EAI Endorsed Transactions
on Energy Web
Research Article EAI.EU

Mimicking Attack Detection at Hybrid Level

V Rama Krishna[1,*] and R Subhashini[2]

[1]Research Scholar, School of Computing, Sathyabama Institute of Science and Technology, Chennai, India.

[2]Professor of Information Technology, Sathyabama Institute of Science and Technology, Chennai, India.

Abstract

Botnets are becoming an easy way of creating multiple attacks. One of them was botnets simulate the behaviour that is very near to the legitimate users. Previous research found through semi-Markov model it was difficult to detect mimicking attack based on browsing statistics if attacking bots were sufficiently large in number [1]. By using Bots attackers will collect the user profiles from multiple systems. Bot master (attacker) will study statistics and Bot master will prepare a common profile (or) multiple profiles similar to the user activities. In the next phase, bot master injects profile into user systems through bots. If bot master injects common profile across all bot injected system then the attack was considered as a homogeneous mimicking attack. If bot master injects multiple profiles to the bot injected systems the attack was considered a heterogeneous mimicking attack. As part of our study, we simulated the mimicking attack and worked on detecting at multiple levels. We have developed algorithms of detecting at a server level [2] and the gateway level [3]. In this paper, we are going to discuss the merits and demerits of these two detection algorithms and proposing another architecture module hybrid level detection. Which will be spread across servers and gateway which will have the bird view of activities happening in the network. It collects the statistics from different network elements and based on the analysis of the trace of the bot activities will identify mimicking attack.

Keywords: Botnet, Mimicking attack, semi-markov model, Ips.

Received on 16 March 2020, accepted on 27 April 2020, published on 20 May 2020

doi: 10.4108/eai.13-7-2018.164630

*Corresponding author. Email: ramakrishnav2525@gmail.com

1. Introduction

Day by day botnets becoming major threats to the cyber world. Most of the recent network DDOS attacks like TCP syn flood, UDP flood. application-level flood, application-level flood attacks, amplification attacks using DNS protocol. [4]-[5] Currently attackers by using bots went to the next level. IDS/IPS/Firewall systems can detect control level attacks, so attackers by using started to mimic legitimate users. Mimic attacks by nature imitate and will look like a genuine user. Intrusions detection systems (or) Firewalls might not able to detect these attacks.

Bot master might not have the complete details about the mimicking profile at the first state, so he will inject it in phase-wise.

Phase 1: Bot master identifies the system that was less secured and will inject initial bot into that system. In this phase, bot will silently reside in the machine and will observe the browsing activity of the user. Bots will pass the browsing history of the user to bot master. The bot master will detect the frequently used server ports and popular server ports. For each server bellow statistics were collected

a) Pages browsed per second.
b) Interval gap for requesting successive web pages

The average amount of bytes transferred and retrieved from the server.

Phase 2: Bot master will analyze collected statistic Bot master will retrieve homogeneous and heterogeneous mimic profiles that are close to the user activates.

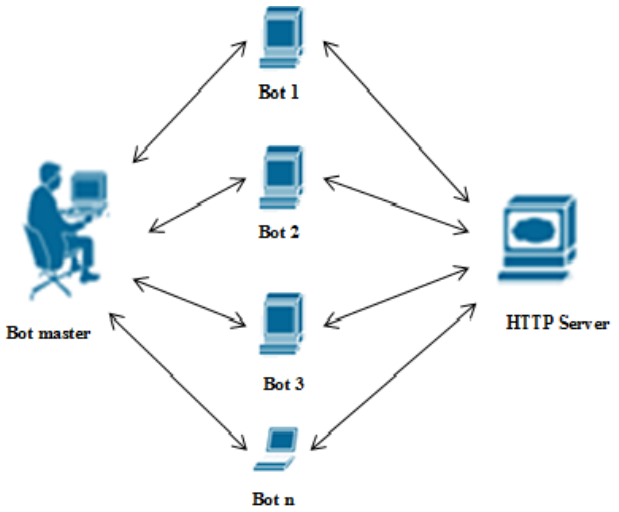

Figure 1. Mimicking attack Architecture

Bot master will continue phase 1 and phase 2 activities based on the need can alter the mimicking profiles. Each Bot can mimic legitimate user so that can fly under the radar of security systems. Bot injection of mimicking profile happens as described in bellow figure 2. First phase bot will retrieve specific user profile, Bot master will analyse the profiles collected from different users and will formulate single (or) multiple mimicking profiles.

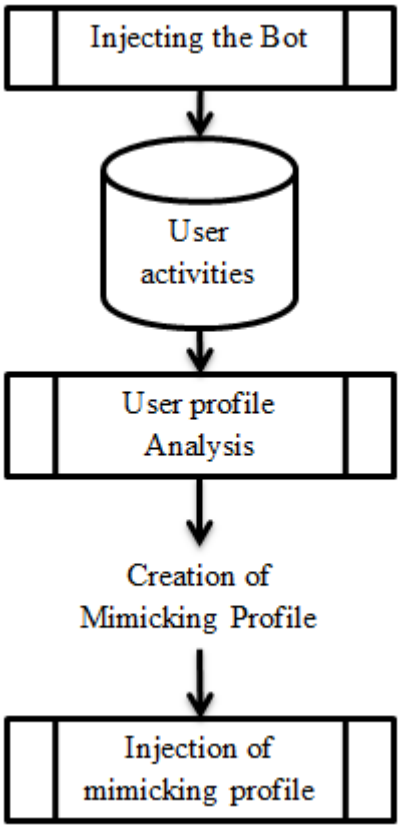

Figure 2. Mimicking attack creation

In this paper, we will discuss Server and gateway level algorithms and merits and demerits of it and hybrid architecture which is better suited for mimicking attacks.

1.1 Mimicking attack detection at the server level

Most of the botnet algorithms mainly focus on anomaly detections. Most of the algorithms like Bot miner, Bot sniffer mainly works on the principle that attacks will have deviations from the normal. behaviour. Mimicking attacks are the different form of anomaly-based attack. So, the existing algorithm might not able to detect mimicking attacks [6]-[11]. This section we are discussing the mimicking detection at the server level [2].

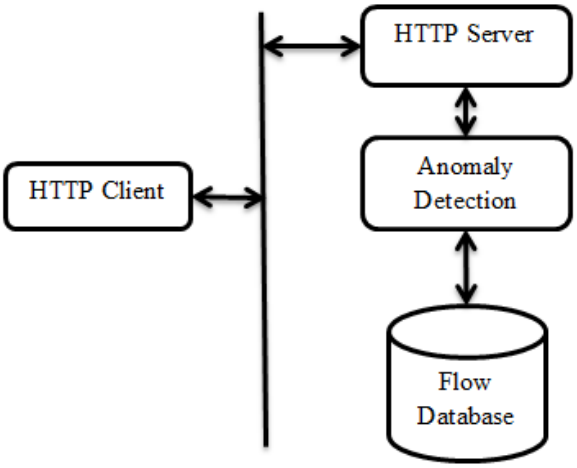

Figure 3. HTTP server Mimicking detection Architecture

This algorithm works in three phases. Feedback from one phase is given to another phase:

.

Process 1: a) At HTTP server, statistics collected based on client created the connection (source IP)

b) The Clustering algorithms were executed on all flows from different clients based on parameters, how much time the flow was kept idle after forming the connection, the successive time interval between two sessions made by the same client, how much time that each flow is actively passing the data, Application methods (current case HTTP) used by the client in each session [12]-[17].

c) The set of clusters formed in step *b)* will be given to Process 2

This process will continue with different source IP flows:

Process 2: From process 1 this process will get different clusters from the different client (source). All these flows will be forwarded to the next phase where the insertion algorithm will be used to find the common client appearing in multiple cluster bucks.

Process 3: Process 2 will give the list of clients IP addresses that are suspicious of doing mimicking attack. This process will send http HTTP redirect for all source IPs fond in process 2. Will wait up to time out, based on the response gray list, while list, block list is prepared.

The blocklist can be blocked at the server, similarly, the list can be populated to IDS/IPS system in the distributed network so that further actives from the suspicious host can be blocked at them [18]-[21].

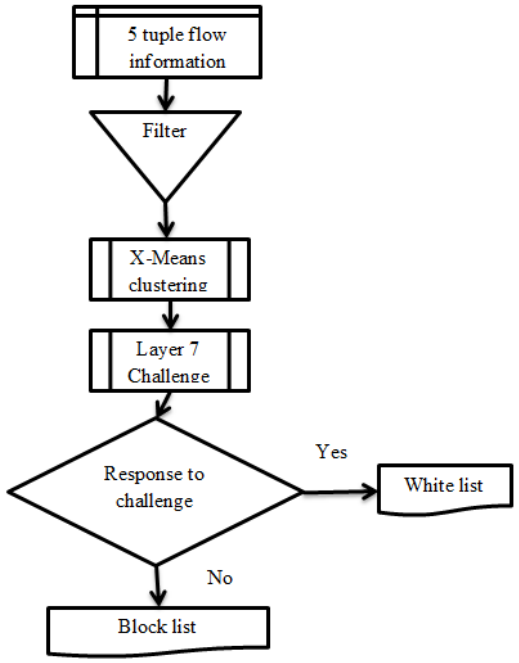

Figure 4. Mimicking detection algorithm at HTTP server

Advantages:

1) Online detection: Can detect the attacker when he is performing the attack.
2) The complexity of processing is less as we are working for only one server.
3) Can detect the attack even though encryption was turned on.

Limitations:

1) Scalability: The algorithm needs to be installed and executed at all servers.
2) Portability: Servers might be running with different operating systems and environments changes needed accordingly.
3) Cost and maintenance: as the need to be present in multiple places the cost and maintenance of this will be more
4) Can identify the attack is initiated within the LAN (local area network) easily, but if source originated from outside the LAN it was difficult to block them.

1.2 Mimicking attack detection at the gateway level

Mimicking attack detection at gateway need to depend on the network layers statistics like Network, Transport, Application header statistics [22]-[24]. This section we are discussing the mimicking detection at the gateway level [3].

Process 1:

a) For each new connection at the gateway, 5 tuple was retrieved and that need to be used as a key for identifying for the flow. The 5 tuple consists of source IP address, destination IP address, source port, destination port, TCP/UDP) for those connections network, transport, application parameters are collected.

b) Run the clustering algorithms on bellow parameters individually.

- Number of flows in 5 min window
- Count of packets in each flow
- Count of bytes in each packet
- The time duration of each session
- Idle time of each connection

c) this cluster data will be given to process 2 and process 1 will continue with a new set of connections in the next 5 min window.

Process 2: Using insertion algorithm the commonality parameters and flows were identified that list of source IP addresses will be given to process 3.

Process 3: For identified hosts in process 2 gateway will send authentication challenge. Based on the response the host will be added to the block list/gray list

These lists will be populated to IDS/IPS systems so that they will drop the connection.

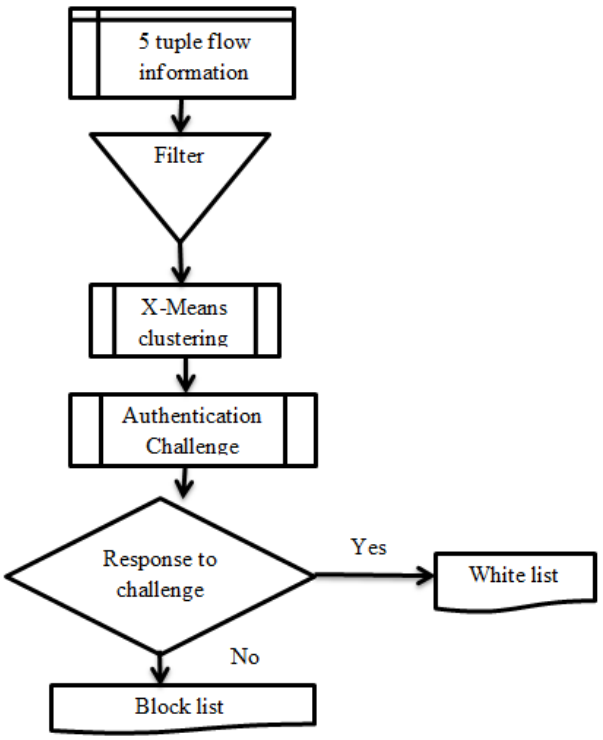

Figure 5. Mimicking Detection Algorithm at the server level

Advantages:

- Less scalability only we need to run this algorithm at the server.
- Cost and maintenance will be easy.
- Can block the attacks coming from outside the LAN also.

Limitations:

- Time for processing: Need an in-depth analysis of the packet which consumes a lot of processing time.
- The complexity of processing: the data will be in huge numbers collecting and analyzing the is a tedious job.
- Offline detection: While the actual attack happens still the analysis might be in progress, we might not able to detect and block it.
- If packets are encrypted at end host, we can't collect the statistics properly.

2. Methodology

Hybrid mimicking detection works individually it will not be installed either in the server/gateway, the detection system will collect the date from the end host and gateways. The analyzer will analyze the data and will make the decision.

Analyzer periodically collects the traffic statistics and activities from HIDS, NIDS, and Firewall. IDS systems will do the known signature analysis and anomaly detection firewall feedback on network activities will be taken. With attack patterns and audit records-analyzer executes X-means algorithm to identify the mimicking patters. Based on the analysis mimicking patterns will be identified and a signature will be passed to the IDS systems.

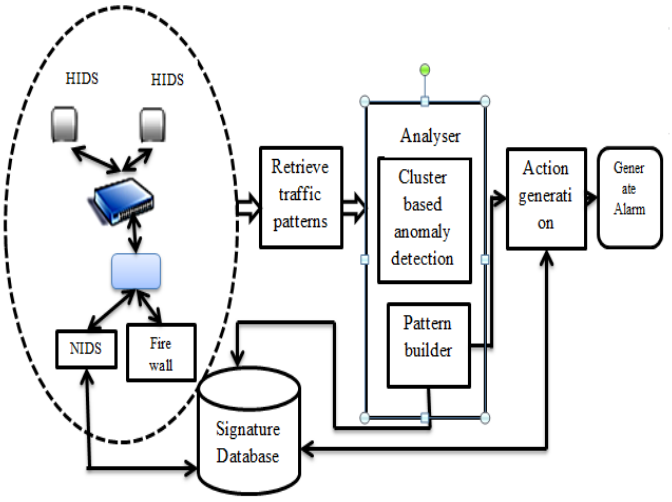

Figure 6. A hybrid threat detection system

Above block diagram shows the hybrid threat detection system. The analyzer will first perform the similar patterns from the data received from the HIDS, NIDS, Firewall detection system uses x-means clustering algorithm. This algorithm provides similar activity performing flows, this is known to be having less false alarms in finding similar patterns.

From HIDS and End host, bellow information will be retrieved:

- Hosts transferring/receiving high/malicious data
- Hosts generating/receiving traffic in the fixed time interval
- A high flow of traffic from specific protocol user
- The host makes a high rate of control connections
- The host having spikes in CPU/low memory because of traffic
- A host sending traffic in non-working hours
- Host getting more frequent changes in configurations
- Host reaching threshold limits more frequently
- Connections staying idle for a long time

From NIDS and gateway, bellow information will be retrieved:

- 5 tuple information (source IP, destination IP, sport, dport,tcp/udp) sending large volumes of traffic
- Hosts passing more control traffic
- 5 tuple information that is staying idle for a long time

- 5 tuple connection passing similar traffic in a similar time frame
- 5 tuple information sending large volumes of traffic
- From the firewall, bellow information will be retrieved
- Traffic coming from unknown ports
- Hosts that violating layer 3/layer 4/ layer 7 security checks

Filter/Converter module: This is the entry point to the analyzer firewalls/IDS will give data ostoref packets/flows/host details that are suspicious from the activity. This module stores them in specific flows 5 tuple format. The flows repeating in the consecutive windows time frame of 5 min will be forward to the next module.

Processing of data: For all flows (TCP/UDP) retrieve 5 tuples (source IP address, source port. destination IP address, destination port, TCP/UDP) collect information for 5 min:

a) Count of flows in 5 min window
b) Count of packets per flow
c) Count of bytes per packet
d) The time duration of each session
e) How much time each session was idle
f) Layer 7 activity happened on each session

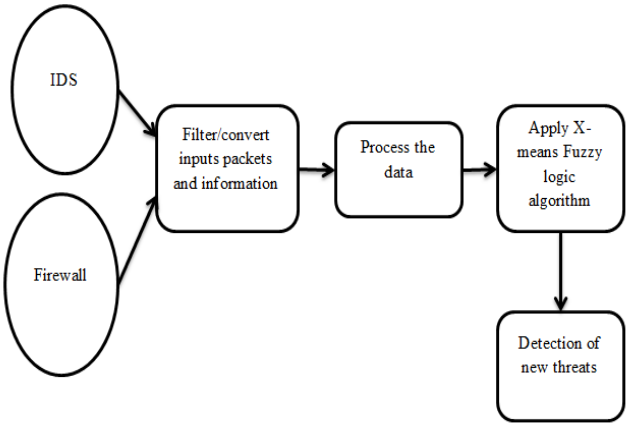

Figure 7. Hybrid threat detection system flow diagram

X-means clustering algorithm will be applied on the network, transport statistics collected. As result, the source ip address that behaving similarly in the network will be retrieved.

Similar activity patterns:

1. For each parameter Li ∈ {l1, l2, l3 .l4 ,l5,l6}

{Count of flows in 5 min window, count of packets per flow, count of bytes per packet, Time duration of each session, how much time each session was idle, Layer 7 activity happened on each session}.

2. In this step, X means clustering algorithm will be applied.

a) C=Cmin

b) For each element of flow fi∈ {f1,f2,…, fn}, compare the parameters and get averages to Cluster (Ci):

c) For all ci∈ {c1,c2,…cn}

Each flow parameter value is added to the sum of the values present in the Cluster (ci); the value calculated after summing will be divided by the total number of elements in the Cluster (ci):

d) The result obtained in step c) is compared with mean value of all clusters. The cluster whose mean value is closest to it will be identified and this flow will be added to that cluster (ci).

e) After adding the new flow, the new mean for Cluster (ci) will be computed

f) Go back to step b) and continue till all elements in fi are completed

g) On each cluster Ci= 1, . . ., K: identify two centroids ck1 and ck2.

h) Again, execute the K-means algorithm this time parameter using as 2 on each cluster Ci.

7. All the iterations were done again, for each parameter

8. In the above steps, each cluster group was formed with one parameter, now we need to find the common flows appearing across the clusters for that we will be using the insertion algorithm. The flows common across the clusters will be added to the mimicking pool [Mp]

a) One flow fi in cluster group ci compared with another cluster group cj with element fj

b) For all Ci∈ {c1,c2,….,cn} in fi and Kj∈ {c1,c2,….,cn} in fj

c) Elements in each cluster group are sorted based on client IP address (source)

d) l ,m=1

e) Length of Ci is Li, length of Ci, is consider as Lj

f) while l ≤ Li and m ≤ Lj

if Ci []l = Cj[m] then add to Pool[Mp] , i+1,j+1

else if C[l] ≥ C[m] then m+1

else l+1

g) go to step a), repeat these same steps for all fi and fj

In 5 min window, 65 mimicking attacks performed on the server. We experimented with different parameters on

the performance of the algorithm.

The above algorithm needs to be fast enough to identify the attack in quick time. We experimented running the algorithm with a single parameter and combination of parameters.

Idle time: Based on the idle time of the connection we can identify the attacks up to 27%. Because of multiple issues like latency and network congestion, idle time may come similar (or) can change.

Methods used: With this success rate was around 25%. Flash crowd sort of activities all users might be performing the same activity. It might be difficult to depend on this alone.

The time gap between two successive sessions: With this, we can identify 30% of mimic attacks. Though attack patterns might generate with same time gap based on the traffic and CPU load on host trigger of the next session with the same gap might not happen all the time.

Session duration: With this parameter able to identify 28% of attacks. Duration of the session might depend on the network latency and the load on server and client activities Results were recorded in Table 1 with these results we concluded it was difficult to identify a good number of mimicking attacks with a single parameter. We experimented with a combination of multiple parameters together.

Table 1. Mimicking attack identification with a single parameter

Parameter	Detection of mimicking attack in 5 min window
Idle time	12827
Methods	62225
Gap between session	93630
Duration of session	82828

We experimented running the algorithm by combining the parameters. the results of that were given in Table 2. The observation was when we combined all parameters only, we are getting the better results, with the combination of idle time, the gap between session, duration of the session, methods used in the session had given us around 83.2% success rate. Then we kept the overlapping of the data between three consecutive windows that has given us better results of all with that able to detect around 95% of the mimicking attacks.

Table 2. Mimicking attack identification by combining multiple parameters

Parameter	Detection of mimicking attack in 5 min window
Idle time & Gap between sessions & Session Duration	28 (43%)
Idle time & Gap between sessions & methods	42(64.5%)
Idle time & Gap between sessions & methods & Session duration	54 (83. 2%)
Idle time & Gap between sessions & methods & Session duration & overlapping time window	62 (95.3%)

Advantages:

- They Can have a complete view of the network, results are more reliable
- There won't be any single point of failures.
- The attacker even tries to bypass the individual threat defences still can be tracked easily
- Distributed and heterogeneous attacks also can be identified here

Limitations:

- It needs the involvement of all elements in the network
- Cost and maintenance of the system will be more
- Offline detection only can happen here. there might be cases by the timing analyzer identifies attack already some damage might have taken place.

3. Conclusion

As discussed, server level and host level detection might not have complete information about the mimic attack. The hybrid detection system will be sitting away from the servers and gateway which will have the bird view of activities happening in the network. It collects the statistics from different network elements and will consider as mimicking attack after observing the trace of the bot activities. We feel this is the better method for detecting mimicking attacks.

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, PP 139 - 151 DOI : DOI: 10.1109/TC.2013.191.

[2] Rama Krishna V, Dr R Subhashini Detecting HTTP Based Mimicking Attacks at HTTP Server, International Journal of Engineering and Technology (IJET), Vol 9 No 4 Aug-Sep 2017, PP 3041-3049, DOI: 10.21817/ijet/2017/v9i4/170904091.

[3] Rama Krishna V, Dr R Subhashini Mimicking attack by botnet and detection at the gateway, Peer-to-Peer Networking and Applications, January 2020, DOI: 10.1007/s12083-019-00854-9.

[4] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-based defence mechanisms countering the dos and DDOS problems", *ACM Computing Survey*, Vol. 39,no1,2007.DOI https://doi.org/10.1145/1216370.1216373.

[5] 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. DOI 10.1145/1592451.1592456.

[6] Rama Krishna V, Dr R Subhashini Botnet Algorithms Adaptability For Mimicking Attacks And Inducing Mimicking ATTACK Jour of Adv Research in Dynamical & Control Systems, Vol. 10, 06, 2018 PP 977-991.

[7] Rolf Oppliger, "Internet Security: Firewalls and Beyond". Communications of the ACM. 40 (5): 94 DOI: https://doi.org/10.1145/.

[8] Compuquip.com/blog/the-different-types-of-firewall-architectures.

[9] KoroshGolnabi, Richard K. Min, Latifur Khan, Ehab Al-Shaer, "Analysis of Firewall Policy Rules Using Data Mining Techniques", Network Operations and Management Symposium, 2006.NOMS 2006. 10th IEEE/IFIP, PP 1-10 doi=10.1.1.131.6122.

[10] E. Al-Shaer and H. Hamed. "Firewall Policy Advisor for Anomaly Detection and Rule Editing." Conference (IM'2003), March 2003, IEEE/IFIP Integrated Management, pp15-29, DOI https://doi.org/10.1007/11535706_15.

[11] Archana D Wankhede et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (1) , 2014,PP 674-678 ISSN:0975-9646.

[12] D.E. Denning, "An Intrusion Detection Model," IEEE Trans. Software Eng., Vol. SE- 13, No. 2, Feb.1987, pp. 222–232. DOI 10.1109/TSE.1987.232894.

[13] Karen Scarfone, Peter Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS)", National Institute of Standards and Technology, Technology administration US department of commerce 2007.

[14] Jennifer Jabbusch, "IDS vs. IPS: How to know when you need the technology", 22 November 2010, DOI 10.1007/978-3-642-22540-6_48.

[15] IEEE 2004 IEEE/IFIP Network Operations And Management Symposium - Seoul, South Korea (19-23 APRIL 2004, Seoul. PP 175-186.

[16] NabeelYounus Khan et al., "Comparative Study of Intrusion Detection System and Its Recovery Mechanism", IEEE, 2010, PP 54-61 doi:10.4236/cn.2010.21008.

[17] Zhisong PAN et al., "An Integrated Model of Intrusion Detection Based on Neural Network and Expert System", Proceedings of The 17th IEEE International Conference on Tools With Artificial Intelligence (ICTAI'05), 2005. doi.org/10.1007/s12083-019-00854-9.

[18] Gisung Kim et al., "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection", Expert Systems with Applications 41 (2014) PP 1690–1700 doi.org/10.1016/j.

[19] Mohammad SanieeAbadeh et al., "Design and analysis of genetic fuzzy systems for intrusion detection in computer networks", Expert Systems with Applications 38 (2011) PP 7067–7075. doi: 10.1016/j.eswa.2010.12.006 7068.

[20] NeminathHubballi et al., "LAN attack detection using Discrete Event Systems", ISA Transactions 50 (2011) PP 119 130. doi: 10.1016/j.isatra.2010.08.003.

[21]. Danielle, Andrew Moore, "X-means: Extending K-means with Efficient Estimation of the Number of Clusters", PP 727–734, January 2012 doi.org/10.1111/1467-9868.00293.

[22] Chaudhary, Alka, V. N. Tiwari, and Anil Kumar. "A new intrusion detection ... "Neuro-fuzzy based intrusion detection systems for network security." Journal of Global Research in Computer Science 5.1, pp. 1-2, 2014.

[23] del NadjaranToosi, Mohsen Kahani, Reza Monsefi, Network Intrusion Detection Based on Neuro-Fuzzy Classification, Computer department, Computing & Informatics, International Conference on Computing & Informatics, 1-5, 2006 PP 65 98.

[24] M. S. Abade, J. Habibi, C. Lucas, "Intrusion detection using a fuzzy genetics-based learning algorithm," Journal of Network and Computer Applications, August 2005. doi: 10.1109/ICOCI.2006.5276608.