ISSN (Print) : 2319-8613
ISSN (Online) : 0975-4024

Detecting HTTP Based Mimicking Attacks at HTTP Server

V Rama Krishna, Dr. R.Subhashini
[1]Research Scholar,School of Computing SathyabamaUniversity, India
[2]Professor , School of Computing , Sathyabama University , India

Abstract - **Botnets are major challenges for cybersecurity. DDOS attacks taken new dimension with botnets. First attacker will target security compromised systems and will inject bot in to them. Those systems will run malicious code and bot master cancontrol them. Bot master forms homogeneous/heterogeneous botnets by this Bots. Botnets are used for many cyber-attacks such as distributed denial of service (DDoS), information phishing and email spamming. Existing Intrusion Prevention / Intrusion Detection (IPS/IDS)systems can detect botnets attacks by using anomaly detection methods. To sustain botnets, bot masters working on bots that can mimic legitimate cyber behavior to fly away from the radar. Most of intrusion detection systems works on assumption the attack traffic is statistically different from normal traffic. Bot owners hack the popular website browsing history with that they will simulate thousands of users through bots and will try to degrade the performance of website. This becomes challenge for existing anomaly detection algorithm to distinguish between legitimate users and attacker. Previous studies carried on browsing behavior by using semi-Markov model prove it is impossible to detect mimicking attacks based on statistics if the number of active bots of the attacking botnet is sufficiently large[1]. It is becoming difficult to identify mimicking attack. We are proposing possible method to do mimicking attack and an algorithm to identify the mimicking attack pattern at server by using HTTP statistic. To prevent the attacks challenging the user, genuine user will respond to the challenge and attackers will fail to respond. This method can be used to distinguish legitimate user and attacker this can be extended to other layer 7 protocols.**

Keywords : Botnet, Mimicking attack, DDoS attack

I. INTRODUCTION

Current decade Botnets are the main contributors to distributed denial of service (DDoS). [2], [3]. DDoS attacks can be classified in two categories

1. Application DDoS attacks like, HTTP floods, slow attacks , and attacks uses operating system vulnerabilities , web applications and attacks using communication protocols. The goal of the attacks is to send more traffic to target application with requests. Which is targeted to eat up resources like CPU and memory eventually downgrade the performance.
2. Network layer DDoS attack like UDP floods, SYN floods, DNS amplification, IP fragmentation.

Botnet mimicking attacks will come under Network layer DDoS attacks. Below are the famous Botnet that caused DDOS attacks

a) Nitol: This is widely spread DDoS botnet which changing it form frequently. It will install in server using a TCP socket and then sends performance information from the victim's machine to bot master. Nitolis one of the most widely spread botnet,infected 60% of all attacking botnet IPs.
b) MrBlack: Mainly targets Linunx based platforms and architecture. It transfer system related information to bot master based on that bot master initiates different types of DDoS attacks like download a file and execute it, and then terminate a process.
c) Cyclone: This IRC-based and its C&C; details are obfuscated. This can perform stealing off FTP credentials , HTTP floods,
d) Pushdo / Cutwail: This bot can effect Microsoft Windows machines In 2010 the botnet effected 300 major site including the CIA, FBI, Twitter and PayPal.
e) Mirai : Recently by using Mirai Botnet many popular websites like Twitter, the Guardian, Netflix, Reddit, CNN and many others websites in Europe and the US

Bot master can execute mimicking attack in two phases

Phase 1:Retrieve the browsing patterns from the effected malicious machine identify the popular websites user is browsing. For selected websites retrieve important characteristics like bellow

a) Number of pages browsed
b) Web page requesting interval
c) Amount of data retrieved/transferred

Phase2: By using the information retrieved in Phase 1. Create new Bots to do Mimicking attacks.

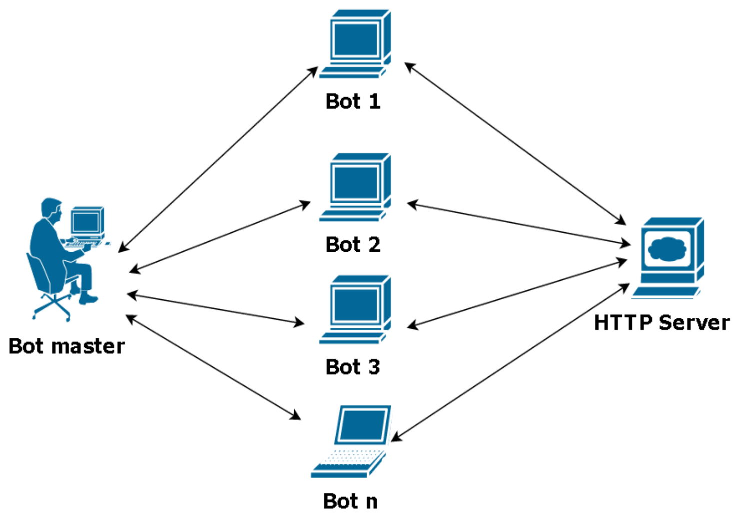

Fig. 1.Mimicking attack Architecture

If bot masters able to establish large number of active bots i.e each bot can simulate one legitimate. It is impossible to differentiate mimicking attacks from the legitimate web browsing of a large number of browsers [1]. Botnet attacks can be detected in two ways signature based and anomaly-based detection. The best suited for above attack is anomaly based after identifying the attack it can be feed to signature data bases[5]. Anomaly detection Anomaly detection also can be further classified as host-based approach [6], each machine is monitored to find malicious activity .This method needs installment toll on machine which may not be scalable. Second approach is, network-based this analyze network traffic [7].
In this paper, we are trying propose host based methodology. Intelligent Analyzer collects the flows and filters them based on originator ip and monitor the flow activities using layer 7 statistics. Cluster the flows having similar characteristics. Derive the flows appearing across the clusters. Validate them using HTTP Challenge. Populate block list to the IDS.

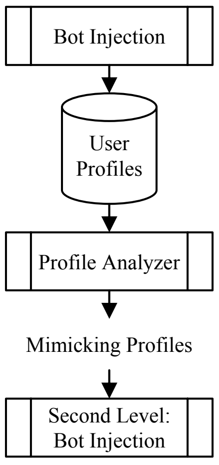

Fig. 2Bot master flow of events

In this paper we are proposing two algorithms First one discuss possible way to do mimicking attack. Second algorithm is for detecting mimicking attack at HTTP server level. As part of Mimicking attack algorithm homogeneous bots will be injected in first step. Using that it will collect the browsing history of the users. Bot master will analyze the user accessing patterns and from that will derive accessing profiles with bellow details

1) Number of request
2) Number of packets on each session
3) Size of the data in each packet
4) Duration of the each session
5) Time interval between successive sessions

With above information different Bots will be designed.Those bots will be heterogeneous in nature. It is not necessary to generate same number of bots as number of systems. Based on the commonality in the profile Bot master can club multiple user profile in single bot. This analysis might need bot master interventions but still bot master can automate this as well. Bot master designs the heterogeneous bots mimicking the user behavior and will insert in to respective system. Bot master starts monitoring the system and system will trigger the mimicking attack. Mean time any variation in the browsing behavior observed in the user profile that feedback will be used in generating the next bot.

Once Botnet started mimicking attack it will be difficult to predict the attack at gateways and IPS/IDS. So we are working on model that can detect at server level. This method mainly focuses HTTP mimicking attack. It will inspect HTTP layer statistics based on the behavior it will form identical groups. If flow similarity found in multipleparameters will be considered possible attack pattern. For those flows HTTP challenge will be send .Genuine user will respond to the challenge attacker who may not implement the complete HTTP stack will fail in responding. That list will be circulated as block list.

We need to cluster the data based on similarities in attributes so we opted for clustering algorithms.Data clustering two types of algorithms are there hierarchical and partition.

As part of the study we analyzed hierarchical (connectivity models) model it starts with assumption that objects will be related to nearby objects than far away objects. The algorithms start with less number of samplesandforms group with one large set having items of similar properties. Whereas partition model will based on a center vector and club the multiple elements close to cluster. The number of partitions (clusters) will be fixed in traditional algorithms. For the above problem we require partition clustering because we need to divide the data objects into non-overlapping subsets. For each parameter we need to come up with multiple clusters. The number of clusters are not deterministic so famous k-means algorithm might not be perfectly suited for this [8].

Whereas X-means clustering algorithm will not work on assumption that number of clusters are known in the beginning.X means algorithm works with three steps [12].

Step 1: Conventional K-means algorithm

Step 2: Adjust the centroids

Step 3: Continue above two steps until k<Kmax

When the clustering process needs to be stopped will be decided by two methods Akaike information criterion (AIC) or Bayesian information criterion (BIC). In Our current requirement we will be not aware of the number of clusters in advance. We need to find the optimal number of clusters that are closely tied up with parameters so using X-means algorithm.

In HTTP server will keep background process to collect the flow information. This process will get the one copy of all the HTTP requests and will collect the Layer 7 information.Analyzer will filter the data based on originator ip address (source ip address). In the filtered data critical components like (Number of session, idle time, amount of data transferred, methods used and duration of the session) will be recorded and that data will be forwarded to clustering algorithm.

X-means clustering algorithm was applied on flow data on different parameters. The quickness of identifying the attack is much needed. X-means algorithm will be executed parley for each parameter after completing the clustering on all parameters that data will be given to insertion algorithm. The insertion Algorithm was designed to identify the common flows occurring across the clusters. If multiple flows occurring across the clusters means there is similarity in the activity they are performing. That set of flows was added to the suspicious list of flows. For those flows Analyzer will give to the challenge process. The HTTP challenges process will walk through all the flows currently alive and will send the HTTP challenge.

HTTP challenger module will use HTTP server to initiate challenge for those suspicious flows for the next subsequent request came from the client. Out of existing HTTP challenges with cookie is difficult to crack.The flows that are not replying on HTTP challenge add them to block list and populate that to the IDS/IPS systems in the network. We can give the trace of those flows so that IDS systems can add them to the block list [13][14]. There is a possibility that bot master can populate program so that Bots can reply to the HTTP challenge. To identify such cases keep the list of suspicious flows identified after insertion algorithm. In multiple iterations of X-means algorithm and insert algorithm the same flow is observed consider them as suspicious mimic attack flowsand add them to the Gray list so that Analyzer will keep watch on those clients, if they are coming further in the iterations the connections coming from those source address will be blocked. By end of each phase we will be having three lists

White list: Flows that are not having any similarities

Black list: Clients HTTP challenge failed, they are not permitted to any further sessions with HTTP server

Gray list: Clients having similar activity pattern. They are under monitoring if they observed further iterations moved to block list.

If any new entry is added in the black list it needs to be populatedto the IDS immediately. If a system was attacked by Bot they might be doing multiple malicious activities going on. The system will be completely blocked in network.

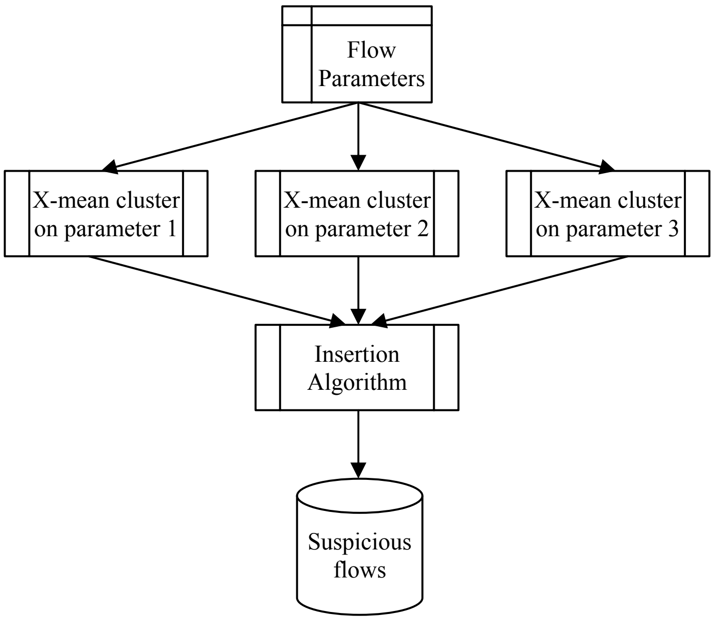
Flow Parameters
X-mean cluster on parameter 1
X-mean cluster on parameter 2
X-mean cluster on parameter 3
Insertion Algorithm
Suspicious flows

Fig. 3 Detection Algorithm work flow

II. Proposed Methodology

Here we are working on proposing two algorithms. First algorithm is about creating the mimicking attack. Second algorithm is to identifythe mimicking attack at host level.

2.1 Algorithm for creating Mimicking attack

As part of the algorithm's N number of bots will be injected to different computers. All these bots are homogeneous. They will collect the browsing statistics of the user. After that browsingstatistics are analyzed and user profile was created. From that target server was identified and create M heterogeneous botnets for N network elements. Each bot simulate the browsing behavior of the user. As part of the second phase Bot master is initiating mimicking attack.

Bot master repeats the process continuously so that IDS/IPS may not able to detect the mimicking behavior.

Algorithm 1:

```
1. Inject Bot (Bi) into N (N1,N2,…. Nn) systems
2. While (TRUE)
   Observe the browsing statistics
   i)   Number of pages browsed
   ii)  Web page requesting interval
   iii) Amount of data retrieved
3. From the analysis of information retrieved in step 2 inject new bots (B1,B2,…Bm)in to system
   Each Bi simulates the browning behavior of the user in the system Ni
4. For each boti∈ {botsn)fallowing action performed
   Decide the browsing length, page request from step 3
   Submit the request and discard the downloaded content;
   Wait for a time interval decided by step 3
5. Remove the current bot(bi) from the set bots;
6. Go to step 2, Introduce new bots
```

Bot master run two phases in parallel while one set of bots are doing mimicking attack. Bot master will collect the new user profiles happening in the system and will change the heterogeneous mimicking pattern.

Explanation of the Algorithm

Step 1: Bot master inject N copies of bot (equal to the number of system in the network). This bot is intended to collect the browsing profile of the user in the system the data looks as bellow

Table ISystem browsing profile

System IP	Number of session per second	Number of packets per session	Number of bytes per packet	Duration of each session	Time interval between two session
1.1.1.1	5	10	250 bytes	50 msec	150 msec
2.2.2.2	12	10	150 bytes	13 msec	70 msec
3.3.3.3	7	12	200 bytes	40 msec	160 msec

Step 2: Analyze the user browsing profile and come up with different unique profiles can be used for mimicking attack. Currently we are proposing this will be done by bot master, he will analyze manually and will keep similar profiles in to one group and prepare common profiles like bellow. This can still be automated.

User profile 1: {Number of session per second: 6, Packets per session: 11, Bytes per session : 225 bytes, Duration of each session : 45 msec, time interval between successive session : 155 msec}

User Profile 2: {Number of session per second: 12, Packets per session : 10, Bytes per session : 150 bytes, Duration of each session : 13 msec, time interval between successive session : 70 msec}

Step3: Bot master will inject the heterogeneous Bots to mimic the user profiles identified in the step2

2.2 Algorithm for detecting mimicking attack

Current Botnet detection algorithm's mainly focuses on anomaly detection. They will work on assumption that attack will have variation in the flow than usual. Most of the botnet detection algorithms like Botminer, Botsniffer will work on this assumption attack pattern will have any deviation with the existing patterns but mimicking attacks will not fall in that category So it is not possible to detect the attack patterns using the traditional algorithms. So we are proposing the new algorithm to detect mimicking attacks

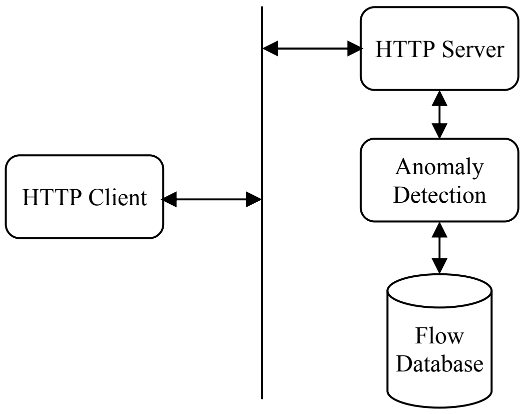

Fig. 4HTTP server Mimicking detection Architecture

The quickness algorithm is muchneeded, the algorithm should be fast enough to detect the attack while attack is in progress. Need to run this algorithm in multiple phases we can have feedback from one phase to another. We can have three processes running in parallel.

Process 1: a) Collect the flow information based on source ip, collect the parameters

b) Run the clustering algorithms on each parameter parley

c) Provide the Cluster sets to the Process 2, and move back to step a) for next set of flows

Process 2: Run the insertion algorithm to find the common flows, provide that information to process 3

Process 3: Send Http redirect for all the common flows find in process 2 , after time out prepare the block list, gray list and populate

Populate the block list to the IDS/IPS. The three processes need to run in parallel so that execution will be fast and detection of the mimicking flow will be quicker.Here we need clustering algorithm based on multiple parameters like idle time, active time, gap between two successive connections, methods used by HTTP sessions. Those clustering on different parameter need to happen in parallel so that input to insertion algorithm happens much quicker.

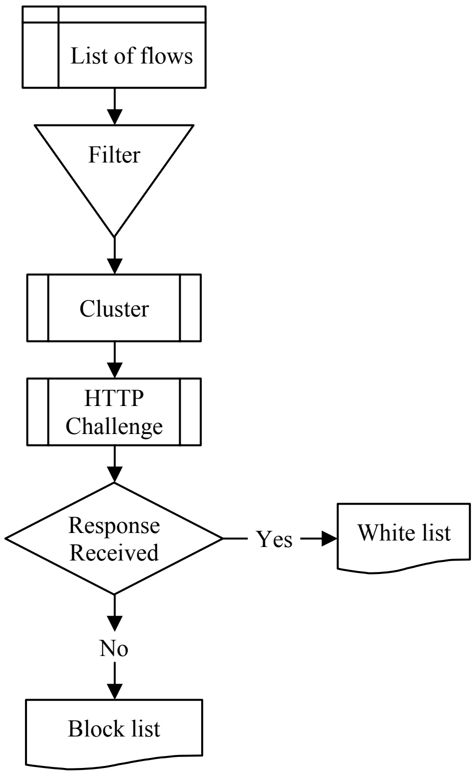

Fig. 4 HTTP server Mimicking detection algorithm flow

Algorithm 2 : Detecting HTTP mimicking attack at HTTP server

1. While (HTTP_SERVER_RUNNING)
2. Group the flows based source ip flow(sip) = {flow1,flow2,flow3,flow4 …}
3. Retrieve HTTP parameters Time stamp, methods from Layer 7 header for each flow in flow (sip)
4. For each (flow (sip)) derive average idle time, active time, Methods used and timeinterval between two successive connections

X means algorithm:

5. For each parameter Pi ∈ {p1,p2,p3.k4}

   {Idle time, active time, gap between two successive sessions,Methods}
6. Use X means algorithm to group them to clusters, Initialize
   a) K=Kmin
   b) For each element Ni∈ {flow1,flow2,…, flow n},get averages for comparison to the Cluster (Ki):
   c) For all ki ∈ {k1,k2,…kn}

   Add individual value of flow to the sum of all values of the individuals in Cluster (ki), then divide by the total number.

   Add individual's value of flow to the sum of all valuesof the individuals in Cluster(kj), then divide by the total number .
   d) The averages found in Step c) is closer to the mean values of which Cluster (ki), then this individual belongs to that Cluster (ki)
   e) Change the new mean vectors for Cluster (ki).
   f) Go to step b) until all flow elements were done
   g) FORki = 1,. . . ,K: Replace each centroid ck by two centroids ck1 and ck2.
   h) Run K-means algorithm with K = 2 over the cluster Ki.
7. All the iterations were done in step 5, different cluster sets will be formed with each parameter

8. Compare one parameter cluster group with other parameter cluster group elements using intersect algorithm keep the common elements in a pool [D].

```
a) Take one group ci in Pi and another group cj  inPj
b)  For all  Ki ∈ {k1,k2,….,kn} in Pi and Kj ∈ {k1,k2,….,kn} in Pj
c) Sort the group Ki and Kj based on source ip
d)  I ,j=1
e) Li is length of ci, Lj is length of cj
f)  whilei  ≤ Li and j ≤ Lj
if K[i] = Ki[j] then add to Pool[D] , i+1,j+1
else if K[i] ≥ K[j] then j+1
else i+1
 g)  go to step a and repeat this for all  Pi and Pj
```

9. All flows (flow∈*D)* identified in step 12 send HTTP challenge (302 redirect message) with cookie.
10. Flows that are responding to HTTP challenge move to Gray list, flows that are not responding move to block list
11. Compare the flows in Gray list if flows are coming more than 3 times in the iterations consider them as possible mimicking flows, add them to block list.
12. Go to step 1 and repeat with new set of flows

Explanation of Algorithm:

Step1: Filter out flows based on source ip' address

flow 1 : SIP1,DIP1,Sport1,DPort1

flow 2 : SIP2,DIP2,Sport2,Dport2

For each source ip calculate

flow (sip1) { (Thursday , 06-June-2017 04:30:01, GET), (Thursday , 06-June-2017 04:30:03, POST) …. }

flow(sip1) {00:03:00 (idle time), 00:01:00(active time), 00:00:50 (gap), GET,POST,PUT,DELETE(methods)}

Table 2 Methods in flow

Method	Number of requests
GET	2
POST	4
HEAD	1
TRACE	12
OPTIONS	15
PUT	1
DELETE	1

Step 2: By using k-Means algorithm [1} groups N data points into different data sets , based on bellow parameters

1) Idle time (P1)
2) Active time (P2)
3) Time between two successive sessions (P3)
4) Number of request per method (P4)

Step 3: Use X-means clustering algorithm on all flows by using above 4 parameters

First run K means cluster Get averages for comparison to the Cluster. For P1 Set the initial partition, and the initial mean vectors for each group

Cluster Group	Individual	Mean
Idle time k1	4 (flow1), 6(flow2)	5
Idle time k2	11 (flow2) 13 (flow1)	12

Step 4: For each individual, get averages for comparison to the Cluster (ki):
Example flow 3 idle time 5

Add individual valueof flow to the sum of all values of the individuals in Cluster (ki), then divide by the total number
Example The mean of K1 is 5
Add individual valueof flow to the sum of all values of the individuals in Cluster (ki), then divide by the total number
Example The mean of K2 is 9.6

Step 6: If the averages found in Step 4 are closer to the mean values of Cluster (ki), then this individual belongs to Cluster (ki), and the averages found now become the new mean vectors for Cluster (ki).
If closer to Cluster (kj), then it goes to Cluster (kj), along with the averages as new mean vectors.
Example : flow 3 belongs to K1 and K1 mean remains as 6

Step 7: If there are still flows present in flow data base , continue again with Step 4. Otherwise go to Step 8.

Step 8: Compare distanceof element to its own cluster's mean and to that of the opposite cluster mean. If cluster is close to its own group keep the element in that cluster else place it in opposite cluster.

Step 9: If any relocation occurred in Step 8, the algorithm must continue again with If no relocations occurred, k-means clustering is completed .

a) Step 10 :forki = 1,. . . ,K: Replace each centroid ck by two centroids ck1 and ck2.

Step 11: Run K-means algorithm with K = 2 over the cluster Ki.

a) Replace each centroid. Use BIC calculation to determine two clusters is needed (or) single cluster is best suited
b) If convergence condition is not satisfied, do it again. Otherwise Stop.

Step 12: By end of the last step for each parameter different clusters are formed
P1 {c11, c21,c31… }, P2 {c21,c22. c23…}. P3 {c31,c32,c33..}, P4{c41,c42,c43,…}
Identify the flows that are falling in common clusters in K1,K2

Parameter P1 : Idle time

Cluster 1	flow1,flow2, flow 5, flow8
Cluster 2	flow 3, flow 4, flow 5, flow 6, flow 7, flow 9

Parameter P2 : Active time

Cluster 1	flow1,flow2, flow 5, flow7
Cluster 2	flow 3, flow 4, flow 6, flow 8, flow 9, flow 10

Parameter P3: Time between two successive sessions

Cluster 1	flow1,flow2, flow 5, flow7,flow 8
Cluster 2	flow 3, flow 4, flow 6, flow 9, flow 10

Parameter P4 : Number of request per method (k4)

Cluster 1	flow1,flow2, flow 10, flow 8
Cluster 2	flow 3, flow 4, flow 5, flow 6, flow 7, flow9

Common flows: flow1, flow2,flow10, flow 8, flow3, flow4, flow6, flow 7, flow 9 flow10

Step 11: Compare the flows that are occurring in both K1 and K2 clusters. For all subsequent requests send initiate HTTP challenge (302 redirect message) with cookie.
Client has to store and resend the cookie. If it is an attacker he fails to respond

Step 12: Flows that are not responding Cookie and having similar pattern considered to be mimicking attack. Server can pass this information to firewalls, IPS/IDS to block the malicious users

Step 13: Maintain Gray list with the ip address that giving reply but if they appeared in multiple times in iterations move those ip address to block list.

III. Conclusion

We discussed the possible ways of mimicking attack and identifying at server level by using HTTP statistics. Will determine the mimic attack and block the malicious machines. If servers are distributed the information need to be exchanged and the current design need to be altered accordingly. This is Host based techniques it may not scalable in large network .We concentrated on HTTP but there is a possibility the attacks can happen in any application protocol in those cases the algorithm need to be altered according to the application protocol. In future works we will be working on the methods of extending to all layer 7 protocols. The quickness of the algorithm is needed if the flows are in large numbers then we may land in blocking after the attack. Working on approach to identify mimicking attackat networkgateways so that it can be much scalable.

REFERENCES

[1] Shui Yu, Song Guo, and Ivan Stojmenovic "Fool Me If You Can: Mimicking Attacks and Anti-attacks in Cyberspace", IEEE Transactions on Computers, Vol. 64, No.1, Jan 2015.
[2] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-based defense mechanisms countering the dos and DDOS problems", ACM Computing Survey, Vol. 39, no. 1, 2007.
[3] M. Edman and B. Yener, "On anonymity in an electronic society: A survey of anonymous communication systems," ACM Computing Survey, vol. 42, no. 1, 2009.
[4] AmuthanPrabakarMuniyandia, R. Rajeswarib and R. Rajaramc, "Network Anomaly Detection by Cascading K-Means Clustering and C4.5 Decision Tree algorithm".
[5] S. Basudev and A. Gairola, "Botnet: An Overview", CERT-In White Paper CIWP-2005-05.
[6] E. Stinson, J.C Mitchell, Characterizing bots' remote control behavior, in: Proceedings of the 4th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, DIMVA'07, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 89-108.
[7] G. Gu, V. Yegneswaran, P. Porras, J. StollandW. Lee, "Active botnetprobing to identify obscure command and control channels", In:Computer Security Applications Conference, ACSAC'09, Annual, 2009, pp. 241-253.
[8] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman and Angela Y. Wu."An Efficient k-means Clustering Algorithm: Analysis and Implementation",
[9] Ma. Del Rocio Boone Rojas and Maria J. SomodevillaGarcia "Research issues on K-means Algorithm: An Experimental Trial using Matlab by Joaquin Perez Ortega".
[10] Asif Khan, IsrafilTamim, Emdad Ahmedand Muhammad Abdul Awal, "Prospective Analysis for Effective Clustering in Wireless Sensor Network (WSN) Using K-Means Algorithm", Wireless Sensor Network, 2012, Vol. 4, pp. 18-24.
[11] "Comparison Between Data Clustering Algorithms", The International Arab JournalInformation Technologies, Vol. 5, No.3, July 2008
[12] "X-means: Extending K-means with Efficient Estimation of the Number of Clusters" ,DanPelleg, Andrew Moore, January 2012
[13] Mohiuddin Ahmed and AbdunNaser Mahmood "Anomaly Detection, DataNovel Approach for Network Traffic Pattern Analysis using Clustering-based Collective",Sci. 2015, Vol. 2, pp. 111. doi:10.1007/s40745-015-0035-y
[14] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-baseddefense mechanisms countering the dos and DDOS problems",ACMComputing Survey, Vol. 39, no. 1, 2007