Agriculture Abeokuta, Ogun State Nigeria. His enthusiasm, knowledge and exacting attention to detail have been in inspiration and kept my work on track.

Most importantly, I wish to thank my loving and supportive wife, and my Four wonderful children (Pamela, Ohi baby, Ebehitale and Ehisoyayan) who provide unending inspiration. They are the ultimate role models.

Declaration of conflicting Interests

The author declared no potential conflicts of interests with respect to the research, authorship and publication of this manuscript.

References

[1] AndreyVishnevsky. Peter Klyucharev (2020): "A Survey Game- Theoretic Approaches to Modeling Honeypots". *Conference: Secure Information Technologies 2017 (BIT 2017): At Mouscow Russia.*

[2] Abigail Paradise; AsafShabtai; Rami Pluzis; AvladElyashar; YuvaiElovici; Mehran Rose (2017): "Creation and Management of Social *Network Honeypots for Detecting Targeted Cyber Attacks." IEEE Transactions on Computational Social Systems.Volume 4, No. 3, pp.65-79, Sept 2017.https://doi.org/10.1109/TCSS.2017.2719705*

[3] **Bakri, A. H., Alkbir MF, Awang N, Januddi F, Ismail MA, Ahmad AN, Zakaria IH, et al. (2021)**. "Addressing the Issues of Maintenance Management in SMEs: Towards Sustainable and Lean Maintenance Approach." *Emerging Science Journal, 5(3), 367–16. https://www.researchgate.net/publication/352371874_Addressing_the_Issues_of_Maintenance_Management_in_SMEs_Towards_Sustainable_and_Lean_Maintenance_Approach*. https://doi.org/10.28991/esj-2021-01283.

[4] **Barylska, K., Koutny, M., Mikulski, L., &Piatkowski, M., et al. (2020).**"*Reversible Computation vs. Reversibility in Petri Nets. Science of Computer Programming, 151 (1), 48–60*. https://doi.org/10.1016/j.scico.2017.10.008.

[5] B. Camiña, R. Monroy, L. A. Trejo, and M. A. Medina-Pérez. (2016): "Temporal and Spatial Locality: an Abstraction for Masquerade Detection. *IEEE Transactions on Information Forensics and Security 11, 9 (2016), 2036–2051.*

[6] Balogh, Z.; Kuchárik, M.(2019): "Predicting student grades based on their usage of LMS moodle using Petri nets." *Appl. Sci. 2019, 9, 4211. [Google Scholar] [CrossRef]*

[7] **Chakraborty, S. (2019).** "Analyzing Peer Specific Power Saving in IEEE802.11s Through Queuing petri nets: Some Insights and Future Research Directions". *IEEE Transactions on Wireless Communications, 15(5), 3746–3754. https://ieeexplore.ieee.org/document/7404028.*

[8] **Consuelo, N. (2020**). "Advanced Design for Manufacturing of Integrated Sustainability "Off-Shore" and "Off-Site" Prototype - MVP "S2_HOME*.". Civil Engineering Journal, 6(9), 1752–1764.*

[9] **P. Cazenave; M. Khifi-Bouassida; A. Togueyeni (2020):** *"* S3PMR Deadlock and Control with Partial Controllability and Observability".*Journal of International Federation of Automatic Control. 15th IFAC Workshop on Discrete Event Systems WOOES 2020-Rio de janeiro, Brazil, Volume 53, Issue 4, 2020 pages 173-179. https://doi.org/10.1016/j.ifaco.2021.04.017*

[10] **Davison, P., Cameron, B., & Crawley, E. F., et al. (2020**). "Technology Portfolio Planning by Weighted Graph Analysis of System Architectures".*Systems Engineering, 18(1), 45–58.https://doi.org/10.1002/sys.21287*

[11**] Dwyer, M., Cameron, B., &Szajnfarber, Z., et al. (2020).**"A framework for Studying Cost Growth on Complex Acquisition programs."*Systems Engineering, 18(6), 568–583.https://doi.org/10.1002/sys.21328*.

[12**] D.Dalla and J. Dheiba (2020**): "Exploration of Various Attacks and Security Measures related to the Internet of Things" *International Journal of recent Technology and Engineering", volume 9, No. 2, pp. 175-184, 2020*

[13] Dlamini, M.T., Venter, H.S., eloff, J.H., Eloff, M. (2020*): "*An Information Behaviour Lens". *In proceeding of the Information Behavior Conference, Pretoria South Africa, 28 september – Ist October, 2020.*

*[***14] Datta D., Garg L., Srinvasan K., Inoue A., Reddy G.T., Reddy ,M.P.K., Ramesh K., Nasser N. (2021):**Efficient Sound and Data Steganography Based secure Authentications System. Computers, Materials, and Continua *Volume 67, No.1,2021, pp.723-751.https://doi.org/10.32604/cmc.2021.014802*.

[15] DimitriosPliatsios; Panagiotis G. Sarigianidis; ThanasisLiatifis; IllasSiniasoglou (2019): *"*A Novel and Interactive Industrial Control System Honeypot for Critical smart grid Infrastructure". SPEAR: Secure and Private Smart Grid (H2020-DS-2016-2017*). Conference: IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks {CAMAD}; At: Limassol Cyprus*

[16] Ellard D., Jones C., Manfredi V., Strayer W.T., Tapa B., Van welleM.and Jackson A. (2015):"A Rebound: Decoy Routing on Symmetric Routes Via Error Messages*". In IEEE 40th Conference on Local Computer Networks (LCN) 2015, pp.91-99*

[17] FAHEEM Ullah, Matthew Edwards, RajivRamdhany, Awais Rashid (2017):"Data Exfiltration: A Review of External Attack Vectors and Counter Measures". *International Journal of Networks and Computer Applications. 101(2),*

https://doi.org/10.1016/j.inea.2017.10.016

[18**] Frederick Weigang Pan and Matthew Caesar (2016**): Salmon: Robust Proxy Distribution for Censorship Circumvention. *Proceedings on Privacy Enhancing Technologies". 2016(4): 4-20*. https://doi.org/10.1515/popsets-2016-0026.

[19] **Freeman, R.E., Phillips, R. and Sisodia, R. (2020**), "Tensions in Stakeholder Theory", *Business and Society, Vol. 59 No. 2, pp. 213-231*

[20] **Gammal E.I Selim; Ezz El-Din Hemdan; Ahmed M. Shehatta; Nawal A. El-Fishawy (2021):** "An Efficient Machine Learning Model for Malicious Activities Recognition in Water-Based Industrial Internet of Things. *Journal Security and Privacy, Volume 4, pp.1-14, Issue 3, May/June 2021.https://doi.org/10.1002/spy2.154*.

[21] Jan Komenda; Aiwen Lai; Jose Godoy-Soto; Sebastian Lahaye; Jean-LoiusBoimond (2020)*: "* Modeling of Safe Time Petric Nets by Internal weighted Automata". *IFAC paper online 53 (4), 187-192.https://doi.org/10.1016/j.ifaco.2021.04.018*.

[22] Konuk, F.A. (2018*),* "Price fairness, satisfaction, and trust as antecedents of purchase intentions towards organic food", *Journal of Consumer Behavior, Vol. 17 No. 2, pp. 141-148*

[23] **K.A. Shin (2019):** "Universal Firgery Attacks on remote Authentication Schemes for Wireless Body Area Networks Based on Internet of Things". *IEEE Internet of Things Journal, Volume 6, No. 5, pp.9211-9212, 2019.*

[24] Liang, X.; Zhang, S.; Liu, Y.; Ma, Y.(2020): *"*Information Propagation Formalized Representation of Micro-blog Network Based on Petri Nets". *Sci. Rep. 2020, 10, 1–20. [Google Scholar] [CrossRef] [PubMed]*

[25] Leyi Shi; Yang Li; HaijieFeng (2018): "Performance Analysis of Honeypot with Petri Nets." *Information Theory and Methodology (2018), {Switzerland}, 9(10):245*.
https://doi.org/10.3390/info9100245

[26] Lama Alhathally; Mohammed A. Alzain; Jchad Al-Amri; Mohammed Baz; MehediMasud (2020): Cyber Security Attacks: Exploiting Weaknesses. *International Journal of Recent Technology and Engineering (IJRTE) ISSN:2277-3878, Volume 8, Issues-5, January 2020, pp.906-913*

[27] MarcinWojnakowski; Remiguisz Wisniewski; GrzegorzBayzydio and Mateusz Poplawskwi (2021):" Analysis of Safeness in a Petri Nets Based Specification of the Control Part of Cyber-physical systems. *International Journal of Applied Mathematics and Computer Science 2021, Volume 31, No.4, 647-657. https://doi.org/10.34768/amcs-2021-0045*.

[28] **Mohammed Y.F (2020):** "Network – Based detection and prevention System Against DNS-Based Attacks." https://doi.org/scholarworks.uark.etd/3970..

[29] **Mouzur Paul (2015**): "Baidu and Cloud flare Boost Users over China's Great Firewall". *The New York Times-Archived (from the original on 24 January 2019* Retrieved 16 September 2017).

[30] **Manuel Cheminod; Luca Durante; Lucia Seno; Adriano Valenzano (2018**): "Performance Evaluation and Modelling of an Industrial Applications Layer Firewall*". IEEE Transactions on Industrial Informatics. Volume 14, No.5, pp. 2159-2170. May 2018.*
https://doi.org/10.1109/TII.2018.2802903.

[31] **PanagiotisRadoglou-Gammaliks; PanagiotisSariagiannidis, Eider Iturbe; Erkuden Rios, et al., (2021**): Spear Siem: A security Information and Event Management System for the Smart Grid". *Computer Networks".Volume193, July 2021,108008, pages1-26.*
https://doi.org/10.1016/j.comnet.2021.108008

[32] Pau Fonseca Casas; Daniel Lijia Hu: AntoniGuasch I Petit and JaumeFigueras Jove (2020): "Simplifying The Verification of Simulation Models through Petri Nets to Flexsim Mapping", *Applied sciences 2020, 10(4), 1395*. *https://doi.org/10.3390/app10041395*.

[33] Qin M, Li ZW and Al-Ahmari AM (2015): "Elementary-Siphon Based Control Policy for Flexible Manufacturing Systems with Partial Observability and Controllability of Transitions. *Asian J Control 2015; 17: 327–342.*

[34] **R. I. Ogie (2017) "**Cyber security incidents on critical infrastructure andindustrial networks," *in Proceedings of the 9th International Conference on Computer and Automation En[45]*

[35] Razzaq M; Ahmad J.(2015*):* "Petri nets and Probabilistic Model Checking Based Approach for the Modeling, Simulation and Verification of Internet Worm Propagations*". Plos ONE 10(12).e0145690.https://doi.org/10.1371/journalpone.0145690*.

[36] Ruotian Liu; Rabah amour: Leonardo Brener; Isabel Demongidin (2020*): "* Event Driven Control for Reaching a Steady State in Controlled Generalized Batches Petri nets. *Journal of International Federation of Automatic Control, volume 53, Issue 4 (2020), 180-186.*
https://doi.org/10.1016/j.ifaco.2021.04.063

[37] **SheetalGokhale; AshwiniDalvi and Irfansuddavatam (2020):** "Industrial Control Systems Honeypot: A formal Analysis of Conpot*". International Journal of Computer Networks and Information Security". 12(6):44-56.*https://doi.org/10.5815/ijcnis.2020.06.04

[38] **Souravlas, S. I., &Roumeliotis, M. (2015)**. "Petri net modeling and Simulation of Pipelined Redistributions for a Deadlock-Free System".*Cogent Engineering, 2(1), 1–22.* 2015.

[39] **Su, Z.; Qiu, M. (2019**): "Airport Surface Modeling and Simulation Based on Timed Coloured Petri net*". Promet-Traffic -Traffico. 2019, 31, 479–490. [Google Scholar] [CrossRef]*

[40] **White, A., Karimoddini, A. andKarimadini, M. (2020**): ".Resilient Fault Diagnosis Under Imperfect Observations—A need for Industry 4.0 Era,IEEE/CAA. *Journal of Automat-icaSinica 7(5): 1279–1288*

[41] **Wisniewski, R., Grobelna, I. and Karatkevich, A. (2020**). "Determinism in Cyber-Physical Systems Specified By interpreted Petri nets, *Sensors 20(19): 1–22, Article no.5565.*

[42] **Wi´sniewski, R., Wi´sniewska, M. andJarnut, M. (2019)**. "C-exact Hypergraphs in Concurrency and Sequentiality Analyses of Cyber-Physical Systems Specified By Safe Petri nets, *IEEE Access7: 13510–13522.*

[43] **Xiaoyang Chen; HongweiHuo; Jun Huan; Jeffrey Scott (2019**): "An Efficient Algorithm for Graph Edit Distance Computation". *Knowledge Based System, Volume 163, 1st January 2019, pages 762-775*. https://doi.org/10.1016/j.knosystem 2018, 10.002

[44] Xia, C. and Li, C. (2021). "Property Preservation of Petri Synthesis net Based Representation for Embedded Systems*", IEEE/CAA Journal of Automatica Sinica8(4): 905–915*.

[45] **Yi-Nan Lin; Cheng-Ying Yang; Gawa-lenCHiou; Sheng-Kuanwang; VivtorR.lShen; Yu-Ying Wang (2022):** "Smart selection from Petri Net Modeling Tools Fast Developing a Manufacturing System". *Cogent Engineering, Volume 9, Issue 1, 2022.*
https://doi.org/10.1080/23311919.2021.2020609.

[46] **YifanHou and KamelBarkaoui (2017**): "Deadlock Analysis and Control Based on Petri nets: A Siphon Approach Review*". Advances in Mechanical Engineering, 2017, Volume 9(5), 1-30.https://doi.org/10.1177/1687814017693542*

[**47] Yang, F., Wu, N., Qiao, Y., Zhou, M., Su, R. and Qu, T.(2018).**"Petri net-Based Efficient Determination of Optimal Schedules for Transport-Dominant Single-Arm Multi-cluster Tools, *IEEE Access6: 355–365.*

[48**] Zareef Mohammed (2020):** "Data Breach Recovery Areas: An Exploitation of Organization's Recovery Strategies for Surviving Data Breaches. Organizational Cyber Security *Journal.2021. https://doi.org/10.1108ocj.05.2021.0014*