IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 30, 2020, accepted October 15, 2020, date of publication October 28, 2020,
date of current version November 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3034383

Dealing With Well-Formed and Malformed Packets, Associated With Point of Failure That Cause Network Security Breach

MOHAMMED ABDULAZIZ AL NAEEM[1], ADAMU ABUBAKAR[2], AND M. M. HAFIZUR RAHMAN[1]

[1]Department of Computer Networks and Communications, College of Computer Science and Information Technology, King Faisal University, Al Ahsa 31982, Saudi Arabia
[2]Department of Computer Science, International Islamic University Malaysia, Kuala Lumpur 53100, Malaysia

Corresponding author: Adamu Abubakar (adamu@iium.edu.my)

This work was supported by the Deanship of Scientific Research at King Faisal University for the financial support under RAE'D Track under Grant 187011.

ABSTRACT It is quite challenging to understand the weakest single points of failure in a network because it is the point where an entire network system can be taken down. The paths leading to a point of failure, and the status of packets that causes network security breaches were examined by Intent-Based Networking approach in this study. Two algorithms are proposed, utilizing single-path and multipath in transmission flow. Every path is potentially weak and a point of failure for which a network security can be breached. Two sets of rules, namely, ''vulnerability rules policies'' with ''rules formulation'' from the regions of connection recognized by Euler's theorem were outlined. The intent is to use these sets of rules in finding the point of failure the packet status that is leading to possible security breaches within network connections. The frequencies of the packets that are liable to create security breaches and the paths where they originate are analyzed. Well-formed packet originating from the least likely weak point of failure is associated the network security breach than malformed packets. This study has contributed to revealing that network security breaches are influenced by the paths with least likely point of failure from well-formed packets.

INDEX TERMS Well-formed packet, malformed packet, point of failure, security breaches.

I. INTRODUCTION

Network security breach happens when confidential configuration information and communication data are exposed by unintentional or intentional means [1]. In a network environment, a security breach is initiated where there is potential access to a network by an unauthorized person or application [1], [2]. At the time where an adversary or attacker gets access to a network, even if nothing damaging happens, a security system has been broken. The majority of the network security breach lies with password hack or leak, virus or malware, reuse of hardware or software storage and transfer of sensitive information, or missing patches and updates. Preventing network security breach is the responsibility of everyone involved in using the network. Software-based network security packages like antivirus and firewalls are fairly available. Similarly, many vendor-based security services for monitoring network are also available.

Checking for abnormal issues in a software development process is very common [3]. In many cases, after a software deployment, certain problems arises, bugs developed over time, and in the most crucial cases, patches and vulnerabilities appear, which leads to some serious security problems. This kind of incident happens in both local and enterprise networks. Immediately after setting up a network, specifically the security setup of the network, administrators usually use various means to monitor any unauthorized access and potential damages that might be caused by an attacker. Intrusion detection system is one of the important and stable technique of identifying any anomaly or malicious or suspicious events that could cause problems to the network system [4]. Intrusion detection system not only find abnormal events, but they also observed network traffic that does not follow the expected pattern [5]. Intrusion detection systems are mostly software-based and are designed without regard

The associate editor coordinating the review of this manuscript and approving it for publication was Eyuphan Bulut.

to flexibility. They typically pick up signatures or pattern based on classifying or clustering an abnormal piece of data that could be dangerous to the security system [6]. They are described as a host-based intrusion detection system, which are usually installed on the client computer, whereas, the network-based intrusion detection system resides on the network. An efficient and flexible monitoring of anomalies is crucial. A dynamic and flexible security system of a network is required in the current network systems. The one that can serve for real-time detection of any irregularities. Any changes within a flow process, the system can be able to respond accordingly. Intent-Based Networking is likely the area that could provide detection of any point of failure in a network system and security breaches [7].

Intent-Based Networking can be described as an enhanced Software-Defined Networking (SDN). It comes as a result of the need for any certain aspect of networking tasks to be handled by user. That is users can define and refine any forwarding and controlling tasks by themselves [8]. The reason why SDN has become an alternative from the traditional network is that it provides an opportunity for resources and bandwidth to be managed instantaneously on-the-fly. SDN helps in solving the problem of configuring each network device separately [9]. This has circumvented the previous need for network hardware requirement if capacity needs increase. The separation of network control and data planes in SDN architecture, makes it programmable, adjustable and dynamically re-configurable [10]. As a result, a large number of leading companies across the world are adopting software-defined solutions in their data centers. It is expected that most of the service providers, will join in the near future. This is due to the new opportunities it provides. Regrettably, for every new SDN it is also accompanied with some issues and concerns, about fault tolerance and recovery [8]–[10].

Intent-Based Networking represents a fundamental change in how networks are designed and managed [11]. Rather than focusing on the process of selecting hardware components of a network, the designers now focus on the applications and their requirements [12]. Intent-Based Networking was established based on the advancements in networking technologies that provide a high-level descriptive language to query network-application requirements [8]. Intent-Based Networking at a software level provides an APIs that enable network control. Various set of rules or standard policies intended to sets what to achieve, rather than mechanism on how to achieve are established Intent-Based Networking [12], [8]. This paper utilizes these advantages that Intent-Based Networking provides, and focuses on network security. The current research intent-concept relies on establishing the paths leading to a point of failure that causes network security breaches. An organization that is implementing Intent-Based Networking can benefit from this concept. It will be able to gain from planning, designing and operating networks in a way that it will improve their network availability, agility and security [13]. The change this research is introducing is a new way of networking technology interactions. Because now is possible for networking systems to be customized in order to suite the global communities' social interaction for the exchange of ideas and information that has the potential to increase opportunities. Network managers now have the opportunity to use Intent-Based Networking capabilities to implement and identify the network policies necessary for achieving a high level of application performance [11]. Intent-Based Networking allows the use of specified policies to automate various network operating tasks. Networking components can be able to integrate services and application. This is a new ways of handling networks and can help to create a network system that is free of any obstacles.

The reaming parts of this paper are organized as follows: Apart from the present section that provides an overview of this study, section 2 presents related works. Section 3 presents the conceptualization. Section 4 presents research methodology and section 5 presents the results while section 6 presents discussion and section 7 is the conclusion of the paper.

II. THE TREND OF COMPUTER NETWORKING

Digital communication is now one of the major components of human existence. Many people have mobile devices that enable them to interact with others. Communication is almost as important to us as other resources that humans required [14]. Digital communication relies on the use of networks, that provide the means to be connected with everyone like never before [15].

Currently the majority of the emerging businesses require a lot of data network resources to improve the quality of life for people everywhere. The wide spread of data networks leads everyone in remote locations to contribute on an equal basis with everyone in every part of the world [16]. The use of the Internet has become an integral part of human activities and has succeeded faster than anyone could have imagined [17]. The way in which people engage in personal, political, social, and commercial interactions has grown rapidly and is changing to keep up with the evolution of the global network. The network technology is growing fast to cover for almost all various platform of communicating with people in social and business relationships [18].

Since the 1980s the centralized architectural network control system has been very robust [19]. As developers push the limits of what is possible, the capabilities of the interconnected networks that form the Internet increases. By the 1990s, programmability in managing network was introduced [20]. Communication over Internet increases and provides an easier way to assess products and services as well as exploring the services, technologies, and issues encountered [21].

Prior to the invention of Software Defined Network (SDN), the transmission control process for a Network are embedded on networking devices by means of some set of rules [22]. During this period, the driving application for the central/programmable network was missing [20].

During this period, many vendors embed set of defined tasks of networking devices which were not flexible and do not provide network managers the opportunities to redefine

any aspect of network management operations [23]. Networking device was like a black box preventing users from modifying them. The management of this early networks is complicated because network device works based on distributed protocols embedded in them, and these protocols require configuration. Furthermore, each device across the entire network needs the configuration of the protocols. The device mode of operation has increased the level of complexity of networking system management. Network administrators find it difficult in a situation where they have to manage hundreds to thousands of network devices. Network faces a lot of difficulty of handling data transfer in various scales of a network [24]. Network managers had to set up each networking device manually through some set of configuration tools where only the parameters defined by the manufacturers are used. The emergence of virtualization and cloud computing in the datacenter provide the SDN to becomes the right application [20].

III. CONCEPTUALIZATION AND ALGORITHM DEVELOPMENT

It is the responsibility of Network Security Administrator to use various techniques to detect, verify, and explore any exploits within its network. These tasks involve some tough activities that require both computer resources as well as human efforts. There are different degrees of network attacks [25]. This paper adopts Euler's theorem, and conceptualized ''paths leading to the point of failure to be associated with the causes network security breaches''. The path attributes and packet status, that causes security breaches are the controlling variables. The path attributes involve ''least likely'' and ''most likely'' weakest path that leads to security breaches. The packet status involves well-formed and malformed packets. The well-formed packet refers to the normal packet, which was transmitted successfully. The malformed packet is the abnormal packet.

The relationship between the Euler's formula and the method of finding the ''least likely'' and ''most likely'' weakest point of failure is that, Euler's formula works on specific planar graph. That is, it defined a bounded area, ''FACE'' in Euler's formula, established focus to certain points/node related to a specific subarea. Within those points, this study conceptualized ''least likely'' and ''most likely'' weakest point of failure. ''Least likely weakest point of failure'' is a point/node within the attack surface of a network that is least likely possible point of security risk exposures, whereas ''Most likely weakest point of failure'' is a point/node within the attack surface of a network that is most likely possible point of security risk exposures. Attack surface is the total sum of all known, unknown, and potential vulnerable possible security risk exposures [26]

In transmission session, a packet is said to be malformed when any of its parts (header or payload) is not as expected to be, or it is not following the standard protocol specifications [27], [28]. Malformed packets, either violate approved protocol or in some certain ways get corrupted. Most of them are distorted and out of order or contains code aimed to confuse or disrupt any network service. Malformed packet can be generated in transmission session, when a packet arrives at the destination ahead of its transmission signal, that is a packet that arrived even before TCP 3-way handshake get established. Usually, these types of packets are discarded in order to avoid any error when it is delivered to the destination [29].

In a network attack scenario, malformed packet is a deconstructed packet that is configured intentionally, targeting a specific destination host, with a length that is not in compliance with the protocol, and the range of the packet field overflowed, with abnormal packet header [30]. Malformed packets are used to breach a security system of a network [31]. This is a special case of an attack called ''malformed packet attack''. A malformed packet attack occurs when an adversary intentionally sends incorrectly formed IP packets to the victim system in order to crash it or to achieve some desired goals [29], [31]. Typically, malicious data in the malformed packets are the major sources of remotely launching an attack. In some critical situation, a single malformed message is capable crashing an entire remote server [29]. There are basically two categories of malformed packet attacks, namely: ''an IP address attack'', and ''packet contains manipulation attack'' The IP address attacks is simple where the attack is on the specific packet for transmission in which both its source and destination IP addresses are same [30]. The effect of this attack is directly felt on the functioning of network operating system, and leads it to crash network. Whereas in the attack directed to the packet content (header or payload), the header content of a packet is modified, specifically, the ''optional fields'' within an IP packet. Usually attackers alter with this field by changing all quality of service bits to one. This can lead to the victim system computation process for analyzing traffic to increases thereby exhausting the processing ability of the victim system.

It is worthwhile to mention that the aforesaid works [27]–[34] focus on detecting and preventing an IP malformed packet, unfortunately, how to employ examining points of failure for which malformed packet sneaked-in has not been greatly discussed. Hence, following the previous successful conceptualizing Euler's characteristic and vulnerabilities discovery and lifecycle model, and motivated by the Intent-Based Networking protocol capability to control sets of networks, this paper proposes techniques for exploring point of failure as well as determining the proportion of malformed packets that can cause problem to a network.

A graph is a connection of dots with lines between them. The ''dots'' represent ''vertices(V)'' or node, and ''lines'' connecting the vertices, are referred to as ''edges (E)'', while the ''regions'' established after connecting the vertices and edges are referred to as ''faces (F)''. Euler's proved that $V - E + F = 2$ [35]. It means that the numbers of vertices, edges, regions and the connected components in graph for any plane connected graph is 2. (see Figure 1).

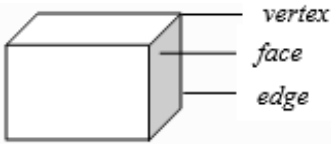

FIGURE 1. An Organizational Euler's Characteristic.

Taking this consideration to the network design and assuming all the faces in each network design to represent potential security breaches, Euler's characteristic proves that the faces f in graph G includes holes (h). This means that if we consider potential security breaches as the ''basic faces'' (f^i) for which the holes are generated. If h and f^i in G, are the crucial security parameter within any established network connection, then, $f = h + f^i + 1$ [36]. This concept allows for generating an Euler number of a security breaches within a network by use of the numbers of vertices, edges and basic faces in its corresponding graph. Similarly, this is an approach from graph theory that the conventional network theory relies, in which a network is developed by a relationship between vertices and edges, ''faces'' is not considered.

A. VULNERABILITY RELATED CONCEPT

There are already existing models suitable of detecting flaws that might endanger software programs. Crucial to this are vulnerability discovery and vulnerability life cycle models. Vulnerability discovery models have been used in the software based for quantifying and classifying abnormal flaws that might exist in a software program [37], [38]. Similarly, vulnerability lifecycle models utilized age of a vulnerability for developing a likelihood of an exploit or patch being available over time after its disclosure date [39], [40]. The vulnerability discovery model is based on time and effort. This approach entails that vulnerabilities arise over time and effort. Typically, while a point of failure is established by an adversary chances of attacking a system increases, from the point of view of this model, attacks can happen because the target system was exposed through some vulnerable points and with the effort made by an attacker to expose those vulnerable points over time, then an adversary can be going towards discovering significant number targets over the time resulting in an amplified growth until the discovery process get saturated [37].

Previous research studies have shown the effectiveness of this model. Specifically, on discovery of detection by feature enhancement and shared code [41], identifying and removing vulnerability, logistic detection rate while discovering vulnerabilities [42], hump-shaped model to capture the vulnerability [43]. These outcomes greatly impacted the use of vulnerability discovery models. The vulnerability lifecycle model relies an understanding of the cycle of processes that yield an experience for a better decision making options. The vulnerability lifecycle model uses a collection of disclosed vulnerabilities over time as an indicator of the increasing risk exposure [39]. The model uses two important control variables: namely, discovery and exploits. The range of time it takes from the beginning of discovering a vulnerability that pose a security risk is described as the time of discovery, whereas, the period which it takes for a system to be exposed from the known vulnerability is described as the time of exploits. An exploit within a network or software system is a piece dangerous data that takes advantage of a vulnerability and performs an undesirable task [44].

The vulnerability discovery model proposed by Alhazmi and Malaiya [37], aims at capturing the number of undetected points of failure over time. The model expresses the association of potential flaws that might be discovered with the flaws that triggers the detection of other supplementary flaws during the discovery process. This means that there is a point of failure where an adversary finds to breach of the security system. Therefore, the detection rate of attempts to breach the security system r depends on the certain number of those points discovered. Additionally, security breaches detected due to the influence of the point of failure over time is t. Frei *et al.* [39] revealed that ''exploit availability before and after the disclosure is found to be best matched with a Pareto distribution'' Hence, the possible distribution of the exploit can be expressed by $\mathrm{F}(t) = 1(k/t)^{\alpha}$ which is the probability of possible security breaches detected due to the influence of the point of failure discovered over time t and k is the security breach being discovered by time t. When α is 0.40 the distribution is under ''before disclosure'' and when it is 0.26 it is under ''after disclosure''. This is based on matching the disclosure time, because it changes the dynamics of releasing exploits. That is why there is a decrease of exploit availability right after disclosure.

The path in a network that lead to a point of failure, might be single or multi-path. Single-path provisioning has been identified to consume more resources than multipath provisioning in terms of transmission flow [45]. However, for security breaches, every path is potentially weak. Vulnerability discovery and lifecycle models are used to establish rules, policies that are made available to an Intent-based controller. The intent is to use them in finding the point of failure for every Euler faces, and the status of the packets within those paths. Therefore, algorithms were formulated based these rules.

B. THE PROPOSED ALGORITHMS

The proposed algorithms were based formulated rules from Euler's characteristics and vulnerability discovery and lifecycle policies. Euler characteristics is the concept that lies in graph theory. This study conceptualized a ''Network Scenario'' with four subnetworks and a control plane that can configure the appropriate rules across the entire subnetworks connections (see Figure 2). The step-by-step process of implementation of the algorithm is presented in Figure 3. The control plane is capable of continuously applying the rules set out in real time as the business intent of the network, and can take corrective actions as well. The network involves four local networks with one outside link through

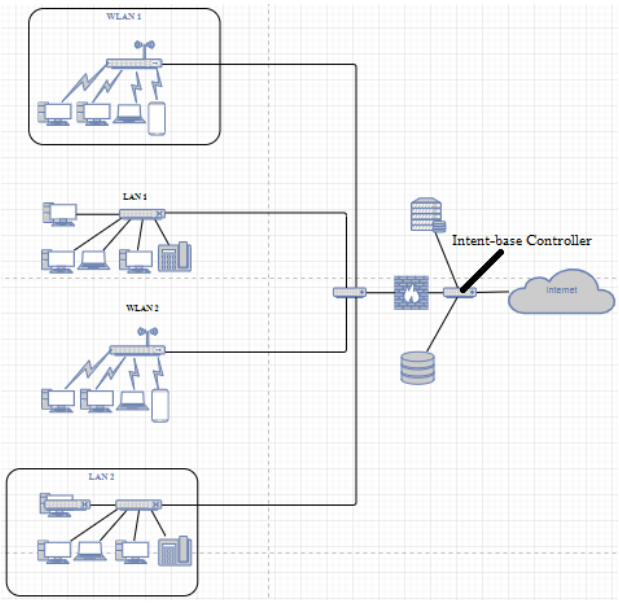

FIGURE 2. Intent-based controller Network Proposed Setup.

the Intent-based controller. Both the local networks and the outside link incoming and outgoing packets pass through the Intent-based controller. IP packets are adopted for this particular analysis. This is because the practical use of network is mainly dominated by IP.

The problem with which this research attempts to examine is the security breach scenarios in connections. Followed by the understanding of how point of failure and packet status can lead to the primary cause of a security breaches in a Network.

Theoretically, a security breach can evolve from any point in a network, this study examines each connecting point in a network as a point of a security breach where the desire ''intent'' is to monitor these breaches and to established a policy for decision making. Also the study attempts to set out the monitoring scheme of critical security breaches. The point of failure where network security is breached and the path leading to this point by Intent-based controller using Algorithm 1 and 2 respectively.

The Intent-based controller also evaluates the attacked coming into a network and the point of failure where the network security is breached. The attack capacity, the sequence of the attack, and the attack time are crucial variable for a flow analysis involving a network attack. All these are analyzed by the Intent-based controller. The Intent-based controller collects all the network communication data for incoming and outgoing for all the four local network and the TCP packets elements like (bytes in flight, connection syn, data-text-lines, duplicate_ack, flags, malformed e.t.c).

The proposed algorithms dwell on rules. Algorithm 1 set out the paths to point of failure, whereas algorithm 2 outlines the point of failure and packet status for security breaches. Considering the ''face'' (F) formed when a vertices (v) and edge (e) are connected in order to form a network as a single monitoring point (M) based on Euler's characteristics, the algorithm proposed set out rules that at every network potential security breaches (S) there is a point of failure ''node'' (G) by the sets of single monitoring rules (M). The network security breaches through the points of failure are generated by all possible ''F'' for which all f changes are observed at (M).

Algorithm 1 Path to Point of Failure

Input: *Path established in a network of* (V, E, F) *for communication of* $A_i \rightarrow B_j$'' (G)
Output: *path to point of failure* (" ") $\leftarrow M$
initialize security breaches $S = \{x_1, \ldots, x_n\}$

1. **Begin**
2. $f = h + fi + 1;$
3. **While** *true* **do**
4. Select x *from* F *for* which $a_i \rightarrow b_j = G$
5. ***if*** $(v_{i-1}$ & e_{i-1} *are connected to* V and $E)$ *THEN*
6. RuleSet $\boldsymbol{i} ==$ number of V and $E \forall\, ai \rightarrow bj \in \mathbf{G}$
7. **ELSE**
8. ***for*** Rule $j \leftarrow 1 \in M$ **do**
9. ***if*** Rule $j | a_i \rightarrow b_j \in S$ &$\Leftarrow\Rightarrow G$
10. Select count $(V$ & $E)$ *connection*
11. **END**
12. **END**

Line 1 through 6 in Algorithm 1 scans the entire flow of connecting nodes (based on Euler's rule) for point of failure where major security breaches ''get in''. The breaches are set out based on vulnerability discovery and lifecycle rules. Similar to the approach performed in Frei *et al.* [39], where the likelihood of exploits in a program is calculated. The correctness of this lies with time of a vulnerability development. This is because it is quite challenging to understand the most dangerous single points of failure. Line 8 to 10 in Algorithm 1, established rules for the connection parameter of the network. These are where the ''faces'' representing a point emerging as a connection of a network are calculated. This represents the major paths of point of failure where the security breaches can ''get in''. Two rules are necessary, for determining the paths to the point of failure:

''alert for TCP traffic length from the connection''
''alert for normal and abnormal record route from both side of the connection''

The rules set out in the algorithms consider the paths and packet lengths. Considering that IP has a specific header length that must be fixed, the total length defines the validity of the packet. The flow rules of IP are influenced by TCP some of the major variable involve are IP packet size and port number. Utilizing this, the Intent–based controller provides more abundant information of the current and future security breach by Algorithm 2. This is set out through the analysis of security breach behaviors and simulating incremental network penetration similar to a study performed in Hu *et al* [5].

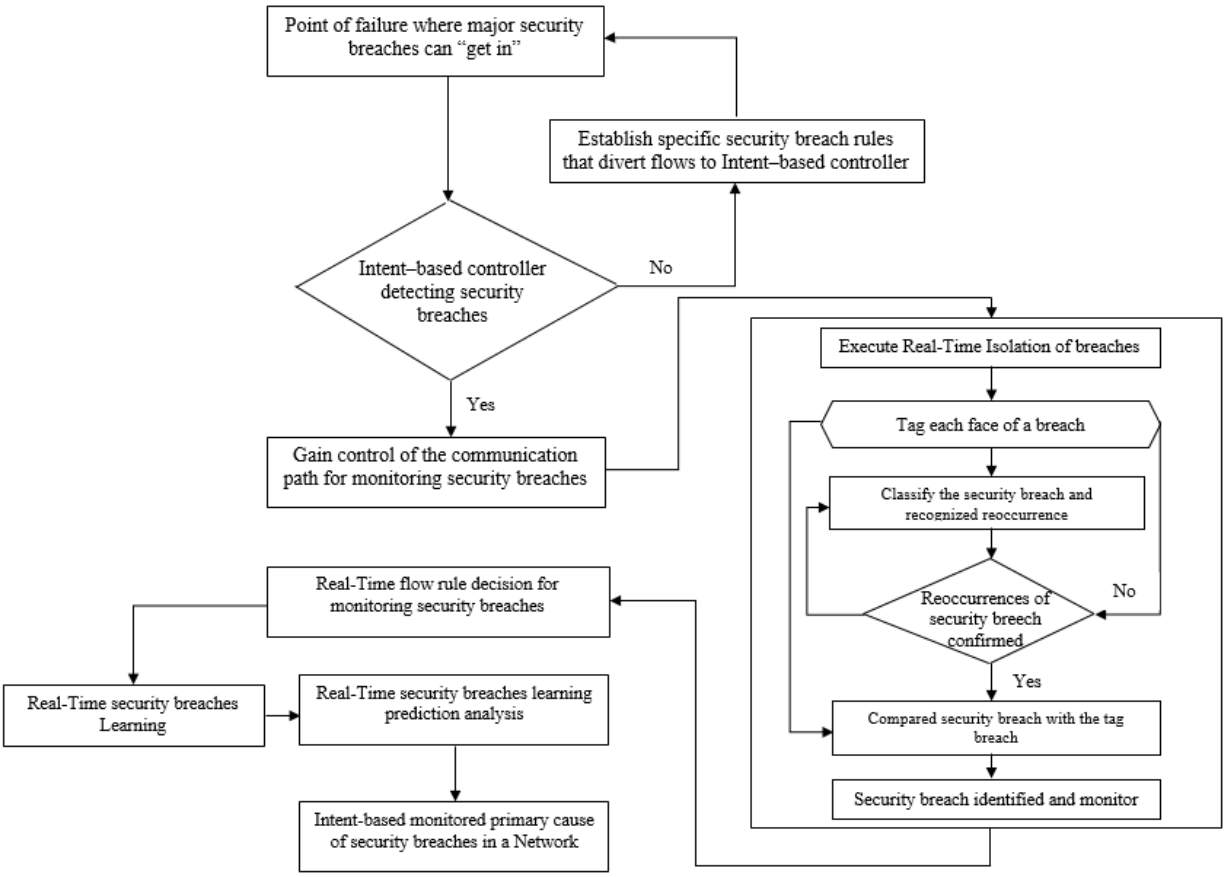

FIGURE 3. The flow Process of the Implementation of the algorithms.

Line 2 to 5 in algorithm 2 scan the connected regions for any abnormal flow to determine the TCP packet status and the point of failure. Line 7 to 11 further analyses the rules for the data integrity of the communication session to determine any abnormal data content. Line 12 to 15 combined both flow and data check to determine point of failure established by the rules:

"alert for TCP Breaking Point of the connection"

"alert for TCP traffic content from the connection"

The Intent-based controller has the capability of timely evaluating security breach through the abnormal flow and abnormal data content. The Intent-based controller identified and detect the security breaches, then the path is monitored for security breaches.

The implementation of the real-time isolation of breaches is carried out for the purpose tagging each part of a network security breach. There might be option of either to ''classify the security breach and recognized its recurrence'', or ''compared the security breach with the tag-breach''. That means if a security breach is identified, they are tagged and monitored. Thereafter real-time flow rule decision for monitoring security breaches will continue. Finally, the algorithm learned all the security breaches (see Figure 3).

After the network communication simulations for all the transmission sessions, link and packet analysis were carried out. This involves all details about the flow and the captured data during the transmission. The analysis of the flow status and the packets associated with the flow, specifically the IP header follows. This analysis depends on the rules set out for determining the paths to the point of failure. The algorithms will raise some IP alerts for the TCP transmission track for all the traffic length of all the connection. This is coded and embedded on Intent-Based Networking controller.

Intrusion detection systems are doing great jobs in detecting well-known attacks on the basis of some setup alerts or signatures. Most them are software based or on certain fixed operation service. The set up rules in them are rather fixed, not flexible. That is why utilizing Intent-Based Networking is crucial. The intent is to provide for recognizing the point of failures and the behaviors associated with suspicious content and activity that may be the result any security breach.

IV. METHODOLOGY

A network scenario is built (see Figure 2). It includes the Intent-based controller with the capability of detecting any communication (outgoing/incoming) from hosts at the southbound out of the network. Four networks and a controller

Algorithm 2 Point of Failure

Input: A communication between *Ai* & B_j (*V, E,* $F \in S$)
Output: *Point of failures on* communication between "$A_i \rightarrow B_j$" (G)

1. **Begin**
2. ***for*** each $\boldsymbol{f}_i \in F(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n)$ *within connected region* **do**
3. ***if*** $h, f_i \in G$ *from the* "$A_i \rightarrow B_j$" *connected region*
4. **While** $G \cup S_n \neq \emptyset \in$ RuleSet *i*;
5. *Get F connects with the exterior of* $f = (h + f^i) + 1$; *//point of failure*
6. **ELSE**
7. ***for*** *every communication between* "$A_i \rightarrow B_j$" (G);
8. ***if*** $(F(f_i) < S(X_i)) \in \boldsymbol{G}$;
9. ***Check*** *the desirable packet size* THEN
10. ***If*** (*packet size* $\neq$ *TCP*$\rightarrow$*min-max)*;
11. "Ai$\rightarrow$Bj" = G *//point of failure*
12. **While** RuleSet *i* true **do**
13. ***if*** $v - e + f = 2 > 0 \mid S(s_{i,\ldots,} s_n)$ THEN
14. ***if*** $f = h + f^i + 1 > 0 \mid S(s_{i,\ldots,} s_n)$ THEN
15. F $\leftarrow$ S == G
16. Update the Change (*h* and *fi* $\in$ G $\leftarrow$ S)
17. ***END***
18. **END**
19. **END**

are set out in the experiment. The security of the network can be broken, but any attempt to do that can be tracked by the Intent-based controller. If an attacker does breach the network, then that attempt within the path is quantify, the "point of failure is recorded and the status if the packet is examined for security breaches". The Intent-based controller accepts all well-formed and malformed packets. The experiments involve sending some series of packets, various kinds of packets over the connections. The Intent-based controller tracked them all based on the rules set out in the algorithms 1 and 2.

The Intent-based controller uses a Python program implemented based on the Euler's characteristics and vulnerability discovery/lifecycle rules. The network simulations were set out based on these rules. An experimental simulation scenario with the following environmental setup: RYU Controller, MININET, Open-V-switch (OVS), and Wireshark. The simulation was built in MININET and Ryu controller, which is a python abstraction on top of OVS and of one of the most popular OpenFlow controllers that have been used in the SDN community. It's an app based framework. An Intent-Based Networking development stack was built with the materials above. Ryu is a component-based software-defined networking framework that offers software mechanisms with a well-defined API that makes it easy for developers to generate new network management and control applications. It supports a wide range of open network protocols such as OpenFlow. This study utilized it as the Intent-based controller, because it enables communication between the "Control Layer" and the "Infrastructure layer" using the OpenFlow protocol. MININET was used as an emulation software because it facilitates creating and manipulating Software Defined Networking components. Open vSwitch (OVS), a multi-layer software that supports standard management interfaces and opens the forwarding functions to programmatic extension and control. Wireshark is a network traffic analyzer. It captures every packet getting in or out of a network interface.

The simulations specification dwells on the algorithms proposed, and is implemented in Ryu controller, because it provides "Python components API" that make it easy for coding the proposed algorithm. The Ryu manager spins up and listen for connections from everywhere. The most important tasks of Ryu manager is to manage the flows and anything on OVS. Once the manager startups it builds a data rebuke and spins up a web services and some of the basic events. The flow functions deal with forwarding and tracking where IP packet is coming from and where it's going to. That is where the interaction with what the controller is going to do for us was set. It first acquired the information on wherever IPs are coming and mapped the data information with the IP. Then when a packet comes in, each flow table contains a set of rules on what is going to happen to the packet. In this specific scenario the rules for isolating "Malformed Packet" and "Point of failure" are set and the controller would map them out and tag as a breach by a variable "tag breach". As a result, when a packet comes in, it inspects it and if it already knows its data path and is sure is on correct interface, it then checks and update the flow table, in order to examine the least and most "weakest" point of the of the communication as far as security is concerned.

The first tasks after writing codes in python was setting out the environmental variables is to check the TCP connection between the local networks and the controller. This opens TCP listening port 6633, then we run the mininet topology, the local subnetworks establishes TCP session with the controller. This was verified by Wireshark filter of TCP port == 6633. The controller and the OVS are connected and set of messages are exchanged. This transmission was captured by Wireshark. The experiment used packets for the transmission session based on the severity of their consequences. The simulations use the design of the point of failure and monitoring of primary cause of security breaches in a network by detecting malformed messages in real time for avoiding any network hazard. Intent based controller exploits any packet within byte-level sequences to determine the malformed packet by using Euler's characteristic and vulnerability discovery and lifecycle model to extract all the discriminative features and uses them to monitor any attacks launched through malformed packets in real-time.

From both directions, the network was flooded with both well-formed and malformed requests, using random time intervals within 30 minutes between the flood bursts.

V. PRESENTATION OF THE RESULTS

The simulation results are presented in this section. All the transmission coming from the four local networks and (both malformed and well-formed) packets are recorded. The Intent-based controller was able to extract the sequence of transmission 30 minutes' simulation. This result is analyzed for point of failure for security breaches and the packet status. The analysis of the results detected the security breaches, with time factor associated with each point of the security as breach is presented.

The entire packets captured for the 30 minutes' transmissions are 84772. The general distributions of the data are found to match Pareto distribution. Out of this packet, the once that are liable to cause security breaches are presented in Table 1. The result indicated that the payload or the bytes-in-flight are the highest and are the well-formed packets (see Table 1). There are many retransmitted packets. These are the packets that are sent and were not transmitted because they did not get an acknowledgment, and they are retransmit again. A duplicate packet was also tracked. The number of the duplicate is presented in Table 1. Since TCP is a connection oriented protocols, and if eventually the first packet was received, and the same one was also received again, TCP will detect the packets as duplicate packets, the protocol is for TCP to ignore them completely [46]. The analysis indicates that TCP did not ignore the duplicate packets because they are intentionally prepared in order to attack the network. Although, some of the duplicate packets are sometimes generated by defective network hardware/software which also causes unintended problems to the network. The total number of duplicate packets transmitted by all the nodes in the transmission is gathered. Each time a TCP detects a retransmission loss, duplicate acknowledgment counts [47]. According to Lin and Kung [48] 4% of the timeouts are due to the huge number of duplicate acknowledgment count.

The results also gathered ''Flags'' which are responsible for indicating any problems that might occur in TCP operation, which include problems related to the port number establishment, issues associated with segmentation ambiguity, and mismatch sequencing of the packet as well as flow being out of order. A huge number of the TCP flags (2324) were examined and they indicated the number of unexpected sequence of unexpected acknowledgement number and unexpected segment length. This will ensure reliability. TCP control flags are in its header.

A flag is raised at the establishment of a new connection, when transmission end and when one end wants to abort connection. TCP controls transmission of packets in accordance with the condition of a network. For instance, TCP would restrict the number of packets to be transmitted if it encounters network congestion [49], [50].

In the TCP transmission session, data packets are free to follow different in order from which they were sent and to arrive their destination in a different order. One of the important functions of TCP is to reassemble out-of-order delivery of data, to in-order at the destination [51]. If the TCP connection does not drop or reorder packets, that means they automatically arrive in-order at the TCP receiver [52]. This could be a point of failure for the fact that the next expected sequence number could be tricked to be greater than the current sequence number, especially in a heterogeneous path of 5G device and the server [53]. After evaluating the frequencies of the packets that are liable to create security breaches, the paths where they comes are also analyzed. The path attributes analyzed are ''least likely (LS)'' and ''most likely (MS)'' weakest point of failure that leads to security breaches. This comes from series of analysis, which captures of the path length, associated with data that can cause security breach. The results presented are for the simulations of 10, 20 and 30 minutes in Figure 4, 5 and 6 respectively.

The path analysis for the first simulation involves: 1) paths where the packets that do not posed any security breach recorded as LS and 2) paths that content packets that most likely will cause security breaches recorded as MS. Series of packets are transmitted from the local network, the simulation indicated that least likely paths are also associated well-formed data and malformed data (see Figure 4). The most likely point of failure steadily drops in packet length and was found to be associated with malformed packet only.

TABLE 1. The distribution of packets.

Name	Number
Bytes_in_flight	37540
Connection_syn	672
data-text-lines	39
Duplicate_Ack	4
Flags	2324
Malformed	14
Option_kind	1330
Out_of_Order	11
Payload	37540
Reassembled PDU in frame	21530
Reassembled_Data	6975
Retransmission Timeout	12
Segment_data	28060
Stream	61
TCP_Port == 80	21461

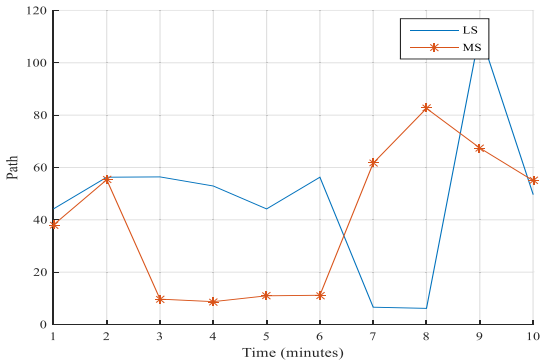

FIGURE 4. First Simulation of Paths to a Point of Failure.

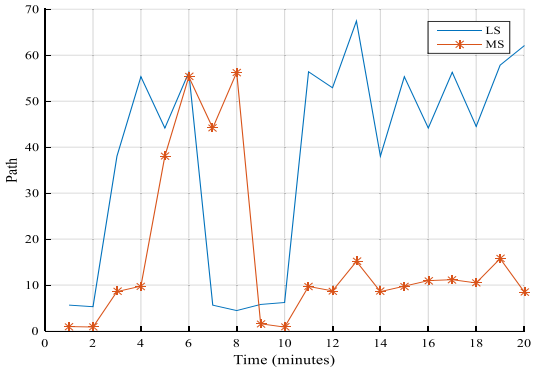

FIGURE 5. Second Simulation Paths to a Point of Failure.

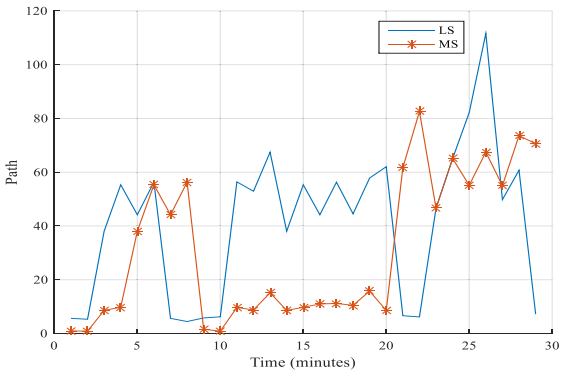

FIGURE 6. Third Simulation of Paths to a Point of Failure.

This suggested that least likely path to detecting point of failures are with security breaches too. That is network can be breached from least likely point of failure.

The second simulation indicates some different behavior compare to the first. The path lengths for the 20 minutes' simulation for which sequence of packets is transmitted from the local network, via the Intent-based controller to the outside network. Both the least and most likely point of failure appears to raise (see Figure 5). In the middle of the simulation scenarios, both the least and most likely point of failure drops rapidly. The most likely point of failure does not raise for the remaining part of the experiment. This means that most of the point of failure is seen in few relative paths, whereas the least likely point of failure keeps appearing. This suggested that the more the transmission session stayed longer the lower the changes of detecting the most likelihood point of failures. Further conclusion drawn from this result, about the network that are breached is that the risk of least likely point of failure in a network is higher than the most likely point of failure.

The longest range for the experimental simulation analysis was set to 30 minutes. This has been designed because the subject under investigation involve a system that changes instantaneously in response to certain discrete events. Hence, the simulation is a discrete event simulation. Similar to the previous simulation analysis, the path lengths for the 30 minutes' simulation for which sequence of packets followed by

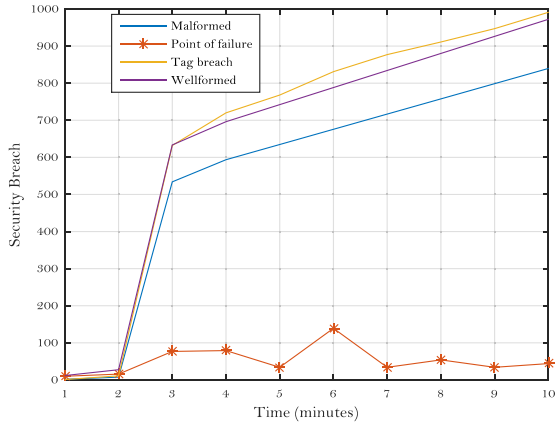

FIGURE 7. Point of Failure in the first simulation.

the local network, via the Intent-based controller has been gathered (see Figure 6).

The analysis of the data drawn shows similar behavior with the previous analysis, except that the current one indicated that both the least and most likely point of failure steadily grow. That means at an extreme length of transmission session, the least and most likely point of failure are the same. This suggests that in a network, least and most likely point of failure emerge from the towards the end of a transmission session.

The decision to transmit messages from both directions of the local networks and the Intent-based controller, was too flooded the transmission session with both well-formed and malformed requests, using different time intervals of 30 minutes for different sets of flood bursts. Malformed, well-formed, point of failure, and tag breach are the controlling variables. Similar to the simulations on determining the paths of point of failures, where 10, 20, and 30 minutes' different simulation ranges were used, this present one is also same.

The analysis of the first round of simulation for the duration of 10 minutes' reveals the highest point of failure to be in the middle of the simulation from where the security is breached (see Figure 7). The malformed packet is lower as compared to the well-formed packets on the cause of the security breach of the network. Throughout the simulation scenario, malformed packets are lower than the rest of the packets. Towards the end of the simulation, the point of failure cause of security breach drops down. It is quite challenging to understand the most dangerous single points of failure for causing breaches in the network security system. This is understanding the point where an entire network system can be taken down. The algorithm runs on the controller and uses the tag of a point at each connection according to the rules set out and then identify the major point of failure where major security breaches ''get in''. This was captured. The burst shows the breach of the network. This suggests that the not only at a malformed packet that security can be breached or the breach might not necessarily be from the point of failure, but even the well-formed packet is susceptible to the cause of network breach.

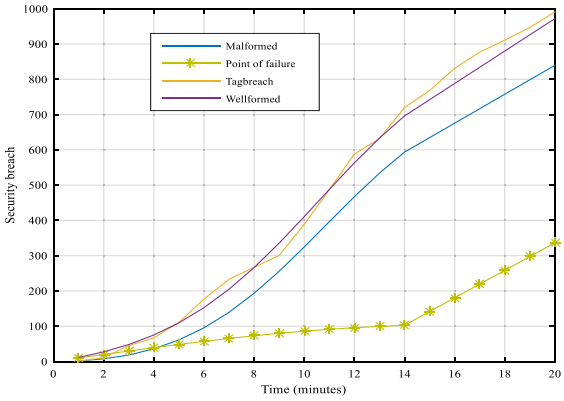

FIGURE 8. Point of Failure in the first simulation.

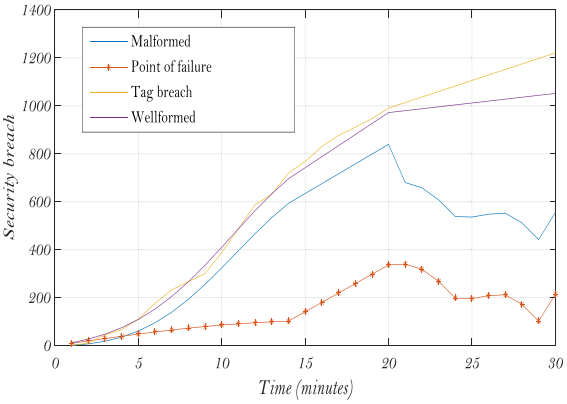

FIGURE 9. Point of Failure in the first simulation.

As the simulation time extends to 20 minutes, the point of failures, numbers increases toward the end of the simulation (see Figure 8). The deviation from well-formed packet to malformed packet is so small according to the results. That's why the transmission gets out of sync and was classified under malformed packet. The packets have been flagged as an unknown opcode error. However, the security breach was seen not to be from dominantly malformed packet. The remaining well-formed steadily increases, indicating that security breach were mostly from an unidentified well-formed packet.

The malformed packet, and point of failure rates in causing security breach has dropped in the final simulation that last for 30 minutes (see Figure 9). The expectation was that corrupt packet in a transmission session would be likely the one that can cause severe security breach. The result of this simulation indicates that in transmission session, corrupt packet might unlikely make it to the destination.

Thre are many resaons to support that. This means that the chances for malformed packet to be use for breaching security is lower as compare to the wellformed packet. The well-formed packet ability to cause security breach is higher than the malformed packets.

VI. DISCUSSION

Network protocols are well-defined and majority of them consider security as the key controlling variable. The nature of the implementation of conventional network security is either software based or "security provision as a service". This is not flexible enough to support unforeseen circumstances that might arise in a network security management. Furthermore, those kind of setting cannot enable flexible monitoring and detecting of anomalies dynamically in a network. The weakness of the conventional network system is also attributed to inability to manage the security on-the fly. Those kind of network do not support changes after initial configurations. That is why SDN comes and provide flexible management of both flow process and security. Later Intent-Based Networking a more formal version of SDN was initiated. It enables creating an on-demand set of rules and policies for intended prescription of what to achieve. That is the desire for any service needed on a network. In order to detect point of failure that causes security breaches, as well as the paths that might lead to security breach in a network system, Intent-Based Networking concept was used.

This current research produces two algorithms, namely: the algorithm that generates the path length toward the point of failure in a communication session, as well as the algorithm that generate the point of failure that leads to the security breaches of a network. In both cases the significance of the research dwells on examining the factors that influence the security breaches of a network. It has been conceptualized that "well-formed, malformed, and point of failure" are the constructs responsible for security breaches over time in a network. Network managers can now be able to use Intent-Based Networking potential to set up some policies necessary in order to automate network security breaches. For this reason, the algorithm proposed are set out to identify how well-formed, malformed, and point of failure are responsible for security breaches of a network

Previous research studies revealed that malformed packet can be used in order to launch an attack on a network [29], [31]. IP addresses and packet header are mostly utilized for this [34], that is why it's crucial to have an efficient evaluation of detecting and preventing network by malformed packet. Even though the results of the previous research study on the effect of malformed packet employ traffic analysis [30], [33], the finding of this study has a new direction.

It was revealed that security breaches evolve from certain paths of the network connections. The path length ranges from the least to the most points of weakness. The finding shows that security breach in most cases are not from the weakest point of failure. There is no directional correlation between path length of and the security breach. The weakest point of failure does not come from any certain path length. Security breaches comes from the paths with least likely point of failure. Recently, it is also proven that compromised network node detection ratio influence a trust computing-based security routing [54]. That is why the path length analysis is crucial for the fact that many protocols of communication session established how normal get transmitted and abnormal packets get discarded. In some protocols abnormal packets are dropped immediately. Whereas others treated them and attempt to correct them before making a decision. Flooding

request from the local network to the outside network through the Intent-based controller allows this study to conceptualized those services. That is why the pattern of the length of the paths for both the least and most likely point of failures in the transmission session is detected.

It was established that it is quite challenging to measure the points of failure, particularly, those points that can cause devastating attack to the network security system. However, the duty cycle of each connecting node in a network based on the energy consumption has been revealed to substantially impact network security [55], similarly trust degree of a network node has been identified to influence network security [56]. The findings of this current study has enabled the research to conclude that the security breach of a network system are influenced by the network communication paths. Furthermore, the data in the transmission session, both the well-formed and malformed are analyzed. Those packets going from the local networks to the outside network through Intent-based controller is captured as well as those coming into the local network via Intent-based controller. The analysis reveals that the malformed packet has been always the least in security breach causes. That is not being the causes of the security breach of the network. Although previous research studies have identified that malformed packet are used to wage an attack on a network [30], [31], but this study reveals that the well-formed packets are the most silent items used in penetrating the security of a network. This has circumvented the previous research finding on malformed packet severity is a network attack. In terms of security breach, it is clear that malformed packets do not play any significant roles, because majority of security breaches are from well-formed packet.

The contribution of this work is based on the fact a network attack on target system can be exposed easily, but the point where the attack comes and the depth from the origin to the targets over the time is challenging. Research studies have identified some models for detecting potential flaws that might be discovered [37]. It is quite normal to start from understanding flaws in a network security system and what triggers those problems. This means that there is a point of failure or a reason that an attacker is used to be able to breach the security system. That is why this study evaluated those concerns. Crucial to this is the detection rate of attempts to breach the security system. This research has been able to set out security breaches influence to the point of failure over time. Intent-based controller evaluates the attacked and the point of failure where the network security is breached.

VII. CONCLUSION

This paper presents a study on the factors influencing network security breaches using Intent-Based Networking. Intent-Based Networking. The negative impact of breaching the network security system is either to destroy the system or to have some data leak. That is why effective monitoring and prevention system is required in any security system. This study has conceptualized that the security breach of a network system is influenced by the network communication path (path length) and the content of the data to be transmitted (well-formed and malformed). As a result, Euler's theorem is utilized in order to propose algorithms for a proving the proposed concept. The algorithms were intended to establish the path length over time for the causes of network security breaches and the point of failure where network security is breached. The path length is for connections within the local to the outside network and the point of failures in the downstream message. Whereas, the point of failure is the point through which attacks take place. The finding of the study reveals the key attributes of the path length ranging from the least likely weak path to the point of failure in the network to the most likely weakest point of failure. Least likely paths of point of failure has been discovered to be the paths that are likely to be used for breaching the network security system. This study has also been able to reveal that malformed packets are not the most causes of the security breach of the network, even though previous findings report that otherwise. That is why it is quite challenging to understand the points of failure that is causing breaches in the network security system. Well-formed packet originating from the least likely weak point of failure manifested network security breach than malformed packet. In another words security breaches are mostly coming from well-formed packet. This study has contributed to uncovering that network security breaches are influence by the paths with least likely point of failure from well-formed packet.

REFERENCES

[1] J. C. Westland, "The information content of Sarbanes-Oxley in predicting security breaches," *Comput. Secur.*, vol. 90, Mar. 2020, Art. no. 101687.
[2] J. Jang-Jaccard and S. Nepal, "A survey of emerging threats in cybersecurity," *J. Comput. Syst. Sci.*, vol. 80, no. 5, pp. 973–993, Aug. 2014.
[3] H. E. Kim, H. S. Son, B. G. Kim, J. Cho, S. M. Shin, and H. G. Kang, "Input-domain software testing for failure probability estimation of safety-critical applications in consideration of past input sequence," *IEEE Access*, vol. 6, pp. 8440–8451, 2018.
[4] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, "Deep recurrent neural network for IoT intrusion detection system," *Simul. Model. Pract. Theory*, vol. 101, May 2019, Art. no. 102031.
[5] H. Hu, H. Zhang, Y. Liu, and Y. Wang, "Quantitative method for network security situation based on attack prediction," *Secur. Commun. Netw*, vol. 19, Jul. 2017, Art. no. 3407642.
[6] A. Warzynski and G. Kolaczek, "Intrusion detection systems vulnerability on adversarial examples," in *Proc. Innov. Intell. Syst. Appl. (INISTA)*, Jul. 2018, pp. 1–4.
[7] M. Kiran, E. Pouyoul, A. Mercian, B. Tierney, C. Guok, and I. Monga, "Enabling intent to configure scientific networks for high performance demands," *Future Gener. Comput. Syst.*, vol. 79, pp. 205–214, Feb. 2018.
[8] N. Paladi and C. Gehrmann, "SDN access control for the masses," *Comput. Secur.*, vol. 80, pp. 155–172, Jan. 2019.
[9] D. Huang, A. Chowdhary, and S. Pisharody, *Software-Defined Networking and Security: From Theory to Practice*, 1st ed. Boca Raton, FL, USA: CRC Press, 2018.
[10] A. Malik, B. Aziz, A. Al-Haj, and M. Adda, "Software-defined networks: A walkthrough guide from occurrence to data plane fault tolerance," *Peer J.*, vol. 7, Apr. 2019, Art. no. e27624v1.
[11] Y. Tsuzaki and Y. Okabe, "Reactive configuration updating for intent-based networking," in *Proc IEEE-ICOIN*, Da Nang, Vietnam, Jan. 2017, pp. 97–102.
[12] T. Szyrkowiec, "Automatic intent-based secure service creation through a multilayer SDN network orchestration," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 10, no. 4, pp. 289–297, Apr. 2018.
[13] A. Campanella, "Intent based network operations," in *Proc. Opt. Fiber Commun. Conf. (OFC)*, San Diego, CA, USA, 2019, pp. 1–3.

[14] E. Drago, "The effect of technology on face-to-face communication," *Elon J. Undergraduate Res. Commun.*, vol. 6, no. 1, pp. 13–19, 2015.
[15] D. Bourgeois and D. T. Bourgeois, *Information Systems for Business and Beyond* (Networking and Communication). Minneapolis, MI, USA: Saylor Foundation, 2014.
[16] M. van Steen and A. S. Tanenbaum, "A brief introduction to distributed systems," *Computing*, vol. 98, no. 10, pp. 967–1009, Oct. 2016.
[17] E. Tranos and C. Stich, "Individual Internet usage and the availability of online content of local interest: A multilevel approach," *Comput., Environ. Urban Syst.*, vol. 79, Jan. 2020, Art. no. 101371.
[18] J. Naughton, "The evolution of the Internet: From military experiment to general purpose technology," *J. Cyber Policy*, vol. 1, no. 1, pp. 5–28, Jan. 2016.
[19] S. Horing, J. Menard, and R. Staehler, "Stored program controlled network," *Bell Syst. Tech. J.*, vol. 61, no. 7, pp. 1759–1778, Sep. 1982.
[20] D. L. Tennenhouse, J. M. Smith, W. D. Sincoskie, D. J. Wetherall, and G. J. Minden, "A survey of active network research," *IEEE Commun. Mag.*, vol. 35, no. 1, pp. 80–86, Jan. 1997.
[21] K. K. Kapoor, K. Tamilmani, N. P. Rana, P. Patil, Y. K. Dwivedi, and S. Nerur, "Advances in social media research: Past, present and future," *Inf. Syst. Frontiers*, vol. 20, no. 3, pp. 531–558, Jun. 2018.
[22] D. Kreutz, F. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.
[23] T. Benson, A. Akella, and D. A. Maltz, "Unraveling the complexity of network management," in *Proc NSDI*, 2009, pp. 335–348.
[24] D. Robinson, *Content Delivery Networks: Fundamentals, Design, and Evolution*. Hoboken, NJ, USA: Wiley, 2017.
[25] F. Ullah, M. Edwards, R. Ramdhany, R. Chitchyan, M. A. Babar, and A. Rashid, "Data exfiltration: A review of external attack vectors and countermeasures," *J. Netw. Comput. Appl.*, vol. 101, pp. 18–54, Jan. 2018.
[26] D. Forte, S. Bhunia, R. Karri, J. Plusquellic, and M. Tehranipoor, "IEEE international symposium on hardware oriented security and trust (HOST): Past, present, and future," in *Proc. IEEE Int. Test Conf. (ITC)*, Nov. 2019, pp. 1–4.
[27] S. Zhang, L. Zhou, M. Wu, Z. Tang, N. Ruan, and H. Zhu, "Automatic detection of SIP-aware attacks on VoLTE device," in *Proc. IEEE 84th Veh. Technol. Conf. (VTC-Fall)*, Montreal, QC, Canada, Sep. 2016, pp. 1–5.
[28] C. P. Ravikumar, S. K. Swamy, and B. V. Uma, "A hierarchical approach to self-test, fault-tolerance and routing security in a network-on-chip," in *Proc. IEEE Int. Test Conf. India (ITC India)*, Bangalore, India, Jul. 2019, pp. 1–6.
[29] M. Z. Rafique and M. Abulaish, "XMiner: Nip the zero day exploits in the bud," in *Proc. IEEE 10th Int. Symp. Netw. Comput. Appl.*, Cambridge, MA, USA, Aug. 2011, pp. 99–106.
[30] B. Peng, Q. Wang, X. Li, J. Cai, J. Fei, and W. Chen, "Research on abnormal detection technology of real-time interaction process in new energy network," in *Proc. Int. Conf. Internet Things (iThings), IEEE Green Comput. Commun. (GreenCom), IEEE Cyber, Phys. Social Comput. (CPSCom), IEEE Smart Data (SmartData)*, Jul. 2019, pp. 433–440.
[31] Z. Tsiatsikas, G. Kambourakis, D. Geneiatakis, and H. Wang, "The devil is in the detail: SDP-driven malformed message attacks and mitigation in SIP ecosystems," *IEEE Access*, vol. 7, pp. 2401–2417, 2019.
[32] M. N. Rajkumar, "A survey on latest DoS attacks: Classification and defense mechanisms," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 1, no. 8, pp. 1847–1860, 2013.
[33] S. Sultana, S. Nasrin, F. K. Lipi, M. A. Hossain, Z. Sultana, and F. Jannat, "Detecting and Preventing IP Spoofing and Local Area Network Denial (LAND) Attack for Cloud Computing with the Modification of Hop Count Filtering (HCF) Mechanism," in *Proc IC ME*, 2019, pp. 1–6.
[34] J. Kaur Chahal, A. Bhandari, and S. Behal, "Distributed denial of service attacks: A threat or challenge," *New Rev. Inf. Netw.*, vol. 24, no. 1, pp. 31–103, Jan. 2019.
[35] G. Stapleton, L. Zhang, J. Howse, and P. Rodgers, "Drawing euler diagrams with circles: The theory of piercings," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 7, pp. 1020–1032, Jul. 2011.
[36] J.-S. Wu, W.-S. Zheng, J.-H. Lai, and C. Y. Suen, "Euler clustering on large-scale dataset," *IEEE Trans. Big Data*, vol. 4, no. 4, pp. 502–515, Dec. 2018.
[37] O. H. Alhazmi and Y. K. Malaiya, "Modeling the vulnerability discovery process," in *Proc. 16th IEEE Int. Symp. Softw. Rel. Eng. (ISSRE)*, Los Alamitos, CA, USA, 2005, pp. 1–10.
[38] Y. Movahedi, M. Cukier, A. Andongabo, and I. Gashi, "Cluster-based vulnerability assessment of operating systems and Web browsers," *Computing*, vol. 101, no. 2, pp. 139–160, Feb. 2019.
[39] S. Frei, M. May, U. Fiedler, and B. Plattner, "Large-scale vulnerability analysis," in *Proc. SIGCOMM Workshop Large-Scale Attack Defense (LSAD)*, Barcelona, Spain, 2006, pp. 131–138.
[40] K. Kritikos, K. Magoutis, M. Papoutsakis, and S. Ioannidis, "A survey on vulnerability assessment tools and databases for cloud-based Web applications," *Array*, vols. 3–4, Sep. 2019, Art. no. 100011.
[41] A. Anand, S. Das, D. Aggrawal, and Y. Klochkov, "Vulnerability discovery modelling for software with multi-versions," in *Proc Adv. Rel. Syst. Eng.* Cham, Switzerland: Springer, 2017, pp. 255–265.
[42] P. K. Kapur, V. S. S. Yadavali, and A. K. Shrivastava, "A comparative study of vulnerability discovery modeling and software reliability growth modeling," in *Proc. Int. Conf. Futuristic Trends Comput. Anal. Knowl. Manage. (ABLAZE)*, Feb. 2015, pp. 246–251.
[43] A. Anand and N. Bhatt, "Vulnerability discovery modeling and weighted criteria based ranking," *J. Indian Soc. Probab. Statist.*, vol. 17, no. 1, pp. 1–10, 2016.
[44] G. V. Marconato, V. Nicomette, and M. Kaâniche, "Security-related vulnerability life cycle analysis," in *Proc IEEE-CRiSIS*, 2012, pp. 1–8.
[45] A. Fischer, J. F. Botero, M. T. Beck, H. de Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 1888–1906, 4th Quart., 2013.
[46] C. Gharat and S. Krishnan, "Effects of Duplicate Packet Transmission in Timer based Co-ordination Opportunistic Routing Scheme," in *Proc. IEEE-ICSSIT*, Nov. 2019, pp. 401–405.
[47] L. Cheng and F. C. M. Lau, "Revisiting TCP congestion control in a virtual cluster environment," *IEEE/ACM Trans. Netw.*, vol. 24, no. 4, pp. 2154–2167, Aug. 2016.
[48] D. Lin and H. T. Kung, "TCP fast recovery strategies: Analysis and improvements," in *Proc. IEEE Conf. Comput. Commun., 17th Annu. Joint Conf., IEEE Comput. Commun. Societies, 21st Century (INFOCOM)*, Mar. 1998, pp. 263–271.
[49] A. I. Abubakar, E. E. E. Mohamed, and A. M. Zeki, "The dynamics of data packet in transmission session," *IEEE Access*, vol. 5, pp. 4329–4339, 2017.
[50] A. M. Al-Jubari, M. Othman, B. Mohd Ali, and N. A. W. Abdul Hamid, "TCP performance in multi-hop wireless ad hoc networks: Challenges and solution," *EURASIP J. Wireless Commun. Netw.*, vol. 2011, no. 1, p. 98, Dec. 2011.
[51] J. Liao, J. Wang, T. Li, X. Zhu, and P. Zhang, "Sender-based multipath out-of-order scheduling for high-definition videophone in multi-homed devices," *IEEE Trans. Consum. Electron.*, vol. 56, no. 3, pp. 1466–1472, Aug. 2010.
[52] F. Yang, Q. Wang, and P. D. Amer, "Out-of-order transmission for in-order arrival scheduling for multipath TCP," in *Proc. 28th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, May 2014, pp. 749–752.
[53] P. Dong, J. Xie, W. Tang, N. Xiong, H. Zhong, and A. V. Vasilakos, "Performance evaluation of multipath TCP scheduling algorithms," *IEEE Access*, vol. 7, pp. 29818–29825, 2019.
[54] A. Liu, N. N. Xiong, and F. Liu, "A trust computing-based security routing scheme for cyber physical systems," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 6, pp. 1–27, 2019.
[55] X. Liu, A. Liu, T. Wang, K. Ota, M. Dong, Y. Liu, and Z. Cai, "Adaptive data and verified message disjoint security routing for gathering big data in energy harvesting networks," *J. Parallel Distrib. Comput.*, vol. 135, pp. 140–155, Jan. 2020.
[56] B. Jiang, G. Huang, T. Wang, J. Gui, and X. Zhu, "Trust based energy efficient data collection with unmanned aerial vehicle in edge network," *Trans. Emerg. Telecommun. Technol.*, vol. e3942, pp. 1–32, Mar. 2020.

MOHAMMED ABDULAZIZ AL NAEEM received the B.Sc. degree in CIS from the College of Management Science and Planning, King Faisal University, in 2005, and the M.Sc. degree in networks and communications (spec. in information security) and the Ph.D. degrees in networks and communications (spec. in wireless networks) from Monash University, Australia, in 2009 and 2015, respectively. He is currently the Chairman for the Department of Computer Networks and Communications, King Faisal University. He is interested in wireless networks, network security, machine learning, artificial intelligence, and pattern recognition.

ADAMU ABUBAKAR received the B.Sc. degree (Hons.) in geography, P.G.D., M.Sc., and Ph.D. degrees in computer science from Bayero University, Kano, Nigeria, and IIUM, respectively. He is currently an Assistant Professor with the Department of Computer Science, International Islamic University Malaysia (IIUM). Prior to his completion of M.Sc. degree programme from 2005 to 2007, he has worked in various area of information technology. During his Ph.D., he worked on many research projects. He received many awards research. He had published many papers in conferences, journals, and book chapters during his Ph.D., Postdoctoral and at the current position of an Assistant Professor. He is a professional member of Association for Computing Machinery (ACM) and Institute of Electrical and Electronics Engineers (IEEE) since 2014. He is working in the areas of computer network, 3D mobile map, information retrieval technologies, digital watermarking, steganography, artificial neural networks, and wireless sensor networks.

M. M. HAFIZUR RAHMAN received the B.Sc. degree in EEE from KUET, Khulna, Bangladesh, in 1996, and the M.Sc. and Ph.D. degrees in information science from JAIST, in 2003 and 2006, respectively. He is currently working as an Assistant Professor with the Department of CN, CCSIT, KFU, Saudi Arabia. Prior to join in the KFU, he was an Assistant Professor with Xiamen University, Malaysia and IIUM, Malaysia, and Associate Professor with CSE, KUET, Khulna, Bangladesh. He was also a Visiting Researcher with the School of Information Science at JAIST and a JSPS Postdoctoral Research Fellow with the Graduate School of Information Science (GSIS), Tohoku University, Japan and Center for Information Science, JAIST, Japan, in 2008 and 2009, and from 2010 to 2011, respectively. His current research interests include hierarchical interconnection networks and optical switching networks, and software defined networks.