Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 4, No. 3, August 2019, Pp. 259-268

Implementing flash event discrimination in IP traceback using shark smell optimisation algorithm

Omoniyi Wale Salami*[1], Imeh Jarlath Umoh[2], Emmanuel Adewale Adedokun[3], Muhammed Bashir Muazu[4]
Ahmadu Bello University, Nigeria[1,2,3,4]

Article Info

Keywords:
Network, Forensic, Traceback, Optimization Algorithm, Flash Event

Article history:
Receive 28 December 2018
Revise 27 February 2019
Accepted 21 June 2019
Published 30 July 2019

Cite:
Salami, O., Umoh, I., Adedokun, E., & Muazu, M. (2019). Implementing Flash Event Discrimination in IP Traceback using Shark Smell Optimisation Algorithm. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 4(3). doi:http://dx.doi.org/10.22219/kinetik.v4i3.740

* Corresponding author.
Omoniyi Wale Salami
E-mail address:
Salamiow@gmail.com

Abstract

Denial of service attack and its variants are the largest ravaging network problems. They are used to cause damage to network by disrupting its services in order to harm a business or organization. Flash event is a network phenomenon that causes surge in normal network flow due to sudden increase in number of network users, to curtail the menace of the Denial of service attack it is pertinent to expose the perpetrator and take appropriate action against it. Internet protocol traceback is a network forensic tool that is used to identify source of an Internet protocol packet. Most of presently available Internet protocol traceback tools that are based on bio-inspired algorithm employ flow-based search method for tracing source of a Denial of service attack without facility to differentiate flash event from the attack. Surge in network due to flash event can mislead such a traceback tool that uses flow-based search. This work presents a solution that uses hop-by-hop search with an incorporated discrimination policy implemented by shark smell optimization algorithm to differentiate the attack traffic from other traffics. It was tested on performance and convergence against an existing bio-inspired traceback tool that uses flow-base method and yielded outstanding results in all the tests.

1. Introduction

Network attacks such as unauthorized use of restricted online assets without permission, stealing or gaining unauthorized access into a system, exposing private resources, or malicious disabling or altering or destroying services of a system on the network, are cybercrimes [1]. Denial of service (DoS) attack and its variants are the largest ravaging network problems. It is the most powerful damaging network attack used to harm a business or organization [2]. In recognition of the serious setbacks that cybercrimes are causing to humanity different countries of the world, including Nigeria, have enacted laws and policies to fight the scourge of cyber-attacks. Examples are the United States Stop Online Piracy Act and Protect IP Act (SOPA/PIPA) [3]. The UK Data Protection Act [4], and the Nigerian cybercrime act 2015 [5]. Resolving DoS attacks requires identifying its perpetrators and engaging legal battle against the perpetrator to serve as deterrent and to be able to compensate the victim. Successful legal battle can only be achieved based on proven infallible facts used to establish criminal offence against a perpetrator. Network forensic professionals use Internet Protocol (IP) traceback tools to acquire network data that can be used as facts about an attack and also detect the source of the attack.

1.1 Background

Denial of service (DoS) attack is a type of cybercrime that require IP traceback scheme specifically designed to take into consideration its intricate attributes and discriminate it from normal transactions on the network that generate large data which may be symptomatically comparable to DoS [6]. A normal network traffic scenario known as flash event (FE) is very similar to Distributed DoS (DDoS) attack, which is a variant of DoS. Flash event (or flash crowd [6]), refers to a situation whereby a circumstance arouses interest of a majority of network users toward accessing a particular network resource on a server [7]. A good example of flash event when legitimate traffic overwhelmed the network is the news of Michael Jackson death on June 25, 2009. CNN reported that “Michael Jackson's death sees Twitter, TMZ, news sites struggle to cope with traffic”, [8].Both flash event and DDoS attack generate heavy traffic from different sources to a particular server. Flash event can be mistaken for DoS attack because of some similar features of the two illustrated in [9]. High packets flow traffic that is caused by flash event can be distinguished from a DoS attack by studying some characteristics of the traffic. Flash event characteristics like rate of request from the same source IP address, the timing between request packets arrival, the sizes of request packets and their contents, and relation between packets will be different from that of DoS. Other characteristics that may be examined to differentiate FE and DoS are packets traffic features including delays, throughput, packets sequences and entropy, and their randomness to deduce if they are from the same node or different nodes [10]. There are usually more packets per IP address in the DoS traffic than what may occur in the case of flash event traffic but their patterns are very similar as shown by [9] illustration. Thus, entropy is wider in the case of flash event than in DoS attack. In the case of the DoS attack the attacker has good knowledge of the system it is attacking and deliberately overloads by exploiting known vulnerability. But flash event is just incidentally caused by users' attraction to particular online items.

Some of the major differences between flash event and DoS attack are;

1. In contrast to the purpose of DoS attack, flash event is not perpetrated to inflict bad consequences on the affected host but happens as a result of large visitors being attracted to the item on the host.
2. DoS attacker knows and target the major metrics that can impact on the performance of the network it is attacking. He understands the impact of its action on the network. User causing flash event are unaware of the effect of their activities on the network.
3. Correlation between DoS traffic packets is stronger than those of flash event traffic packets [9]. This is evident because flash event traffic packets generation is purely random and highly unpredictable but DoS attack packets are generated automatically with or without deliberate randomization.

Since flash event by itself is not an unwanted occurrence but just that it may have undesirable effects on the system like causing significant latency on the network that may disrupt the network services [11], identifying the individuals at its sources is unwarranted. The solution to it is better planning and upgrading of the capacity of the system to be able to cope with such situations. Thus, efficient IP traceback scheme should be able to differentiate flash event traffic from DoS attack traffic during the traceback process.

The process by which data packets are traced back to their source using available information on the packets, e.g. source IP address, is called IP traceback. IP traceback technique employs a mechanism for storing routing path information of the packets, e.g. marking schemes [12], so that it can later be used to trace the packet back to its source. IP traceback can use a single packet as used in [13], or many packets as in [3], to acquire adequate information that can used for tracing its source. Challenges usually faced by IP traceback schemes are caused by IP spoofing, concealment of source address in the packet header by attackers to hide their identity [14], and flash event that may be confused with DoS attack because of the similarity of its characteristics to those of Distributed DoS.

1.2 Statement of the Problem

Accurate detection of source of attack is essential to prevent further malicious transmissions from the same source or expose the perpetrator for the purpose of taking other appropriate actions as may be necessary. Many IP traceback schemes for detecting the source of DoS attacks have been reported in literature but they do not have the facility to differentiate flash event from DoS attack. This may cause them to mistakenly identify flash event traffic on the path as attack during traceback process. This research work proposes an IP traceback scheme for acquiring accurate data about an attack and detect genuine source of the attack by avoiding other network traffics, including flash event traffics, that may cause false error in its results. This will ensure acquisition of genuine data about an attack for detecting the source.

1.3 The Relevant Literature on the Research Subject

There are various methods for implementing IP Traceback scheme [15]. Some of the common methods available in the literature are; Packet marking that may be Deterministic Packet Marking (DPM) or Probabilistic Packet Marking (PPM). DPM has large convergence time and PPM requires large number of packets. Link testing is another method used for IP Traceback which could be either input debugging or control flooding. Input debugging challenges include vulnerability to topology changes, the attack should be active, it also requires the coordination from the network administrators which can make it slow or may not even complete traceback if the coordination failed or the topology changes [15]. Control flooding can cause network overhead problem. ICMP Traceback sends ICMP message together with packet that is sampled. It is a proactive measure but creates additional overhead on the network. IP Logging traceback method may require large memory space for storing packets digest, signature, and fields of IP header packets of flows that passed through it. Some IP traceback schemes are developed based on these principles. Among them are Efficient Traceback Technique (ETT) for detecting DDoS attack in cloud-assisted healthcare environment [16], and the scheme using completion condition to determine the minimum required packet for IP traceback [17], they both used packet marking method.

In recent time IP traceback schemes are developed based on nature inspired algorithms. Nature inspired algorithms are developed for complex computing where exact results may be hard to achieve or not determinable due to non-availability of enough input parameters for conventional computation [18]. Among nature inspired algorithm-based IP traceback scheme are Improved Ant Colony Optimization (ACO) algorithm-based system for solving IP traceback by [19]. It used ant colony optimization algorithm for flow-based traceback. Shark Smell Optimization Algorithm (SSOA) is another nature inspired optimization algorithm that was developed to be used for obtaining optimal. The fundamental concepts of this algorithm were based on the superior ability of shark to find its prey in a large search space in, the sea, within a limited period of time using its strong smell sense. The algorithm was evaluated by comparing its performance against thirty-two (32) other popular optimization techniques often employ to solve engineering problems and produced better results [20]. The algorithm comes in hand for a quick scrutiny of large data with close similarity for selecting the ones with the best fit within a limited time constrain. Like other nature inspired algorithms, the SSOA user-define parameters can be used to control its speed and result accuracy.

IP traceback tools use information obtained from the received packets and other relevant information available on the network to trace the source of the packet. Network data are very volatile because most of the important data that may be vital for traceback or use as legal evidence often get overwritten within short time. This necessitated the need for IP traceback tool that can collect reliable network data and detect the attack path accurately within reasonable time.

DoS attack IP traceback finds the edges of attack packets route and use it reconstructs the same path back to the attacker. One of the metrics often used for the path reconstruction include the flow traffic whereby edges with higher packets flow are selected as probable attack path [21]. There are mechanisms that keep records of known compromised hosts and query the hosts if any of them sent packets to the victim during the time of the attack [22]. Some mechanisms use NetFlow tools on the router to traceback the DoS attack source [14].

1.4 The Proposed Solution

The search process by the iSSOA-DoSTBK starts when the DoS detector on the victim system detected DoS attack and alert the iSSOA-DoSTBK. The detector will pass the parameters of the attack packet to it. iSSOA-DoSTBK will assign weight to each parameter according to their importance to the traceback. The weighted parameters are used to build a discernment policy that is incorporated into the SSOA search agent module. The discernment policy is used by the SSOA search agent to differentiate the other confusing traffics from the identified attack traffic. While iSSOA-DoSTBK implements hop-by-hop search traceback process the SSOA search agent scrutinizes each hop and select the most probable hops on the attack path for reconstructing the attack path. Among assumptions on which SSOA was developed are that there is an injured fish in the sea, and the fish is ejecting blood which shark is using to trace it, [20]. This make it suitable for real time tracing of an attack source. But it can be used after the attack as long as relevant data has not been lost on the network. By analogy, the shark is the victim using iSSOA-DoSTBK to trace source of an attack. The injured fish is the DoS attacker and the sea is the network. The blood is the attack packets and the attacker is assumed to be sending attack packets when the IP traceback is going on, in the case of real-time traceback. Shark execute local search at each position along the path by moving round in cycles to exploit all the points around each position and detect the point with highest smell at a position to be able to determine the direction to follow to the next position [20]. This is similar to hop-by-hop search of IP traceback. The closeness of the SSOA search process to the tracing of the DoS attack path make it considered suitable for improving IP traceback for detecting source of an attack.

1.5 The Novelty of Research

Many IP traceback schemes for detecting the source of DoS attacks have been reported in literature but they do not have the specific facility for differentiating FE flow from DoS attack flow. This may cause them to mistakenly identify FE traffic paths as attack path during traceback process. This research work proposes an IP traceback scheme that is incorporated with discernment policy for acquiring accurate data about an attack and detect genuine source of the attack by avoiding other network traffics, including FE traffics that may cause false error in its results. This will ensure acquisition of highly accurate data about the attack for detecting the original source in order to be able to take appropriate actions against the real perpetrator of the attack. The novelty of this work includes:

1. Development of SSOA-DoSTBK IP traceback scheme that can avoid flash event and other legitimate flows that may be symptomatically similar to an attack traffic.
2. Incorporation of Discrimination policy into IP traceback scheme for focused tracing to improved performance and obtain more accurate results.
3. This work mitigates the challenge of distinguishing the sudden surge in normal network traffic from the DoS attack traffic. It avoids misleading IP traceback scheme during traceback process if the surge is encountered along the attack path that is being traced.

2. Research Method

The prototype of the iSSOA-DoSTBK proposed in this research was simulated on a laptop PC; x64-based processor, Intel Core i3-3110M CPU 2.4 GHz, 4.0 GB RAM, running Ubuntu 16.04 LTS operating system. Network Simulator version 2 (NS2), ns-allinone-2.35, was used for the simulation. C++ and OTCL were used for the necessary programming. GNUPlot were used for the graph plot. MS Excel was used to generate the bar charts.

2.1 Research design

The simulated network consists of 25 autonomous systems with 8 nodes each. The network topology was generated using a random topology simulator. Wired Network was used for the path tracing simulation to enable physical

Cite: Salami, O., Umoh, I., Adedokun, E., & Muazu, M. (2019). Implementing Flash Event Discrimination in IP Traceback using Shark Smell Optimisation Algorithm. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 4(3). doi:http://dx.doi.org/10.22219/kinetik.v4i3.740

visibility of the path tracing process. The random topology simulator randomly placed the 200 nodes at the coordinate points in the 20 × 20 meters network area.

$$p(i,\ j) = \eta.e^{\left(\frac{d(i,j)}{L\gamma}\right)} \tag{1}$$

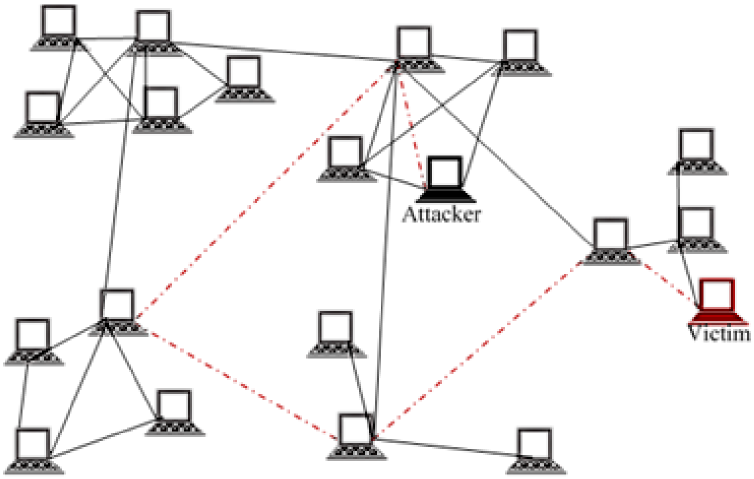


Figure 1. A Randomly Connected Network

The Waxman's connectedness probability Equation 1 [19], was used to establish connection links attributes which include transmission rates and bandwidth between nodes based on their distance apart. The nodes distances apart, d(i,j) was set to 1 to connect each node to its one hop distance neighboring node. L is the longest possible distance between any two nodes. Different values of η were selected from [0.3, 0.6, 0.9, 1.2, 1.5] for different generations of 100 simulations each in that order. The value of γ = 0.1 was used for all values of η selected. Ping and messaging agents were set up using OTCL classes. Messaging was used for generating normal network transaction between other nodes. Ping agent was used to generate the ping of death attack used as the DoS attack by the randomly selected attacker on the randomly selected victim. Other communication agents including TCP, FTP, and CBR were establish for different network transactions on the network. Figure 1 is a simple network with randomly connected nodes showing an attacker and a victim, and the attack path (red dotted line) through which the attack packets were transmitted.

```
GET attack packets attributes from an attack detector
SET rules for the discrimination policy using the parameters of the attack packets obtained
WHILE Not HALT instruction encountered REPEAT the following steps:
FIND edges that carry ingress traffic to this node
APPLY discrimination policy rules to the packets on the identified ingress edges
SELECT the most probable ingress edge as part of the attack path
ADD the edge to the attack path edges list
MOVE to the node on the other end of the just selected edge
IF this node is the host or the closest router to the host of the attack packets HALT
END IF clause
END WHILE loop
RETURN attack path and the attack packets host.
```

2.2 Research Procedure

2.3 Data Acquisition and Test Process

Routers log vital information that are useful for attack path reconstruction [23]. Based on the Cisco Systems NetFlow Services Export V9 the Internet Engineering Task Force (IETF) in October 2004 published Request for Comments (RFC) 3954 [24], to establish a standard for NetFlow by different vendors and remove interoperability problems. Logging of some of the information may require to be enabled if they are not enabled by default, [23]. This is why cooperation of Autonomous Systems (AS) owners is vital for a successful IP traceback for source detection. Appropriate authority can make regulations that will make AS owners to cooperate and assist in crime investigations. Cisco routers can keep some data packet traffic record for as much as 30 minutes if IP flow-capture is enabled [25].

2.4.1 The Discrimination Policy

The discrimination policy parameters were stored in an array, x_0 Equation 2.

$$x_0 = [4,5,*,1,*,*,4,*,0,1,2] \tag{2}$$

The elements in x_0 in Equation 2 are arranged to corresponds to the position of the parameter assigned the weight in Table 1. The parameter with no weight assigned has value * in the array.

Table 1. Weight Values Assigned to Discrimination Parameters

Parameter	NS2 Parameter	Value
ICMP	Pkt Type	4
IP-ID	Pkt Id	5
MAC-addresses	Not used	
Packet-Length	Pkt_sz	1
TTL	Not used	
VLAN-ID	Not used	
Dst IP address	Dst Node Id	4
Dst Port Msk AS	Not used	
Pkts	Packets Count	No Value (NV)
Active	Host log Time	1
NextHop	Next hop node	2

When reconstructing the attack path, each neighboring node is examined against the flow parameters stored in array, x_0, and matching parameters were assigned values according to it corresponding weights. The value obtained for the parameters of packets flow records on each node examined was stored in $x_{e,k}$,

$$\chi_e = \sum_{k=1}^{K} \left(x_{e,k} \cap x_0 \right) \tag{3}$$

The sum of matching parameters of all packets on the edge e with the attack packet parameters in x_0 is determined by Equation 3.

$$\chi_j = \left[\chi_e, \ldots, \chi_E \right], \text{ } 1 \le e \le E, \text{ E is the total edges on the node} \tag{4}$$

Each χ_e for all edges on node j on the attack path was stored as array in Equation 4. It is considered that there may be two or more χ_e values in Equation 4 that may be accidentally the same. This is because different sets of numbers can sum up to the same value, e.g. the additions (5 + 5), (3 + 7), (6 + 4) all give 10. Further steps are considered to avoid such situation that may confuse the IP traceback in determining a probable edge that is part of attack path. The average value of the matching features for edge e is calculated.

$$\beta_e = \frac{\chi_e}{E} \tag{5}$$

Equation 5 determines the most common set of values. It will be small if smaller values are more than bigger values, or big if otherwise. But it will still be the same for χ_e values that are equal if those χ_e values have the same number of terms.

Since the occurrence of traffic flows in a packet switch network is random, it is assumed that the path that different attack packets will follow may not be the same. Thus those χ_e with same values may not have exactly the same set of addends. Variance can show the difference in the terms of different χ_e. It will also magnify differences that are very close.

Cite: Salami, O., Umoh, I., Adedokun, E., & Muazu, M. (2019). Implementing Flash Event Discrimination in IP Traceback using Shark Smell Optimisation Algorithm. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 4(3). doi:http://dx.doi.org/10.22219/kinetik.v4i3.740

$$\alpha_e = \frac{\sum_1^M \left(x_{e,m} - \mu\right)^2}{K} \tag{6}$$

The variance of the distribution of the sums of matching parameters on edges e was derived from Equation 6.

$$p_e = \frac{\chi_e}{\sum_{e=1}^{E}(\chi_e)} \tag{7}$$

The probability of edge e been on the attack path, p_e was calculated using Equation 7.

$$f\left(\chi_j\right) = max\left[\left(\frac{\beta_e}{\alpha_e}\right)\cdot\left(\frac{\chi_{j,e}}{\alpha_e}\right)\cdot e^{\left(\left(\frac{\chi_{j,e}}{\alpha_e}\right)^{p_e}\right)}\right]_{e=1}^{E} \qquad \chi_{j,e} \geq 0 \tag{8}$$

Using Equations 4, to Equations 7 the fitness of the edge been on attack path, $f\left(\chi_j\right)$, is determined from Equation 8. In Equation 8, $\left(\frac{\beta_e}{\alpha_e}\right)$ further differentiate edges with same values of χ_e based on the distributions of their matching parameters. As the shark advances from the victim towards the attacker, at every node (*j*) on the attack path each edge e carrying packets to (*j*) was examined using Equation 3 and Equation 8.

3. Results and Discussion

Performance was measured in terms of correctness based on the number of attack packets available on the return path as was used in [19]. The results obtained for SSOA-DoSTBK were compared with similar results obtained from ACS-IPTBK for the same tests under the same conditions. Performance in attack packets detection, and convergence time were used as the metrics for comparing the developed scheme with the benchmark. These values form the data that were used for calculating *percentage improvement* and *correctness* used for plotting the results.

3.1 Performance Evaluation

The performance of the developed scheme was calculated and compared to the ACS-IPTK that it was benchmarked against as follow;

To illustrate the computation of the comparison metrics, if the average of the results obtained from SSOA-DoSTBK in a test is S_{AVE} and A_{AVE} was obtained from ACS-IPTBK in the same test, e.g. average of attack packets on paths returned by iSSOA-DoSTBK is S_{AVE} and for ACS-IPTBK is A_{AVE},

$$pecentage\,improvement = \frac{S_{AVE} - A_{AVE}}{A_{AVE}}\% \tag{9}$$

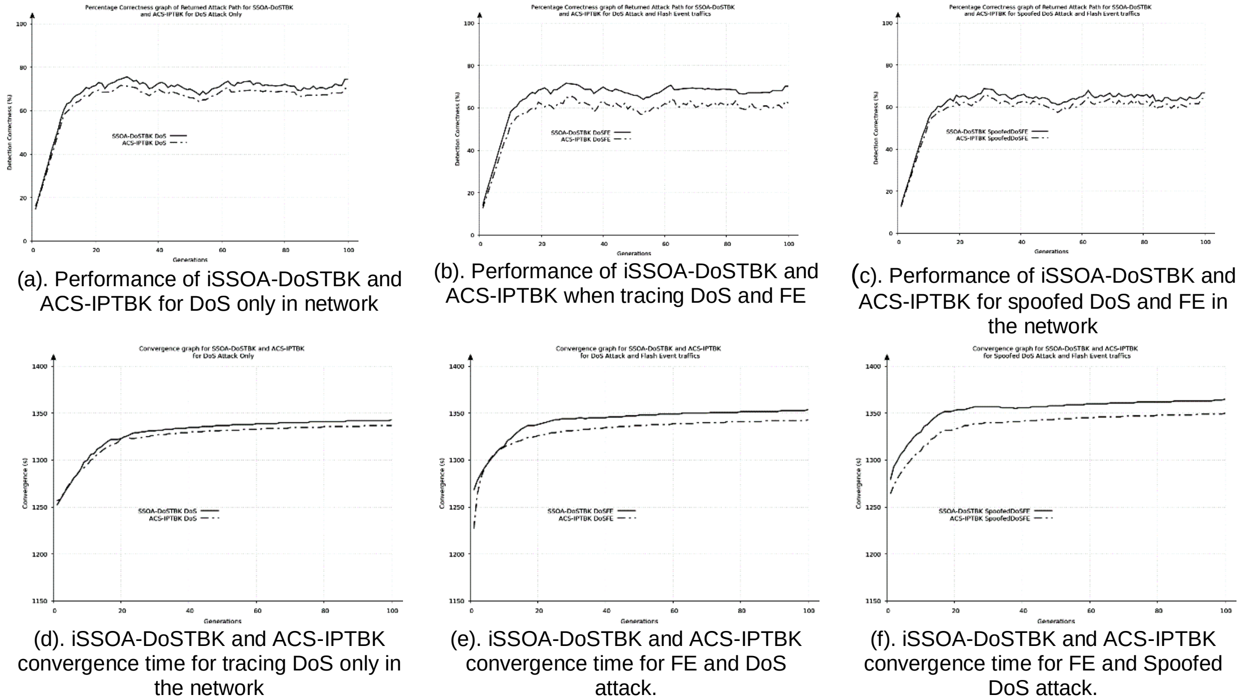
The comparison of the average performance of iSSOA-DoSTBK over ACS-IPTBK is calculated in Equation 9.

$$Correctness = \frac{Average\ attack\ packets\ on\ returned\ path}{Total\ packets\ routed\ on\,the\ path} \times 100\% \tag{10}$$

The performance of the schemes in traceback of the simulated attacks, including spoofed packets, under different conditions were measure in terms of correctness of the attack path returned by examining the number of attack path on the returned path in Equation 10.

3.2 Analysis of the Results

The tests were carried out under three conditions which are; (1) DoS was traced back to source there is no flash event in the network, (2) DoS was traced when there was flash events in the network, (3) DoS attack packets were spoofed and traced when there are flash events in the network. Two dimensional (2D) graphic plots and bar charts of the results are presented for clearer view of the performance of the developed scheme and benchmark as shown in Figure 2. Results of iSSOA-DoSTBK is shown with continuous line and ACS-IPTBK is with dotted line in Figure 2.



(a). Performance of iSSOA-DoSTBK and ACS-IPTBK for DoS only in network

(b). Performance of iSSOA-DoSTBK and ACS-IPTBK when tracing DoS and FE

(c). Performance of iSSOA-DoSTBK and ACS-IPTBK for spoofed DoS and FE in the network

(d). iSSOA-DoSTBK and ACS-IPTBK convergence time for tracing DoS only in the network

(e). iSSOA-DoSTBK and ACS-IPTBK convergence time for FE and DoS attack.

(f). iSSOA-DoSTBK and ACS-IPTBK convergence time for FE and Spoofed DoS attack.

Figure 2. iSSOA-DoSTBK and ACS-IPTBK Returned Path Correctness and Convergence Rate Data Compared

3.2.1 Performance of iSSOA-DoSTBK and ACS-IPTBK

Figure 2(a) shows that iSSOA-DoSTBK recorded 4.76% average performance improvement over the ACS-IPTBK for the test with DoS attack without spoofed packet and no flash events in the network. Also, in Figure 2(b) showing the result of the test when DoS packets were not spoofed but there were flash events in the network when DoS attack was traced, the iSSOA-DoSTBK recorded better performance than the ACS-IPTBK by 11.6% on average result. It can be seen on Figure 2(c) that iSSOA-DoSTBK showed better performance over the ACS-IPTBK by 5.2% for the case when the DoS attack packets wre spoofed and there were flash events in the network during the attack traceback process by the two schemes.

The developed scheme, iSSOA-DoSTBK, returned better results than the benchmark the scheme, ACS-IPTBK, in all the tests conducted for Performance in terms of the correctness of the path returned by each of the schemes based on the number of attack packets present on the attack paths returned. This is due to the fact that iSSOA-DoSTBK actually scrutinize the traffics found on the hops using the discernment policy as the template of the attack features to ensure the actual attack traffic was identified. But ACS-IPTBK estimates the fitness of a traffic based on its size in terms of the amount of its packets. Also, the improvement margin of SSOA-DoSTBK performance over the benchmark, ACS-IPTBK, was highest when DoS was traced in the presence of flash event surges for the False Error Rate tests and Performance tests. This confirm the fact earlier stated that flow-based method cannot effectively mitigate the effects of flash event surge in normal traffic during IP traceback process unless a feature is specifically incorporated to address it.

3.3 Evaluation of Convergence time for iSSOA-DoSTBK and ACS-IPTBK

The efficiency of the developed scheme, iSSOA-DoSTBK, was also tested on how fast it could accomplish the task compared to the scheme used for benchmarking it, ACS-IPTBK. The results obtained under the three conditions in which the tests were carried out are show in Figures 2(d) to Figures 2(f). ACS-IPTBK converged faster than iSSOA-DoSTBK in the three tests conducted for convergence. In Figure 2(d) showing convergence result for the case of DoS only in the network, ACS-IPTBK recorded a faster convergence than iSSOA-DoSTBK by 0.4% on average values. So also, in Figure 2(e) for the presence of flash events during attack traceback, it was faster than the developed scheme by 0.78% on average, and was faster than the iSSOA-DoSTBK by 1.2% on average in Figure 2(e), which was the highest margin recorded for all the convergence tests.

Convergence time is not a measure of accuracy of a computation results but it may indicate the relative computational efforts expended to compute a result based on certain conditions like the number of data involved in the computation. The convergence results test shows that iSSOA-DoSTBK performed more computations than the benchmark. It also shows that the more the attack packets the more the computation required for it. This is expected since it employs a deep search to thoroughly examine the hops. Hop-by-hop search is an exhaustive method that takes time. Also, packet level examination is another time-consuming task. But the use of shark smell optimization algorithm for the scheme paid off by recording relatively close convergence with the benchmark.

4. Conclusion

An Internet Protocol traceback scheme for detecting source of DoS attack based on shark smell optimization algorithm that can mitigate the effects of flash event surge traffic on traceback process was developed. It was tested against ACS-IPTBK developed by [19] using the same test procedures under the same conditions. The result obtained from the proposed scheme when compared with those of ACS-IPTBK showed improvement over the ACS-IPTBK in terms of Performance tests with as much as 11.6%. ACS-IPTBK converge a little faster than the proposed scheme by the maximum recorded difference of 1.2% in convergence time at the worst test condition when there is flash event on the traceback routes and the DoS attack packets were spoofed.

The tests results show that ACS-IPTBK deviated further than SSOA-DoSTBK from the true attack path and that SSOA-DoSTBK is more effective for detecting source of spoofed IP attacks. The time difference between SSOA-DoSTBK and ACS-IPTBK convergence was negligibly small. The results indicated that SSOA-DoSTBK performed better in the detection of true DoS attack path because ACS-IPTBK works on the basis of parallelism whereby different agents examined different segments of the network concurrently to estimate a probable attack path. However, SSOA-DoSTBK performed a sequential search to detect the most probable attack path which resulted in a little longer convergence time. ACS-IPTBK examined more areas most of which are not on attack path but SSOA-DoSTBK narrowed its search to the most relevant area of the network based on defined heuristics and avoided confusing traffic flows. The detailed examination of the traffics enhanced SSOA-DoSTBK performance to return a more correct attack path when attack packets were spoofed and flash event traffics were encountered during traceback process. Further research on better methods for the choice of parameters for the discrimination policy is expected to improve the performance of the scheme. Also, further research on swarming shark agents will make the scheme to be applicable for tracing flooding DoS attacks.

5. Notation

e : network edge counter
E : the total edges on the node
J : network hop counter
k : the counter for the flows on each edge
K : the total traffic flows examined on the node connected by edge e
m : the counter for the matching parameters found on each edge e
M : the total number of matching parameters on edge e
µ : the mean of all matching parameters obtained for the edges on hop j

References

[1] ISO/IEC., "Information technology — Security Techniques — Information Security Management Systems — Overview and Vocabulary.," ISO/IEC, Switzerland, 2009.

[2] D. S. N. Mary and A. T. Begum, "An Algorithm for Moderating DoS Attack in Web Based Application," in *2017 International Conference on Technical Advancements in Computers and Communications (ICTACC)*, Melmaurvathur, India, 2017. https://doi.org/10.1109/ICTACC.2017.17

[3] S. Saurabh and A. S. Sairam, "Increasing Accuracy and Reliability of IP Traceback for DDoS Attack Using Completion Condition," *International Journal of Network Security,* Vol. 18, No. 2, Pp. 224-234, 2016.

[4] Data Protection Act 1998, "Data Protection Act 1998," Data Protection Act 1998 Website, 1998. [Online]. Available: http://www.legislation.gov.uk/ukpga/1998/29/contents. [Accessed 22 September 2018].

[5] Cybercrime Act 2015, "Cybercrimes (Prohibition, Prevention, Etc) ACT, 2015,"Centre for Laws of the Federation Of Nigeria, 2015. [Online]. Available: http://lawnigeria.com/LawsoftheFederation/Cyber-Crime-Act,-2015.html. [Accessed 3 May 2018].

[6] A. Bhandari, A. L. Sangal and K. Kumar, "Characterizing flash events and distributed denial-of-service attacks," *Security and Communication Networks,* Vol. 9, No. 13, Pp. 2222-2239, September 2016. https://doi.org/10.1002/sec.1472

[7] A. Dhingra and M. Sachdeva, "Recent Flash Events: A Study," *in International Conference on Communication, Computing & Systems (ICCCS-2014),* Chennai, India, 2014.

[8] R. Linnie and H. Nick, "Cable News Network," Cable Network News, 26 June 2009. [Online]. Available: http://edition.cnn.com/2009/TECH/06/26/michael.jackson.internet/. [Accessed 26 December 2018].

[9] S. Chawla, M. Sachdeva and S. Behal, "Discrimination of DDoS attacks and Flash Events using Pearson's Product Moment Correlation Method," *International Journal of Computer Science and Information Security,* Vol. 14, No. 10, Pp. 382-389, 2016.

[10] M. A. Mohamed, N. Jamil, A. F. Abidin, M. M. Din, W. W. N. S. Nik and R. A. Mamat, "Entity-Based Parameterization for Distinguishing Distributed Denial of Service from Flash Events," *International Journal of Engineering & Technology,* Vol. 7, No. 2.14, Pp. 5-8, 2018. https://doi.org/10.14419/ijet.v7i2.14.11142

[11] M. S. a. K. Kumar, "A Traffic Cluster Entropy Based Approach to Distinguish DDoS Attacks from Flash Event Using DETER Testbed," *ISRN Communications and Networking,* Pp. 15, 2014.

[12] Y. Bhavani, V. Janaki and R. Sridevi, "IP Traceback Through Modified Probabilistic Packet Marking Algorithm Using Chinese Remainder Theorem," *Ain Shams Engineering Journal,* Vol. 6, No. 2, Pp. 715-722, 2015. https://doi.org/10.1016/j.asej.2014.12.004

[13] Kamaldeep, M. Malik and M. Dutta, "Implementation of Single-Packet Hybrid IP Traceback for IPv4 and IPv6 Networks," *IET Information Security,* Vol. 12, No. 1, Pp. 1-6, 01 February, 2018. https://doi.org/10.1049/iet-ifs.2015.0483

[14] M. H. Bhuyan, D. K. Bhattacharyya and J. K. Kalita, "E-LDAT: A Lightweight System for DDoS Flooding Attack Detection and IP Traceback Using Extended Entropy Metric," *Security and Communication Networks,* Vol. 9, No. 16, Pp. 3251-3270, 2016. https://doi.org/10.1002/sec.1530

[15] V. Murugesan, M. Shalinie and N. Neethimani, "Brief Survey of IP Traceback Methodologies," *Acta Polytechnica Hungarica, Vol*. 11, No. 9, Pp. 197-216, 2014.

[16] R. Latif, H. Abbas, S. Latif and A. Masood, "Distributed Denial of Service Attack Source Detection Using Efficient Traceback Technique (ETT) in Cloud-Assisted Healthcare Environment," *Journal of Medical Systems,* Vol. 40, No. 161, Pp. 1-3, 2016. https://doi.org/10.1007/s10916-016-0515-4

[17] S. Saurabh and A. S. Sairam, "A More Accurate Completion Condition for Attack-Graph Reconstruction In Probabilistic Packet Marking Algorithm," in *2013 National Conference on Communications (NCC)*, New Delhi, India, India, 2013. https://doi.org/10.1109/NCC.2013.6488043

[18] N. Siddique and H. Adeli, "Nature Inspired Computing: An Overview and Some Future Directions," *Cognitive Computation,* Vol. 7, No. 6, Pp. 706-714, 2015. https://doi.org/10.1007/s12559-015-9370-8

[19] P. Wang, H.-T. Lin and T.-S. Wang, "An improved ant Colony System Algorithm For Solving the IP Traceback Problem," *Information Sciences,* Vol. 326, Pp. 172-187, 2016. https://doi.org/10.1016/j.ins.2015.07.006

[20] O. Abedinia, N. Amjady and A. Ghasemi, "A New Metaheuristic Algorithm Based on Shark Smell Optimization. Complexity," *Complexity,* Vol. 21, No. 5, Pp. 97-116, 2016. https://doi.org/10.1002/cplx.21634

[21] M. Hamedi-Hamzehkolaie, R. Sanei, C. Chen, X. Tian and M. K. Nezhad, "Bee-based IP Traceback," in *2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, Xiamen, China, 2014. https://doi.org/10.1109/FSKD.2014.6980970

[22] R. Chen, J.-M. (. Park and R. C. Marchany, "TRACK: A Novel Approach for Defending Against Distributed Denial-of-Service Attacks," *Technical Report TR ECE-06-02.Dept.of Electrical and Computer Engineering,* Vol. 14, No. 10, Pp. 382-389, 2006.

[23] R. D. F. Overview, "Router Data Flow Overview.Flow of Routing Information; Juniper networks," Juniper, 31 August 2017. [Online]. Available: https://www.juniper.net/documentation/en_US/junos/topics/concept/firewall-filter-router-data-flow-overview.html. [Accessed 06 May 2018].

[24] B. Claise, V. Valluri, D. Martin and S. Ganesh, "Request for Comments: 3954.," Network Working Group, Internet Engineering Task Force (IETF), 2004. [Online]. Available: https://www.ietf.org/rfc/rfc3954.txt. [Accessed 06 May 2018].

[25] N. Commands, "Cisco IOS NetFlow Command Reference (Release 12.3 T ed.)," CISCO, 2009.