Chapter 14
Fuzzy Observation of DDoS Attack

Łukasz Apiecionek

Abstract DDoS attacks are able to block Web servers. Such attacks could be started from anywhere in the network. This chapter presents the possibility of using Ordered Fuzzy Numbers (OFNs) for observation of a DDoS attack. The proposed algorithm could be implemented on routers and predict the moment of the attack. Such prediction gives a possibility for the network administrators to protect server resources. In the chapter the author presents the real test results made on a prepared IP network. The presented results prove that OFNs have a huge potential for usage in observation of DDoS attacks.

Keywords DDoS · Security · Fuzzy logic · Fuzzy observation

14.1 Introduction

Today the main network used is the Internet which could be described as a wide area network (WAN). Many huge companies have many offices in different locations connected to the Internet. In this situation often some virtual private networks (VPN) are created to connect the locations in a secure way. This gives a possibility to block the company using a distributed denial of services (DDoS) attack on such systems. Such an attack could block the server from managing connections and thus resources and could generate financial loss for this company. That is why this problem is very common and should be solved.
14.2 DDoS Attack Description and Recognition

These attacks are well described in the literature [1–3]. There are many possible types of such attacks. One of them could use TCP/IP sockets vulnerabilities [2, 4], whereas others could use domain name system (DNS) server vulnerabilities. As mentioned, the main principle of such attacks is to try to utilize all server resources by generating a lot of ordinary user connections. The number of such connections can exceed the servers’ capability to handle them. Many papers [1–3, 5, 6] provide methods for dealing with DDoS attacks. These methods could be described as detection and the necessary cooperation between network providers. This is because the attackers send their packets through the network which belongs to a network provider. Therefore only network providers can block the attacker traffic. If they do not block such traffic, that traffic will saturate data links. This saturation causes blockage of the connection to the server [7] which could block some valuable portals such as e-learning platforms [8, 9]. Some of the methods for detecting DDoS attacks use general-purpose computing on graphics processing units [10], whereas others recommend protecting networks with a firewall [11, 12]. Intrusion prevention systems and intrusion detection systems use huge databases that consist of data collected during simple attacks from one place on the network [11]. But detecting attacks in a real worldwide network, such as a WAN, is a very complicated process. Some authors claim that it could be ensured by cooperation of routers, firewalls, and Internet providers [13–15]. Other possibilities include the use of fuzzy logic for this purpose [16, 17]. Such a solution requires a great deal of effort from experts who have to provide the rules describing the possible attack. Then, these rules are used for attack observation.

14.3 The Idea of Attack Recognition and Prevention

The proposed idea of attack observation, for recognition and prevention, is to limit performance of a network device during the attack that will limit the number of connections made to the target of the attack. Nowadays there are methods for limiting incoming traffic on a firewall and they allow the servers to deal with the already established connection. This should let the users finish their work and enable new users to connect to the server. Such known methods are quality of service (QoS) methods. QoS methods let the network administrators limit data traffic that could be described by many parameters such as source address, destination address, protocol, port, and so on. This solution also makes it possible for the router to count incoming traffic and decide which packet will be transferred as first and which will be the last.

There are some papers describing the QoS method idea that can work on one router and try to protect network resources locally [18]. But this solution does not recognize the source of the attack and does not solve the problem. When the packets are not blocked the hacker is still able to send packets to the server and block its resources. Nowadays routers are, of course, exchanging a lot of information between them.
The main information describes the reachability of the IP networks. This is done by routing protocols such as OSPF, EIGRP, BGP, or multicast routing protocols [19, 20]. This mechanism can be used for recognizing the DDoS attacks with some Ordered Fuzzy Numbers (OFNs) implemented. As has already been mentioned, QoS methods are able to count the number of packets, but they are not able to decide if the packet is part of a DDoS attack on a server. There is a need for new services for network providers. Those services should provide mechanisms for detecting the attackers’ packets to enable the router to block them on all the network routers, not only on the firewall that operates in front of the target of the attack. Such services should be implemented in an easy way, and it should use some well-known mechanism such as exchanging information between routers via routing protocols. Such a solution is the simple network management protocol (SNMP) which is used for acquiring knowledge on traffic statistics. The solution proposed by the authors for recognizing DDoS attacks using OFNs could be implemented on routers and require collecting traffic statistics. This method should not utilize too many resources of the routers, therefore it should also cooperate with the possible target of the attack. The author defines this method in the following steps.

- Server collects information about its traffic statistics (1) via SNMP from network routers.
- Using OFNs, its traffic statistics, and operating status, the server detects that it is under attack (2).
- Server establishes IPSec channel to network provider’s router (3).
- Server passes information about the type of traffic that has to be blocked by the network using SNMP over the IPSec channel (4).
- Router starts to block specific traffic (5).
- Router spreads the information about specific traffic that has to be blocked via a SNMP trap message to other routers (6).
- The specific traffic of the attacker is blocked via the network resources (7).

These steps are presented in Fig. 14.1. This idea is very simple and can be implemented very easily. The proposed algorithm uses SNMP and IPSec encryption. Some known aspects of the attack are used. The first is that the target of the attack is a server. The consequence of that fact is that the server should recognize it is under attack. The network alone is not able to decide if the server is under attack. Second, if the routers get the information that they have to block some specific traffic, they could do it. Routers could block specific traffic to the server using an already implemented mechanism such as random early detection. The process of passing the information from the server to the router that there is an attack has to be protected and both sides of the communication have to be authenticated. This could be achieved using a second already existing mechanism such as IPSec encryption and public-key infrastructure. Nowadays many organizations possess X.509 certificates signed by authorized certification authorities. The aforementioned secure communication channel, established by IPSec, could be used for passing the information from the server to the router that
the server is under attack. This message can also include information about the kind of traffic to be blocked in the network. There could also be SNMP used in a secured IPSec channel to pass the information. As mentioned, SNMP is also widely implemented on routers. Thus the network providers can implement this solution on their routers. The only thing that has to be done is to decide if the server is under attack. This process could be achieved using the aforementioned Ordered Fuzzy Numbers.

14.4 Attack Observation Using OFNs

As has already been mentioned, the system administrator has the possibility to check how many and which users are already connected to the server. He or she is also a person who possesses the knowledge about the prime time of the day in which users usually work with the system. Another issue that could be checked is how many TCP SYN connections actually come to the server by attempts to establish a TCP session. The last area that could be checked by the proposed algorithm, as mentioned before, is router statistics of packet transmission. That complete set of information should be enough to decide whether the server is under attack. In the proposed algorithm, the administrator will not use the fact that the number of connections is growing. The method will measure the packet count during network operation based on router statistics provided by SNMP. In the proposed algorithm, the administrator should measure the specific packet count four times:
where \( t_i \) is a current timeslot.

All four measures together give a fuzzy number in OFN notation where

- \( f_A(0) \) corresponds to \( t_{i-3} \)
- \( f_A(1) \) corresponds to \( t_{i-2} \)
- \( g_A(1) \) corresponds to \( t_{i-1} \)
- \( g_A(0) \) corresponds to \( t_i \)

This fuzzy number in OFN notation is presented in Fig. 14.2. This is a definition of fuzzy observance of a router.

**Definition 1**  Fuzzy observance of an R router in time \( t_i \) is a set

\[
R/t_i = \{ f_R(0)/t_{i-3}, f_R(1)/t_{i-2}, g_R(1)/t_{i-1}, g_R(0)/t_i \}
\]

(14.2)

where

\[
|t_i - t_{i-1}| = |t_{i-1} - t_{i-2}| = |t_{i-2} - t_{i-3}| = t_n, \text{ timeslot of the measurement}
\]

\[
|t_i - t_{i-1}| = |t_{i-1} - t_{i-2}| = |t_{i-2} - t_{i-3}| = t_n, \text{ timeslot of the measurement}
\]

This provides Lemma 1.

**Lemma 1**

\[
R_{\text{positive}} = \begin{cases} 
  f_R(0) < f_R(1) < g_R(1) \\
  f_R(1) < g_R(1) < g_R(0)
\end{cases}
\]

(14.3)

in other situations, \( R_{\text{negative}} \).
According to this definition during router observance the counters should give:

- Positive order of OFNs when the packet count **increases**
- Negative order of OFNs when the packet count **decreases**

The interpretation of these orders is presented in Fig. 14.3. Then the statistics collected on the routers are prepared and the appropriate counters give the results for preparing fuzzy numbers, and a fuzzy observance of the group of routers can be defined. Fuzzy observance of the group of routers is defined as

**Definition 2** Fuzzy observance of the group of routers is described by the formula:

\[
S_m = \sum_{i=1}^{n} \left\{ \frac{R_{positive}}{R_i \times w_i} \right\} - \left\{ \frac{R_{negative}}{R_i \times w_i} \right\}.
\]

(14.4)

where \(w_i \in \{w_1, \ldots, w_n\}\) describes an impact on all routers.

This provides a possibility to define the situation when DDoS should be detected on the router:

**Definition 3** An attack on the router is detected in the following conditions:

*If Ri is positive AND Ri is negative THEN Attack = true*

where \(R_i\) is an order from the history of statistical results according to that router on the time of day of the observance.

According to this, the situation when DDoS could be detected on a group of routers could be defined as follows.

**Definition 4** An attack on a group of routers is detected in the following conditions.

*If Sm is positive AND Sm’ negative THEN Attack = true*

where \(S_m\) is an order from the history of statistical results according to that group of routers on the time of day of the observance.
14.5 Experiment Test Results

In the following two subsections we provide descriptions of the test together with the results of attack detection by the proposed method.

14.5.1 Test Description

To test and prove the idea a special IP network was prepared. This test was inspired by the situation in Poland, when the Polish government attempted to sign the ACTA regulations. This situation caused a huge attack on Polish government websites. In consequence, those websites were blocked. The method of attack was very simple, because it was sufficient that many people tried to visit such websites which generated massive traffic to the servers. These servers were not prepared to manage so many connections exceeding their capacity. Finally the servers stopped responding to users’ queries. Such a method of attack could be simulated in an easy way. Some simple tools could be found on the Internet [21]. One of them is DDOSIM, Layer 7 DDoS simulator [22]. This solution is provided with the source code, thus it is possible to analyze it and check how it works. In a real situation, the attacker has to collect an appropriate number of hosts that could be used as sources of the attack. Once she has them, the attack can be started. In the prepared simulation, it is enough to just run the DDOSIM program with the appropriate parameters. After that, this tool will generate the defined amount of connections to the defined IP address. To perform this test a special network with mesh topology (shown in Fig. 14.1) was prepared. The user hosts labeled from 1 to 5 were running DDOSIM software in the appropriate cycle. This station was equipped with an Intel i3 processor running under Windows 7 64-bit system control. But the DDOSM software was running on a virtual machine in a VMWare environment. That virtual machine was equipped with 512 MB of memory and 1 processor and was running under Debian operating system control and the Web server was equipped with two Intel Xeon processors with a Windows 2008 server operating system. An Internet information service was started on the Web server. It was used to provide an HTTP server functionality. The routers R1 to R6 were Cisco 2600 series routes. To simulate the attack the DDOSIM software requested information from the Web server using a TCP connection in the following steps.

- User sends TCP SYN packet to port 80.
- Web server answers with TCP SYN ACK and reserved resources.
- User sends TCP ACK packet.
- User sends HTTP/GET as a request for information packet.
- Server attempts to answer this request for information packet.
The cycle of simulation was prepared in the following steps and according to the following conditions.

- The IP address of the Web server was 192.168.10.12.
- The user hosts from 1 to 5 have got IP addresses 192.168.x.4, where x belongs to the set 1, 2, 3, 4, 5.
- During the attack the packets were sniffed using Wireshark; there were six places where packets were sniffed: user 1 to 5 machines and the Web server.
- The network was running normally.
- After 1 min of normal operation the host of user 1 started the attack.
- The target of the attack, of course, was the Web server.
- The user machines were sending request packets using DDSOIM software with these parameters: there were 1,000 HTTP GET messages passed to the Web server every 30s.
- Another user host joined the attack every minute.
- Once all the user’s hosts started the attack, all of them attacked together for five minutes.
- When the attack stopped, the packets were sniffed for another five minutes.

This test provides a lot of data from the network node using files with whole IP packets from six places of the network. The database from this simulation including the traffic from all the connections can be downloaded from data resource page [23]. Additional packets that could be recognized in the collected database include the open shortest path first routing protocol which was running between routers as in ordinary networks [18, 23, 24]. In the network prepared for the simulation, to ensure better test conditions, no quality of service method was implemented. Also there was no firewall or IDS/IPS implemented in the network whereas it should normally be implemented [25–27]. This was because the authors claim that in a real network the packets which come from ordinary users are not treated as special packets. Moreover the Web servers should not be under any protection in order to check that the method of the attack is strong enough and the database collected could be treated as a valuable amount of data (Fig. 14.4).

14.5.2 Attack Detection Using Proposed Method

Statistics of packets have to be provided to use a proposed method for detecting attacks using OFNs. According to the collected database, the results could be calculated from sniffed packets, and this was done. There is a requirement for defining timeslots for which the statistics will be calculated. Those timeslots were defined to be one minute. Then the amount of packets directed to the Web server and that pass through the routers R1 to R6 was normalized by dividing by 1,000. The achieved results of this simulation are presented in Table 14.1.
To calculate a sum described by Definition 2, the OFNs have to be defined by the statistics collected from the routers. In the timeslots $t_0$ the normalized value is 0. It means that all numbers are $[0, 0, 0, 0]$. Thus, accordingly:

$$S_m = 0.$$  \hspace{1cm} (14.5)

This has not got any order, but the results provide the information that there is no packet directed to the Web server, therefore there is no attack on it.

In the timeslot $t_1$ it is possible to provide the OFN with an approximation. This approximation could be made by providing the OFN with only two measurements and is utilized as a four-part number by using the values as follows.

- $f_A(0)$ corresponds to $t_0$.
- $f_A(1)$ corresponds to $t_0$.
- $g_A(1)$ corresponds to $t_1$.
- $g_A(0)$ corresponds to $t_1$.
In this situation the OFN for the routers 1 to 6 could be described as

- For router R1 it is \([0, 0, 1, 1]\).
- For router R2 it is \([0, 0, 0.5, 0.5]\).
- For router R3 it is \([0, 0, 0.5, 0.5]\).
- For router R4 it is \([0, 0, 0.5, 0.5]\).
- For router R5 it is \([0, 0, 0.5, 0.5]\).
- For router R6 it is \([0, 0, 1, 1]\).

Therefore the \(S_m = [0, 0, 2.5, 2.5]\) with positive order. Those results according to Definition 4 have to be compared with the historical results in the part corresponding to an appropriate day. In the performed simulation the results indicate that the Web server is under attack, but the probability of the attack is 50% because the OFN was prepared with the aforementioned approximation. In the timeslot \(t_2\) it is still possible to provide the OFN only with an approximation. This approximation could be done by providing the OFN from only three measurements and used as a four-part number by making use of the following values.

- \(f_A(0)\) corresponds to \(t_0\).
- \(f_A(1)\) corresponds to \(t_1\).
- \(g_A(1)\) corresponds to \(t_2\).
- \(g_A(0)\) corresponds to \(t_2\).

In this situation the OFN for the routers 1 to 6 could be described as

- For router R1 it is \([0, 1, 1, 1]\).
- For router R2 it is \([0, 0.5, 1, 1]\).
- For router R3 it is \([0, 0.5, 1, 1]\).
- For router R4 it is \([0, 0.5, 1, 1]\).
- For router R5 it is \([0, 0.5, 1, 1]\).
- For router R6 it is \([0, 1, 2, 2]\).

Therefore \(S_m = [0, 2.5, 6, 6]\) with positive order. Those results according to Definition 4 have to be compared with the historical results in the part corresponding to an appropriate day. In the performed simulation the results indicate that the Web server is under attack, but the situation of the attack could be described with the probability of 75% because the OFN was also provided with an approximation. In the timeslot \(t_3\) it is possible for the first time to provide the OFN without any approximation. It means that for the OFN four measurement results will be used as follows.

- \(f_A(0)\) corresponds to \(t_0\).
- \(f_A(1)\) corresponds to \(t_1\).
- \(g_A(1)\) corresponds to \(t_2\).
- \(g_A(0)\) corresponds to \(t_3\).

In this situation the OFN for the routers 1 to 6 could be described as

- For router R1 it is \([0, 1, 1, 1]\).
- For router R2 it is \([0, 0.5, 1.5, 1.5]\).
### Table 14.2  The sum calculation results

| Time | Sum          | Trend     | Situation          |
|------|--------------|-----------|--------------------|
| $t_0$| $[0, 0, 0, 0]$| –         | No attack          |
| $t_1$| $[0, 0, 2.5, 2.5]$| Positive | Attack in 50%     |
| $t_2$| $[0, 2.5, 6, 6]$| Positive | Attack in 75%     |
| $t_3$| $[0, 4, 7.5, 9.5]$| Positive | Attack            |
| $t_4$| $[4, 7.5, 9.5, 12.5]$| Positive | Attack            |
| $t_5$| $[7.5, 9.5, 12.5, 14.5]$| Positive | Attack            |
| $t_6$| $[9.5, 12.5, 14.5, 14.5]$| Positive | Attack            |
| $t_7$| $[12.5, 14.5, 14.5, 14.5]$| Positive | Attack            |
| $t_8$| $[14.5, 14.5, 14.5, 14.5]$| Positive | Attack            |
| $t_9$| $[14.5, 14.5, 14.5, 14.5]$| Positive | Attack            |
| $t_{10}$| $[14.5, 14.5, 14.5, 0]$| Negative | No attack         |
| $t_{11}$| $[14.5, 14.5, 0, 0]$| Negative | No attack         |
| $t_{12}$| $[14.5, 0, 0, 0]$| Negative | No attack         |
| $t_{13}$| $[0, 0, 0, 0]$| Negative | No attack         |

- For router R3 it is $[0, 0.5, 1, 2]$.
- For router R4 it is $[0, 0.5, 1, 1]$.
- For router R5 it is $[0, 0.5, 1, 1]$.
- For router R6 it is $[0, 1, 2, 3]$.

Therefore $S_m = [0, 4, 7.5, 9.5]$ with positive order. Those results, as in previous timeslots, have to be compared with the historical results in the part corresponding to an appropriate day. In the performed simulation the results indicate that the Web server is under attack. Using Table 14.1 the OFN for each router in subsequent timeslots was defined and $S_m$ was calculated. Those results are presented in Table 14.1. Obviously, the order was described and the decision about a possible attack was made. The timeslot of interest is $t_{10}$. In this situation the OFN for the routers 1 to 6 could be described as

- For router R1 it is $[1, 1, 1, 0]$.
- For router R2 it is $[1.5, 1.5, 1.5, 0]$.
- For router R3 it is $[2.5, 2.5, 2.5, 0]$.
- For router R4 it is $[2, 2, 2, 0]$.
- For router R5 it is $[2.5, 2.5, 2.5, 0]$.
- For router R6 it is $[5, 5, 5, 0]$.
- All of them with negative order.

Therefore $S_m = [14.5, 14.5, 14.5, 0]$ with negative order. Those results, as in previous timeslots, have to be compared with the historical results in the part corresponding to an appropriate day. In the performed simulation the results indicate that the Web server is no longer under attack. The same situation could be recognized in timeslot $t_{11}$. In this situation the OFN for the routers 1 to 6 could be described as follows.
• For router R1 it is \([1, 1, 0, 0]\).
• For router R2 it is \([1.5, 1.5, 0, 0]\).
• For router R3 it is \([2.5, 2.5, 0, 0]\).
• For router R4 it is \([2, 2, 0, 0]\).
• For router R5 it is \([2.5, 2.5, 0, 0]\).
• For router R6 it is \([5, 5, 0, 0]\).
• All of them with negative order.

Therefore \(S_m = [14.5, 14.5, 0, 0]\) with negative order, which means that there is no attack recognized (Table 14.2).

### 14.6 Conclusions-Method Comparision

The presented method could be compared with the method proposed in the literature as provided in Table 14.3.

As shown in Table 14.3, the proposed method does not require an expert to define the rules of possible attack as in a system with fuzzy logic proposed in some papers [16, 17]. Such an expert would have to possess extended knowledge about security and IP networks. This is something that allows us to use the proposed method in a very quick way. The second thing is the manner of gathering results of the observance of a DDoS attack. In the proposed method, the decision about the attack is made using simple calculations provided by the OFN description. This allows achieving the results very quickly and easily. The methods found in the literature make decisions by comparing a list of rules [16, 17]. Of course there are some solutions that use mathematical models [28], but they are much more complicated than the method using OFNs. The last thing that could be compared is the possibility of being implemented in a real environment, which means in real networks. The method proposed in the literature requires a lot of processing power. The method proposed in this chapter requires only solving a simple mathematical equation. This is very important, because it lets us use the proposed method in a real network, on real routers.

| Functionality                        | Existing method with fuzzy logic | Proposed method                  |
|--------------------------------------|----------------------------------|----------------------------------|
| Expert work                          | Method requires an expert, who has to define the rules of the attack | Method does not require an expert to define the rules of the attack |
| Result of DDoS attack observance     | Decision made by comparing a list of rules | Decision made by simple calculations |
| Possibility to implement a in real environment | Requires lot of processing power | Requires only solving a simple mathematical equation |
The presented new concept of detecting DDoS attacks was introduced using fuzzy numbers. As has already been mentioned, it consists in determining the server activity trend changes by specification of the direction of changes using OFNs. The test was performed on the real dataset collected in the prepared network. This dataset is about 2 GB in size, but the information about the possible attack could be achieved without utilizing a lot of CPU performance.

References

1. Apiecionek, L., Romantowski, M.: Secure IP network model. Comput. Method Sci. Technol. 19(4), 209–213 (2013)
2. Moore, D., Shannon, C., Brown, D.J., Voelker, G.M., Savage, S.: Inferring internet denial-of-service activity. ACM Trans. Comput. Syst. (TOCS) 24(2), 115–139 (2006)
3. Rocky, K., Chang, C.: Defending against flooding-based distributed denial-of-service attacks: a tutorial. IEEE Commun. Mag. pp. 42–51 (2002)
4. Schuba, C.L., Krsul, I., Huhn, M.G., Spafford, E.H., Sundaram, A.: Analysis of a denial of service attack on TCP. Computer Science Technical reports. paper 1327. http://docs.lib.purdue.edu/cstech/1327 (1996)
5. Cert advisory CA-1996-21 TCP SYN flooding and IP spoofing attacks. http://www.cert.org/advisories/CA-1996-21.html
6. http://www.cert.org/advisories/ca-1996-01.html
7. Cert advisory CA-1996-01 UDP port denial-of-service attack, September 1997. http://www.w3schools.com/browsers/browsers_stats.asp (1997)
8. Mikolajewska, E., Mikolajewski, D.: E-learning in the education of people with disabilities. Adv. Clin. Exp. Med. 20(1), 103–109 (2011)
9. Świerczewski, L.: Intel manycore testing lab - hardware and software environment focused on didactic of development and efficiency testing in software paralleling [in polish]. Scientific Bulletin of Wroclaw School of Information Technology 03, Wroclaw (2013)
10. Vokorokos, L., Ennert, M., Hartinger, M., Radušovský, J.: A survey of parallel intrusion detection on graphical processors. In: Proceedings of International Scientific Conference INFORMATICS 2013, 5–7 November 2013, Spišská Nová Ves, Slovakia (2013)
11. http://www.ll.mit.edu/mission/communications/cyber/cstcorpora/ideval/data/ (2014)
12. Thatte, G., Mitra, U., Heidemann, J.: Detection of low-rate attacks in computer networks. In: INFOCOM Workshops 2008, pp. 1–6. IEEE (2008)
13. Apiecionek, L., Czerniak, J.: Qos solution for network resource protection. In: Proceedings of International Scientific Conference INFORMATICS 2013, 5–7 November 2013, Spišská Nová Ves, Slovakia (2013)
14. Apiecionek, L., Czerniak, J., Dobrosielski, W.: Quality of service method as a DDoS protection tool. Advances in Intelligent Systems and Computing, pp. 225–234. Springer, Berlin (2015)
15. Apiecionek, L., Czerniak, J.M., Zarzycki, H.: Protection tool for distributed denial of services attack. Beyond Databases, Architectures and Structures, BDAS, vol. 424, pp. 405–414. Springer, Berlin (2014)
16. Dickerson, J.E., Dickerson, J.A.: Fuzzy network profiling for intrusion detection. In: Fuzzy Information Processing Society, 2000 NAFIPS, 19th International Conference of the North American, pp. 301–306. IEEE (2000)
17. Dickerson, J.E., Juslin, J., Koukousoula, O., Dickerson, J.A.: Fuzzy intrusion detection. In: Joint 9th IFSAS World Congress and 20th NAFIPS International Conference 2001, vol. 3, pp. 1506–1510. IEEE (2001)
18. KOVÁČ, D.: Modern internet based production technology. In: Joo Er, M. (ed.) New Trends in Technologies: Devices, Computer, Communication and Industrial Systems, pp. 145–164. SCIYO (2010)

19. Kosiński, W.: On fuzzy number calculus. Int. J. Appl. Math. Comput. Sci. 16(1), 51–57 (2006)

20. Kosiński, W., Prokopowicz, P., Ślężak, D.: On algebraic operations on fuzzy numbers. Intelligent Information Processing and Web Mining, pp. 353–362. Springer, Berlin (2003)

21. http://resources.infosecinstitute.com/dos-attacks-free-dos-attacking-tools/ (2014)

22. http://sourceforge.net/projects/ddosim/ (2014)

23. Piechowiak, M., Zwierzykowski, P.: The evaluation of unconstrained multicast routing algorithms in ad-hoc networks. In: The International Science Conference: Computer Net-works CN2012, Szczyrk, Poland (2012)

24. Piechowiak, M., Zwierzykowski, P.: The evaluation of multicast routing algorithms with delay constraints in mesh networks. In: 8th IEEE, IET International Symposium on Communication Systems, Networks and Digital Signal Processing CSNSDP 2012, Poznań, Poland (2012)

25. Chapman, B., Zwicky, E.: Building Internet Firewalls. O’Reilly & Associates, Inc. (1995)

26. Cheswick, W., Bellovin, S.: Firewalls and Internet Security: Repelling the Wily Hacker. Addison-Wesley, Reading (1994)

27. Wrzesień, M., Olejnik, L., Ryszawa, R.: IDS/IPS: detection and prevention systems of hacking the computer networks [in polish]. Stud. Mater. Appl. Comput. Sci. 4(7), 16–21 (2012)

28. Shangytbayeva, G.A., Srazhatdinovich, B., Akhmetov, M.P.K., Beysembekova, R.N., Ospanov, E.A.: Research distributed attacks in computer networks. Biosci. Biotechnol. Res. Asia 12(1), 737–744 (2015)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.