ISSN:0975-9646
Gamidi Pavan Babu et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (3) , 2011, 1197-1201

# Anomaly Detection On User Browsing Behaviors Using Hidden Semi-Markov Model

Gamidi Pavan babu[1], Jayavani.V[2,] C.P.V.N.J. Mohan Rao[3]
[1]*Jawaharlal Nehru Technological University Kakinada*
[2,3]*Computer Science & Engineering, Avanthi Institute of Engineering Technology, Tamaram, Vishakhapatnam, Andhrapradesh, India*

***Abstract:*** **- CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) is widespread security on the World Wide Web to prevent Application layer DDOS attacks and abusing online services. Generating of CAPTCHA statically to WebPages in the website might annoy users and introduce additional service delays for legitimate users."Puzzle" also has the effect of denying web crawlers access to the site. That causes denying the search engines indexing the content of the websites. In this paper, we introduce a new scheme to achieve early attack detection and countering the attack by generating CAPTCHA dynamically for abnormal HTTP requests of the website, on a popular event generation in the corresponding WebPages. A Hidden semi-Markov model used to describe the browsing behaviors of different users. A novel P-algorithm is introduced for serving the CAPTCHA to attacker to a specific WebPages, based on entropy of the users' HTTP request rate, page viewing time, Page request sequence. If users are requesting for service from proxy servers, any user fall in to the deviated behavior category we generate CAPTCHA dynamically instead of dropping all the requests from behind the proxy server.**

**Index Terms:-CAPTCHA, App.DDoS attacker, HTTP request rate, entropy, Hidden semi Markov Model, P-Algorithm.**

## I.INTRODUCTION

IN November of 2010, a series of massive Distributed Denial-of-Service (DDoS) attacks incapacitated several high-visibility Internet e-commerce sites, social networking websites including Wiki Leaks, Twitter, Facebook, and Orkut. Distributed Denial of Service (DDoS) attacks are becoming an increasingly frequent disturbance of the global Internet. DDOS attacks limit and block legitimate users' access by overwhelming victim servers' resources. DDoS attack directs hundreds or even thousands of zombies [1] [2] affected innocent hosts against a single target. From these innocent hosts enormous HTTP requests impact website performance. DDoS attacks are carried out at the network layer, such as TCP flooding, ICMP ECHO flooding, and UDP flooding, which are called Network Layer DDoS attacks. The main aim of Net-DDoS is the disruption of services by attempting to limit access to a machine, consuming bandwidth and creating unwanted traffic in subnet. App.DDoS attacks are carried out at the application layer, such as HTTP flooding. By requesting enormous HTTP GET requests to the victim server, will bring the victim server down.

The Application layer DDoS attacks can target one particular page or part of the website or whole website. Whenever the flash crowd event [3] [4] occurred in the website, attacker specifically targets that website or flash crowd event occurred webpage. To the best of our knowledge, few existing papers focus on the detection and counter the App. DDoS attacks. History based IP filtering [5] and Monitoring the App. DDoS attack using Hidden Semi Markov Models [6] [7], Traffic load balancing at switching elements in Internet.

In this paper, we introduce a stream to capture the statistical information of page accessing by the users and calculating the entropy of page access using vector quantization. Our contribution in this paper 1) Capture the user browsing behaviors to monitor the App. DDoS. 2) Based on previous work we use Vector Quantization [8] [9] [10] algorithm to find the Entropy value.3) we apply the entropy value for each user to find the anomaly behavior. 4) Based on deviated entropy value of users, we dynamically generate CAPTCHA [11] [12] to the WebPages of particular requests from users.

The remainder of this paper is organized as follows. In Section II, we put our ideas within the context of prior and ongoing research related to DDoS attacks detection and counters. In Section III, we discuss the App. DDoS attack and detail their properties. In Section IV, we explain our technique to detect the App-DDoS attacks. In Section V, based on the hidden semi-Markov model (HsMM) a new model introduced for describing the browsing behavior of web users and detection of the App-DDoS attacks. In Section VI, we explain the P-algorithm for dynamic generation of CAPTCHA to the web pages of deviated users. In Section VI, we concluded our paper.

## II. RELATED WORK

A DDoS attack is a large-scale, coordinated attack on the availability of services of a victim system or network resource, launched indirectly through many compromised computers on the Internet. The services under attack are those of the "primary victim", while the compromised systems used to launch the attack are often called the "secondary victims."

Most DDoS-related research has focused on the Network layer. These mechanisms attempt to detect attacks by analyzing specific features, e.g., arrival rate or header information. Other statistical approaches for detection of DDoS attacks include IP addresses and time-to-live (TTL) values, request rate. The Transport layer is another main layer for detecting DDoS attack. For example, ICMP, UDP, and TCP packet statistical abnormalities to specific DDoS attack. However, more work has been done on the detection of App-DDoS attacks because there were few

such attacks in the past. Other researchers combated the App-DDoS attacks by static "CAPTCHA," and "puzzles", Honey pots [5]. "CAPTCHA and "puzzles" mainly concentrate on website access with registration and initial login and other important WebPages. These techniques can defense the attacker at particular services, and if any abnormalities found services permanently stopped for those users. If all users are requesting for services behind proxy, dropping the traffic from that proxy will cause service termination to legitimate users behind that proxy. Therefore the conventional DDoS prevention schemes designed at initial entry point of the website and proxy. So the schemes developed by early researchers become ineffective.

## III. Application Layer DDoS Attack

The goal of the attacker is to exhaust one or more server resources so that legitimate clients experience high delays or lower throughputs thereby reducing or eliminating the capacity of the servers to its intended clients. Application layer attacks categorized into three classes:1) *request flooding attacks* that send application-layer requests enormously than that of normal sessions; 2) *asymmetric attacks* that send high workload request types; example drawing large images or tend to process the large queries in to the database and 3) *repeated one-shot attacks* in which the attacker spreads its workload across multiple sessions instead of multiple requests per session, and initiates sessions at rates higher than normal. For example, an HTTP flood can stress server resources as an asymmetric attack if the attack sessions send requests involving high-computation database queries.

Besides the flooding attack pattern, App-DDoS attacks may focus on exhausting the server resources such as Sockets, CPU, memory, disk/database bandwidth, and I/O bandwidth.

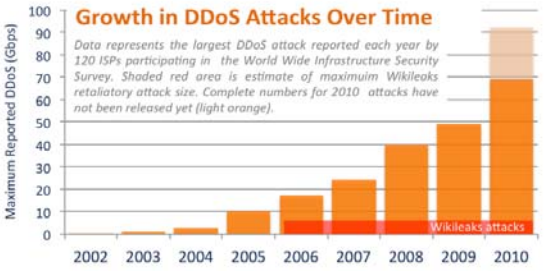

Fig: 1. DDoS attack Structure

Thus, the App-DDoS attacks may cause catastrophic problems in the high-speed Internet. The first characteristic of App-DDoS attacks is that the application-layer requests originating from the compromised computers on the Internet are indistinguishable from those generated by legitimate users. App-DDoS attacks can be mounted with legitimate requests from legitimately connected network machines. Usually, App-DDoS attacks utilize the weakness enabled by the standard practice of opening services such as HTTP and HTTPS (TCP port 80 and 443) through most firewalls to launch the attack. Many protocols and applications, both legitimate and illegitimate, can use these openings to tunnel through firewalls by connecting over a standard TCP port 80. Legitimate users may request services to the website, but these clients are unable to complete their transactions, website will be put busy giving responses to the Zombie processes. App.DDoS attacks can be identified by the user browsing behaviors, the user browsing behavior elements are HTTP request rate, page viewing time, page requesting sequence.

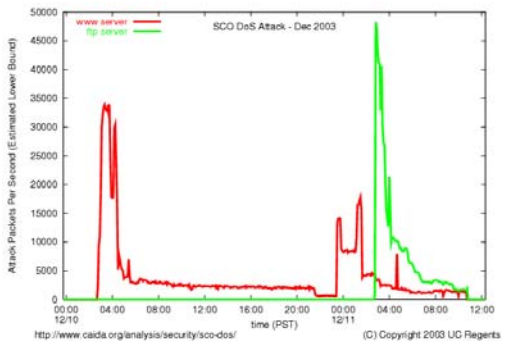
The static CAPTCHA, Puzzles generation for some populared WebPages defense App. DDOS attacks. Those schemes are not effective for DDoS attack detection because they may annoy legitimate users and introduce additional service delays. They may deny search engines access and legitimate user access behind proxy to the Website. App.DDoS attacks are more dangerous than Network based DDoS attacks, it would not need more bandwidth to attack victim server in the Internet. Therefore, the conventional DDoS detection and countering schemes become ineffective.

## IV. Anomaly Detection

Web user behavior is mainly influenced by the structure of Website (e.g., the Web documents and hyperlink) and the way users access web pages. In this paper we are mainly concentrated on App.DDoS attacks to websites; our prevention method considers the anomaly behavior of the variety of users. In this paper we investigate the browsing behaviors of the variety of users based on the HTTP requests and the page viewing time when popular event occurred in the website. The normal user behavior and deviated behavior will be identified by the log information. The below figure describes the growth rate of attack over time.




*(a)*




*(b)*

Fig: 2. (a) DDoS attack growth rate over time. (b) DDoS attack on SCO [13].

These results are significant to our work. They show that the users' access behavior profile can be used to detect the abnormal varieties of users' browsing process during the popular event. Since the document popularity has been widely used to characterize the user behavior and improve the performance of Web server and reduce the internet traffic. We will extend it to our detection in the rest of this paper.

A. Data Capturing

Users browsing behavior can be identified with elements HTTP request rate, page viewing time. Normal browsing behavior profile is first importance to our project. The profile is the structure that characterizes the browsing behavior with respect to the WebPages in terms of statistical metrics and models of observed behavior. To create the normal usage profile of some spatial-temporal network traffic, the similarity measures of network traffic usage are necessary in our project. According to the similarity measures of network traffic data, the data space of network traffic is partitioned into different subspaces and the similar traffic data grouped into the same subspace.

B. Classification of The Browsing Behaviors

Browsing behavior profiles creation based on Vector quantization.

Vector Quantization (VQ) is an efficient compression technique especially used in data compressions of image and speech based on the **similarity measures between feature vectors**. In this paper we are classifying the user browsing behaviors based on VQ. In the following we extended the Vector quantization and detection principle.

A vector quantizer Q is a mapping function to *k-dimensional* vectors in the vector space $R^k$ into a finite subset C of finite set of vectors $Y = \{y_i: i = 1, 2, N\}$. Each vector $y_i$ *is* called a code vector or a *codeword*, and the set of all the codeword's C is called a *codebook* and m is the codebook dimension.

$Q: R^k \rightarrow C, C = \{Y0, Y1, \cdot, Ym-1 \mid Yi \in R^k\}$

Given the input vector $X = (x_1, x_2, \cdot \cdot x_n)$ and the codeword $Yi = (y_{i1}, y_{i2}, .. , y_{in})$ $(i = 1, 2, . . . ,m)$, every input vector will be assigned with a codeword in which the distortion (Euclidian distance) between this codeword and the input vector is smallest among all codeword's. The distortion is defined as Quantization Error (*QE*):

$$QE = [\sum (x_{ik} - y_{jk}(t))\,2]^{1/2} \quad \text{---------- (1)}$$

Webpage popularity is defined by the page Request rate as $p_{it} = a_{it}/\sum_{i=1}^{N} a_{it}$, where $a_{it}$ is the request number of the *i*th document at *t* th time unit, and N is number of the Website pages. Let $C_{it}$ is the number of users who request the *i* th page at the *t* th time unit, users average revisitation to the *i* th document at t th time unit is calculated $U_{it} = a_{it}/C_{it}$ where $p_i$ is the *i* th page in the website at *t*th time unit, and S is the number of users at the *t*th time unit.

The potential of VQ application to the usage user behavior profile establishment of network traffic is that VQ can compartmentalize the behavior feature vector space to groups by comparing the similarities of feature vectors. The user profile can be recapitulated and indexed by the codebook.

The following steps describe the algorithm for finding the anomaly behavior.

The input vector: $X = (x_1, x_2, \ldots, x_n)$

The codeword: $W = (w_1, w_2, \ldots, w_n)$

1. Select a single set of best matching codebook vectors.
2. Compute the distance the input vector $X_i$ and the codeword $W_j$ (t), designate the winner node j* with the smallest distance.
   $J^* = \arg\min 1 \leq j \leq m \;\; X_i - W_{j(t)}$ -------- (2)
3. The Euclidean distance is chosen as Quantization Errors (QEs).
   $D = \| X_i - W_{j(t)} \| = [\sum (x_{ik} - w_{jk}(t))\,2]^{1/2}$ - (3)
4. Update the winner vectors of the winner node and its neighborhood.
   $Wjk(t+1)=w_{jk}(t)+\alpha(t)[Xik-w_{jk}(t)], j \epsilon N(t)$ ---- (4)
   N (t): non-increasing neighborhood function;
   α (t): learning rate function, $0 < \alpha(t) < 1$.
5. Repeat Step 2 and step 3 until all values under one corresponding codeword.
6. Sort the codeword's and repeat the steps 1 to 5.

Here we classify the browsing behaviors that detect the normal behavior and abnormal behavior. There exist significant differences in entropy distributions between two groups: the normal behavior entropy and abnormal behavior entropy.

## V. HIDDEN SEMI-MARKOV MODEL:

HsMM is an extension of the hidden Markov model (HMM) with explicit state duration. It is a stochastic finite state machine, specified by (S, **π**, **A, P**) where:

- S is a discrete set of hidden states with cardinality N, i.e. S = {1,….,N}.
- **π** is the probability distribution for the initial state $\boldsymbol{\pi}_m \equiv \mathbf{P_r}[s1 = m]$, $s_t$ denotes the state that the system takes at time and $m \in S$. The initial state probability distribution satisfies $\sum_m \pi_m = 1$;
- **A** is the state transition matrix with probabilities: $a_{mn} \equiv Pr[st = n \mid s_{t-1} = m]$, m, n Є S, and the state transition coefficients satisfy $\sum_n a_{mn} = 1$;
- **P** is the state duration matrix with probabilities: $pm(d) \equiv Pr[r_t = d \mid s_t = m]$, $r_t$ denote the remaining ( or residual) time of the current state st, m Є S, d Є {1,…,D}, D is the maximum interval between any two consecutive state transitions, and the state duration coefficients satisfy $\sum_d p_m(d) = 1$.

Then, if the pair process $(s_t, r_t)$ takes on value (m,d), the semi-Markov chain will remain in the current state m until time t + d-1 and transits to another state at time t+d, where d>=1.The states themselves are not observable. The accessible information consists of symbols from the alphabet of observations $o = (o1,\ldots,o_T)$ where $o_t$ denotes the observable output at time t and T is the number of samples in the observed sequences. For every state an

output distribution is given as $b_m(k) \equiv pr[o_t = k \mid s_t = m]$ where m Є S and k Є V ={1,…,K}. K is the size of the observable output set. The output probabilities satisfy $\sum_k b_m(k) = 1$. $b_m(O_{a|b}) = \prod_t{}^{b}{}_{=a}\, bm(o_t)$ when the "conditional independence" of outputs is assumed, where $O_{a|b} = \{o_t : a <= t <= b\}$ represents the observation sequence from time a to time b. Thus, the set of HsMM parameters λ consists of initial state distributions, the state transition probabilities, the output probabilities and state duration probabilities. For brevity of notation, they can be denoted as $\lambda = (\{\pi m\}, \{a_{mn}\}, \{b_m(k)\}, \{p_m(d)\})$.

We use the Markov state space to describe the webpage set of the victim website. Each underlying semi-Markov state is used to present a unique webpage clicked by a web user. Thus, the state transition probability matrix A presents the hyperlink relation between different WebPages. The duration of a state presents the number of HTTP requests received by the web server when a user clicks the corresponding page. The output symbol sequence of each state throughout its duration presents those requests of the clicked page which pass through all proxies/caches and finally arrive at the web server. We use a simple example to explain these relations by Fig. 2. The unseen clicked page sequence is (page 1, page 2, and page 3). Except those responded by the proxies or caches, the corresponding HTTP request sequence received by the web server is $(r_1,r_2,r_3,r_4,r_5,r_6,r_7,r_8,r_9,r_{10},r_{11})$. When the observed request sequence is inputted to the HsMM, the algorithm may group them into three clusters $(r_1,r_2,r_3,r_4)$,( $r_5,r_6,r_7$), $(r_8,r_9,r_{10},r_{11})$ and denote them as a state sequence ((1) ,(2),(3)). The state transition probability $a_{12}$ represents the probability that page 2 may be accessed by the user after accessing the current page 1. The duration of the first state (1) is d =4, which means 4 HTTP requests of page 1 arrived at the web server, i.e., $(r_1,r_2,r_3,r_4)$.

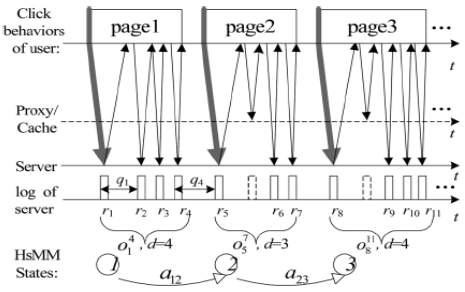



Fig 3: web browsing model

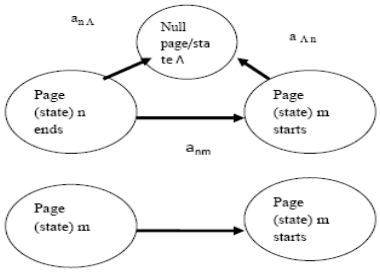



Fig 4: State transitions

The web browsing behavior can be formulated in terms of the HsMM as follows. We use a random vector $o_t = (q_t, r_t)$ to denote the observation at the th request in the request sequence, where rt Є V = {1,…., K} is the index of the object that the $t^{th}$ request intends to get from the original web server, such as an HTML page or an embedded image, and qt Є C = {1,2…..,} is interval between $r_t$-1 and $r_t$. K is the size of accessible object set. $O_t$ forms the observation vector sequence { $o_t$ : t=1,….T}, where T is the total length of the observation sequence. We use a underlying semi-Markov process $\{s_t : t=1,\ldots,T\}$ to describe the user's click behavior, where $s_t$ Є S is the index of the webpage browsed by the user at the $t^{th}$ request is the Markov state space which presents webpage set of the website. We use N to denote the size of S. Transition from $s_t$-1 to $s_t$ is considered as the user's browsing behavior from one webpage to another, following a hyperlink between the two pages. The state transition probability as t-$1^{st}$ means the probability of the user browsing from page $s_t$-1 to page $s_t$.If $s_t$-1 = $s_t$, then we assume the (t-1) th request ($r_t$-1) and the th request ($r_t$) are made for the same page.

In considering that the user may jump from one page to another page by typing URLs or opening multiple windows without following the hyperlinks on the WebPages in browsing the website, we introduce a Null page which is denoted as Λ to describe these browsing behaviors. That is, if page n has no direct link to page m (i.e., the transition probability $a_{mn}$ =0), and the user jumps from n to m directly in browsing the website, we then assume that the user transits from page n to the Null page Λ at first, and then transits from the Null page Λ to page m with the transition probabilities $a_n$ Λ and a $\Lambda_m$, respectively, as shown in fig.3.

The parameter estimation of HsMM can be done by the forward and backward algorithm [6]. We define the entropy of observations fitting to the HsMM and calculate the average logarithmic entropy (ALE) per observation. Use the outputs of VQ module as the model training data set to estimate the parameters of HsMM. Compute the entropy of the training data set and the threshold.

## VI.CAPTCHA GENERATION

Different priorities are given to the clusters according to their abnormalities and serve them in differently. Based on the entropy value if some abnormities found then the CAPTCHA based web service render to the attacker. Up on detection of the abnormalities all the requests are serviced with CAPTCHA to the deviated behavior user. An authentication mechanism that relies solely on CAPTCHAs has one disadvantage. The attacker can force the server to continuously send graphical tests, imposing an unnecessary overhead on the server. When presented with a CAPTCHA test, legitimate users may react as follows: (1) they solve the test, immediately or after a few reloads; (2) they do not solve the test and give up on accessing the server for some period, which might happen immediately after receiving the test or after a few attempts to reload. The zombies have two options; (1) either imitate human users who cannot solve the test and leave the system after a few trials, in

which case the attack has been subverted, or (2) keep sending requests though they cannot solve the test. However we are detecting the anomaly behavior in this paper, the deviation behavior identified again on servicing CAPTCHA, it confirms the attacker. The following P-algorithm describes the dynamic CAPTCHA serve to the users.

1. Compute the Entropy.
2. Classify the traffic legitimate or illegitimate.
3. If the traffic is illegitimate then serve the user with CAPTCHA.
4. If CAPTCHA answered then continue Service with CAPTCHA
5. Else Drop the traffic
6. Else
7. Serve the user without CAPTCHA.

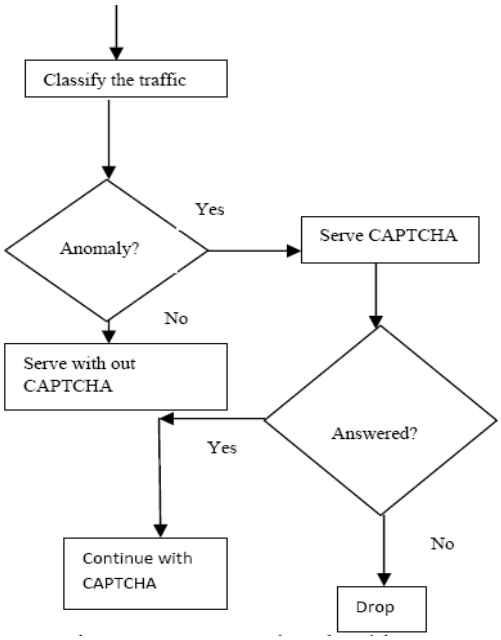



Fig 5: Dynamic CAPTCHA service algorithm

## VI. Conclusion and Future Work

This paper focuses on protecting Web servers from DDOS attacks based on CAPTCHA service. We presented novel algorithm that distinguish the normal and deviated behavior users. A set of real traffic data collected from an educational website and applied the p-algorithm to differentiate the normal and abnormal behaviors. Several issues will need further research: 1) if all clients are getting service from one proxy and Zombie is behind that proxy among the legitimate clients, blocking the IP results the service annoy and service delays to the legitimate users also.2) applying this model for other schemes to detect the App.DDoS attacks, such as FTP attacks.

## REFERENCES

[1] Shui Yu, Wanlei Zhou, Robin Doss, Weijia Jia,"Traceback of DDoS Attacks Using Entropy Variations" IEEE/ACM Tran. ON Parallel and Distributed Systems" vol. 22, no. 3, March 2011.

[2] John Elliott "Distributed Denial of Service Attacks and the Zombie Ant Effect" IEEE Computer Society April 2000.

[3] I. Ari, B. Hong, E. L. Miller, S. A. Brandt, and D. D. E. Long, "Modeling, Analysis and Simulation of Flash Crowds on the Internet," Storage Systems Research Center Jack Baskin School of Engineering University of California, Santa Cruz Santa Cruz, CA,Tech. Rep. UCSC-CRL-03-15, Feb. 28, 2004[Online]. Available: http://ssrc.cse.ucsc.edu/, 95064

[4] J. Jung, B. Krishnamurthy, and M. Rabinovich, "Flash crowds and denial of service attacks: Characterization and implications for CDNs and web sites," in *Proc. 11th IEEE Int. World Wide Web Conf.*, May 2002,pp. 252–262.

[5] Kostas G. Anagnostakis, Stelios Sidiroglou,Periklis Akritidis, Michalis Polychronakis,Angelos D. Keromytis, Evangelos P. Markatos. "Shadow Honey Pots" *IJCNS Vol. 2, No. 9, September 2010.*

[6] S.-Z. Yu and H. Kobayashi, "An efficient forward- backward algorithm for an explicit duration hidden Markov model," *IEEE Signal Process.Lett.*, vol.10, no. 1, pp. 11–14, Jan. 2003.

[7] Y. Xie and S. Yu, "A dynamic anomaly detection model for web user behavior based on HsMM," in *Proc. 10th Int. Conf. Comput. Supported Cooperative Work in Design (CSCWD 2006)*, Nanjing, China, May 3–5, 2006, vol. 2, pp. 811– 816.

[8] Dr.S.T.Gandhe , Mr.J.H..Nirmal," Performance of Different Techniques for Text Independent Speaker Identification in Noisy Environment" *IJCNS vol.2, No.7, July 2010.*

[9] Marcus Karnan and T.Logeswari "An Improved Implementation of Brain Tumor detection using Soft Computing "Vol.2, No. 1. January 2010.

[10] G.Suresh, P.Epsiba,Dr.M.Rajaram, Dr.S.N.Sivanandam "Scalable ACC-DCT Based Video Compression Using Up/Down Sampling Method" Vol.2, No.6, June 2010.

[11] N. Chan. BYAN: Sound Oriented CAPTCHA, Website http://drive.to/research.

[12] Greg Mori, Jitendra Malik, "Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA"

[13] [Online]. Available: http://www.caida.org/analysis/security/sco-dos/

Gamidi Pavan babu received the B.tech Degree from Jawaharlal Nehru Technological University Hyderabad. He is currently working towards the M.tech degree. His research interest areas are in Network Security and Web Security.

Jaya Vani .V is working as an Assistant professor in Avanthi Institute of Engineering & Technology. She received the M.tech Degree from Andhra University Vishakhapatnam. Her interest areas are Data Mining, Computer Networks.