Development of Firewall Optimization Model Using by Packet Filter

Myo Thant[1]
[1]*Defence Services Science and Technology Research Center.* PyiOoLwin, Mandalay Division, Myanmar, mmthant62@gmail.com

Kyaw Myat Thu[3]
[3]*Department of Automatic Control Systems,* Bauman Moscow State Technical University, Moscow, 105005 Russia, mr.kyaw.myatthu@ieee.org

Kyaw Zaw Ye[2]
[2]*Department of Informatics and Computer Software System,* National Research University of Electronic Technology, Zelenograd, Moscow, Russia, Kyaw.Z.Ye@ieee.org

Si Thu Thant Sin[4]
[4]*Department of Computer Science,* National Research University of Electronic Technology, Zelenograd, Moscow, 124482, Russia, sithuthantsin86@gmail.com

Abstract—This paper studies one particular aspect of providing communication security: firewall technology. This article provides a security framework in the form of packet filtering model within the firewall technology system and its components can be designed and evaluated. This paper introduces a reference model based on packet filtering firewall technology. All components are governed by a centralized security policy and they can be deployed in a distributed fashion to achieve scaling. This packet filtering firewall design is proven technology that provides confidentiality, integrity and availability in own information and network resources. The required design implement to set-up packet filtering firewall for network to protect and unauthorized access to our network. We describe the filtering process design depends on its underlying network technologies. The resulting security mechanism can be used as a bastion-host in the construction of firewall system.

Keywords— network; protocol; firewall; packet filtering; optimization.

I. INTRODUCTION

A packet filters is an active firewall element that analyzes and controls the inbound and outbound packets in the Network Access, Network, and Transport layers. It records and analyzes the packets (for example, Ethernet or token ring) that are transmitted over the physical cable. The interposition of the packet filter between the networks keeps them physically separate. A packet filter normally behaves like a simple bridge. Packet filters are not confined to TCP/IP protocols [1, 2].

A packet filter interprets the content of the packets and checks whether the data in the corresponding communication layer headers complies with the defined rule base. The rules are defined so that only necessary communication is allowed and settings know to pose a risk to security, such as IP fragmentation, are avoided. The packet filters occur at the network level so they are transparent to users.

The function of a packet filter can be com-pared to those of a security guard. When a supplier's truck drives up to the factory gates with a delivery, the packet filter security guard examine the logo on the side of the vehicle to see if it is familiar to him. If the vehicle passes this summary inspection, the filter security guard allows the truck to pass through the gate without checking the shipping manifest.

II. PACKET FILTER GENERAL MODE OF OPERATION

Figure illustrates the general manner in which packet filters work. It is important to know what information from the packet is used in the analysis.

Different checks can be performed in the different communication layers as determined by the specific capabilities and rule set in place.

- The party whom the packet is received is checked (this information is obtained from the integration module).
- In the network access layer, the following checks are performed:

A. The source and destination addresses

- The protocol type used
- In the network layer, depending on the protocol, the following checks are per-formed:

  - IP protocol (for example, the source and destination addresses and the layer 4 protocols well as the checkbox and flags)
  - ICMP protocol, such as ping and network redirect
  - IPX protocol (for example, network/ node)
  - OSI protocol (the OSI network address)
- In the Transport layer, the following checks are performed:

978-1-5090-0888-9/16 $31.00 © 2016 IEEE
DOI 10.1109/UKSim.2016.45

- With UDP/TCP, the port numbers (source and destination ports)

- Services such as FTP, Telnet and http have fixed port numbers associated with them.

- With TCP, the direction of the connection setup

- In addition, a check can be performed on packets as to whether access through the packet filter occurs within predefined time bands.

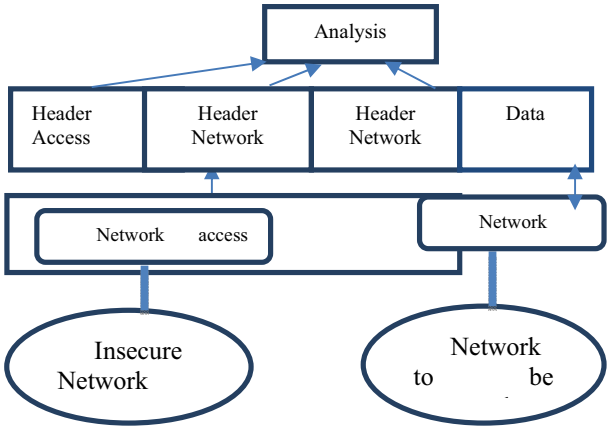


Fig. 1. General Mode of Operation of a Packet-Filter

The verification information is taken from the rule set (access list, authorization list) and compared with the analysis results. If the have been violated, a security - relevant event is logged, and if the relevant option has been set, a spontaneous message is set to security management with the log data related to the security-relevant event to appropriate and prompt action can be taken.

The following sections explain in greater detail what checks can be performed in the various communication layers. With intranets, checks in the network access layer are generally performed in the local area, while in the network and transport layers, checks are aimed at controlling communications over the Internet and intranets [3-5].

III. Checks in the Network Access Layer

Different standard must be support in the network access layer. The possibilities with an Ethernet are shown in Figure 1.

With Ethernet packets, the packet filter can analyze the destination and source addresses and check the corresponding rule set to see whether the computer system, servers, and routers to which the addresses (MAC addresses) belong are direct communication partner in the lowest communication layer (for example, application gateway, mail server, or DNS server) can be defined here. The data type, or DSAP/SSAP, field is used to specify which communication protocol for example, the IPX, IP or DECnet protocol will be used during communications in the next higher layer. The data type field definitions are specified in RFC1700.

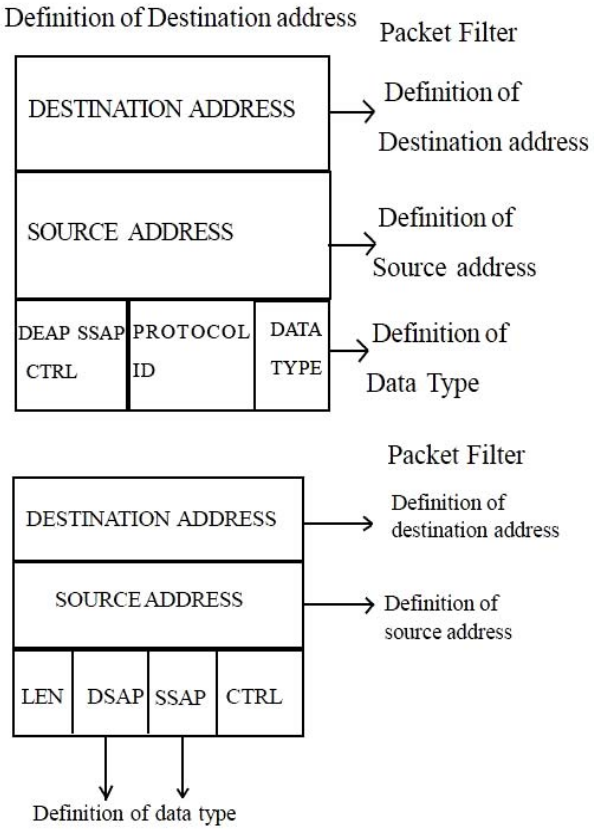


Fig. 2. Ethernet MAC Frame Structures

IV. Checks in the Network Layer

In the network layer, the destination and source addresses and the transport protocol are checked for an IP protocol. The network and node are checked for an IPX protocol.

Figure 3 shows the possibilities available with IP frames for performing an analysis to control communications through the packet filter (RFC791).

With an IP frame, destination and addresses are checked and compared with the ruleset to establish whether the communication is allowed to pass through the packet filter. You can also deduce which transport communications protocol is used from the protocol field. Here again, it is possible to check against the authorization list whether the corresponding transport communications protocol (such as TCP or UDP) may be used or not. The Flags field can them tell you whether the IP packets are fragmented. Because

fragmentations are vulnerable to attack, you should prohibit them by defining the rights appropriately[6,7].

The options field determines which options (source routing, and so on) are allowed through the packet filter. Source routing can and should be forbidden, as it is possible for attack to be carried out using this function.

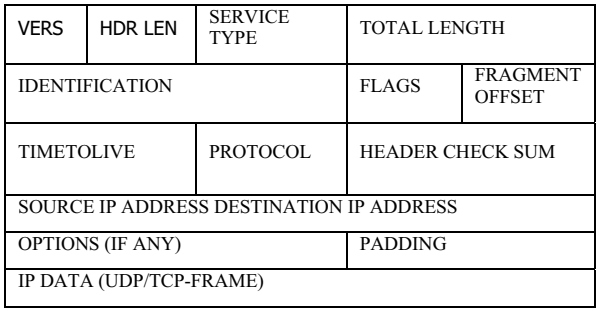


Fig. 3. Structure of the IP Frame

With ICMP (RFC792) it is possible to analyze the type field in which the commands are defined. Here, commands such as echo Request, redirect, and destination unreachable can be either allowed or forbidden. For example, echoRequest and echoReply, which are used for ping command, might be permitted, but the redirect command, which are used for attacks, should be forbidden. The commands are defined in RFC 792.

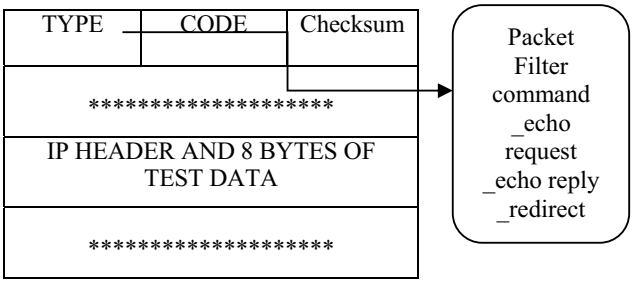


Fig. 4. Structure of the ICMP Frame

V. Checks in the Transport Layer

In the transport layer, a check of the port numbers is performed in the case of UDP/TCP (and indirectly for the TCP/IP applications HTTP, FTP, Telnet, and so on). In the case of TCP, the direction of the connection setup is also checked.

A. The UDP Transport Protocol

UDP is a connectionless communications protocol (seen figure 1.5). In other words, the UDP packets are transmitted independently of each other, with no guarantee or check that the packet is delivered correctly. No distinction is made between a new UDP connection setup and the packets within an existing UDP connection. With the UDP frame (RFC768), the packet filter can analyze the source and destination ports. Using an authorization list, it is possible to specify which services can be run over UDP (for example, SNMP, TFTP and so on).

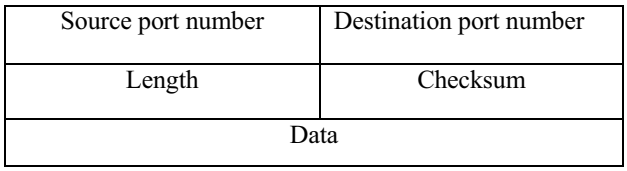


Fig. 5. Structure of the UDP Frame

As a rule, prohibit UDP packet if possible, because they can be used to carry out numerous attacks.

B. The TCP Transport Protocol

In figure 6. shows what information can be analyzing and checked with a TCP frame.

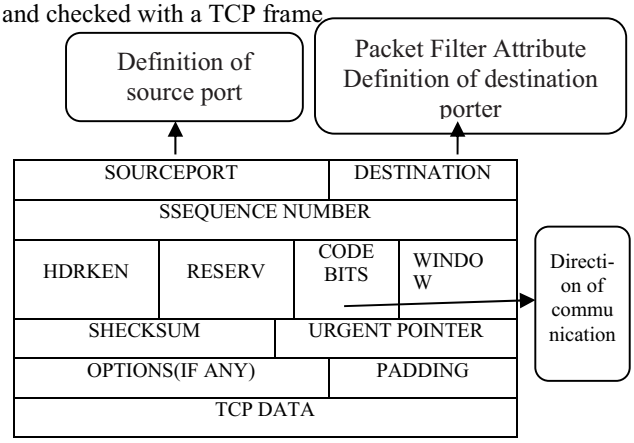


Fig. 6. Structure of the UDP Frame

With the TCP frame (RFC793), the packet filter can analyze the source and destination ports. Services allowed through the packet filter (at par-ticular times) are specified in an authorization list, Moreover, the direction in which the connection has been established can be identified from the code bits' field through interpretation of the ACK (ac-knowledge) bit. In this way, connections can be established in only one direction for security reasons.

C. Checking the Connection Setup

TCP is a connection-oriented communications protocol. During connection setup, TCP always works the ACK bit in the code bits' field (that is, ACK=0. All the other packets in a TCP connection have the ACK bits set (ACK=1) (Chapman Zwicky, 1996). This makes TCP-based applications easier to through a packet filter (see Figure-1.7).

D. Filtering with FTP Connection

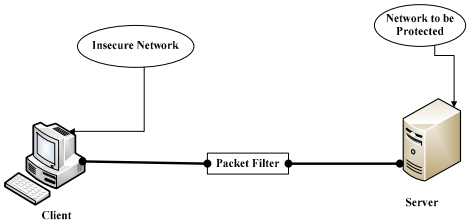
FTP applications (RFC959) work with two logical TCP connections: one for exchanging commands and the other for exchanging data. These logical TCP connections can be

set up using either an active or a passive method-depending on the viewpoint of the FTP client.

E. FTP Connection Setup

During an FTP connection setup, the client uses two port numbers above 1024 (for example, 4320 and 4321). The client establishes the TCP connection for the commands over the first port (for example, 4320). The server receives the command over the defined port 21.

The two methods by which the data channel can be established by the computer systems with FTP applications are described below.



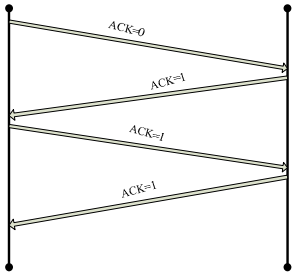


Fig. 7. Checking the Connection Setup

VI. ACTIVE METHOD

With the command PORT 4321, the client tells the server the number of the port over which it intends to process the data. The server sends the data from its defined port 20 to the client's port, number 4321.

A packet filter that controls these connections must enable the establishment of a TCP connection from the insecure network to the protected network. However, because this method poses a security risk, it should not be used if at all possible. Instead, using a passive FTP connection setup by the clients its recommended [8-11].

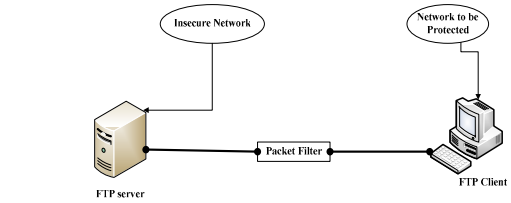
VII. PASSIVE METHOD

With the passive method, the client establishes the TCP connection. When combine with a packet filter, this method can result in a higher level of security. The passive method is illustrated in Figure 1.6.

Keep in mind that the passive method is not offered by all client/server FTP implementations.

A. Other Possible Stipulation

The times that each filter rule applies should be specified in the packet filters (for example, Monday to Friday form 8 am and 5 pm, Saturdays from 8 am to noon and at no time on Sundays.



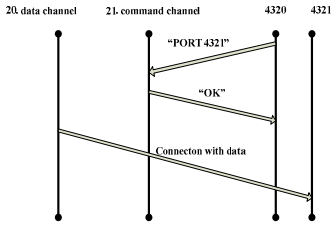
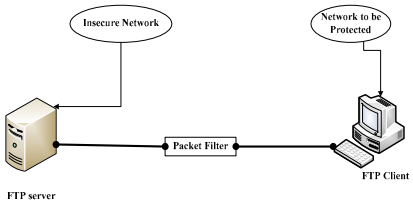


Fig. 8. Active method



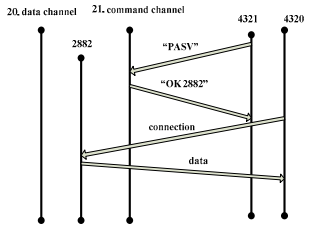


Fig. 9. Passive Method

B. Strategies for Setting Up and Evaluating the Filter Rule

You can take several possible approaches when determining the strategy for setting up and evaluating filter rules. Two strategies are presented in the following section.

C. Specify Positive Filter Rules

With the positive filter rules strategy, you specify exactly what types of packet are allowed.

- You must specify exactly what is to be allowed
- Everything that is not explicitly allowed is automatically for bidden.
- The firewall element only allows what is explicitly characterized as "allowed" in the access list.
- Specify Negative Filter Rules
- With the negative filter rules strategy, you specify exactly what packet are denied.

- Begin with the principle that every thing is allowed.
- Using specific entries, specify what is to be forbidden.
- The packet filter only prohibits what is explicitly characterized as "not allowed" in the access list.

D. Protocol Elements

The protocol elements that are exchange between transmitter and receiver can be defined as follows; sum of all protocol element E= 2n

Protocol element $X = \{ X_1...X_1, X_{t+1} + ... Xu, .X_{u+1}.....$

$$X_v, ,X_{v+1}, \ldots\ldots\ldots + X_n\}$$

The security policy specifies which protocol elements are permitted and which are not.

When considering the security of protocol elements {x1 xn}, the following additional points must be considered;

- Not all the fields in the protocol elements (for example, sequence numbers, random values, and so on) are security- relevant in relation to the capabilities of a firewall system which has the effect of sharply reducing the number of protocol elements that need to be considered in practice.
- Certain protocol elements are either permitted or not permitted, depending of a communications protocol. Protocol elements can be permitted in a particular state, yet not be permitted in another state.
- The possibility of transferring information over concealed channels is not considered in this model.
- Permitted protocol elements {X1....Xt}, can only be exchanged between permitted transmitters {t1 tg), and receivers{r1...rh}, at permitted times.

E. Action

An action (A) = {a1....af} consists of a defined number of sub action. The writing of a file to the receiver's hard disk using FTP is one example of an action. Sub actions include, for example, selecting the subdirectory, receiving data subsets, and saving data. These actions can all be categorized as either permitted or non-permitted for the purpose of evaluation.

- Permitted actions {a1...at}- Permitted actions are actions that are necessary for permitted application (task). Allowing permitted applications to access defined assets constitutes a calculable risk regarding the vulnerability of the assets. The range of permitted actions also includes error handling in connection with undefined states and events.
- Non-permitted actions { at+ 1...af}. Non-permitted actions are actions that, although they enable implementation of a communication protocol or service at the receiver's end, are not necessary for the actual task of the host system and therefore are not permitted in the interest of preventing intentional or unintentional damage.

The security policy specifies which actions are not permitted actions {at+1,....af} can only be initiated through permitted protocol elements {X1....Xt} which are exchange between permitted transmitters {t1,…,tg} and receivers {r1,....,rh} at permitted times.

VIII. FILTER FUNCTION

In this stage, we must filter the IP packets according to the repaired rules. We decided each forward or drop according to the rules. First we extract the IP header and examine the protocol TCP. We accept all packet of established connections due to bit syn active then we pass the packets. Otherwise we compare the packet with our rules. If the incoming protocol is the same we examine in source address destination address, source port & destination port. After that we decided what to do with the packets based on TCP, UDP protocols. Then we decided what to do required rules and decided to packet drop or packet forwards. The implementation stage of the filter function is shown below (Figure 10).

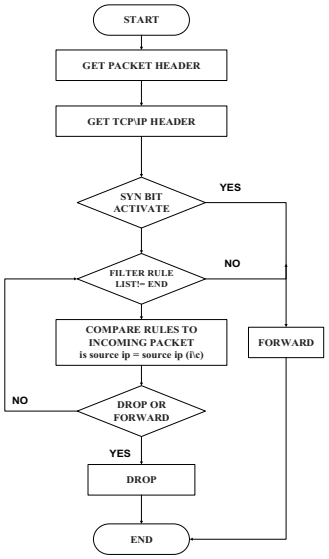


Fig. 10. Algorithm of filter function

IX. CLEAR FILTER FUNCTION

Finally, we remove the linked list where the rules were saved. Then we freed form the linked list. The implementation stage of the clear filter function is shown below (Figure 11).

A. Filter Hook Example

This section shows a simple filter hook that makes forward and drop decisions based on certain packet properties. This filter hook specifies to drop Transmission Control Protocol (TCP) packet and to forward packets from all other protocols.

If packets with specific IP address or TCP/ UDP port numbers must be filtered, consider creating a user-mode application that uses the Packet Filtering API instead This API optimize the system-supplied filter driver to process packets without the overhead that is associated with a filter-hook driver.

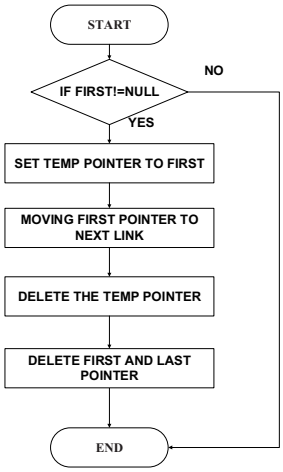


Fig. 11. Algorithm of clear filter function

```
#define PROT_ TCP 6
//Drop all TCP packets
PF FORWARD ACTION
Drop TcpPacket (
    unsigned char *Packet Header,
    unsigned char *Packet,
    unsigned int Packet Length,
    unsigned int RecvInterface Index,
    unsigned int SendIntereace Index,
    IPAddr Recvlink Next Hop,
    IPAddr Sendlink Next Hop,
    ) {
    If(PacketHeader -> iph_protocol = PROT_TCP){
        Return PF_DROP;} Return PF_FORWARD; }
```

X. CONCLUSION

Besides creating and initializing a device object, as all kernel mode drivers must, the filter hook driver's Driver Entry routine can register the driver's filter hook with the IP filter driver as described in setting and clearing a filter hook in the IP filter driver.

If user-mode applications or higher-level drivers send I/O control requests down to the filter-hook driver setup the filter hook as described in implementing IOCTLs for applications, driver entry must specify export an entry point that enables device control. This entry point is an IRP_MJ_DEVICE_CONTROL dispatch routine. If Driver Entry enables device control in this way, this device-control routine registers the drivers the driver's filter hook rather than Driver Entry.

Driver entry must specific and export an entry an entry point that unloads the filter-hook driver. This unload routine removes the device that created in Driver Entry but must not clear the previously registered filter hook when the operating system unloads the filter-hook driver.

ACKNOWLEDGEMENTS

The research work was supported by Laboratory of Information management systems and complexes of National Research University of Electronic Technology, Moscow, Russian Federation.

REFERENCE

[1]Kyaw Zaw Ye, Kyaw Zin Linn, Ba Hla Than, Hein Tun, Portnov E. M. Hierarchical dataflow control systems using by cloud computing network// ITA15 Sixth International Conference on Internet Technologies & Applications, Tuesday 8 - Friday 11 September 2015, Wrexham, North Wales, UK, 2015, pages 256–260

[2] Kyaw Zaw Ye; Naing Lin Zaw, "Modeling and Analysis of the Power Line Communication Channel," Engineering and Telecommunication (EnT), 2014 International Conference on , vol., no., pp.46,50, 26-28 Nov. 2014//doi:10.1109/EnT.2014.21, URL:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7121432&isnumber=7121413

[3] Kyaw Zaw Ye, Htike Aung Kyaw, Bain A. M., Portnov E. M. The efficiency of detecting the failures and troubleshooting while applying technical diagnostics for multi-computer systems// Archives of Control Sciences Volume 25(LXI), 2015, No. 1, pages 5–25

[4] Douligeris C. and Mitrokotsa A. " DDoS Attacks and Defense Mechanisms: Classification and State of the Art, " Computer Journal of Networks, vol. 44, no. 5, pp.643-666, 2004.

[5] S. Lee, G. Kim, and S. Kim, " Sequence-OrderIndependent Network Profiling for Detecting Application Layer DDos Attacks " , EURASIP J. Wireless Comm. and Networking, vol. 2011, no. 1, p. 50, 2011.

[6] T. Peng, C. Leckie, and K. Ramamohanarao, " Survey of Network Based Defense Mechanisms Countering the Dos and DDos Problems " , ACM Computing Surveys, vol. 39, no. 1, p. 3, 2007. 5, Mar. 2011.

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

[8] Y. Xie and S. Yu, " A Large-Scale Hidden Semi-Markov Model for Anomaly Detection on User Browsing Behaviors " , IEEE/ACM Trans.Networking, vol. 17, no. 1, pp. 54-65, Feb. 2009.

[9] Y. Xie and S. Yu, " Measuring the Normality of Web Proxies Behavior Based on Locality Principles, Network and Parallel Computing " , vol. 5245, pp. 61-73, 2008.

[10] Yash Pravinkumar Raithatha, Chirag Suryakant Thaker, " Various Methods used for the Protection, Detection and Prevention Layer DDoS Attacks, " IJCSIT, ISSN:2278-733X.

[11] Yi Xie, S. Tang, Y. Xiang and J. Hu, " Resisting Web Proxy-Based HTTP Attacks by Temporal and Spatial Locality Behavior " , IEEE transactions on parallel and distributed systems, VOL. 24, NO. 7, JULY 2013.