STUDIA INFORMATICA 2017
Volume 38 Number 3 (132)

Daniel CZYCZYN-EGIRD, Rafał WOJSZCZYK
Koszalin University of Technology, Poland

DDOS ATTACKS PREDICTION IN A SIMULATION ENVIRONMENT BY MEANS OF DATA MINING TECHNIQUES

Summary. The notion of Internet attacks has been well-known in the area of computer networks for a long time now. The effects of these actions can be difficult to rectify and also very expensive. Therefore, these harmful attacks should be detected in the shortest time possible when the effects are still quite easily reversible. The article presented the results of the research on predicting the occurrence of DoS attacks on the selected network resources by using data mining techniques.

Keywords: computer networks, data mining, DDoS attack

PRZEWIDYWANIE ATAKÓW DDOS W ŚRODOWISKU SYMULACYJNYM PRZY UŻYCIU TECHNIK EKSPLORACJI DANYCH

Streszczenie. Pojęcie związane z atakami sieciowymi jest znane w tematyce sieci komputerowych już od dawna. Efekty ataków sieciowych są trudne do naprawienia i bardzo drogie. Dlatego też wskazane jest jak najszybsze wykrywanie ataków, tak aby ich skutki były jak najmniej dotkliwe. Artykuł przedstawia wyniki badań dotyczących przewidywania wystąpienia ataku DDoS na wybranych zasobach sieciowych przy użyciu technik eksploracji danych.

Słowa kluczowe: sieci komputerowe, eksploracja danych, atak DDoS

1. Introduction

In this day and age of constant and instant development of computers, computer systems and networks there are many risks related to network attacks aiming at data stealing, destroying and denying access to data [1]. One may avoid many of these attacks by observing the basic principles of information security; however, there are also those attacks with which

one has to use special protective strategies and systems which not always guarantee security with one hundred percent certainty.

In the whole list of network attacks, the Distributed Denial-of-Service (DDoS) attack can be found as one of more serious threats and more common attacks aiming at denying access to information services. DDoS attacks consist in generating enormous packages by a great number of systems-agents in order to exhaust computing and communication assets in a quite short time. The most common effect of these actions is making resources and services unavailable to a victim.

The article aims at investigating the relationship and attempting to predict DDoS attack in the selected test area of the computer network. The results of the research may be used to determine parameters and trends having the influence on the behaviour of network traffic, while the scope and methods of the research may show effective usefulness of data mining tools.

The second chapter presented general information concerning DDoS attacks, their goals and possible countermeasures. The third chapter included the definition of the research environment and plans of action. The fourth chapter introduced the research results and drawn conclusions. The summary was included in the last, fifth, chapter or the article.

2. Introduction to Distributed Denial-of-Service (DDoS) attacks

2.1. Safety is important

Nowadays, the Internet is the prevailing information medium, and not only as a mass medium but also as a platform of access to entertainment and culture. One may observe that today people are being increasingly active in the virtual zone and this often happens at the expense of a real life. The Internet has become an integral part of human life. More new social services are being established [2] and they enable users to communicate freely and establish new contacts; shopping can be done in online stores without living one's house, and gaining knowledge does not necessarily have to do with thumbing voluminous encyclopaedia's pages – everything is available in the global network. Some of the today's activities are slowly becoming essential in an electronic form – it saves a lot of time. An interesting example of online activities is online banking thanks to which one does not have to stand in a line in the bank's seat anymore to perform financial operations. Money can be transferred with the use of a computer, and, if help is needed, there is a so-called virtual customer assistant one can use. It is easy to imagine the situation where one transfer all his

money to the bank to pay by cards and payment terminals. It all works until there is a banking system failure – an example here can be an unpredicted DDoS attack which cut off the users from their banking systems. Disabled cash machines and payment terminals result in no access to financial funds of many people – panic breaks out.

Therefore, the issues of security in the virtual world are very important. Unawareness of a number of threats possible in the computer sphere may lead to grievous losses, also in the financial dimension. Consequently, one should predict certain dangers in advance and try to minimize losses through quick reactions to undesired actions unless preventing the attacks is possible.

2.2. Specific character of DDoS attacks

Apart from the dangers mentioned in the previous section, network resources are exposed to another type of attacks for which a common thread is denying access to network services. These are Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks [3]. The visualization of DDoS attack is illustrated in Fig. 1. These are the dangers that may result from the attacks:

- interrupting HTTP requests – problems with access to websites and server applications,
- interrupting data transferring supported by database servers,
- stopping print enqueuing in the case of a print server,
- mail servers cannot send and receive messages,
- overloading the line of network devices (e.g. router) which may result in shutting local networks from the Internet.

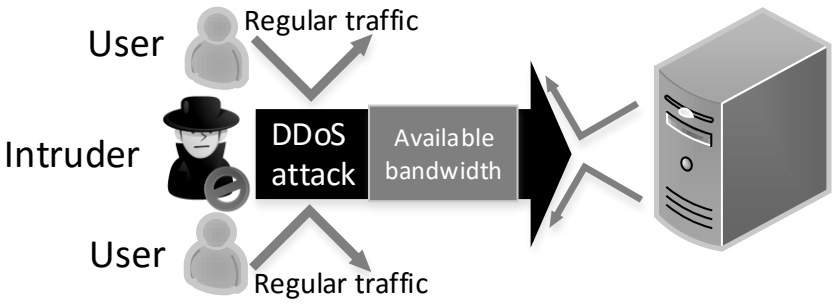

Fig. 1. The visualization of DDoS attack
Rys. 1. Wizualizacja ataku DDoS

In bygone days, the aim of DoS attacks was disabling the service by means of different mechanisms taking advantage of deficiencies of Transmission Control Protocol/Internet

Protocol stack and security vulnerabilities in the specific operating systems. Currently, to deny service, one uses the possibility of generating great traffic interrupting the operation of network applications, server resources and the network itself, and also defects of mechanism of establishing the session of TCP/IP connection. Links or servers are not able to support and process too many requests sent in a short time.

A part of DoS attacks are possible because network hosts use a source IP address or certificates (which can be copied) with authentication. Another problem concerns control mechanisms and routing protocols which use poor authentication methods for the source of information or they are not applied at all. DoS and DDoS attacks may be divided into 3 groups:

- attacks based on TCP/IP standards – taking advantage of flaws in specification in a particular operating system,
- attacks based on TCP/IP standards – regardless of an operating system,
- brute force attacks. Attacks of this kind generate great traffic, seize a network band or also server resources.

The results of DoS and DDoS attacks can be divided into three main groups [4]:

- destruction of resources – damaging selected objects in a data stream through their destabilisation, causing them to function improperly. Improper input objects may lead to destruction of system infrastructure. The reason for this situation can be improper size or improper options of received packets which cannot be handled by a port,
- exhaustion of resources – overloading the resources in a way that the information sent is not received in a specific time. Server resources (mainly processor time and operation memory assigned to a request) are limited, therefore, any process requesting more resources than provided can be blocked,
- denial of services – using the processes of resetting devices, disabling them temporarily or giving the control over them to other process. The denial of service is managed by a system, in order to maintain reliability, by shutting TCP connections. Consequently, for particular source and target addresses, connections are rejected for a specific time.

2.3. Countermeasures concerning DDoS attacks

The basic defensive technique is filtering of incoming packets – securing the network by using firewall tools including sets of network traffic rules on edge routers, analysing the current flow of packets. The protection consists in blocking the traffic that seems to be

suspicious. There are certain good practices applied in the first basic stage of network protection, for instance:

- disabling the possibility of generating network traffic with source addresses that do not belong to the previously provided pool of IP addresses – the network secured in that way could not participate in the attack,
- rejection of packets of source addresses that do not belong to our network or to a pool of private class addresses which are not reserved,
- limitation of attempts of logging to routers; for instance, after three authorization failures, a particular IP address is temporarily disabled – there are no requests from it processed (it is important during DDoS attacks on edge routers),
- forbidding others to send insets with a broadcast address and rejecting ICMP packets.

Another countermeasure concerning mass requests is an option of configuring services so that the maximum number of simultaneous connections is defined. This setting can be applied to one computer-client (one IP address); however, this will not be effective in case of distributed attack launched by multiple attackers of multiple addresses. An additional problem can be the fact that many hosts can properly use one common address, for example, by using the so-called NAT – network address translation. Consequently, different hosts visible by a server as one IP may be treated as a potential attacker and cut off from resources (mail, FTP server, Internet services). One should bear in mind that routers' and firewalls' security may fail to provide sufficient protection against more sophisticated attacks. A hint that could help is additional throughput provided by an Internet service provider which will be activated as a substitute one in case of failure. In case a rapid increase in traffic is observed, link reserves are applied for a time of attack. Still, this is only an emergency solution easing the symptoms and not fighting the causes of the problem.

The countermeasures should be more advanced since firewalls and routers themselves do not protect the user from a compromised client station's access in an internal network. The security should take a form of a vast multi-level architecture that should prevent DDoS attack attempts from the network which includes servers enabling their services. Therefore, another element of defence can be adding subsequent nodes filtering internal traffic which will not allow unverified clients to use the services. Thanks to this action, requests will not be sent to servers directly but will be first filtered and routed to them through indirect stations - agents. When client wants to communicate with server it should first be authorized and determine whether it has proper authorization for selected services. This approach is illustrated in Fig. 2.

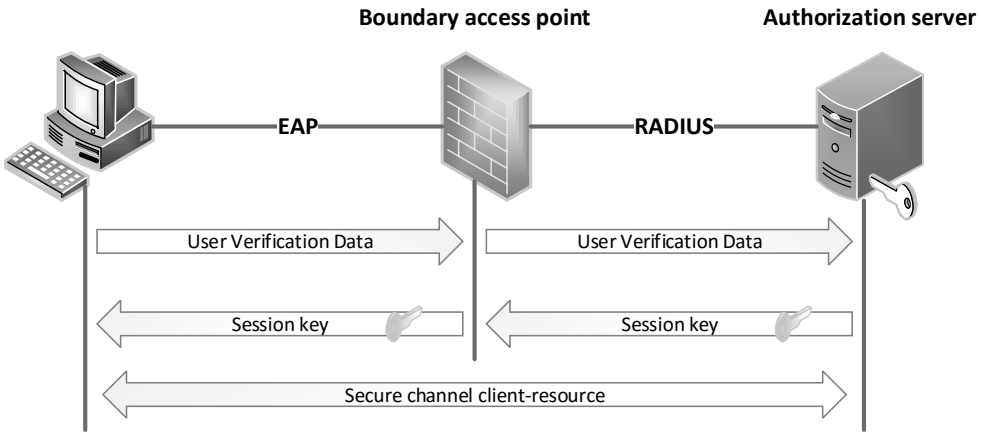

Fig. 2. Typical authorization protocol architecture
Rys. 2. Typowa architektura protokołu autoryzacji

2.4. Related works and background

Several authors have focused on the detection of signatures of DDoS attacks using various methods such as artificial neural networks, statistical methods, data mining techniques, etc. Literature surveys of these techniques are presented below.

Neural network based methods aggregate already identified patterns related to DDoS attacks and develop a neural network that can analyse the traffic in a network and decide whether a DDoS attack is in progress or not. An example of this technique has been shown in [5]. An Author described operates in two stages. In the first stage it monitors various features of the traffic and estimates the ratios for a DDoS attack. In the second stage the algorithm combines the result of each feature identified and the results are forwarded to the neural network that provides the final decision whether a DDoS attack has been detected or not. The result of such techniques are heavily dependent on the selection of network parameters.

Statistical-based methods monitor and model normal traffic patterns by using advanced statistical analysis techniques and are able to detect anomalies based on a predefined threshold [6]. This type of technique may provide relatively accurate results depending on the statistical analysis technique used. This approach however, it may increase the false positive detections or false negative detections [7].

Data mining based techniques have also been introduced in the detection of DDoS attacks. Data mining algorithms can be employed in the automatic feature selection for monitoring and the classification of the network traffic patterns [1]. The author proposed data mining algorithms such as Support Vector Machine, K-Nearest Neighbour Algorithm and Nave Bayes to make predictions of DDoS attacks. Support Vector Machine (SVM) is a classification data mining algorithm which can use to group entities by using SVM it can

easily group packet received. Network packets contain source mac address in the header part of the data packet. When sniffing network from monitoring machine all packets can be recorded in a database. It can do in real-time. By analysing that data with SVM algorithm it generates a graph. Analysing is doing considering frequently of mac addresses recorded and content inside data packet. If similar packet receive it can detect by visualization that data. If similar packets received from different networks it can be a DDOS attack.

Second option which has been used by author [1] is K-Nearest Neighbour as a data mining algorithm that make predictions. It takes a decision by comparing most nearby element in a graph. Using nearby element input can be classified into one of a group. By using this factor geographically nearby positions can be detected in real-time. This research suggesting to use Google API for to find location geographical location of IP addresses. If visualization of graph with longitude against latitude shows there is extremely high density in some geographical location. It can be a DDOS attack. However, intruder can impersonate different location, this is the disadvantage of this approach.

In another publication [5], the author touches the issue of predicting DDoS attacks using data mining algorithms such as FCM cluster algorithm and Apriori association algorithm. However, the author focuses only on the use of the input coming from network software such as sniffer or firewall (network traffic data and network packet protocol data), therefore the author focuses strictly on network parameters.

Most of the methods presented in this section are based on the network parameters and their analysis. The proposed paper would enlarge these above-mentioned ideas to develop methods that operate on the hardware parameters (instead of network parameters) which were generated in the original test environment.

3. Preparation for the research

3.1. System of increased resistance

The system of increased resistance to DDoS attacks, unlike typical solutions, e.g. 802.1X standard, which actually does not define an identity verification protocol, has been enriched with a developed algorithm of multi-stage bidirectional authentication [9]. This process takes place in case of every occurring pair of components. The aforementioned requirement that components of each pair identify each other has to be met and will provide the security of sent data even in case of taking over an edge component, that is, agent, by an intruder. This additional assumption makes the authentication procedure more complex. Instead of one-

level client identity verification in relation to server, the described protocol offers three times longer process.

The adopted solution involves three-stage procedure of verifying identity. The first stage includes authentication between a client (user component) and an agent (an edge device equivalent) then between an agent and server enabling resource. Only after a positive execution of the two stages, there is a connection established between a client and an authorization server. To verify identity of each pair of protocol components, the developed authentication algorithm was used. First component (it can be client, agent or server) generates SHA-1 checksum for a sent message. Then the resultant character string is encoded with a private key which is only in the first component. The message and encoded checksum (signature) is sent to the second component which decrypts the signature with the public key of the first component (keys are symmetric, RSA-generated, of 1024 length). Subsequently, the checksum is calculated for the message by means of an identical hash function as in case of first component. If the calculated checksum and the one decrypted from the message are the same, then the first component is authenticated. Otherwise, the request is rejected.

What is more, one should notice that in the assumed solution this is server which establishes a connection with a client and not the other way round as it happens in the standard solutions. Consequently, the external users do not have a direct access to the main point of the system architecture - they may not even know it exists.

This execution causes that DDoS attack launched by an intruder from the outside may only disable the agent component and, in the worst case scenario, disconnect authorized clients which are connected with it. The main protected resource, that is, the server, stays intact. However, there is still a possibility of attempting DDoS attack with multiple usage of the same certificate from many clients. On the other hand, this threat can be relatively easily eliminated by introducing limitation for the maximum number of simultaneous connections (sessions) for a client.

3.2. Simulation environment

For the simulation purposes, the system of increased resistance to DDoS attacks was implemented as three applications of Microsoft .NET technology [10], in C# language. In the simulation application of this type, there is no typical domain model as in case of business applications, therefore, implementation with the use of architecture patterns as MVC or MVP or a popular three-layer implementation were groundless [11]. However, application modularity was maintained and the selected areas (cryptographic functions, static data, simple

data model) were allocated to separate libraries which are shared by all components. What is more, .NET platform standardized libraries were used, including [12]:

- *System.Net* provides essential libraries to handle basic network mechanisms, e.g. operation of a server waiting for connection and a client which will connect to the server; it also provides classes representing an IP address or network data stream object,
- *System.Net.Socket* is a subspace for *System.Net* space. It includes classes responsible for communication by means of sockets,
- *System.Security.Cryptography* includes classes responsible for generating keys and handling a digital signature.

Figure 3 shows the first stage of the whole process. The marked piece represents an area of activity when the agent application is waiting in an infinite loop for messages from clients. Each client is handled in a separate thread of application thanks to which the requests are handled concurrently. The parameters recorded for the purposes of further experiment concern one thread of application (that is, one request from a client) or a state of the whole agent application. The example parameters selected for the experiment are as follows [13]:

- *DateTime.Now* (*DateTime* type) – time of request,
- Difference between *GC.GetTotalMemory()* readed befor and after handle request (*Float* type),
- Number of active requests (*Integer* type),
- The amount of *Process.***memory* (*NonpagedSystemMemory*, *PagedMemorySize*, *PagedSystemMemorySize*, *VirtualMemorySize*, *WorkingSet* – physical memory usage, *PrivateMemorySize*), in bytes, allocated for the associated process (Each record has been standardized, because some of the recorded are saved incrementally) (*Float* type),
- *Process.Threads.Count* – gets the number of threads that are running in the associated process (*Integer* type),
- *Process.***ProcessorTime* – gets the total, user or privileged processor time for this process (*Float* type),
- *Customer* type – intruder or an authorized user (*String* type).

The experiment was done only for the parameters recorded in the first stage, that is, communication between a client and an agent, since here is the place of first contact with a potential intruder. However, on this basis, recording parameters and predicting attacks can be transferred to any other pair of components or even an independent library which can be disseminated as a verified solution.

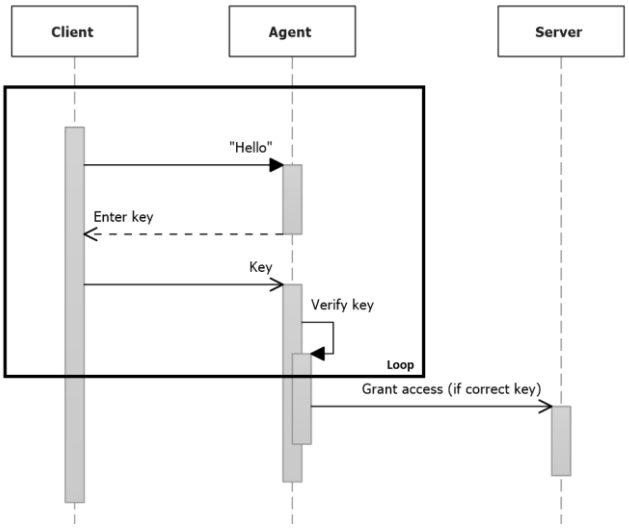

Fig. 3. Sequence diagram with first stage of protocol
Rys. 3. Diagram sekwencji z pierwszą fazą protokołu

3.3. Assumptions and the course of the experiment

The assumptions of the experiment aimed at emphasizing autonomy of the system, that is, the fact that it can be used as an independent solution, e.g. in dedicated made to measure business applications. The main assumptions are as follows:

- ignoring network analysers and tools of third entities in general,
- possibility of using the system from different devices and places; therefore, filtering requests on the basis of an IP address of ACL access list is impossible,
- disabling internal defensive mechanisms which are embedded in the simulation environment,
- total compromising of a client, that is, taking over all client data by an intruder with a requirement that a real (trusted) client should also have an access.

The course of the experiment consists in performing simulation with the use of the previously described simulation environment. During the 24-hour simulation, one agent component and one server component were assembled; additionally, one of the client components was simulating real traffic while additional instances of the client component simulated DDoS attack according to the aforementioned assumptions.

After the simulation process, a data set was obtained; subsequently, this set was structured for the purposes of Microsoft Excel software which was used to predict the attacks. Techniques classification were confined entirely to Naive Bayes classifier and K-Nearest Neighbours classification, since they declare bigger effectiveness [14] and popularity rather than different, what translate to better support to application programs.

4. Research results

4.1. Preliminary data processing

The data set obtained from the simulation included more than 227 000 unique records. Thereafter, the data set was divided into three subsets, corresponding to various period of time, which each containing attack. The subsets are characterized by different intensity of the trusted clients net traffic:

- Set 1 – the requests registered in the morning i.e. 9:38-10:12, with average exertion of the net traffic,
- Set 2 – the requests registered in the rush hours i.e. 14:52-15:29, with the greatest net traffic,
- Set 3 – the requests registered at night i.e. 22:11-22:59, with the infinitesimal net traffic of trusted clients.

Each collection have been submitted for further described classification. Unfortunately , on the grounds of technical reasons, it was necessary to divide mentioned sets into smaller, fulfilling max size of the training set and max size of the prediction set from table 1 conditions. What means, that in the case of set 2 and Naive Bayes classier was necessary to conduct five classification, which results were summed up.

4.2. The classification of the clients requests

Naive Bayes classifier and K-Nearest Neighbours classification [15] were used for each subset. The analysis was conducted in the public environment of Microsoft Excel using a dedicated additive package XLSTAT [16]. Each classification has been made with the selection of different values for the parameters, according to table 1.

Table 1

Classification parameters

Naive Bayes classifier		K-Nearest Neighbours	
Ties handling	Rnd. breaker	Number of neighbours	3-10
Prior distribution	Empirical	Metrics/Distance	Euclidean
Smoothing parameter	1	Ties handling	Smallest index
Training set	6336	Training set	12672
Prediction set	1584	Prediction classes	3168
Cross-validation / Number of folds	2	Cross-validation / Number of folds	2
		Weighted vote	Inverse squared distance

Table 2

Classification results

Overview	Naive Bayes classifier	K-Nearest Neighbours	The sample size of data	
			Trusted	Intruders
Set 1	29%	30,5%	950	34000
Set 2	33,4%	36,3%	17367	76000
Set 3	25%	25,9%	352	22000

Table 2 shows the classification results of the three subsets in regard to two methods of the classification. The present result is the percentage of correctly classified requests.

The analysis of the obtained results indicates that for the test environment of choice and recorded parameters, the most effective is the KNN [14]. However, in the critical case the most effective is the KNN classification with 7 number of neighbours [17].

The effectiveness in the range of 25% - 36,3% is relatively low, compared to other studies [8]. However, it is worth mentioning that, there was an extremely critical case in the used environment test, i.e. the intruder completely intercepted client's identity, not only the verification of IP addresses but also intervals of requests from the client or intruder were omitted. In that case, different methods may prove to be insufficient, moreover, they require additional tools connected with analysing TCP/IP network, what is inconsistent with the main assumption of the experiment.

4.3. The effects of a DDoS attack

During the simulations, each attack at the culminant moment, proved to be effective, i.e. the movement of trusted clients was blocked partially. On the basis of the mechanisms built into the simulation environment, it was estimated that, at the peak of 7% trusted clients did not gain the access to the agent's component, at the same time, up to 38.2% of intruder's requests did not gain the access to the agent. This tendency indicates that, the developed simulation environment may be a fundament to a generative implementation in business. Unfortunately, the sum of abovementioned results indicates that as many as 45.2% of the data has not been registered by the agent's component and is not included in the analysed data sets. A potential solution of this problem is to launch simulations in the cloud, for example Azure. Then the scalability of the cloud can provide better capability than in the case of a single computer, what leads to recording more data input.

5. Summary

The proposed paper develops methods that operate on the hardware parameters instead of network parameters. In the experiment the simulation environment consisting of three components increasing resistance to DDoS attacks was used. The environment was realized as three independent applications in the Microsoft. NET technology. During the experiment the movement of trusted clients and attacks of intruders on agent's component was simulated. For each registered connection to agent's component selected parameters were saved. Based on registered data the analysis with using techniques of data mining was conducted. The analysis' result shows that KNN method characterizes the most effectiveness in the classification of demands , it will do as device for further research. The correctness of the classification in the range of 25% - 36,3%, taking into consideration, the critical assumptions of the simulations give gratifying results.

The experiment shows that 7% of the trusted clients have been rejected in the culminant moment of the attack. Further work envisage the extension of used data mining techniques to ensure greater effectiveness of the attacks detection. It is worthy to extend the simulation environment of the recording additive parameters. Furthermore, the integration of the simulation environment with the cloud computing could provide the access to more specific data.

BIBLIOGRAPHY

1. Bandara K.R.W.V., et al.: Preventing DDoS Attack Using Data Mining Algorithms. International Journal of Scientific and Research Publications, vol. 6, issue 10, 2016, p. 390÷400.
2. Czyczyn-Egird D., Wojszczyk R.: Determining the Popularity of Design Patterns Used by Programmers Based on the Analysis of Questions and Answers on Stackoverflow.com Social Network. 23rd Conference on Computer Networks, CCSI, Springer, vol. 608, 2016, p. 421÷433.
3. HeeKyoung Yi, et al.: DDoS Detection Algorithm Using the Bidirectional Session. 18th Conference on Computer Networks, CCIS, Springer, vol. 160, Ustroń 2011, p. 191÷203.
4. Giovanni C.: Topology of Denial-of-Service. Endeavor Systems Inc. 2000.

5. Gulay O., Georgios L.: A Denial of Service Detector Based on Maximum Likeli-hood Detection and the Random Neural Network. Computer Journal, vol. 50, issue 6, November 2007, p. 717÷727.
6. Thapngam T., Yu S., Zhou W., Makki S. K.: Distributed Denial of Service (DDoS) Detection by Traffic Pattern Analysis. Peer-to-Peer Networking and Applications, 2012, p. 1÷13.
7. Rahmani H., Sahli N., Kamoun F.: DDoS Flooding Attack Detection Scheme Based on F-divergence. Computer Communications, vol. 35, 2012, p. 1380÷1391.
8. Zhong R., Guangxue Y.: DDoS Detection System Based on Data Mining. Proceedings of the Second International Symposium on Networking and Network Security ISNNS ’10, Jinggangshan, P. R. China 2010, p. 62÷65.
9. Górski G.: Novel Multistage Authorization Protocol. Information Systems Architecture and Technology: Service Oriented Networked Systems, Wroclaw University of Technology, Wrocław 2010, p. 221÷230.
10. Griffiths I., Adams M., Liberty J.: Programming C# 4.0: Building Windows, Web, and RIA Applications for the .NET 4.0 Framework. O'Reilly Media, 2010.
11. Wojszczyk R.: The Process of Verifying the Implementation of Design Patterns – Used Data Models. Advances in Intelligent Systems and Computing, vol. 521, 2017, p. 103÷116.
12. Troelsen A.: Pro C# 2008 and the .NET 3.5 Platform. Apress, New York 2007.
13. https://msdn.microsoft.com/en-us/library/ms123401.aspx
14. Ashari A., Paryudi I., Tjoa M.: Performance Comparison Between Naïve Bayes, Decision Tree and k-Nearest Neighbor in Searching Alternative Design in an Energy Simulation Tool. International Journal of Advanced Computer Science and Applications, vol. 4, no. 11, Bradford UK 2013, p. 33÷39.
15. Bishop C. M., Pattern Recognition and Machine Learning, Springer, 2006.
16. https://www.xlstat.com/en/
17. Hassanat A. B., et al.: Solving the Problem of the K Parameter in the KNN Classifier Using an Ensemble Learning Approach. International Journal of Computer Science and Information Security, vol. 12, no. 8, Pittsburgh USA 2014, p. 33÷39.

Omówienie

Pojęcie związane z atakami sieciowymi jest znane w tematyce sieci komputerowych już od dawna. Efekty ataków sieciowych są trudne do naprawienia i prowadzą do bardzo dużych,

nieprzewidzianych kosztów. Dlatego też wskazane jest jak najszybsze wykrywanie ataków, tak aby ich skutki były jak najmniej dotkliwe.

Artykuł przedstawia wyniki badań dotyczących przewidywania wystąpienia ataku DDoS na wybranych zasobach sieciowych przy użyciu technik eksploracji danych. Na potrzeby eksperymentu zostało opracowane środowisko symulacyjne z zaimplementowanym protokołem o podwyższonej odporności na ataki DDoS. Przebieg pierwszej szczegółowo przeanalizowanej fazy protokołu przedstawia rysunek 3. W trakcie symulacji rejestrowane były różne wartości, dostarczone przez technologię implementacji, tj. Microsoft .NET. Zarejestrowane wartości posłużyły następnie do próby przewidywania ataku.

Ataki były przewidywane przez dwa algorytmy z dziedziny eksploracji danych. Użycie klasyfikatora Bayesa dostarczyło wyniki skuteczności do 33,4%, natomiast KNN wykazało skuteczność do 36,3%. Tabela 2 przedstawia szczegółowe wyniki, natomiast w tabeli 1 zostały przedstawione parametry klasyfikacji.

Addresses

Daniel CZYCZYN-EGIRD: Koszalin University of Technology, Department of Computer Engineering, ul. Śniadeckich 2, 75-453 Koszalin, Poland,
daniel.czyczyn-egird@cicomputer.pl

Rafał WOJSZCZYK: Koszalin University of Technology, Department of Computer Science and Management, ul. Śniadeckich 2, 75-453 Koszalin, Poland,
rafal.wojszczyk@tu.koszalin.pl