A novel traceback model for DDoS attacks using modified Floyd-Warshall algorithm

Sherif F. Emara, Sayed Abdelhady and M. Zaki*

Computer and Systems Engineering Department,
Al-Azhar University,
Cairo, Egypt
Email: sherif_emara@hotmail.com
Email: sayed.nouh07@gmail.com
Email: mzaki.azhar@gmail.com
*Corresponding author

Abstract: Distributed denial of service, DDoS, attacks are drastically increasing, therefore, they cause serious threats for information networks. This paper proposes, for the first time, the use of a graph theoretic approach to exploit the entropy techniques for detecting and tracing back DDoS attackers. It presents a novel approach to traceback DDoS attacks using modified Floyd-Warshall algorithm, TDA/MFWA. Such model starts by feeding the network adjacency matrix in which the link weights are changed to comply with the network traffic entropy, accordingly the reachability from node to node can be examined. Then we borrowed the idea of enumerating all the intermediate points between every pair of network nodes from Floyd-Warshall algorithm and modified it to find out the victim node(s). The fact that entropy at network nodes is systematically accounted using a modified Floyd-Warshall algorithm contributes to the smartness and dependability of TDA/MFWA.

Keywords: packet logging; packet marking; entropy variation; modified Floyd-Warshall algorithm; traceback; DoS/DDoS attack; adjacency matrix; reachability; zombie traffic; link weights.

Reference to this paper should be made as follows: Emara, S.F., Abdelhady, S. and Zaki, M. (2023) 'A novel traceback model for DDoS attacks using modified Floyd-Warshall algorithm', *Int. J. Information and Computer Security*, Vol. 20, Nos. 1/2, pp.84–103.

Biographical notes: Sherif F. Emara is a researcher at the Faculty of Engineering, Al-Azhar University at Cairo. He received his BSc from Military Technical College, Cairo, Egypt. He received his MSc and PhD in Electrical Engineering from Al-Azhar University. His fields of interest include artificial intelligence and security of computer networks.

Sayed Abdelhady is a Professor of Computer Networks and Communications, Faculty of Engineering, Al-Azhar University at Cairo. He received his BSc and MSc in Electrical Engineering from Al-Azhar University. He received his PhD in Computer Engineering from Krakov University, Poland. His fields of interest include artificial intelligence and distributed systems.

M. Zaki is a Professor of Software Engineering, Computer and System Engineering Department, Faculty of Engineering, Al-Azhar University at Cairo. He received his BSc and MSc in Electrical Engineering from Cairo

University in 1968 and 1973, respectively. He received his PhD in Computer Engineering from Warsaw Technical University, Poland in 1977. His fields of interest include artificial intelligence, soft computing, and distributed systems.

1 Introduction

Attacks that employ source address spoofing represent a growing threat to the internet infrastructure. Denial of service, DoS, attacks and the more complicated version known as distributed DoS, DDoS, are the most common to take advantage of source address spoofing (Bhuyan et al., 2013). These attacks deny regular Internet services from being accessed by legitimate users either by blocking service completely or by disturbing it such that users become not interested in the service anymore. Meanwhile, detection and prevention of such attacks and misuse is of prime importance and it is a complex task because these attacks can be conducted anywhere and at anytime with varying intensity.

IP traceback is the process of identifying the actual source(s) of attack packets (Balyk et al., 2015). This has the benefit of holding attackers accountable for abusing the Internet. However, DDoS attack spoofing is a major challenge that needs an effective approach for attack detection and traceback rather than the traditional ones like packet marking, packet logging or both to perform traceback which is difficult, expensive and complex solution.

Accordingly, we present a traceback of DDoS Attacks using modified Floyd-Warshall algorithm model, TDA/MFWA, which is based on a logical graph theoretic approach to confirm the fact that a node is victim if and only if it is reachable from the attacker node and it is subjected to tremendous amount of packet overflow. These two circumstances represent the necessary and sufficient conditions for realisation of DoS/DDoS attacks. Our model utilises the graph theory (Ruohonen, 2013) to build up a model for DDoS detection (and consequently traceback) by adapting Floyd-Warshall algorithm (Floyd, 1962) that runs at the application layer (layer 7). The adaptation is carried out by making use of the entropy concept (Yu et al., 2011). From the graph theory, we borrow the mental graphical exercise that starts from weighted graph to adjacency matrix to Floyd-Warshall, F-W algorithm that enumerates the graph nodes in order to compute the shortest path between every pair of graph nodes.

However, we are not interested in finding shortest paths but attacks overflows, therefore we transform the network into a weighted graph in which the edge weight x is substituted by $h = -p \log p$ so that the summation of x at a potential victim yields Shannon entropy (Shannon, 1948). Then a weighted adjacency matrix is applied to a modified Floyd-Warshall algorithm which performs graph nodes enumeration to compute the flow entropy at every graph node and uses this entropy value to decide whether the underlying node is a victim. Starting from the victim node, if any, TDA/MFWA affords a traceback to the real attack source. It is worth noticing that the single packet attacks are out of scope of TDA/MFWA. Nevertheless, if the sequence of one-packet attack comprises a condition that the traffic flow is considerably increased so that it may cause a denial of service, then TDA/MFWA will be capable to detect the DoS attack.

This paper is organised as follows: Section 1 introduces the goal of the work, the approaches used and the paper layout, Section 2 introduces the related work reviewing different IP Traceback schemes, Section 3 introduces entropy calculation where the adaptation is carried out by making use of the entropy concept. In section 4 we introduce the graph theoretic solution where the theoretical idea of our proposed model for tracing back DDoS attack using Modified Floyd Warshall Algorithm, TDA/MFWA, is explained, where we use the Floyd-Warshall idea (Floyd, 1962) for graph nodes enumeration to compute the flow entropy at every graph node and use this entropy value to decide whether the underlying node is a victim then starting from the victim node, if any, traceback to the real attack source, Simulation and performance evaluation is introduced in section 5 and the conclusion is presented in section 6.

2 Related work

The summary of the existing DoS/DDoS traceback methods can be found in Balyk et al. (2015) and Nagesh et al. (2017). Such traceback strategies can be categorised as follows:

2.1 Packet logging

In logging methods, the routers keep some specific information of travelling packets, such information represents a fingerprint of the packet (Aghaei-Foroushani and Zincir-Heywood, 2013), based upon the invariant portions of the underlying packet (source, destination, etc.). During the traceback, the routers can verify whether or not a suspicious packet has been forwarded. To achieve improvement in logging, only a small portion of each travelling packet at the transient routers has been considered. One of the major problems of the logging method is the requirement for high amount of memory and CPU usage on the routers in the attack paths.

2.2 Packet marking

There are two methods of packet marking: the probabilistic packet marking (PPM) and the deterministic packet marking (DPM).

2.2.1 Probabilistic packet marking

In such mechanism, packets are probabilistically marked with partial path information as they are forwarded by the routers. Accordingly, the victim can reconstruct the paths that the attack packets went through The PPM method is vulnerable to attackers, as pointed out in Aghaei-Foroushani and Zincir-Heywood (2013), as attackers can send spoofed marking information to the victim to mislead the victim. The accuracy of PPM is another problem, because the marked messages by the routers who are closer to the leaves (which means far away from the victim) could be overwritten by the downstream routers on the attack tree (Al-Duwairi and Govindarasu, 2006). At the same time, most of the PPM algorithms suffer from the storage space problem to store large amount of marked packets for reconstructing the attack tree (Goodrich, 2008; Savage et al., 2001). Based on the PPM mechanism, Law et al. (2005) tried to traceback the attackers using traffic rates of packets which were targeted on the victim (Yaar et al., 2005). Their model is based on the assumption that the traffic pattern has to obey the Poisson distribution which is not always true in the internet.

2.2.2 Deterministic packet marking

The deterministic packet marking mechanism tries to mark the spare space of a packet with the packet's initial router's information, e.g., IP address. Therefore, the receiver can identify the source location of the packets once it has sufficient information of the marks. The major problem of DPM is that it involves modifications of the current routing software, and it may require a very large amount of marks for packet reconstruction (Snoeren et al., 2001, 2002; Belenky and Ansari, 2003; Dean et al., 2006).

2.3 Hybrid approach

In Belenky and Ansari (2003), two hybrid schemes that combine the packet marking and packet logging afford a method to traceback the attack sources. Attack sources:

1 distributed link-list traceback (DLLT)

2 the probabilistic pipelined packet marking (PPPM).

The first one preserves the marking information at intermediate routers in a specific way so that it can be collected using a link-list-based approach. The second method targets propagating the IP addresses of the routers that were involved in marking certain packets by loading them into packets going to the same destination, therefore, preserving these addresses while avoiding the need for long term storage at the intermediate routers.

2.4 ICMP traceback

In internet control message protocol (ICMP), traceback routers can, with a low probability, generate a traceback message that is sent along to the destination (Bellovin, 2000). With enough traceback messages from enough routers along the path, the traffic source and path can be determined. It was considered an industry standard by Internet Engineering Task Force (IETF). ICMP traceback does not require any change in the existing infrastructure, however, ICMP traceback requires an out_of_band message. The messages generated for the purpose of traceback will pollute the network with additional packets during large scale DDoS attacks, as they rely on numerous packets to trace an attack origin (VijayalakSshmi and Shalinie, 2014).

2.5 Statistics and entropy variation

A covariance analysis model is discussed in Jin and Yeung (2004) and Manikopoulos and Papavassiliou (2002). The simulation results show that this method is accurate in detecting malicious network traffic in DDoS attacks of different intensities. This method can effectively differentiate between normal and attack traffic. Moreover it can detect very subtle attacks only slightly different from the normal behaviours. The linear complexity of the method makes its real time detection practical. The covariance model of Jin and Yeung (2004) verifies the effectiveness of multivariate correlation analysis for DDoS detection, however, some open issues still exist in this model for further research. Such model can be extended to provide principle component analysis that can be also exploited for detecting DDoS attacks (Kaur and Gauravdeep, 2017).

By making use of particular statistics of information theory different entropy-based algorithms can be conducted. Entropy measures the unpredictability of a distribution (Koay et al., 2018). Sudden variations in the measured entropy allow detecting anomalies in the distribution of traffic features.

To this end, Shannon entropy (Shannon, 1948) can be exploited to detect and traceback DDoS attacks (Ruohonen, 2013). Statistically high incidence for a given flow leads to a reduced entropy and conversely, low and dispersed incidences translate to higher entropy values. It follows that entropy-based algorithms are widely used for the detection and traceback of attacks in communication networks (Giotis et al., 2014). By identifying significant changes in the randomness of consecutive traffic features distributions, this statistical approach can detect DDoS, with better accuracy than methods based on volume metrics (Giotis et al., 2014).

2.6 Machine learning

Learning paradigms, such as artificial neural networks (ANNs), radial basis functions and genetic algorithms are widely used in DDoS attack detection because of their ability to classify traffics intelligently and automatically (Liu et al., 2007). Other researchers have proposed an effective defensive system called NetShield to protect client hosts, network routers and network servers from becoming victims (Mohan and Angamuthu, 2018). Their work protects any IP-based public network using rate limiting to eliminate system vulnerabilities on target machines. It enforces dynamic security policies to secure the underlying network resources against DDoS flood attacks.

The scheme that has been proposed here in Sections 3 and 4 consists of two parts. The first part, Section 3, is concerned with entropy calculations while the second part is devoted to the explanation details of the underlying graph theoretic solution.

3 Entropy calculation

Our model operation starts with the calculation of probability p_{ij} of particular flows at local router L_k, where k is an intermediate point between the a source i and a corresponding destination j, then we calculate the entropy of flow H(F)

$$HF = H(p_1 \dots p_N) = -\sum_1^N p_i \log p_i \tag{1}$$

where i = 1, 2, …, N. For each source (destination) address (port), we calculate the probability;

$$p(x_i) = \frac{\text{Number of packets with } x_i \text{ as source (dst) address}}{\text{Total number of packets}} \tag{2}$$

From which one can obtain the flow entropy expressed by:

$$H(F) = \sum_i \sum_j p_{ij}(u_i, d_j) \log p_{ij}(u_i, d_j) \tag{3}$$

where F is the underlying flow, u_i is the i^{th} upstream source, d_j is the j^{th} destination and

$$p_{ij}\left(u_i, d_j\right) = \frac{N_{ij}\left(u_i, d_j\right)}{\sum_i \sum_j N_{ij}\left(u_i, d_j\right)} \tag{4}$$

where N_{ij} is the number of packets from source i to destination j denoted by u_i and d_j, respectively. Accordingly

$$\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} p_{ij}\left(u_i, d_j\right) = 1.$$

4 Graph theoretic solution

First we explain the theoretical idea of the proposed model, however, such idea in its theoretical form is not fairly useful due to practical limitations in memory size and processing power. Therefore we followed the theory by a practical form of TDA/MFWA.

4.1 Theoretical idea of TDA/MFWA

The proposed model, TDA/MFWA is illustrated in Figure 1. It starts by reading the underlying network topology G(V, E). If we are given an adjacency matrix of the graph G = (V, E), one can construct a weighted adjacency matrix in which an edge weight, x_{ij}, between nodes i and j is represented by:

$$x_{ij} = h_{ij} = p_{ij} \log p_{ij} \tag{5}$$

Then, we replace such entropy values with ones to get matrix E on which Algorithm 1, that is a special version of Floyd-Warshall algorithm, is applied to get the reachability matrix R. Matrix R = (r_{ij}) of G can be formed, where

$$r_{ij} = \begin{cases} 1, & \text{if G has a directed } v_i\text{-}v_j \text{ path} \\ 0, & \text{otherwise} \end{cases}$$

where, $V = \{v_1, \ldots, v_n\}$. We should note that if $r_{ii} = 1$, then v_i is in a directed circuit. F-W algorithm constructs a series of $n \times n$ matrices $E_1, \ldots, E_n$, where

1 elements of E_i are either zero or one

2 $E_i \le E_{i+1}$ $(i = 0, \ldots, n - 1)$

3 E_0 is obtained from the adjacency matrix D by replacing the positive elements with ones

4 $E_n = R$.

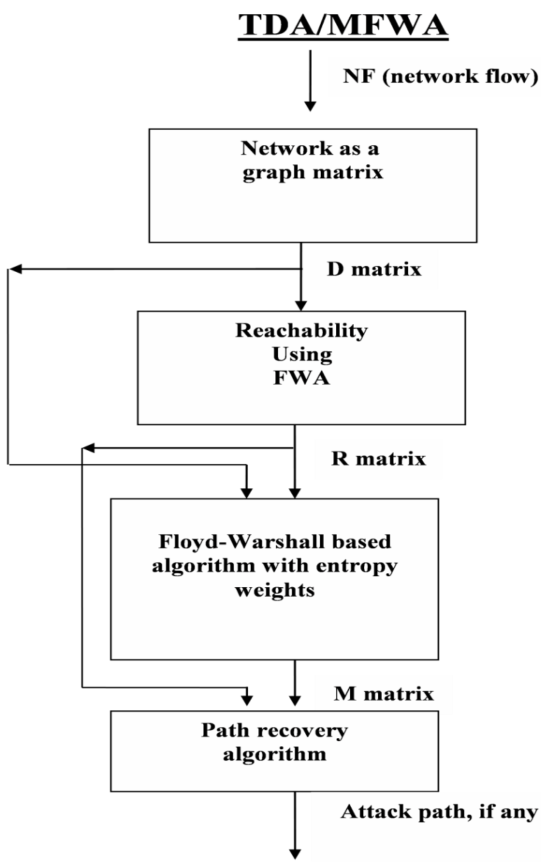
Algorithm 1 Reachability

```
Input : Adjacency matrix D
Output : Reachability matrix R
begin
    E := E0
    for i := 1 to n do
```

```
        for j := 1 to n do
            if (E)ji = 1 then for k := 1 to n do
                (E)jk := max((E)jk, (E)ik)
            End if
        End for
    End for
End Reachability
```

To detect an attack from a source $\mathcal{S}$ to victim $\mathcal{V}$, it is necessary (but not sufficient) that matrix R should include a path that makes $\mathcal{V}$ reachable from $\mathcal{S}$. If the network has n nodes then TDA/MFWA considers the corresponding set of graph nodes V, and for every node it applies equation (1) to obtain matrix M.

Figure 1 Flowchart of attack paths, if any



Actually, M is a matrix in which the flow entropy $H_{ij}(F)$ values at graph nodes (i, j), i = 1, 2, …, n; are preserved. M is constructed as follows:

$$M[i][j] = \begin{cases} H_{ij}(F), & \text{if } H_{ij}(F) <= Thr_{ij} \\ 0, & \text{otherwise} \end{cases}$$

The Thr_{ij} is predetermined manually by the network manager depending on his experience. Consequently, by ANDing (i.e., multiplying) R (that contains the reachable paths to the victim) by M (that contains reduced entropy values at infected graph nodes) one can obtain a final matrix with its entries are nodes satisfying both necessary and sufficient conditions of DoS/DDoS attack. At this end TDA/MFWA is theoretically completed and the attack path can be directly determined using any path recovery procedure (e.g., the procedure illustrated at the end of Algorithm 2).

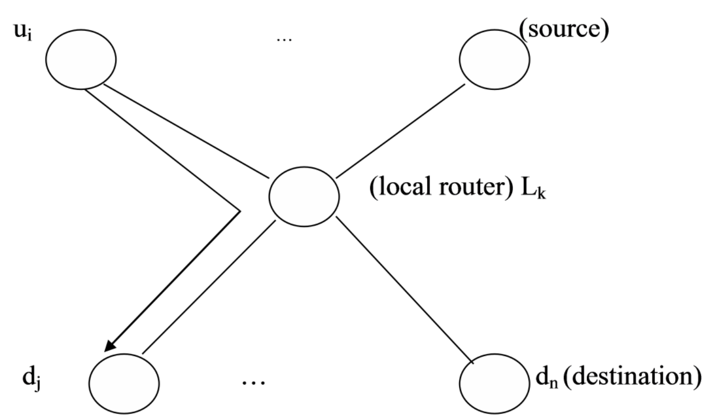
Thus by ANDing the reachability and entropy drop inequality conditions attack necessary and sufficient conditions are obtained. At this point a deep insight for R and M matrices indicates that for every vertex, if a victim node is determined it will be associated with a large amount of connections (reachability links in R). The study of these tremendous amounts of edges for every vertex is impractical from both processing power and memory size view points. It is obvious that there is a need for a practical solution to mitigate that problem.

4.2 Practical model using modified Floyd-Warshall

4.2.1 Algorithm

Actually Floyd-Warshall algorithm is an algorithm, from graph theory, for finding shortest paths in a weighted graph represented by matrix D. A single execution of that algorithm will find out the shortest (summed weights) paths between all pairs of network vertices. Applying that algorithm on a network G(V, E) with vertices, V, numbered from 1 to n, it returns every shortest path from i to j (i, j = 1, 2, …, n) using nodes only from a set {1, 2, …, k} where ks act as intermediate points between i and j. In fact, the use of k (between i and j) represents the heart of Floyd-Warshall algorithm for finding out the required paths. Here the algorithm is modified so that its objective function is changed from finding shortest paths between every pair of graph nodes to finding each node between every pair of graph nodes at which the entropy is less than a predefined threshold. Accordingly, the modified Floyd-Warshall idea for graph nodes enumeration is used to compute the flow entropy at every graph node and use this entropy value to decide whether or not the underlying node is a victim. Then starting from the victim node, if any, a traceback is carried out to the real attack source S.

Figure 2 Local router k as intermediate point between i and j



The proposed algorithm borrows that idea by relying upon k as intermediate node (router between i and j) to compute the flow entropy H(F) at every network vertex (rather than the shortest path) as shown in Figure 2, in which the local router, k, is the intermediate point between i and j.

Therefore the input to the proposed algorithm is a weighted graph in which any i, j edge weight is denoted by $h_{ij} = h_{ij}(u_i, d_j)$, given by equation (5). Also it should be noticed that a typical Floyd-Warshall algorithm does not provide path reconstruction between any pair of nodes. However a straightforward modification that is possible to recover the path between any two end points is presented in the traceback DDoS attack model using TDA/MFWA algorithm which is coded as Algorithm 2. It is worth noticing that such algorithm includes three nested for loops. If we consider a pair of nodes i and j, then according to Floyd-Warshall idea the outer most loop should be 'for k', otherwise that algorithm never works.

Algorithm 2 TDA/MFWA

```
Input: adjacency matrix
Output: nodes of recovered path
begin
    for every edge (u, v)
        h[u][v] := x(u, v) // the weight of the edge
        parent[u][v] := v
        for k from 1 to n
            for i from 1 to n
                for j from 1 to n
                    H[k] := H[k] + h[i][j]
                    if H[k] < thr, M[k] := H[k]
                        parent[i][j] := parent[i][k]
                end for
            end for
        end for
    end for
    Algorithm path (u, v)
    if parent[u][v] := null, return[]
        path[u]
    end if
    while u != v
        if H[u] < thr, u := parent[u][v]
        end if
    path.append(u)
    return path
end TDA/MFWA
```

The F-W algorithm in its original form considers every node in a graph as an intermediate point between a pair of nodes. A single execution of the algorithm will find out the lengths (weights) of shortest paths between all pairs of vertices (nodes). On the contrary, Dijkstra algorithm (Ruohonen, 2013) finds one shortest path at a time. Here the intermediate points are replaced by routing nodes and the weights are substituted by entropy values. Consequently, the entropy at 'all' network nodes are found out in a single algorithm execution.

4.3 Traceback time complexity

The node of the victim may exploit the reachability concept to reach the source of the attack. Therefore, one may be intended to store the actual path from each vertex to each other vertex. This is not necessary and in fact is costly in terms of time and memory. Instead the path tree can be utilised, as shown by algorithm path(s), to recover the attack path. The path tree as such could be found out using the following steps:

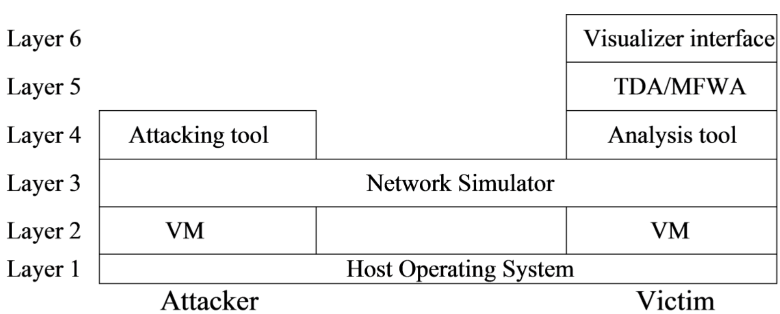
1. determine path(s) from root v to vertex node u if: v and u are connected and entropy at u is less than the threshold
2. for all non-root vertices u, assign to u a parent vertex that satisfies these two conditions
3. construct the attack path from the edges between every node and its parent, if that path is not unique, then investigate all of the resulting paths.

Accordingly, algorithm path(s) can obtain the path nodes in O(n) time units where n is the number of the network nodes, in addition to $O(n^3)$ for applying the modified Floyd-Warshall algorithm.

4.4 System architecture

System architecture for the underlying model is shown in Figure 3.

Figure 3 System architecture for TDA/MFWA



Such architecture contains two individuals: one represents attacker while the second represents the victim. It consists of a hierarchy of six layers. The lowest layers are responsible for O.S. where layer 1 presents the host environment that allows a virtual machine to run as a process belonging to the current operating system. Layer 2 is the virtual machines that can behave as separate standalone computing system. In Figure 3, layer 3 represents a network simulator that might allow the combination of virtual and real elements used to simulate complex network. The simulator as such, should be scalable, adaptable and capable to allow on demand configuration. On the top of the simulator, for the victim individual, there is the assisting tool that consists of analysing and tracing back tools. Also, in the same layer namely layer 4 there are the analysis tools that are able to monitor and profile the network traffic in normal and abnormal operational conditions. It basically lets the administrator capture the traffic parameters and distribution in the underlying network.

On the other hand, for the attacker individual, the attacking tool can launch an attack, e.g., DDoS attack using the following steps:

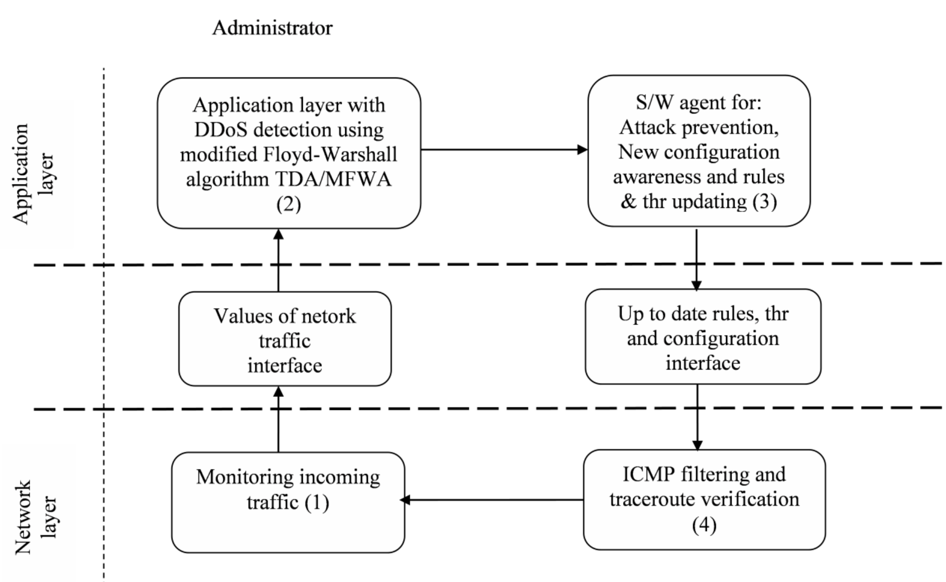
1. the attacker establishes a machine and/or a network to be responsible for the required volume of traffic
2. the attacker then discover vulnerable hosts in the sense that they are running without software protection
3. the attacker now apply his attack tool(s) to throw down this victim(s).

Layer 5 in the victim individual shows the tool which is concerned with detecting and tracking back any DDoS attack (here, TDA/MFWA). It can be implemented using versatile concepts, methods and techniques. Layer 6, in the victim side, is the highest layer in the hierarchy. It contains a visualise interface to provide visualisation of the network traffic data that can be displayed and manipulated in real-time.

4.5 Relation between TDA/MFWA and other network components

Sections 4.2 and 4.4 explain TDA/MFWA from inside but in Section 4.5 it is explained from outside. Therefore, an example for its applicability is given in Figure 4.

Figure 4 Cross-layer control loop



TDA/MFWA can work properly in a cross-layer control loop, as shown in Figure 4. Apart from the given example, it is worth noticing that other cross layer configurations can be also implemented. Such cross-layer approach is exploited here to connect two layers in this work, namely, the network layer where the traffic overflow is sensed and the application layer where the modified Floyd-Warshall algorithm is computed and the corresponding decisions are taken. The control loop starts from block 1, at the network layer, where the incoming traffic is monitored and special ICMP signals are utilised to determine the nodes that can reach the underlying victim. Then value of the incoming traffic is received by an interface, that reads the traffic value from the IP layer and sends it to TDA/MFWA at block 2, that can detect either DoS or DDoS attacks if any. Moreover, TDA/MFWA exploits the path tree to determine the traceback path(s). If an attack is found out, then TDA/MFWA sends a signal to the attack prevention, block 3. Actually, several software agent platforms are available where any of them can be employed to:

1 activate the corresponding firewall to prevent the attack
2 inform the network administrator to dynamically adjust thresholds and filtering rules

Moreover, such software agent sends the traffic statistics and network configuration based on updated thresholds to the corresponding interface that feeds them back to the IP layer for traceback verification using the ICMP traceroute, block 4. Thus, the traffic is monitored accurately by the network layer, block 1. Now the network administrator obtained the traceback paths from two reliable sources, the Modified Floyd-Warshall Algorithm and the ICMP traceback. If the paths of the two independent sources match then the trace back is verified, otherwise, this result is considered inconsistent and should be disregarded.

5 Simulation and performance evaluation

Here GNS3 is used as an appropriate simulator to simulate DoS/DDoS attacks. The underlying setup is illustrated, the model performance is evaluated and a comparative study between TDA/MFWA and other traceback tools is presented.

5.1 Set-up

The simulation setup is inspired from the proposed architecture that has been given in Section 5. That instance is shown in Figure 5.

The underlying hardware configuration is a machine with a processor Core i5 and a RAM of 8 GB. On which the host environment is based on Ubuntu operating system at which virtual machines are running where every virtual machine supports a network node or router. On Ubuntu GNS-3 runs as a simulator to build and access the underlying network. GNS-3 is chosen because of the following:

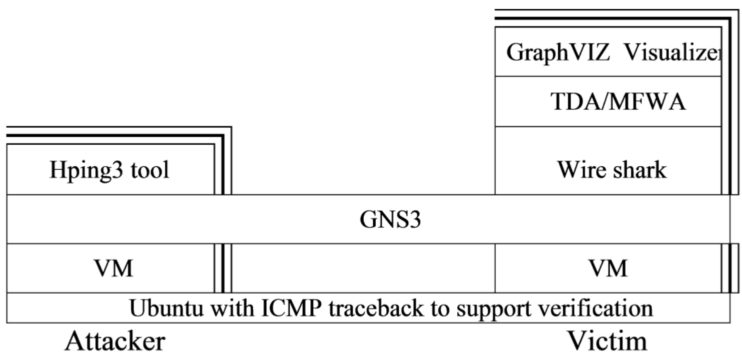
1 it is simple and unique virtualisation capability provides a precise and flexible simulation platform (Lal et al., 2016)
2 it can be used as a scalable tool for emulating computer networks

3 it can be used to easily simulate a more complex network, which operates in real world scenarios.

The GNS3 software is open-source and free of cost that can be easily downloaded.

The attacker individual provided by hping3 tool exists on the top of GNS-3, while in the victim side we find the well-known traffic analyser, Wire Shark (win64-2.6.1). The results obtained from Wire shark are passed to TDA/MFWA module that detects the DoS/DDoS attacks and traceback them. GraphVIZ (2.38) is used as a visualiser to facilitate user friendly interface.

Figure 5 The simulation setup



5.2 Simulation results evaluation

An important concern here is to show that the traffic flow is normal and entropy variation is stable for non-attack cases, and find out the fluctuations for normal situations. Accordingly the relationship between the drop of flow entropy variation and the increase of attack strength is demonstrated. Further, the whole attack tree and its corresponding traceback time is simulated and evaluated. The underlying topology, shown in Figure 6, is chosen as a network topology since it represents an illustrative simple realistic example. That topology contains 11 typical machines in addition to a server, from which six machines can be attackers whilst five machines can act as victims. The configuration contains four routers and four switches. It should be noticed that the network example of Figure 6 appears to be simple. However, it is carefully chosen and constructed to satisfy the following:

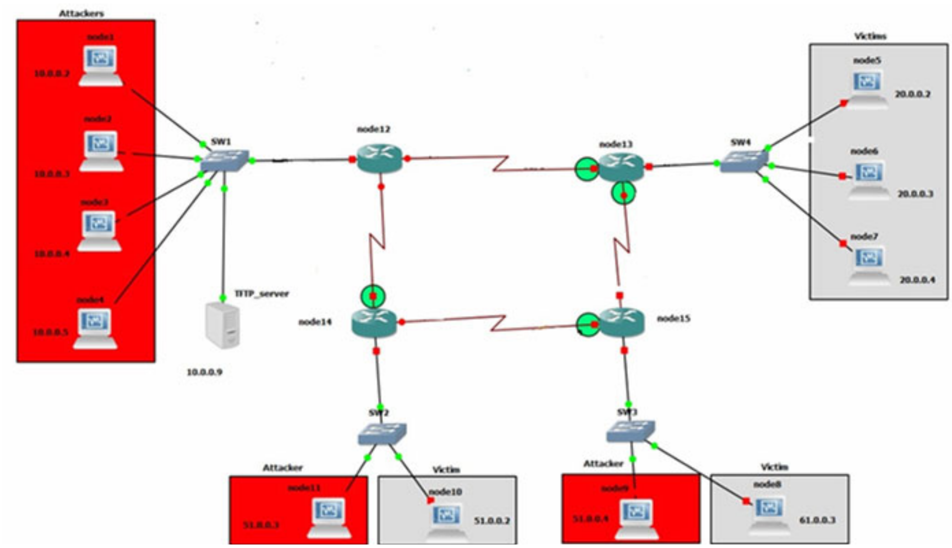
1 ability to simulate all types of DoS/DDoS attacks including the cases of: single attacker only, attacker that exploits zombies and distributed attackers

2 traceability so that either the victim or the network analyst can accurately trace back the underlying attack without confusion

3 ability to simulate different operational circumstances including: normal, abnormal, overloading or network partitioning.

In this work DoS/DDoS threats, as network-based attacks, are considered and informally modelled. The given model is implemented via an attacker-centric approach in which either attackers or zombies can be detected and traced back using F-W algorithm. In addition, it utilises a virtual representation of the network infrastructure, see Figure 6, to identify the potential attack and to avoid security vulnerabilities. Although such model does not provide a formal security framework it affords a rigorous means for improving readability, mitigating DoS/DDoS threats and enabling comparative studies with other researches.

Figure 6 The network topology (see online version for colours)



The single DoS flow case is illustrated in Figure 7, where, node 5 represents the victim.

Figure 7 The single DoS flow attack (see online version for colours)

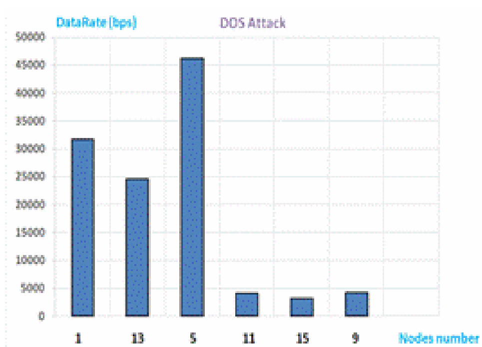


Figure 8, represents zombie attack where all zombies target the same victim (node 5).

The DDoS is indicated in Figure 9, where the attack paths are 1, 12, 5 from attacker (node1) to victim (node5) and 11, 14, 8 from attacker (node11) to victim (node 8).

Figure 10 is interesting as it represents the entropy variation on the nodes along a path in the cases of no attack and attack. In the normal case the entropy changes from node to node while in the case of attack the entropy variation is nearly constant along the path nodes. This result confirms the propositions that are given in Yu and Zhou (2008), upon attack, the values of randomness at the attack path routers are nearly the same. To confirm our result Figure 10 is plotted between entropy and path nodes to present the constancy of entropy in case of attack.

Figure 8 Zombies attacks (see online version for colours)

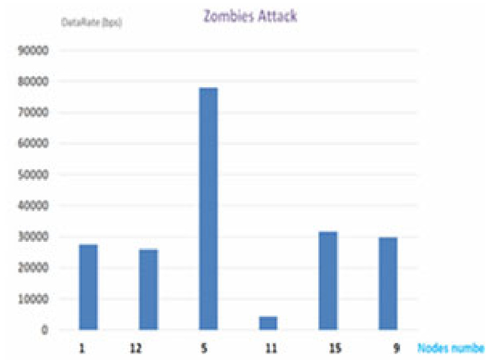


Figure 9 DDoS attacks (see online version for colours)

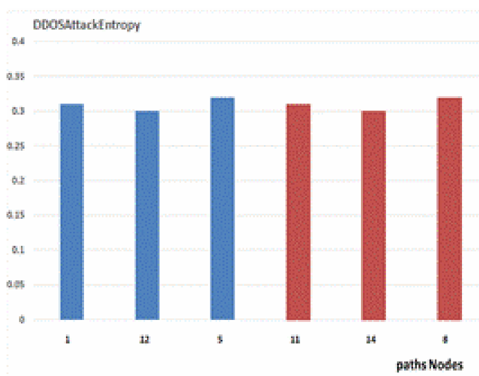


Figure 10 Entropy variation at attack and no attack (see online version for colours)

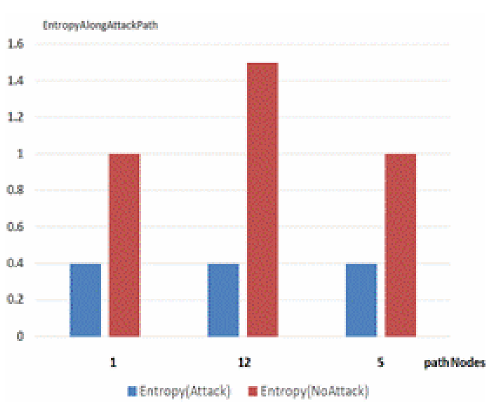
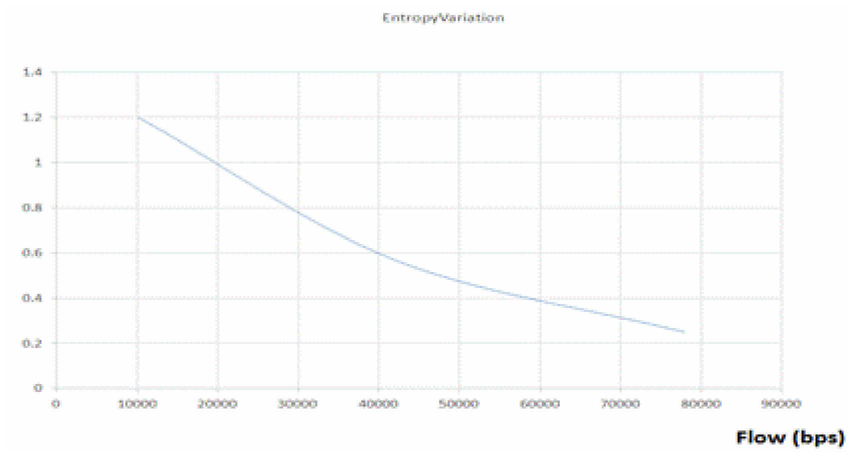


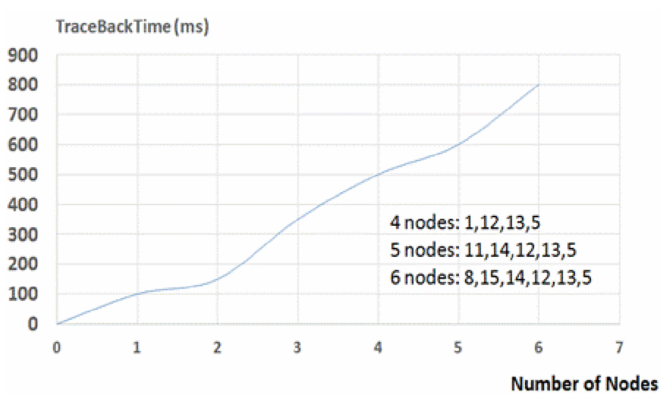
Figure 11 indicates that the entropy variation drops clearly and steadily with the increase in the attack strength represented by a flow in bps.

Figure 11 Entropy variation with attack strength (see online version for colours)



After detecting an attack the attacker is traced back. Different attacks have been carried out during different paths. Figure 12 illustrates traceback time against number of nodes of each path.

Figure 12 Traceback time with number of nodes at different paths (see online version for colours)



For that figure, it should be noticed that:

$$\text{time to detect attack} = \text{time of finishing proposed algorithm} \\ -\text{time at which attack reaches victim}$$

$$\text{turnaround time} = \text{time to capture data rate files} \\ +\text{time to detect attack} + \text{traceback time}$$

5.3 *Comparative study*

This study concentrates on similar traceback tools given in (comparative references). The basic comparison metrics are illustrated in Table 1, where DPM, PPM, entropy variation and machine learning (ML) approaches are considered.

The reasons that made this table qualitative (not quantitative) are:

1. lack of any common suitable data sets of real DDoS attacks so that they can be used by different research groups
2. it is even harder to find data sets reasonable to our algorithms
3. it is not sensible to simulate networks using dissimilar environments.

Despite the fact that CAIDA dataset is available it has not been used here because in this dataset non-attack traffics are removed. However, TDA/MFWA depends on real traffic flows including attack and non-attack traffics (Figure 4) to compute entropy variations.

In Table 1, both DPM and PPM are considered despite the fact that they are old techniques. But they are still working and dependable until now. The existence of these old techniques emphasises the considerable saving in storage and workload that had been achieved by the new techniques (entropy variation and machine learning). In some cases the severity of attack may make sense, however, such parameter has not been taken into consideration here.

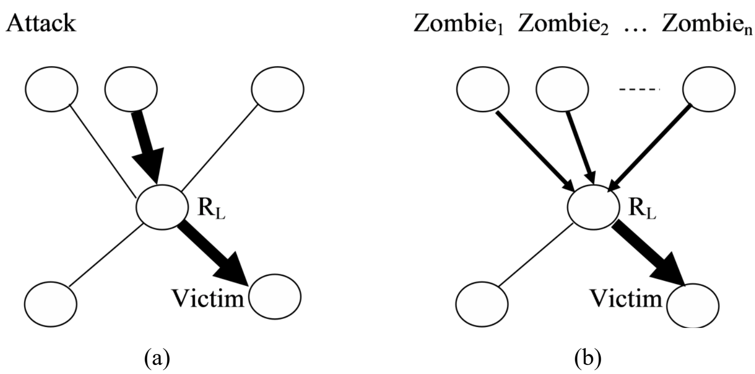
Table 1 Comparative study*

	DPM (Snoeren et al., 2002; Belenky and Ansari, 2003)	*PPM (Aghaei-Foroushani and Zincir-Heywood, 2013)*	*Entropy variation (Yu et al., 2011)*	*Entropy-based collaborative DDoS (Yu and Zhou, 2008)*	*ML (Liu et al., 2007; Mohan and Angamuthu, 2018)*	*TDA/MFWA*
Storage	Very high	High	Very low	Very low	Low	Very low
Traceback time	Low	Medium	Low (network delay)	Not measured	Not measured	Very low (linear complexity)
Operation workload	Very high	Very high	Very low	Very low	Needs training and testing	Very low
Purpose	Finding traceback path	Finding traceback path	Finding traceback path	Attack detection	Attack detection	Attack detection and attack traceback
Tool approach	Heuristic	Heuristic	Heuristic	Detection only	Artificial intelligence	Graph theoretic
Entropy distribution along routers	Not measured	Not measured	Decreases near the victim	Nearly constant	Not measured	Nearly constant

Note: *The numbers in the columns' headers are the numbers of the references that we compare with.

It is worth noticing that, depending on the network topology, the local router R_n may receive one attack flow (very high) out of many incoming flows Figure 13(a). Then it passes these flows to their destination, in addition to the attack flow, to the victim. Another situation is presented by Figure 13(b) where several zombies target their victim via R_l.

Figure 13 Attack situations



Since entropy variation, equation (3) is sensitive to both incoming and outgoing flows the two cases can be directly handled using the proposed graph theoretic approach. For both situations the underlying approach, as such can discriminate the attack from normal flow without need to the heuristics given in Bhuyan et al. (2013) and Yu and Zhou (2008). The DoS attacks that are chosen here belong to the pattern given in Figure 13(a) because it is more realistic for small networks.

Actually, this study is based on Shannon entropy, thus the decision whether the underlying event is an attack or not is determined not only by packet overflow but also by input and output distributions of packets at a network node. Actually, this fact increases the reliability and decreases wrong identifications. Shannon entropy has been employed in several previous works, however, here the situation is different, since in all previous works the attacker trace back is conducted depending upon a heuristic way but in this study the entropy is embedded in a graph theoretic approach that yields a practical and dependable traceback methodology. In addition, in this approach a node might not be falsely identified (or overflow is wrongly considered an attack) since an event at a particular node is recognised as attack if and only if:

- the entropy is considerably reduced at that node
- the entropy values are nearly the same at all nodes of the attack path (see Figure 10).

6 Conclusions

In this paper TDA/MFWA has been presented as a security tool against DoS/DDoS attacks. In fact such tool modifies Floyd-Warshall algorithm and integrates it properly with the entropy variation concept to afford a unique graph theoretic model that can be used efficiently and effectively with the following advantages:

1 DoS and DDoS are treated homogenously.

2 not only the DoS/DDoS attack is detected (at the victim node) but also the attacker is traced back

3 the traceback path is verified using ICMP trace route

4 neither logging nor marking is needed for the flow packets

5 the tool response is fast and reliable as well as it enables network administrators to discriminate between DDoS attacks and legitimate flows.

On the other hand, the main disadvantage of TDA/MFWA is that it is an emerging model with insufficient practical tests that proposed model is designed, simulated and verified for a computer network under various operational and attack conditions. Accordingly, its performance is evaluated and compared with other similar tools that exist in the literature.

In future, TDA/MFWA will employ the graph theoretic approach to examine other entropy formulas, rather than Shannon entropy to find out the best formula and the corresponding attack conditions. The use of multiple formulas is attractive, however, online switching from one formula to another is not an easy task.

References

Aghaei-Foroushani, V. and Zincir-Heywood, A.N. (2013) 'On evaluating IP traceback schemes: a practical perspective', *Security and Privacy Workshops (SPW)*, San Francisco, Vol. 2, pp.127–134, IEEE.

Al-Duwairi, B. and Govindarasu, M. (2006) 'Novel hybrid schemes employing packet marking and logging for IP traceback', *IEEE Transactions on Parallel and Distributed Systems*, May, Vol. 17, No. 5, pp.403–418.

Balyk, A., Iatsykovska, U., Karpinski, M., Khokhlachova, Y., Shaikhanova, A. and Korkishko, L. (2015) 'A survey of modern IP traceback methodologies', *Proc. of 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, 24–26 September, Warsaw, pp.484–487,.

Belenky, A. and Ansari, N. (2003) 'IP traceback with deterministic packet marking', *IEEE Communications Letters*, April, Vol. 7, No. 4, pp.162–164.

Bellovin, S.M. (2000) *ICMP Traceback Messages*, March, Internet Draft: draft-bellovin-itrace-00.txt.

Bhuyan, H., Kashyapm, H.J., Bhattacharyya, D.K. and Kalita, J.K. (2013) 'Detecting distributed denial of service attacks: methods, tools and future directions', *The Computer Journal*, Vol. 57, No. 4, pp.1–20.

Dean, D., Franlin, M. and Stubblefield, A. (2006) 'An algebraic approach to IP traceback', *ACM Transactions on Information and System Security*, May, Vol. 5, No. 2, pp.119–137.

Floyd, R.W. (1962) 'Algorithm 97 shortest path', *Communications of the ACM*, Vol. 5, No. 6, p.345.

Giotis, K., Argyropoulos, C., Androulidakis, G., Kalogeras, D. and Maglaris, V. (2014) 'Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments', *Computer Networks*, Vol. 62, No. 4, pp.122–136.

Goodrich, M.T. (2008) 'Probabilistic packet marking for large-scale IP traceback', *IEEE/ACM Transactions on Networking*, Vol. 16, No. 1, pp.15–24.

Jin, S. and Yeung, D.S. (2004) 'A covariance analysis model for DDoS attack detection', *IEEE International Conference on Communications*.

Kaur, R. and Gauravdeep, R. (2017) 'Statistical based DDoS detection methods: a review', *International Journal of Engineering Technology, Management and Applied Sciences*, April, Vol. 5, No. 4, pp.301–320.

Koay, A., Chen, A., Welch, I. and Seah, W.K.G. (2018) 'A new multi classifier system using entropy-based features in DDoS attack detection', *The 32nd International Conference on Information Networking (ICOIN)*, Chiang Mai, Thailand.

Lal, D., Ghorban, B. and Vaghr, S. (2016) 'A survey on the use of GNS3 for virtualizing computer networks', *International Journal of Computer Science and Engineering (IJCSE)*, December–January, Vol. 5, No. 1, pp.49–58.

Law, T.K.T., Lui, J.C.S. and Yau, D.K.Y. (2005) 'You can run, but you can't hide: an effective statistical methodology to traceback DDoS attackers', *IEEE Transactions on Parallel and Distributed Systems* September, Vol. 16, No. 9, pp.799–813.

Liu, Y., Cukic, B. and Gururajan, S. (2007) 'Validating neural network-based online adaptive systems: a case study', *Software Quality Journal*, Vol. 15, No. 9, pp.309–326.

Manikopoulos, C. and Papavassiliou, S. (2002) 'Network intrusion and fault detection: a statistical anomaly approach', *IEEE Communications Magazine*, October, Vol. 40, No. 10, pp.76–82.

Mohan, S.V. and Angamuthu, T. (2018) 'Association rule hiding in privacy preserving data mining', *International Journal of Information Security and Privacy*, Vol. 12, No. 3, pp.141–163.

Nagesh, K., Sumathy, R., Devakumar, P. and Sathiyamurthy, K. (2017) 'A survey on denial of service attacks and preclusions', *International Journal of Information Security and Privacy*, Vol. 11, No. 4, pp.1–15.

Ruohonen, K. (2013) *Graph Theory*, pp.52–61, Tampere University of Technology.

Savage, S., Wetherall, D., Karlin, A. and Anderson, T. (2001) 'Network support for IP Traceback', *IEEE/ACM Transactions on Networking*, June, Vol. 9, No. 3, pp.226–237.

Shannon, C. (1948) 'A mathematical theory of communication', *Bell System Technical Journal*, July–October, Vol. 27, No. 3, pp.379–423, doi:10.1002/j.1538-7305.1948.tb01338.x.

Snoeren, A.C., Partiridge, C., Sanchez, L.A., Jones, C.E., Tchakountio, F., Schwartz, B., Kent, S.T. and Strayer, W.T. (2002) 'Single-packet IP traceback', *IEEE/ACM Transactions on Networking*, December, Vol. 10, No. 6, pp.721–734.

Snoeren, A.C., Partridge, C., Sanchez, L.A., Jones, C.E., Tchakountio, F., Kent, S.T. and Strayer, W.T. (2001) 'Hash-based IP traceback', *AMC SIGCOMM*.

VijayalakSshmi, M. and Shalinie, S.M. (2014) 'Single packet ICMP traceback technique using router interface', *Journal of Information Science and Engineering*, Vol. 30, No. 6, pp.1673–1694.

Yaar, A., Perrig, A. and Song, D. (2005) 'FIT: fast internet traceback', *IEEE INFOCOM*, Miami, USA, pp.1395–1406.

Yu, S. and Zhou, W. (2008) 'Entropy-based collaborative detection of DDoS attacks on community networks', *Sixth Annual IEEE International Conference on Pervasive Computing and Communications*, Hong Kong.

Yu, S., Zhou, W., Doss, R. and Jia, W. (2011) 'Traceback of DDoS attacks using entropy variations', *IEEE Transactions on Parallel and Distributed Systems*, March, Vol. 22, No. 3, pp.412–425.