BOX COUNTING BASED MULTI-FRACTAL ANALYSIS OF NETWORK TO DETECT DOMAIN NAME SERVER ATTACK

[1]V.RAJAKUMARESWARAN, raja@admiresolution.com, Assistant Professor, Department of Computer Science and Engineering, Jay Shriram Group of Institiutions, Dharapuram Road, Avinashipalayam, Tirupur- 638 660, Tamilnadu, India.

[2]DR.S. NITHIYANANDAM, princenithi@gmail.com, Principal, Jay Shriram Group of Institiutions, Dharapuram Road, Avinashipalayam, Tirupur- 638 660, Tamilnadu, India.

Abstract

Domain Name Server (DNS) is a type of server, used to maintain and process the IP addresses of all the domains in internet. It works by responding with corresponding IP addresses when a client requests with a domain name. The DNS can be attacked by redirecting all the incoming traffic to a fake server by returning fake IP address when requested by a client. In this work, a novel work has been employed to detect DNS attack using Box Counting Method (BCM) based multi-fractal analysis. A set of network features are selected and rules are created using CISCO's Flowspec model and those features are analysed using BCM technique to find the attack in the network traffic. To the best of our knowledge, this is the first work which implements Flowspec based monitoring of DNS attack using fractal analysis.

Keywords: Domain Name Server, Box Counting Method, multi-fractal analysis, Flowspec, NXDOMAIN response

1. INTRODUCTION

1.1 NETWORK TRAFFIC ANALYSIS

Network monitoring is a most daunting task for a network administrator. He should constantly strive to maintain the operation of the networks. If a network is down even for a minor period of time, the impact will be high. In order to be proactive rather than reactive, admins need to monitor traffic movement and performance throughout the network and validate that security gaps do not occur within the network.

Network analysis is performed for recording, reviewing and analysing of network traffic for security, performance analysis and general network based actions and supervision. It can also be used to capture network traffic and inspect it closely to determine anomalies in the network. Anomalies can be defined as some patterns in data which do not imitate network traffic's regular behaviour. Network operators frequently face a wide range of such patterns in network traffic. Anomalous patterns could be benign abnormalities due to technical or physical issues, such as network outage, high-rate flows and sudden deviations due to flash crowds. On the other hand, they could be due to malicious illegal activities like cyber intrusions, Domain Name Server (DNS) attacks, worm propagation, port scanning, credit card frauds which could lead to catastrophic consequences and threaten the proper operation of networks.

Numerous methods are availed to analyse the network traffic for attack. This research proposes a novel method of fractal based Box Counting Method to carry out an in depth study of the characteristics of network traffic to detect the attacker in the basis of self-similarity.

1.2 FRACTAL ANALYSIS USING BOX COUNTING METHOD (BCM)

Multifractal analysis probe the nature of fractality (which can be either singularity, or non-integer behaviour or fractional) of the network. Mono-scale analysis suits any time series, but could not describe the relationship among various levels of resolutions or scales. If the time series is self-affine, single scale analysis would not be adequate and multifractal analysis is required to excerpt the features.

In general, the fractal analysis is performed to discover structures and characteristics of the network in order to understand the network better. By gathering such details, a systematic infection or attack can be possibly prohibited as mentioned by Stephanie[1]. According to Song *et al*[2], complex networks could have self-similar structures. According to these authors, the box-counting algorithm is a suitable method to examine global features of complex networks. Fractal analysis helps in calculating and understanding the fractal dimensions of complex networks.

The fundamental relation of fractal scaling is based on the box-counting method which analyses the self-similarity among the boxes and counts the total number of boxes that are needed to cover a network with boxes of certain size. The aforementioned method contains a random process for selecting the position of the centre of each box. Let $N_{\mathrm{B}}(r_{\mathrm{B}})$ be the least number of boxes required to tile the whole network, where the adjacent size of the boxes is the measure of radius r_{B} as follows, using Pengkui et al[3].

$$N_B(r_B \text{ rB}) \sim r_B^{-d_B} \quad (1)$$

Where d_B is the fractal dimension.

By implementing a multi-fractal analysis of DNS traffic, one can detect the in-depth characteristics of DNS based traffic attacks. From many multi-fractal analyses, Box-Counting Method (BCM) is implemented to trace the network attack.

In this work, the DNS traffic is divided into number of boxes each with a regular interval of time. On each interval, the traffic is analysed for some features of the network and the fractal dimension of such features are noted for each box. The comparison is made with all the boxes of the network and self-similarities are noted. If any box contains a fractal dimension which exceeds a threshold value, it will be noted and reported as bursty traffic. This process is repeated, until all the samples are completed and the deviation in traffic is analysed for attacker.

The flow of the paper can be summarized as follows:

In section 2, the tabulation of all the related works on detection of attack in network is made.

In section 3.1, all the features related to the detection of DNS based attack are analysed.

In section 3.2, Fractal analysis of DNS attack is made. In this section, the non-stationary characteristics of DNS traffic are analysed and discussions are carried out on how to detect attack in the flow using this model. It also discussed on how fractal analysis of NXDOMAIN can be made.

In section 3.3, it is described how BCM based FD analysis of DNS traffic can be made.

In section 3.4, Flowspec based and PARETO based traffic monitoring for burstiness is performed and the simulation of the attack detection is performed.

In section 4, Result is generated which proved that our Flowspec model is best which makes early detection of attack.

In Section 5, conclusions are derived and Section 6 gives the list of references made.

2. RELATED WORKS

TABLE I: DISCUSSION OF WORKS RELATED TO ATTACK DETECTION IN THE NETWORK

Work	Burstiness detection	Rule Creation	Mitigation of attack	Diversion or Dissemination Mechanisms	Fractal Dimension based attack detection
Hsiao-Wen[4]	Yes. Using Pareto Model	Yes. Using threshold based burstiness monitoring	No	No	Yes. Using BCM approach
Ziad El Jamous[5]	Yes. Using Flowspec model	Yes. Rule is created to update routing tables and to divert traffic.	Yes. By redirecting unwanted traffic for analysis and allowing normal traffic flows in	Yes. By diverting harmful traffic to safe location for further analysis	No.

			the network.		
Alakiri[6]	Yes. By using Pareto Model	Yes. By using inter-arrival time	Yes. By directing only regular traffic to fall into acceptable region	No	No
Muhammad[7]	Yes. By using variance analysis	Yes. By detecting changes in stationary characteristics of network	No	No	Yes. By using Variance Fractal Dimension (VFD) technique
Shriram[8]	Yes. By using connection level information	Yes. By analysing the alpha and beta traffic	No	No	Yes. By using Fractional Gaussian noise technique
Proposed system	Yes. It used a threshold value to detect the burstiness of DNS traffic	It creates rules by assessing FD of 5 distinct features of DNS to detect the attack	DNS routes all incoming requests to the provider's scrubbing servers, where malicious requests are dropped and legitimate ones are forwarded	DNS routing is activated by changing the CNAME and A record, so as to point them to the IP(s) of the mitigation provider.	Yes. Implemented BCM approach using information dimension of complex networks.

3. METHODOLOGY

3.1 Features used to detect DNS traffic attack

Any features which are significant for the detection of DNS attack can be considered for FD based analysis. The list of features that are more relevant to DNS attack with its legitimate and attacked values are mentioned in Table II.

TABLE II: FRACTAL ANALYSIS OF SOME FEATURES OF DNS TRAFFIC BY BOX COUNTING METHOD

Features Extracted	Reference network	Attacked network
Number of DNS servers contacted	420.17	560.23
Number of NOERROR responses	47.20	78.45
Number of NXDOMAIN responses	400	600
Average number of answers	42.21	85.34
Average number of authority answers	21.21	42.56
Average number of additional answers	39.14	86.52
Average number of resolved IPs	18.05	28.67
Mean of the value of TTL (Time to Live) field	27.45	57.89
Standard deviation of the value of TTL field	13.34	35.67

Here are the details of each features that are supportive to detect the DNS attack.

*a. **Number of DNS servers contacted***

Stephanie[1] stated a way to detect the DNS attack by monitoring the number of DNS servers being communicated for a request. He states that by doing a time series analysis on the requests done by DNS servers, an increase in the number of DNS servers being contacted by a specific host might be detected when an attack takes place.

Observation:

Thus, in our simulation, the attacked network had an average of 560 servers contacted per milliseconds, whereas the legitimate network had 420 servers, contacted for a request.

*b. **Number of NOERROR Responses***

The NOERROR response is made by the DNS when a query is finished successfully and a valid domain is created for the IP addresses requested. The attacker corrupts a DNS server by substituting a legitimate IP address in the server's cache with that of rogue address, in order to forward traffic to a malicious website, collect information or initiate another attack. Definitely, this may take some more time than that of the legitimate domain creation. It is justified by the following formula.

Time taken for an attacker to poison the DNS

To calculate the time needed for at least one forged response, accepted by a recursive NS, the following formula holds:

$$H = \frac{N}{1000/W} \quad (2)$$

Where, H is the time requirement of an attack in seconds.

N is the number of 'windows of opportunity' (the time between the query of the recursive NS and a genuine response from an authoritative NS) necessary for at least one fake response

W is the length of the 'window of opportunity' in milliseconds (ms) + overhead for the next 'window of opportunity' in milliseconds (ms).

Observation:

Thus, in the simulation, it is mentioned as the time taken for a legitimate reference network to generate valid domain name is 47.20ms, whereas the attacked response had taken 78.45ms to generate the cooked domain name which is the time taken by the attacker to get one forged response accepted by a NS.

*c. **NXDOMAIN responses***

NXDOMAIN is the situation where the DNS is unable to resolve a domain name due to its absence. NXDOMAIN attack arises when an attacker attempts to flood the DNS server with false queries to resolve a non-existent domain name. The DNS server looks for the domain that doesn't really exist, and hence never finds it. While the server efforts to

find the false domains sent to it, the cache gets obstructed with NXDOMAIN results, and hence slows down the reply to the legitimate requests. Hence during attacks, the NXDOMAIN response will be vast as there could be some form of malware influence domain generating algorithms (DGA) to try and reach the Command and Control (C&C). It is possible to see hundreds, and sometimes thousands of requests per day, being created by the DGA utilized by the malware.

Observation:

Hence in the simulation, the NXDOMAIN responses for the legitimate network is 400 per minute, when the network is attacked, it is 600 responses which is definitely due to DGA used by the attacker to create fake domains to engage the Name Server.

d. ***Average number of answers***

In a DNS attack, the offender tries to overbear a given DNS server or servers, with seemingly valid traffic produced by scripts, running on several compromised botnet machines. So during an attack, the average count of answers made by the DNS server will be great when compared to the legitimate network.

Observation:

Thus in the simulation, the average number of requests handled by DNS server is 42 per milliseconds, whereas the attacked traffic had got 85 responses handled by the server.

e. ***Average number of authority answers***

A DNS server that is NOT authoritative for a domain can provide an 'authoritative response' to a DNS query for a domain, it does not serve. Non-authoritative responses originate from DNS servers that have cached an answer for a given host, but received that information from a server that is not authoritative for the domain. By this way, an attacker can fake a DNS query by sending a bogus domain.

Observation:

In the simulation, it is observed that a legitimate network had an average number of authenticated reply as 21 responses per millisecond, whereas the attacked network had an average of 42 responses per millisecond.

*f.****Average number of additional answers***

In most cases, when a name is being resolved in the DNS, it offers some additional answers to explain why the name is being resolved. This may let the authoritative name server to guess what other answers a recursive resolver will soon query for. But the issue is that, an attacker may misinform the DNS to store bad evidence into the resolver's cache by referring inappropriate records with the actual response.

Observation:

In simulation, it is observed that a legitimate network had average number of additional answers in DNS responses as 39 per millisecond, whereas the attacked network had an average of 86 responses per millisecond.

g. ***Average number of resolved IPs***

The attacker can custom DNS ID hacking to find the ID number of the user to alter the cache of the user. It is a kind of redirecting domain name to another IP which can be the publishing page of an attacker. So when a network is compromised, minor number of IP addresses will only be resolved than unaffected network.

Observation:

In the simulation, it is observed that when a legitimate network had average number of resolved IPs by DNS is 28 per millisecond, whereas the attacked network had an average of 18 resolved IPs per millisecond.

h. **Mean and Standard deviation of the value of TTL**

TTL value expresses local resolving name servers, of how long a record should be kept locally before a new copy of the record must be fetched from DNS. This record storage is called DNS cache, and the act of storing the records, is known as caching.

Observation:

Observation reveals that some IP packets have an extraordinary TTL value that is much more than the TTL value of legitimate network, which will be more vulnerable for the attacker to underplay in the network. So in the network, the mean and standard deviation of TTL value for legitimate reference network is lower than the attacked network.

From the aforementioned statistics, it could be apprehended that if any of the features of a network exceeds the above mentioned legitimate values, it could be considered as an attacked network. In order to analyse the DNS traffic using fractal dimension, the DNS traffic should obey the non-stationary characteristics which is explained in detail in the next section.

3.2 FRACTAL ANALYSIS OF DNS TRAFFIC

3.2.1 Detection of Non- Stationary Characteristics

The DNS traffic should be non-stationary in its performance to undergo FD analysis. In a non-stationary process, there would be a variance in the mean, variance and autocorrelation structure of any network feature. Stationary tools must be applied in a sliding window fashion, to analyse statistical variations of DNS traffic. Variance Fractal Dimension (VFD) is a method, availed to demonstrate the change in the fractal structure of the network and the Hurst phenomenon. To measure long term memory of time series, Hurst exponent is used. VFD is one of the information based fractal dimension used to extract the variance feature of an object.

VFD is generated using Hurst Exponent (H) which is distinguished by fractional Brownian motion process (fBm). A fBm $\{B_t, t \geq 0\}$ is a self-similar stochastic process that has stationary gain. The fBm is ruled using Hurst parameter H ∈ [0,1]. The stationary increments with zero mean and variance which is dependent on the time stamp't' has a normal distribution. The fractional Brownian motion process with H= ½ is called a standard process and the stationary increments become independent.

In Hurst parameter H ∈ [0,1] with zero mean and covariance function, the fractional Brownian motion process is defined as $\{B_t, t \geq 0\}$. The function is as follows:

$$E(B_t B_{t+\tau}) = \frac{1}{2}((t+\tau)^{2H} + t^{2H} - |\tau|^{2H}) \quad (3)$$

where $H = \frac{1}{2}$, $E(B_t B_{t+\tau}) = \min(t, t+\tau)$ (4)

It is the covariance of a zero mean Gaussian process which indicates the independence of increments. It is the hallmark of a quality Brownian motion process. When H ≠ ½, the increments depend on each other.

Power law relationship is used to do VFD calculation. This calculation is done between the amplitude increments of the time series. It is important to note that the time series needs to be stationary in the statistical sense for a valid calculation of VFD. Hence, a sliding window of information samples is selected for a VFD calculation such that the stationarity is fortified in the weak sense and a trajectory of the VFD is obtained which differs within the embedding dimensions of the time series. This trajectory is called Variance Fractal Dimension Trajectory (VFDT).

x(t) indicates a data time series, sampled at equal intervals. x(t) is denoted as a discrete and sampled output of a stochastic process. It is important to note that the sampling frequency must be selected in a way, that it protects the information content of the process. For a process x(t):

$$\text{Variance} = \text{var}[x(t)] = E[(x - \overline{x})^2] \quad (5)$$

where E(.) is the statistical expectation operator and $\overline{x}$is the statistical mean (first moment) of the processes X(t). Hence, according to power law[7]:

$$\log(var[x(t_2) - x(t_1)]) \sim 2H \log[\Delta t] \quad (6)$$

The relationship between $\log(var[x(t_2) - x(t_1)])$ and $\log[\Delta t]$ can be found. The half slope of linear interpolation of the plot provides Hurst Exponent (H), which is mathematically equivalent to the following

$$H = \frac{1}{2} \lim_{\Delta t \to 0} \frac{log[var(\Delta X_{\Delta t})]}{\log[\Delta t]} \quad (7)$$

The variance dimension $(D_\sigma)$is calculated using H as:

D_σ = E + 1-H (8)

where the Embedded Euclidean dimension is E, E=1 will be obtained in a single Euclidean dimension, (i.e.) single independent variable. Hence,

D_σ=2-H (9)

(D_σ) varies between 1 and 2 for a data time series with one measurable parameter. The process will indicate a standard fractional Brownian motion (FBM)[34] if D_σ = 1.5. The process is called mono-fractal, when the process is not showing any multifractal and D_σ = 1. Considering that the sampling interval is fixed as M and the total time of the data series is T, then the points on a log-log plot are represented as follows:

$$(x_i, y_i)=\big(\log(\Delta t_i), \log(\Delta B_i)\big) \quad (10)$$

where ΔB_i is the amplitude of the first and last samples of the interval Δt.

Therefore, the Linear Least Square regression of the log-log plot is done as follows[28,29]:

$$\text{slope}=2H=\frac{K\sum_{i=1}^{i=K} x_i y_i-(\sum_{i=1}^{i=K} x_i)(\sum_{i=1}^{i=K} y_i)}{K\sum_{i=1}^{i=K} x_i 2-\big((\sum_{i=1}^{i=K} x_i)^2\big)} \quad (11)$$

$$\text{slope}=\frac{Cov(X,Y)}{Var(Y)} \quad (12)$$

It is imperative to note here that in order to compute finite sequence of time increments [Δt_1, Δt_2.. Δt_T], the time gap T is segregated into a sub-window sizes of N_w each in a dyadic order.

The algorithm to calculate VFD is illustrated in detail in [28] and [29].

Estimates of the difference in variance using least square Euclidean measures is provided by variance based stationary change detection models, using least square Euclidean measures. This study uses online data to find change in variance, which is used as an indicator of Non-stationary characteristics. The method depends on minimizing the following cost function:

$$J(\tau,\theta)=\frac{1}{n}\sum_{i=1}^{i=n} \frac{\|x_i-\overline{x}\|^2}{var(x_i)}+n_i log[var(x_i)] \quad (13)$$

where:
1. J(.) is the cost function
2. x_i are the samples of the stochastic process x
3. $\overline{x}$ is the first moment/mean of x
4. var(xi) is the second moment or variance of x

5. $\theta = \text{var}(x_i)$
6. n is the data window size

By minimizing the above cost function, the value of θ indicates the change in variance in the data series.

Variance Fractal Dimension Trajectory (VFDT) Computation
1) Set the maximum and minimum scaling levels.
2) Calculate the step size at each level according to dyadic criterion.
3) Loop through each level and
    a. Using equation 9 to calculate points on the log-log plot.
    b. Availing equation 11 to calculate slope.

The forthcoming section deals with computation of VFD for DNS traffic which proves to obey the non-stationary features.

3.2.2 TEST FOR NON-STATIONARITY IN DNS TRAFFIC AND ATTACK DETECTION

The non-stationary of DNS traffic is tested by analysing the variance fractal dimension for 60 minutes with 5 minutes of each sampling. The analysis in Fig. 1 indicates the variance fractal dimension trajectory of the DNS count series of first sample which reveals the outlier from time 2 to 5 minutes. The computation is done availing an adaptive window based on wavelet based change detection algorithm. Moreover, the adaptive window slides with a count of 1. This is performed in order to computer the long range correlation effect on the current window samples. It is also noted that the VFDT calculation falls below the valid range of dimension 1. This occurs because of the availability of outliers in the data series, which rotates the regression of log-log plot from a positive slope into negative one. Hence the log-log plot reveals the negative slope and thus VFD calculation falls below the invalid topological dimension of a line. Further, it is noted that the variance fractal dimension trajectory (VFDT) is showing very rapid differences within valid topological dimension range of [1,2]. This is because of the high correlation effect introduced by the sliding window with a log of 1 sample.

There are a very few such high varying occurrences in Fig. 1, within the range of 2 and 5. However it can also be seen in Fig. 2 that the trend is showing invalid fractal dimensions which is given to the negative slopes of the log-log plot and multi-fractality of the time series.

When Fig. 2 exhibits lot of high frequency variations, it can be deduced that the VFDT calculation window is effected with numerous changes in variances within a window size because of the presence of outliers.

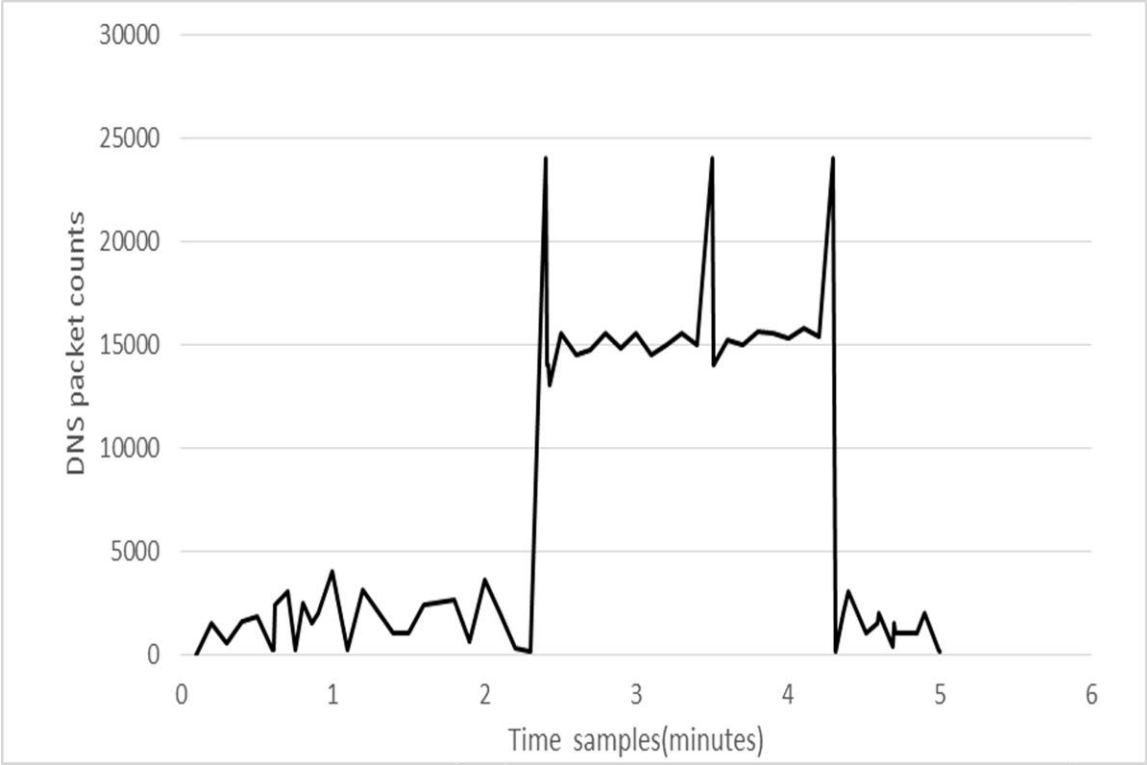

Figure 1. Time series plot of DNS counts

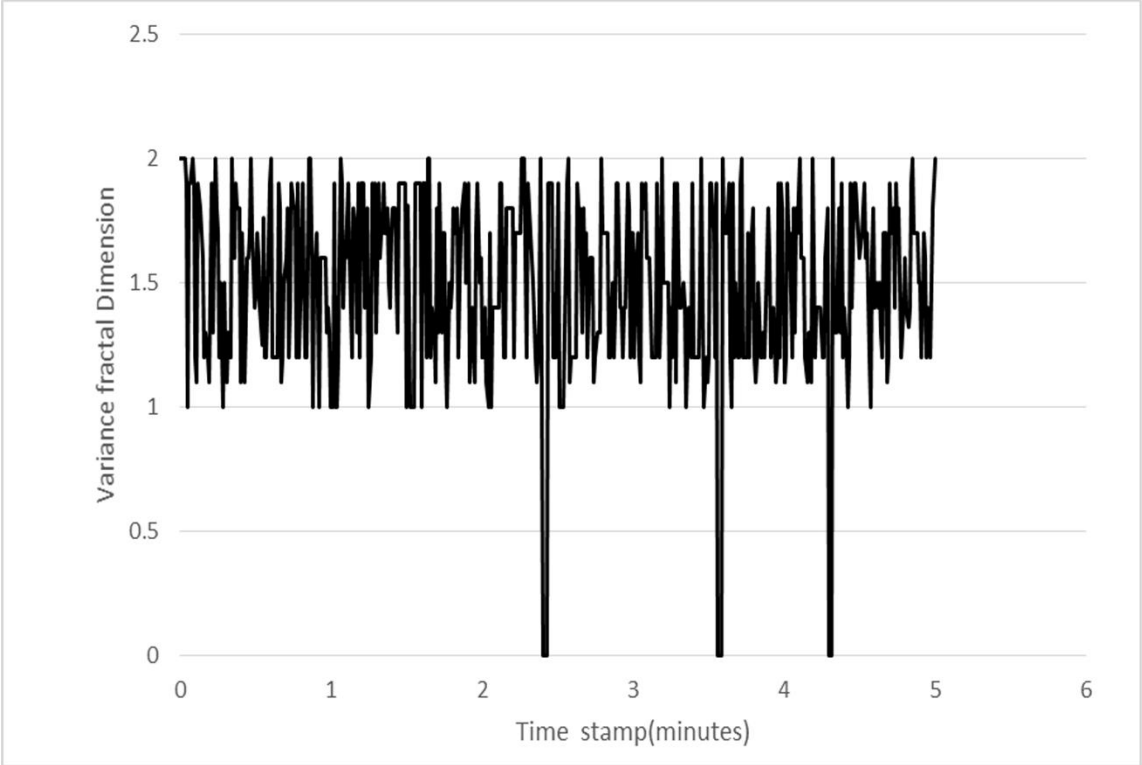

Figure 2. Variance fractal dimension trajectory

This study characterized the DNS time series to extract the varying features of a non-stationary time series. Our study proves the presence of an attacker in the network from 2 to5 minutes. With this evidence, we are implementing this fractal analysis in DNS traffic to detect attack through NXDOMAIN Responses which is discussed in the next section.

3.2.3 FRACTAL ANALYSIS OF NXDOMAIN RESPONSES

This work considered the NXDOMAIN ("the domain does not exist") responses based attack detection in the network. The NXDOMAIN is a type of message received by the client when a request for a domain is sent to the DNS and cannot be resolved to an IP address. An NXDOMAIN error message means that the domain which is requested does not exist.

Flooding of requests for non-existing domain shall be made by attacker by randomly producing subdomain strings and sent to DNS requests. The volume and type of attack might vary, marginally based on what the attacker's planned target is – which can be either authoritative server of a target domain or the recursive DNS server. When the aim is the recursive server, the objective is to consume available resources of the server and poison the cache with NXDOMAIN results. When the aim is the authoritative server of another genuine domain, it causes attack and can influence performance, especially for servers that have insufficient memory resources or have to query the disk to seek these non-existent domain names.

Using multifractal analysis of a network, which analysed different features and characteristics of elements in a network, we are analysing some features which are related to the NXDOMAIN. In the observation, it is noted that 5 components of a DNS request is more relevant to NXDOMAIN, whose results are more proportional to the NXDOMAIN responses. The following are the five components:

a. TTL
b. Refresh
c. Retry
d. Expire
e. Minimum

All these features are monitored and fractal analysis are made, to detect the attack that takes place in the NXDOMAIN RESPONSE. These 5 components of DNS requests that are related to NXDOMAIN response is analysed by gathering the legitimate and abnormal values.

a. TTL is the time (in seconds) period that slave DNS servers should stock the record in a cache. For a normal network it will be 30 sec to 300 sec.

b. Refresh parameter shows how often (in seconds) the slave name servers cross check with the primary name server, to find if any modifications have been made to the site zone file. For a normal network, it will be 1200 to 43200 sec.

c. Retry parameter specifies the time (in seconds) a slave (or secondary) DNS server pauses before retrying a failed zone transfer. This time is normally fewer than the refresh interval. Usual values vary from 180 (three minutes) to 900(15 minutes).

d. Expire parameter specifies the time in seconds a master or slave will delay before considering the data stale if it cannot accomplish the primary name server. For a normal network it will be 1209600 to 2419200 sec.

e. Minimum parameter is defined as the time (in seconds) through which a secondary server should store a negative response. For a regular network, it will be 10800 sec.

It is detected that whenever an attack occurs in the DNS, the above features retains more time than their legitimate time period, which symptoms the movement of attacker. So in this study, these features are observed for all the requests to the DNS server and the requests were counted, which have the values of these features, which crossed their legitimate limits and tabulated. The finest part of the operation is that this abnormality of these values coincides with the abnormality of NXDOMAIN responses. The Box Counting Method (BCM) is considered to be the most efficient way of making the fractal dimensional analysis of all the above mentioned network features which is discussed in the next section.

3.3 A Traffic distribution analysis using BCM based fractal analysis

The traditional information dimension and the BCM is referred[26] to calculate the fractal dimension of complex networks. The traditional BCM algorithm comprises of different number of nodes in a given box size. This method considers different number of nodes in boxes. The probability of information containing the ith box is denoted by $p_i'(l)$ and defined as follows:

$$p_i'(l) = \frac{n_i(l)}{n} \tag{14}$$

where $n_i(l)$ is the count of nodes in the ith box and n is the count of nodes of the complex networks. Information dimension of the network can be defined as

$$I'(l) = -\sum_{i=1}^{N_b} p_i'(l) \ln p_i'(l) \tag{15}$$

d_l' is obtained as follows

$$d_l' = -\lim_{l\to 0} \frac{l'(l)}{\ln(l)} = \lim_{l\to 0} \frac{\sum_{i=1}^{N_b} p_i'(l) ln\, p_i'(l)}{\ln(l)} \tag{16}$$

where d_l' is information dimension of complex networks. Using Eqs. (6) and (8), we have

$$d_l' = \lim_{l \to 0} \frac{\sum_{i=1}^{N_b} \frac{n_i(l)}{n} ln \frac{n_i(l)}{n}}{\ln(l)} \quad (17)$$

Eqs. (15) and (16) are theoretic formulations. The 'l'value is negligible in planar network. In real complex network, the value of 'l' cannot be small,as the distance between nodes will not be less than one. Their relationship ln(l) and $I'(l)$ are linear in a log-log plot[26]. Limited number of box size 'l' is considered. And then, the value of d_l' is provided by the slope of the straight line in the log-log plot.

The BCM based fractal analysis of all the above described features are performed, monitored and controlled using both Flowspec model and Pareto model. We consider every DNS request per regular interval as a box and the total values obtained from the above fractal analysis for all 5 features are plotted on a geometric plane with just enough square boxes all with a certain side length.

If the fractal dimension is greater than a certain threshold't', definitely it can be suspected as a movement of attacker at that particular time, as mentioned in Fig 3. The threshold value't' of the complete fractal dimension, can be detected by finding the average of the entire fractal values over a given time.

$$t = \frac{\sum_{i=1}^{n} d_I'}{n} \quad (18)$$

where n is the number of intervals.

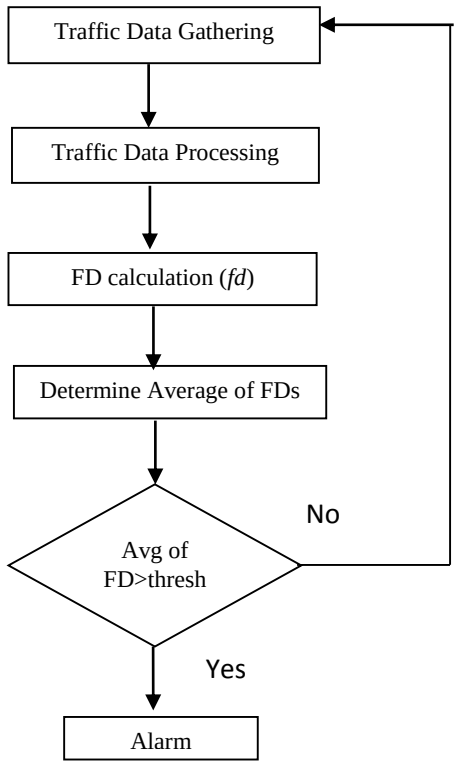

Figure 3. STEPS TO DETECT THE ATTACK IN NETWORK USING FRACTAL DIMENSION ANALYSIS

The attack monitoring and detection, which is stated in the next section is made by noting fractal dimensions of networks in terms of boxes, which crosses the threshold value.

3.4 NETWORK TRAFFIC MONITORING USING BCM TECHNIQUE

The threshold value of all these features of the DNS requests of network traffic are noted and monitored using Flowspec based and PARETO generation based technique to find the burstiness of the traffic. The basic topology of our network is shown in Fig. 4. Network Simulator Version 2 (NS2) has been used to simulate both the network models.

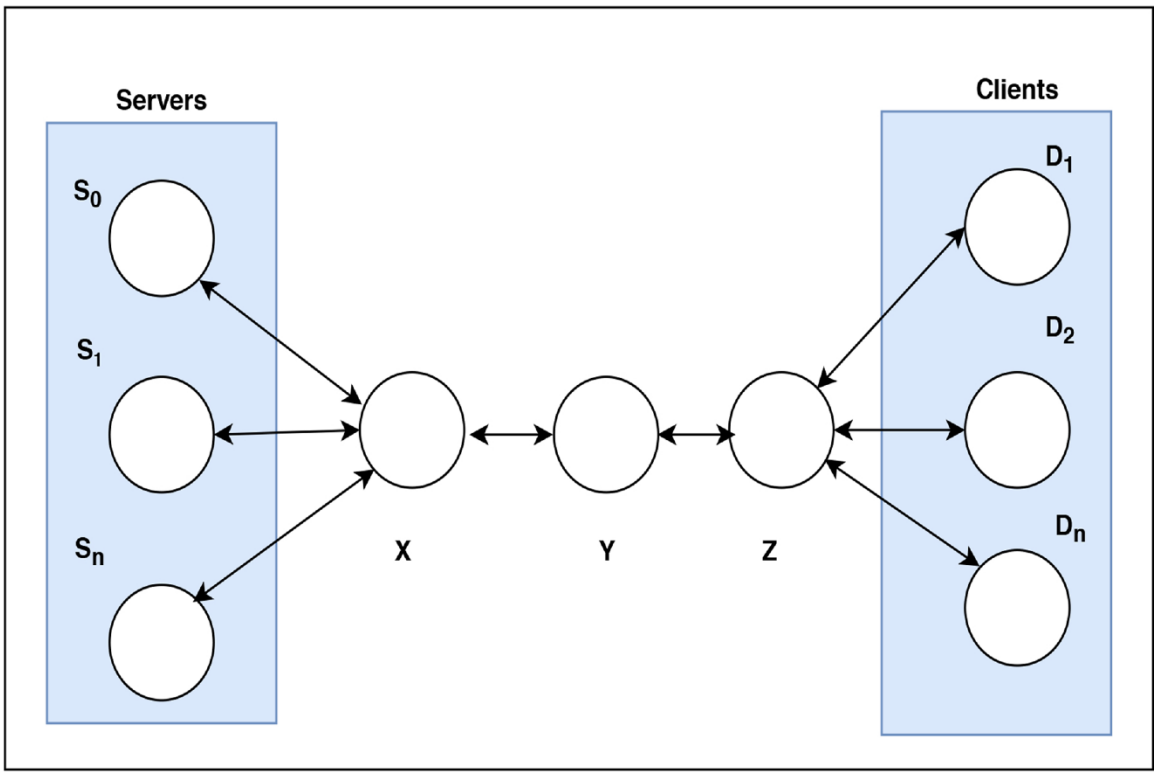

FIGURE 4: TOPOLOGY OF THE TARGET NETWORK

3.4.1 Flowspec based monitoring and controlling of traffic

Flowspec based network monitoring technique has been proposed to monitor the above 5 features to give alarm when it crosses the threshold value. The flowspec based monitoring allows one to rapidly deploy and propagate filtering and policing functionality among a large number of peer routers to mitigate the effects of DNS attack over the network. The link parameter and topology parameters of Flowspec model are displayed in

Table III and IV. In this work, Flowspec based network monitoring performs the following operations.

1. DNS based network traffic monitoring

As mentioned earlier, we are making the fractal analysis of 5 features of the traffic which are related to detection of NXDOMAIN response based DNS attack. The FD is calculated for features, before and after the DNS attack.

2. Rule set generation to handle the DNS attack

Rule sets are created using threshold value in order to handle the attacker at a particular time. At any point of time, if total FD value crosses this threshold value, it can be determined as the movement of attacker. If attacker is detected, then the query will be diverted to resolve.

3. Diversion and Dissemination Mechanisms

Whenever the network is suspected with the flow of attacker, DNS redirection is done to reduce the effect of attack. DNS routing is activated by changing the CNAME and A record, so as to point them to the IP(s) of the mitigation provider. Afterward, DNS initially routes all incoming HTTP/S requests to the provider's scrubbing servers, where malicious requests are dropped and legitimate ones are forwarded. DNS redirection is truly effective in the mitigation of application layer attacks. It also has the benefit of hiding the domain's IP address. This gives some measures of safety against direct-to-IP network layer attacks.

4. Mitigation of Suspicious/Malicious Traffic Flows

It puts down unexpected or unsolicited DNS queries which had not been noticed earlier. These queries may be because of lame delegations, taking a server to resolve, for probing, because of incorrect configurations, for debugging or to simply attack the traffic. In any case, it makes sense to drop them. During non-flood times, one can construct a table of legitimate queries that has been reacted with a positive response. Such a table can be availed to prevent queries under flood that have been unnoticed before. This can ensure that no one gets flooded with drip, phantom-domain and phantom-subdomain DNS attacks.

This can also ensure that authoritative name servers will find queries only for domain names within or below zones they are authoritative for, hence preventing the so-called unwanted DNS queries.

TABLE III. LINK PARAMETERS FOR FLOWSPEC BASED TRAFFIC MODEL

Link	**Bandwidth (Kbps**)	**Latency (ms)**

Z-D_1	12000	25
Z-D_1 to Z-D_n	Unif(50,120)	23
D_i-clients	6500	15
X-Y	10000	25
Y-Z	30000	25
X-S_0 to X-S_n	15000	25
S_0-servers to S_n-servers	20000	Unif (10,100)

TABLE IV: **TOPOLOGY PARAMETERS FOR FLOWSPEC BASED TRAFFIC MODEL**

Parameter	Value
Number of servers per E-H	10
Number of nodes P_i	15
Number of clients per P_i	6
Packet Size	15kbits

3.4.2 PARETO based traffic analysis

The Pareto ON/OFF traffic generator invents traffic according to Pareto ON/OFF distributions. Packets are provided at a fixed rate during ON periods, and packets are not provided during OFF periods. For a Pareto distribution with uniform size packets, both OFF and ON periods are considered. The research in this study involved 10 source nodes, 6 destination nodes and 15 middle nodes as found in Table IV. Pareto parameter α is defined[8] as 1.2.

This model is taken into account to control the FD of our 5 features in the network. Equation 7 has been used to calculate the threshold for PARETO. If the FD value goes beyond the provided threshold level at any interval, it will be detected as attacker.

It has also been proved here that this model is inefficient than the proposed Flowspec model, as this model cannot monitor the flow of packets for a long time. So it will be

inaccurate if the data flow is huge and the duration to monitor is high. The link parameter and topology parameters of Pareto Model are displayed in Table V and VI.

TABLE V. LINK PARAMETERS FOR PARETO ON/OFF MODEL

Link	**Bandwidth (Kbps)**	**Latency (ms)**
Z-D_1	12000	25
Z-D_1 to Z-D_n	Unif(50,120)	23
D_i-clients	6500	15
X-Y	10000	25
Y-Z	30000	25
X-S_0 to X-S_n	15000	25
S_0-servers to S_n-servers	20000	20

TABLE VI: **TOPOLOGY PARAMETERS FOR PARETO ON/OFF MODEL**

Parameter	Value
Number of servers per E-H	10
Number of nodes P_i	15
Number of clients per P_i	6
Mean ON Time	0.5sec
Mean OFF Time	0.5sec
Pareto Parameter α	1.2
Packet Size	15kbits

The above configuration is adapted and the simulation results of both Pareto and Flowspec model are discussed and proved Flowspec as best in the next section.

3.4.3 SIMULATION OBSERVATIONS

The fractal dimension of the network has been analysed using 4 datasets, NXDOMAIN Response monitoring using flowspec before attack, after attack, NXDOMAIN Response monitoring using Pareto before attack and after attack. Each datasets have the statistics of total number of NXDOMAIN Responses, attacked NXDOMAIN Responses, number of abnormal features and FD values of 4 features of the DNS network and the threshold value.

The simulation is made for 1 hour with 5 minutes of sampling. The reason for this hour lasting analysis is to maintain granularity of the analysis. Also it is found[33] that a longer analysis period is helpful to make a better understanding of malware behaviour of a network. It is planned to make a day long analyse of DNS attack in future.

1. TOTAL NUMBER OF NXDOMAIN RESPONSES

   It is noted to find the number of responses that cannot retrieve any domain from the DNS. In our work, in initial 5 minutes, the total NXDOMAIN responses arrived are 3350.

2. Attacked NXDOMAIN Responses

   It is noted how many NXDOMAIN responses are caused by the attacker. In this work, at initial 5 minutes, total attacked NXDOMAIN responses arrived are 1340.

3. Number of abnormal features

   Here, whenever there is a request in the network, the features that exceed their legitimate values are noted. For example, on considering TTL feature, the TTL's legitimate time is between 30 sec to 300 sec. if any requests arrives with TTL, exceeds this legitimate values, it will be counted and noted. In our work, at initial 5 minutes, total TTL features that exceeds the legitimate value are 1295.

4. FD values of a feature

   The fractal dimension of a feature using BCM is done using the formula stated in eqn. 17

$$d'_l = \lim_{l \to 0} \frac{\sum_{i=1}^{N_b} \frac{n_i(l)}{n} \, ln \frac{n_i(l)}{n}}{\ln(l)} \tag{19}$$

i.e. the FD of the probability of attack due to all the 4 features can be drawn by using the summation of FD of probability of attack by individual features and the total attack in the NXDOMAIN responses. Thus, the FD of total features of our network is calculated.

5. Threshold value

   The threshold value can be calculated by the standard deviation of the total FDs of all samples.

a. Dataset 1 using fractal analysis of NXDOMAIN using flowspec BEFORE ATTACK

We have analysed the NXDOMAIN Response using Flowspec before attack in TABLE VII. The total FD should be noted to check whether it crossed the threshold value at any period of time. There are no such values found, which means the network is safe and no attack is found.

TABLE VII. FRACTAL ANALYSIS OF NXDOMAIN USING FLOWSPEC BEFORE ATTACK

Time slice (min)	# of NXDOMAIN	# of attacked NXDOMAIN	# of abnormal TTL	FD of TTLcount (1)	# of abnormal Refresh	FD of refresh count (2)	# of abnormal retry	FD of retry count (3)
5	3350	1340	1295	0.01	1259	0.04	1341	0.02
10	15375	6150	5788	0.01	5752	0.03	5642	0.03
15	31500	12600	12515	0.00	12479	0.01	12561	0.01
20	57000	22800	22115	0.01	22079	0.01	22231	0.01
30	65075	26030	25915	0.00	25879	0.01	26161	0.00
35	135125	54050	53815	0.00	53779	0.00	54261	0.00
40	153000	61200	60135	0.00	60371	0.01	60381	0.01
45	176125	70450	69015	0.01	68979	0.01	69261	0.01
50	189050	75620	74535	0.00	74499	0.01	74781	0.00
60	210375	84150	4015	0.04	4251	0.04	4261	0.04

Time slice (min)	expire	FD of expire (4)	minimum	FD of minimum (5)	total FD (1+2+3+4+ 5)	Threshold Value
5	1353	0.02	1335	0.02	0.11	0.059
10	5846	0.03	5828	0.03	0.13	
15	12573	0.01	12555	0.01	0.04	
20	22173	0.01	22155	0.01	0.05	
30	25973	0.00	25955	0.00	0.02	
35	53873	0.00	53855	0.00	0.01	

40	60193	0.01	60175	0.01	0.03
45	69073	0.01	69055	0.01	0.03
50	74593	0.00	74575	0.00	0.02
60	4073	0.04	4055	0.04	0.18

b. Dataset 2 using fractal analysis of NXDOMAIN using flowspec AFTER ATTACKED

On observing TABLE VIII, it is obvious that the total FD exceeds the threshold value from 40th minute to 1 hour, which shows that there is a movement of attacker in the network. In the below calculation, 0.0257 is the threshold value which is the SD of total FDs of all samples.

TABLE VIII. FRACTAL ANALYSIS OF NXDOMAIN USING FLOWSPEC AFTER ATTACK

Time slice (min)	# of NXDO MAIN	attack in NXDO MAIN	Ttl	FD of ttl (1)	refresh	FD of refresh (2)	retry	FD of retry (3)
5	3350	1340	1350	0.02	1350	0.022	1402	0.014
10	15375	6150	6500	0.00	6500	0.004	6400	0.007
15	31500	12600	12450	0.01	13010	0.001	12890	0.003
20	57000	22800	23100	0.00	22150	0.012	23003	0.003
30	65075	26030	25900	0.01	26010	0.004	26002	0.004
35	135125	54050	52100	0.01	51200	0.015	51200	0.015
40	153000	61200	58200	0.01	58200	0.014	61200	0.002
45	176125	70450	69999	0.00	69521	0.005	70344	0.002
50	189050	75620	75200	0.00	75001	0.003	75120	0.003
60	210375	84150	82005	0.01	8320	0.057	82520	0.006

Time slice (min)	expire	FD of expire (4)	minimum	FD of minimum (5)	total FD (1+2+	Thres hold Value

					3+4+5)	
5	1380	0.02	1400	0.015	0.091	0.0257
10	6500	0.00	6500	0.004	0.022	
15	12985	0.00	12860	0.004	0.020	
20	23010	0.00	23100	0.002	0.020	
30	26040	0.00	26350	0.001	0.020	
35	51000	0.02	51200	0.015	0.020	
40	61062	0.00	61510	0.001	0.032	
45	68500	0.01	68400	0.009	0.027	
50	73000	0.01	73520	0.008	0.028	
60	83900	0.00	84250	0.001	0.074	

Dataset 3 using fractal analysis of NXDOMAIN using PARETO ON/OFF MODEL BEFORE ATTACK

On observing TABLE IX, it is obvious that there is no total FD that exceeds the threshold value, which shows that there is no movement of attacker in the network and the network is safe.

Table IX. Fractal analysis of NXDOMAIN using PARETO ON/OFF MODEL BEFORE ATTACK

Time slice (min)	# of NXDOMAIN	attack in NXDOMAIN	Ttl	FD of ttl (1)	refresh	FD of refresh (2)	Retry	FD of retry (3)
5	3350	1340	1280	0.011	1300	0.007	1326	0.003
10	15375	6150	6008	0.006	5672	0.019	5800	0.014
15	31500	12600	12500	0.002	12450	0.003	12546	0.001
20	57000	22800	22100	0.008	22064	0.008	22600	0.002
30	65075	26030	25900	0.001	25864	0.002	25946	0.001
35	135125	54050	53800	0.001	53764	0.001	53846	0.001

40	153000	61200	60120	0.004	60084	0.005	60166	0.004
45	176125	70450	69000	0.005	68964	0.005	69046	0.005
50	189050	75620	74520	0.004	74484	0.004	74566	0.003
60	210375	84150	4000	0.036	2600	0.027	2600	0.027

Time slice (min)	expire	FD of expire (4)	minimum	FD of minimum (5)	total FD (1+2+3+4+5)	Threshold Value
5	1338	0.000	1313	0.005	0.03	0.04469
10	6150	0.000	6120	0.001	0.043	
15	12558	0.001	12533	0.001	0.01	
20	22158	0.007	22133	0.007	0.03	
30	25958	0.001	25933	0.001	0.01	
35	53858	0.001	53833	0.001	0.01	
40	60178	0.004	60153	0.004	0.02	
45	69058	0.005	69033	0.005	0.03	
50	74578	0.003	74553	0.004	0.02	
60	3100	0.030	4033	0.036	0.16	

c. **Dataset 4 using fractal analysis of NXDOMAIN using PARETO ON/OFF MODEL UNDER ATTACK**

In Table X, we can observe that in PARETO model, the analysis of attack is less efficient when the duration of monitoring the traffic is high. It detected only one attacked traffic during the initial level of request flow, though attacker is found from 40 to 60th minutes of the network traffic which is found by Flowspec model.

Table X. Fractal analysis of NXDOMAIN using PARETO ON/OFF MODEL UNDER ATTACK

Time slice (min)	# of NXDOMAIN	attack in NXDOMAIN	Ttl	FD of ttl (1)	refresh	FD of refresh (2)	retry	FD of retry (3)
5	3350	1340	850	0.02	920	0.005	910	0.008

10	15375	8250	8190	0.00	8100	0.005	8110	0.004
15	31500	12600	12580	0.00	12500	0.002	12550	0.001
20	57000	22800	22100	0.01	22712	0.001	22780	0.001
30	65075	26030	26014	0.00	26000	0.001	26004	0.001
35	135125	54050	54001	0.00	54000	0.001	54008	0.001
40	153000	61200	61050	0.00	61100	0.001	61005	0.001
45	176125	70450	67450	0.01	70000	0.002	70210	0.001
50	189050	75620	75520	0.00	75420	0.001	75110	0.002
60	210375	84150	84015	0.00	84102	0.000	83456	0.002

Time slice (min)	expire	FD of expire (4)	minimum	FD of minimum (5)	total FD (1+2+3+4+5)	Threshold Value
5	850	0.023	920	0.005	0.064	0.01795
10	8120	0.004	8200	0.002	0.016	
15	12580	0.000	12520	0.002	0.005	
20	22750	0.001	22780	0.000	0.010	
30	26009	0.000	25030	0.009	0.010	
35	54002	0.000	53900	0.001	0.002	
40	61102	0.000	60251	0.004	0.006	
45	70301	0.001	70310	0.000	0.014	
50	75142	0.002	75020	0.002	0.006	
60	82450	0.005	83520	0.002	0.009	

4. RESULT AND DISCUSSION

It is observed from this research, that fractal analysis of five features of NXDOMAIN responses can be used to detect DNS attacks, as the fractal analysis of characteristics of any features of network can bring more details. It is also gathered that BCM based FD analysis of these features helps to detect the attack more accurately, because the BCM helps in finding the self-similarity of the feature and any contradiction in the values can be instantly determined. As it has been analysed over time, one can make a micro level analysis of entry and exit point of the intruder in the network. In order to show the efficiency of the work, it has compared the proposed Flowspec model with Pareto model.

We have calculated the fractal dimensions at different time scales and the average fractal dimension by (18). To measure the changes in flow, both models sampled the flow for every 5 minutes. Fig.5 shows that the fractal dimension increases at 40 to 60 minutes. It also shows that the fractal dimensions exhibits only minor obvious trend.

In Fig. 5, the points with black shades are the positions which exceed the thresholds which show the attack in the network. The flowspec model shows larger differences in the FD value, when compared to the traffic distribution. The Pareto model cannot show variation in the FD, though there is an attacker in the network.

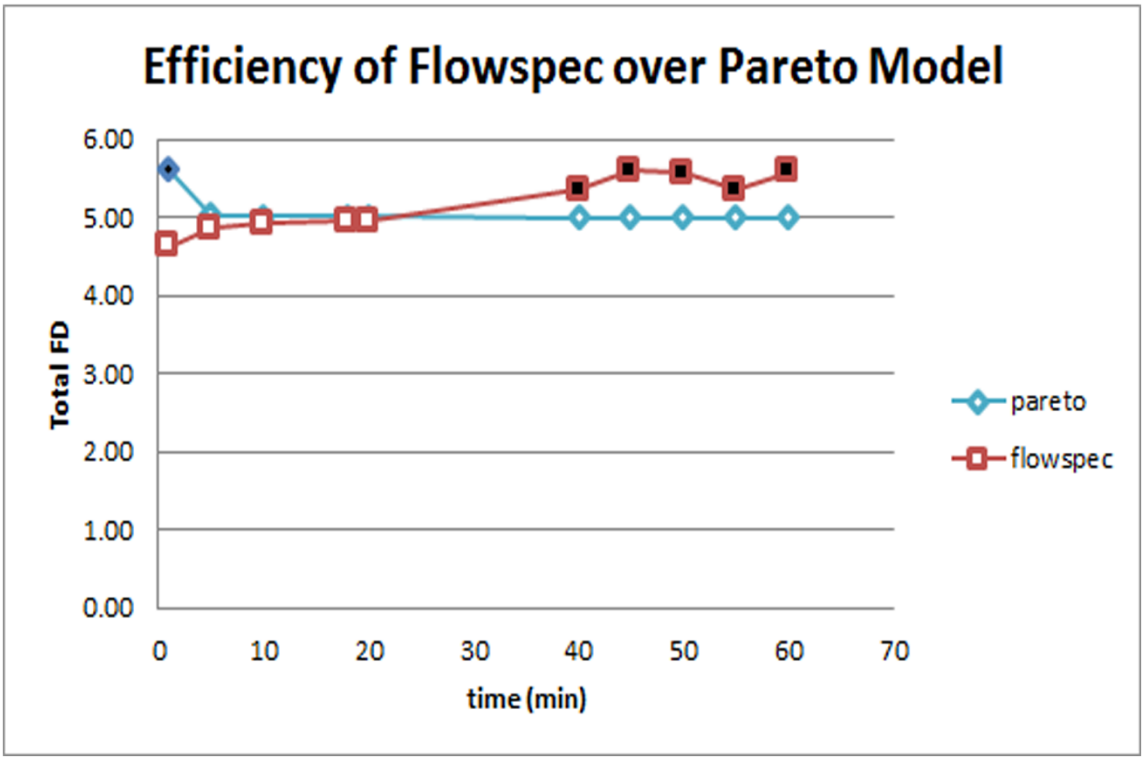

Figure 5. Efficiency of Flowspec model over Pareto Model

The major advantage in Flowspec model compared to PARETO model is that, it detects the DNS based attack and redirects the query from attacker. The DNS-based redirection has several advantages. The important one is that, it attains transparency without losing scalability. It is transparent because the clients are obliged to use the addresses provided by the authoritative DNS server, and cannot establish whether these addresses belong to the home machine of the service or to any of its replicas.

Another vital advantage of using DNS to redirect clients, is that it is a natural way of informing the clients about the service addresses. Once this infrastructure is made to work, both efficiency and availability of the redirector considerably increases. The important advantage of DNS redirecting is that it allows multiple replica addresses to be returned, enabling the client to choose one from them.

The last advantage of DNS-based redirection is its good maintainability. Deployment of the complete redirection mechanism boils down to launch a single modified DNS server and subsequently delegating a service domain to this server. From this moment, this server is responsible for answering requests for the service address. No other modification of the DNS infrastructure is necessary. With the assistance of this method, attack can be identified and its influence can be mitigated at the earliest which is stated in the next section.

4.1 Alarm timing

Fig.6 and 7 indicates alarm points to detect the attack in the initial stage. The alarm in this study was set at 4minutes from observation. Fig. 6 and 7 also shows the traffic distribution after taking alarm procedure for 1 minute time scales. These results prove that the burstiness decreased after the alarm. It was also noted that the variations of fractal dimensions after the alarm procedure decreased in the Flowspec based traffic model when compared to Pareto model.

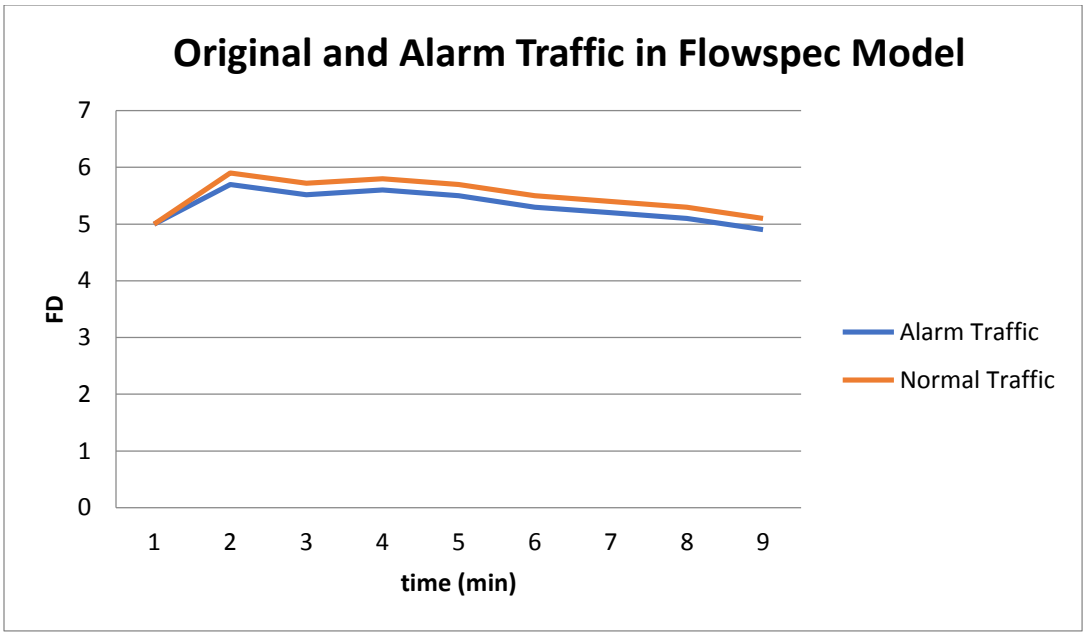

Figure 6. Comparison of FD of Normal and Alarm traffic for Flowspec model

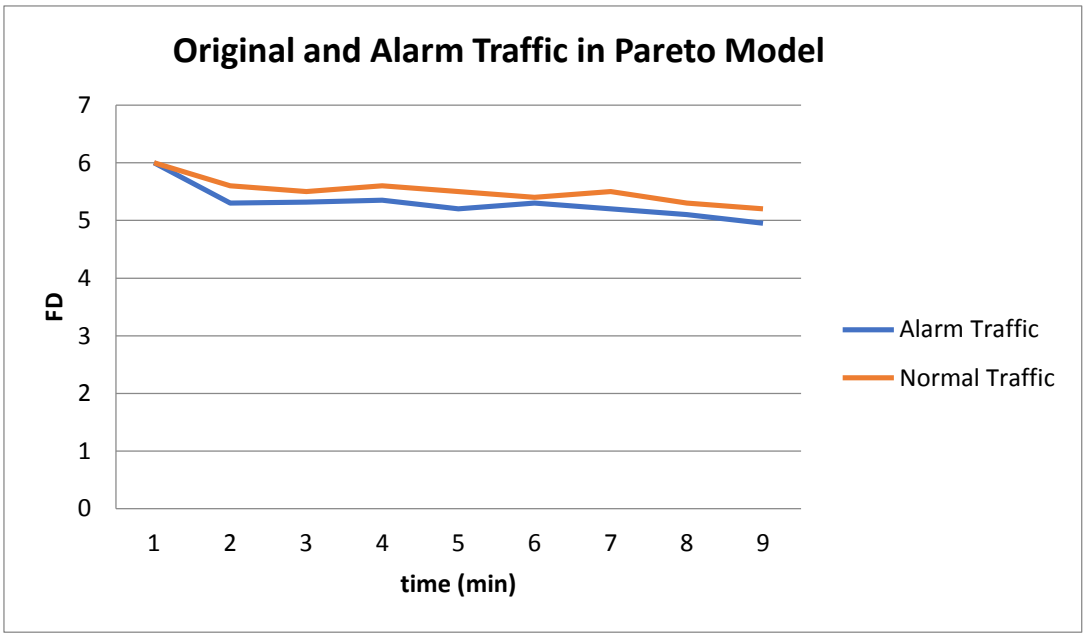

Figure 7.Comparison of FD of Normal and Alarm traffic for Pareto model

5. CONCLUSION

This work aimed at making a more in-depth analysis of characteristics of network, using fractal dimensions to detect the attack much earlier, than any other techniques. Thus, we equipped selectively 5 features for the detection. We utilized Flowspec model to handle the attack, once it is detected. We chose NXDOMAIN responses based attack detection because it is the most common attack in the DNS traffic. This study shows that even when

a slightly distinct pattern occurs in the network, our model can easily predict and take action accordingly.

6. REFERENCES

1. Stephanie Rendón de la Torre, JaanKalda, Robert Kitt, JüriEngelbrecht, "Fractal and multifractal analysis of complex networks: Estonian network of payments" ,School of Science, Department of Cybernetics, Tallinn University of Technology, Akadeemia tee 21, 12618, Tallinn, ESTONIA, Swedbank AS, Liivalaia 12, 15038, Tallinn, ESTONIA
2. C. Song, L.K. Gallos, S. Havlin, H.A. Makse, J. Stat. Mech.: Theor. Exp, 3, 4673 (2007), http://dx.doi.org/10.1088/1742-5468/2007/03/P03006.
3. Pengkui Luo, Ruben Torres, Zhi-Li Zhang, SabyasachiSaha, Sung-Ju Lee, "Leveraging Client-Side DNS Failure Patterns to Identify Malicious Behaviors", University of Minnesota, Symantec Corp.
4. Hsiao-Wen Tin, Shao-Wei Leu, Shun-Hsyung Chang and Gene Eu Jan, "Measurement of Flow Burstiness by Fractal Technique", 2014.
5. Ziad El Jamous, SohraabSoltani, YalinSagduyu, and Jason Li, "RADAR: An Automated System for Near Real-Time Detection and Diversion of Malicious Network Traffic", Intelligent Automation, Inc., Rockville, MD, USA
6. [1]Alakiri Harrison, [2]O, Okolie Cletus. C, [3]Oladeji Florence. A, [4]Okikiola Folasade.M, [5]Benjamin Benjamin. C, "The Desirability of Pareto Distribution for Modeling Modern Internet Traffic Characteristics", [1]Department Computer Technology, Yaba College of Technology, Lagos, Nigeria, [2]University of Lagos, Nigeria, [3]University of Lagos, Nigeria, Yaba College of Technology, Lagos, Nigeria, [5]University of Jos, Nigeria.
7. Muhammad Salman Khan, Ken Ferens, and WitoldKinsner, "A Cognitive Multifractal Approach to Characterize Complexity of Non-Stationary and Malicious DNS Data Traffic Using Adaptive Sliding Window", Dept. of Electrical and Computer Engineering, University of Manitoba, Winnipeg, ME, Canada.
8. ShriramSarvotham, Rudolf Riedi, and Richard Baraniuk, "Connection-level Analysis and Modeling of Network Traffic", ACM SIGCOMM Internet Measurement Workshop, pp. 99-103, Nov. 2001.
9. Roland Molontay, "Networks and fractals", Budapest University of Technology and Economics", Institute of Mathematics, 2013.

10. VisheshGuptay, Cayman Simpson, BharadRaghavan, “An Investigation of Network Fractality”, December 2014.
11. Muhammad Salman Khan, Sana Siddiqui, Ken Ferens, and WitoldKinsner, “Spectral Fractal Dimension Trajectory to Measure Cognitive Complexity of Malicious DNS Traffic”, Dept. of Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB, Canada, 2016.
12. M. Aiello, M. Mongelli and G. Papaleo, “DNS tunneling detection through statistical fingerprints of protocol messages and machine learning”, INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS, *Int. J. Commun. Syst.* 2015; **28**:1987–2002, published online 28 July 2014 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/dac.2836.
13. R. Murdzek, “The box-counting method in the large scale structure of the universe”, Physics Department, Al. I. Cuza University.
14. Jones, C, Jelinek, “HF: Wavelet Packet Fractal Analysis of Neuronal Morphology”, Meth 2001
15. D. Schertzer, S. Lovejoy, J. Appl. Meteor., 36, 1296 (1997)
16. V.V. Anh, K.S. Lau and Z.G. Yu. Phys. Rev. E 66, 031910, (2002)
17. Forest, SR, Witten, “TA: Long-range correlations in smoke particle aggregates”.
18. J Physics A Math and Gen 1979; 12:L109-L117.
19. Z.G. Yu, V.V. Anh and K.S. Lau, Phys. Rev. E 68, 021913 (2003)
20. L.Q. Zhou, Z.G. Yu, J.Q. Deng, V.V. Anh and S.C. Long, J. Theor. Biol. 232, 559-567, (2005)
21. Droppo, IG, Flannigan, DT, Leppard, GG, Liss, “SN: Microbial floc stabilization and preparation for structural analyses by correlative microscopy”. Water Science and Technology, 1996; 34 (5-6), 155162.
22. Alisha Cecil, “A Summary of Network Traffic Monitoring and Analysis Techniques”, https://www.cse.wustl.edu/~jain/cse567-06/net_monitoring.htm.
23. Roberto Perdisci, Igino Corona, and Giorgio Giacinto, Senior Member, IEEE, “Early Detection of Malicious Flux Networks via Large-Scale Passive DNS Traffic Analysis”, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 9, NO. 5, SEPTEMBER/OCTOBER 2012.

24. ZOU Futai, ZHANG Siyu, RAO Weixiong, “Hybrid Detection and Tracking of Fast-Flux Botnet on Domain Name System Traffic”, China Communication, November 2013.
25. Stephanie Rendón de la Torre, JaanKalda, Robert Kitt , JüriEngelbrecht, “Fractal and multifractal analysis of complex networks: Estonian network of payments”, School of Science, Department of Cybernetics, Tallinn University of Technology, Akadeemia tee 21, 12618, Tallinn, ESTONIA
26. DaijunWei[a,b], BoWei[a], YongHu[c], Haixin Zhang, Yong Deng[a,d], “A new information dimension of complex networks”, [a]School of Computer and Information Science, Southwest University, China, [b]School of Science, Hubei University for Nationalities, China, [c]Institute of Business Intelligence and Knowledge Discovery, Guangdong University of Foreign Studies, China, [d]School of Engineering, Vanderbilt University, USA.
27. M. ZubairShafiq, LushengJi, Alex X. Liu, Jia Wang, “Characterizing and Modeling Internet Traffic Dynamics of Cellular Devices”, Department of Computer Science and Engineering, Michigan State University, USA
28. W. Kinsner and W. Grieder, "Amplification of Signal Features Using Variance Fractal Dimension Trajectory," International Journal of Cognitive Informatics and Natural intelligence, vol. 4, no. 4, pp. 1-17, Oct. 2010.
29. A. Phinyomark, P. Phukpattaranont and C. Limsakul, "Applications of Variance Fractal Dimension: A Survey", Complex Geometry, Patterns, and Scaling in Nature and Society, vol. 22, no. 1, 2014.
30. Yue Zhang, Ning Huang, Liudong Xing, “A novel flux-fluctuation law for network with self-similar traffic”, [1]School of Reliability and Systems Engineering, Beihang University, China, [2]Science and Technology on Reliability and Environmental Engineering Laboratory, PR China, [3]Department of Electrical and Computer Engineering, University of Massachusetts, USA
31. https://www.bgp4.as/tools
32. [1]Jin-Long Liu, [1,2]Zu-Guo Yu and [2]Vo Anh, “Determination of multifractal dimensions of complex networks by means of the sandbox algorithm”, [1]Xiangtan

University, Xiangtan, Hunan 411105, China, [2]Queensland University of Technology, China.

33. [1,2]Christian Rossow, [1,3]Christian J. Dietrich, [2]Herbert Bos, [2]Lorenzo Cavallaro, [2]Maarten van Steen, [3]Felix C. Freiling, [1]Norbert Pohlmann, "Sandnet: Network Traffic Analysis of Malicious Software", [1]Institute for Internet Security, University of Applied Sciences Gelsenkirchen, Germany, [2]Department of Computer Science, VU University Amsterdam, The Netherlands, [3]Department of Computer Science, University of Erlangen, Germany
34. P. Zhang, "Fractal dimension estimation of fractional Brownian motion" in IEEE proceedings of Southeastcon, 1990.