2015 IEEE 13th International Scientific Conference on Informatics

INFORMATICS 2015

PROCEEDINGS

Editors

Valerie Novitzká
Štefan Korečko
Anikó Szakál

Organized by

Slovak Society for Applied Cybernetics and Informatics

Faculty of Electrical Engineering and Informatics, Technical University of Košice

Association of Slovak Scientific and Technological Societes

IEEE SMCS Technical Committee on Computational Cybernetics

IEEE

informatics 2015

November 18-20, 2015
Poprad, Slovakia

ISBN 978-1-4673-9867-1
IEEE cat. no. CFP15E80-PRT

2015 IEEE 13th International Scientific Conference on Informatics

INFORMATICS 2015

November 18-20, 2015, Poprad, Slovakia

PROCEEDINGS

Editors
Valerie Novitzká
Štefan Korečko
Anikó Szakál

Organized by
Affiliated branch of the Slovak Society for Applied Cybernetics and Informatics at Department of Computers and Informatics, FEEI TU of Košice
Faculty of Electrical Engineering and Informatics, Technical University of Košice
Association of Slovak Scientific and Technological Societes
IEEE SMCS Technical Committee on Computational Cybernetics

ISBN 978-1-4673-9867-1
IEEE catalog number CFP15E80-PRT

2015 IEEE 13th International Scientific Conference on Informatics

IEEE Catalog Number CFP15E80-PRT
ISBN 978-1-4673-9867-1

Additional copies of this publication are available from
Curran Associates, Inc.
57 Morehouse Lane
Red Hook, NY 12571 USA
+1 845 758 0400
+1 845 758 2633 (FAX)
email: curran@proceedings.com

COMMITTEES

General Chair

prof. Ing. Liberios Vokorokos, PhD., *Dean of Faculty of Electrical Engineering and Informatics, Technical University of Košice, SK*

Honorary Committee

akad. prof. Ing. Ivan Plander, DrSc., *Slovak Society for Applied Cybernetics and Informatics, SK* (chair)
Imre J. Rudas, *Óbuda University, Budapest, HU*
C. L. Philip Chen, *University of Macau, CN*

Program Committee

Valerie Novitzká, *Technical University of Košice, SK* (chair)
Mikuláš Alexík, *University of Žilina, SK*
Miklós Bartha, *Memorial University of Newfoundland, CA*
Andreas Bollin, *University of Klagenfurt, AT*
Dmitriy B. Buy, *National University of Taras Shevchenko,* Kyiv, UA
Svetlana Cicmil, *University of the West England, Bristol, UK*
Zbigniew Domański, *Częstochowa University of Technology, PL*
Erik Duval, *Katholieke Universiteit Leuven, BE*
János Fodor, *Óbuda University, HU*
Zoltán Fülöp, *University of Szeged, HU*
Gianina Gábor, *University of Oradea, RO*
Ján Genči, *Technical University of Košice, SK*
Andrzej Grzybowski, *Częstochowa University of Technology, PL*
Klaus Haenssgen, *Leipzig University of Applied Sciences, DE*
Tamás Haidegger, *Óbuda University, Budapest, HU*
Aboul Ella Hassanien, *Cairo University, Giza, EG*
Zdeněk Havlice, *Technical University of Košice, SK*
Pedro Rangel Henriques, *University of Minho, Braga, PT*
Pavel Herout, *University of West Bohemia, Pilsen, CZ*
Ladislav Hluchý, *Slovak Academy of Sciences, Bratislava, SK*
Elke Hochmüller, *Carinthia University of Applied Sciences, AT*
László Horváth, *Óbuda University, Budapest, HU*
Zoltán Horváth, *Lóránd Eötvös University, Budapest, HU*
Štefan Hudák, *Technical University of Košice, SK*
Péter Kádár, *Óbuda University, Budapest, HU*
Waldemar W. Koczkodaj, *Laurentian University, CA*
Milan Kolesár, *Slovak University of Technology, Bratislava, SK*
Levente Kovács, *Óbuda University, Budapest, HU*
Jiří Kunovský, *Brno University of Technology, CZ*
Sandra Lovrenčić, *University of Zagreb, HR*
Ivan Luković, *University of Novi Sad, RS*
Dragan Mašulović, *University of Novi Sad, RS*
Karol Matiaško, *University of Žilina, SK*
Marjan Mernik, *University of Maribor, SI*
Igor Mokriš, *Slovak Academy of Sciences, Bratislava, SK*
Hanspeter Mössenböck, *Johannes Kepler University Linz, AT*
Günter Müller, *Albert-Ludwigs-Universität Freiburg, DE*
Hiroshi Nakano, *Kumamoto University, JP*
Mykola S. Nikitchenko, *National University of Taras Shevchenko, Kyiv, UA*
Lucia Pomello, *University of Milano-Bicocca, IT*
Jaroslav Porubän, *Technical University of Košice, SK*
Stanislav Racek, *University of West Bohemia, Pilsen, CZ*
Sonja Ristić, *University of Novi Sad, RS*
Imre J. Rudas, *Óbuda University, Budapest, HU*
Gábor Sági, *Hungarian Academy of Sciences, Budapest, HU*
Abdel-Badeeh M. Salem, *Ain Shams University, Cairo, EG*
Elena Somova, *University of Plovdiv, BG*
William Steingartner, *Technical University of Košice, SK*

Jiří Šafařík, *University of West Bohemia, Pilsen, CZ*
Petr Šaloun, *University of Ostrava, CZ*
Michal Štepanovský, *Czech Technical University in Prague, CZ*
József K. Tar, *Óbuda University, Budapest, HU*
Katarína Teplická, *Technical University of Košice, SK*
Tsuyoshi Usagawa, *Kumamoto University, JP*
Mladen Vouk, *North Carolina State University, USA*
Neven Vrček, *University of Zagreb, HR*
František Zbořil, *Brno University of Technology, CZ*
Jaroslav Zendulka, *Brno University of Technology, CZ*
Doina Zmaranda, *University of Oradea, RO*

Organizing Committee

Milan Šujanský, *Technical University of Košice, SK* (chair)
Anikó Szakál, *Óbuda University, Budapest, HU* (financial chair)
William Steingartner, *Technical University of Košice, SK* (general manager)
Katarína Feciľaková, *Technical University of Košice, SK*
Sergej Chodarev, *Technical University of Košice, SK*
Štefan Korečko, *Technical University of Košice, SK*
Csaba Szabó, *Technical University of Košice, SK*
Marek Čopjak, *Technical University of Košice, SK*

PREFACE

It is with pleasure that we hereby present the proceedings of the *2015 IEEE 13th International Scientific Conference on Informatics*. Its success builds upon our ever improving efforts to publish with higher standards in the various areas of informatics. As part of this, our conference becomes a significant international forum for presenting original research results, sharing experience, and exchanging new ideas. The topics of this conference cover theoretical and practical results, along with methods for transferring these research results into real-life domains, by scientist and experts working in computer science and informatics. The conference also provides an opportunity for young researchers to demonstrate their achievements and to discuss their results at an international scientific forum. The main topics of the conference are as follows:

- Computer Architectures
- Computer Networks
- Theoretical Informatics
- Programming Paradigms, Programming Languages
- Software Engineering
- Distributed Systems
- Computer Graphics and Virtual Reality
- Artificial Intelligence
- Knowledge Management
- Information System Research
- Applied Informatics and Simulation

The proceedings start with keynote lectures from three eminent experts in their respective fields. Every paper in these proceedings has been peer-reviewed by two independent external referees. On behalf of the Programme and Organizing Committees, we would like to thank all the reviewers for their time and effort in reviewing the papers. Their contribution has ensured the high quality of publications in these proceedings. We would also like to extend our thanks to all the authors and keynote speakers, who contributed and guaranteed the high and professional standard of this conference.

The *2015 IEEE 13th International Scientific Conference on Informatics* has been organized by the following organizations:

- Faculty of Electrical Engineering and Informatics, Technical University of Košice
- Slovak Society for Applied Cybernetics and Informatics at Department of Computers and Informatics
- Association of Slovak Scientific and Technological Societies

We also thank the following sponsors:

- IEEE Hungary Section
- IEEE SMC Chapter, Hungary
- IEEE Joint IES/RAS Chapter, Hungary

and technical co-sponsor:

- IEEE SMC Society

The conference is held in Poprad, an historical city nestled at the foot of the High Tatra Mountains, a region rich in culture and natural beauty. We are confident that inside these covers, you will find papers relevant to your field of interest. We also look forward to your participation at the next event of this conference.

Poprad, 2015 November

On the behalf of the Programme and Organizing Committees
Valerie Novitzká

TABLE OF CONTENTS

Invited Papers

Regular Papers

Multi Path Transmission Control Protocols as a security solution

Łukasz Apiecionek
Institute of Technology
Kazimierz Wielki University
Bydgoszcz, Poland
lukasz.apiecionek@ukw.edu.pl

Wojciech Makowski
TELDAT Sp. z o.o. sp.k.
Bydgoszcz, Poland
wmakowski@teldat.com.pl

Marcin Sobczak
Institute of Mechanics and Applied Computer Science
Kazimierz Wielki University
Bydgoszcz, Poland
marsobczak@gmail.com

Tibor Vince
Department of Theoretical and Industrial Electrical Engineering, Faculty of Electrical Engineering and Informatics
Technical University of Kosice
Kosice, Slovak Republic
tibor.vince@tuke.sk

Abstract— **This article presents the potential of using Multi-Path Transmission Control Protocols for more secure data transfer. Data transfer over the network could be sniffed and then some decryption process could be made. Nowadays lot of network have more than one connection to others. The authors presents idea of Multi-Path TCP which could be used in such networks which have more than one connection to public network. The authors proposed method for special data transfer which could increase security of data and according to Multi-Path TCP the speed of transfer will increase too. The main advantage of this idea is that it could be used without a lot of work. This article is a part of authors papers focused on IT security.**

Index Terms—**security, TCP, Multip Path TCP**

I. Introduction

There is a necessary to protect data against lost nowadays because losing data costs companies a lot. That is why data are well protected against unauthorized access to network [[1]] or cloud systems [[2]] where it could be stored. Data are mostly encrypted before they are transfer over the network. Usually the stronger crypto algorithms are used according to amounts of data and time in which data should be transferred. To be sure that data reach their destination there are Transmission Control Protocols used.

Nowadays network consist more than one connection to public network as Internet is. This connection are used not in parallel, but worst for a company is used as a second when first will stop working. Omitting more than one network connection could increase transfer speed and security level of transfer data process. When only one connection is used, hacker could sniff packets in one place of network and collect data. Then some decryption method could be used to get real information. More than one network connection will force hacker to work on more place to sniff packets. Sometimes it will require knowledge on more than one transfer technology. Without it, hacker will lost possibility to collect all data packets, so it will not be possible do decode it. There is a one technology ready to use which could provide such possibility Multi-Path TCP (MPTCP in short). MPTCP work on operation system level which is very important according to simplicity of using it. MPTCP technology was proposed in this paper to increase security level of transmitted data. Chapter II describe MPTCP technology according to its features and implementation status. Chapter III presents algorithm and concept of using MPTCP in security aspects for faster and more secure transfer. It also presents results of real test made between Kazimierz Wielki University in Poland and Technical University of Kosice. Finally some conclusions and remarks for future work are presented.

II. Multi-Path TCP

To start talking about MPTCP should briefly explain the concept of TCP [8], which is used to transfer data between processes running on different machines. TCP can send data in two directions between two hosts. Unique identifier of the TCP connection are two pairs of values (one for each side of the connection) – IP and port number. TCP using checksums and sequence numbers provides a complete and orderly data exchange for higher layer applications. The header contains all the necessary details to establish a connection.

978-1-4673-9868-8/15/$31.00

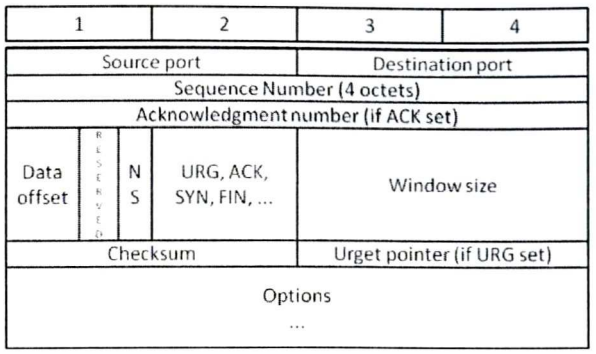

Fig. 1. TCP header

Before application starts to transmit information, it is necessary to exchange initialization data. Host A sends a segment with set SYN flag, then host B confirms that it received packet and send back SYN and ACK flag. Finally, host A send empty segment with only ACK flag [8].

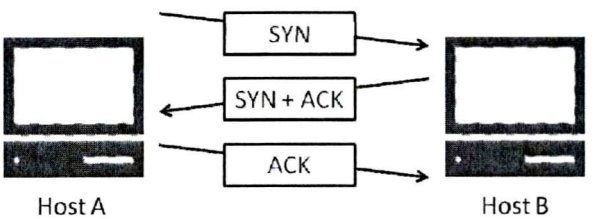

Fig. 2. Three-way handshake

TCP connections cannot move from one IP address to another. When a PC switches from Ethernet to Wi-Fi it obtains another IP address. All existing TCP connections must be shut down and reconnected.

MPTCP protocol is a set of extensions to the specification of TCP which allows the client to make multiple connections using different network cards with the same destination host. In this way fault-tolerant are formed and efficient data connections between hosts that are compatible with existing network infrastructures.

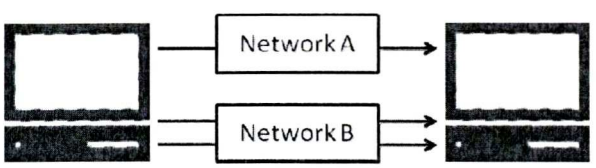

Fig. 3. N-different TCP connections is represented as one logical data

Main goal of this solution is that it is possible to use multiple network paths for a single connection. Another advantage is increases throughput of transport connections. This approach should significantly improve balance congestion between network paths. Simultaneously enabling MPTCP must not prevent connectivity on a path where regular TCP works [[9]].

MPTCP is located at the transport layer and aims to be transparent to both higher and lower layers. It is an additional function of higher layers TCP standard.

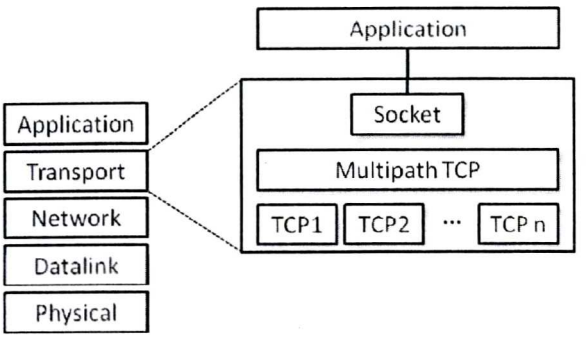

Fig. 4. MPTCP in the stack

New connection of MPTCP is established in the same way as a standard TCP. Protocol is enhanced by new feature. MP_CAPABLE option informs both hosts if MPTCP connection can be established and if data can be transmitted.

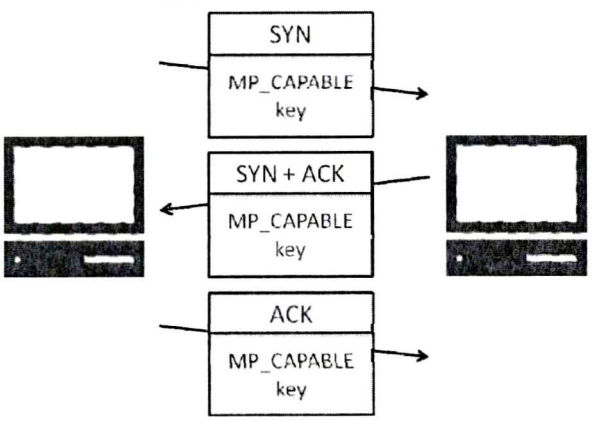

Fig. 5. Establishing connection

Nowadays, establishment of new connection is complicated by middle boxes (switches, routers). Pair (IP, port) of source and destination hosts is not enough to identify connection. Therefore MPTCP extended its functionality with the additional option – MP_JOIN. Adding new subflow is made in three steps.

First, MP_JOIN option contains a token which is generated with the key (truncated hash of the key), which is formed during the initial connection. The exchange of HMAC (hash-based message authentication code) is the second step.

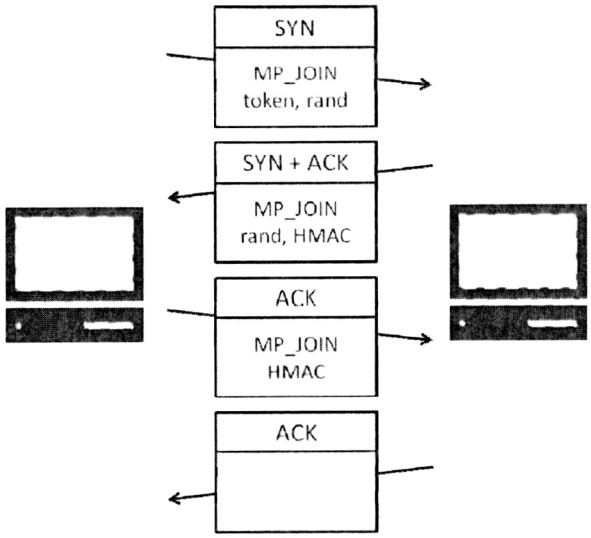

Fig. 6. Adding new subflow into MPTCP

Now that the subflows have been established, MPTCP can use them to exchange data. Each host can send data over any of the established subflows. Furthermore, data transmitted over one subflow can be retransmitted on another to recover from losses.

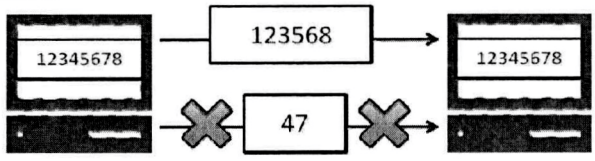

Fig. 7. Error control in MPTCP

Standard TCP 'subflow sequence number' provides the reception of single subflow and ensures if any loss of data is detected. MPTCP uses "data sequance number" to sort received data before passing it to the application [[10]].

Source port	Destination port
Data Sequence Number (8 octets)	
...	
Subflow Sequence Number (4 octets)	

Fig. 8. MPTCP header (in short)

When host's source wants to inform host's destination that it has no more data to send it signals this "Data FIN". It has the same semantics and behavior as a regular TCP FIN, but at the connection level.

Traditionally, people use the internet on smartphones via Wi-Fi or 3G, but not by both. If TCP's connecting fails for some reason, it must be re-established. Multipath TCP avoids this by dynamically switching to the link, and user do not waste time for his reconnecting. It may also select the optimum speed.

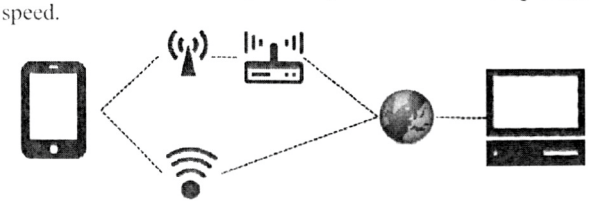

Fig. 9. MPTCP in smartphones

iOS 7 - first mobile system that supports MPTCP [[11]]. It's ensure uninterrupted transmission when one of connection works weakly or the connection is aborted. At the moment, MPTCP is used only for data transfers of Siri. Siri is an intelligent personal assistant that helps in various matters – all users have to do is just ask. It allows user to send messages, schedule meetings, make phone calls and perform many other tasks using voice commands. With high probability this will be extended to the entire operating system.

III. MPTCP AS A SECURITY SOLUTION

MPTCP could increase security level of transmitted data because of fact that it uses many different links to reach destination unlike methodology which are only using network protection [3][4][5] and securing possibility of connection into this network [6][7]. Transmitted data are treated as row binary data, which could be divided into blocks which are passed to transmission layer. To protect data from being sniffed via hacker, authors propose algorithm which consists of such steps:

- Step 1 – data are encrypted,
- Step 2 – data are divided into blocks,
- Step 3 – random sequence of blocks to be passed are determined,
- Step 4 – blocks are collected in random sequence determined in previous step,
- Step 5 – blocks of data are passed to MPTCP socket which will transmit it do destination,
- Step 6 – receiver side collect blocks of data,
- Step 7 – receiver side are connecting blocks of data in right order,
- Step 8 – data are decrypted.

Process of dividing data into blocks and putting them in random sequence according to step 2 and 3 is shown on figure 10. Step 4 of proposed algorithm is presented on figure 11. Data passed to MPTCP socket are transmitted using different data connection. According to MPTCP working schema, transmission process is made via operating system. In potential place of sniffing data, only part of data could be sniffed and hacker is not able to determine which part of data was sniffed. Sniffed data could not be used for hacking encryption algorithm because of fact, that hacker does not know which part of data possess.

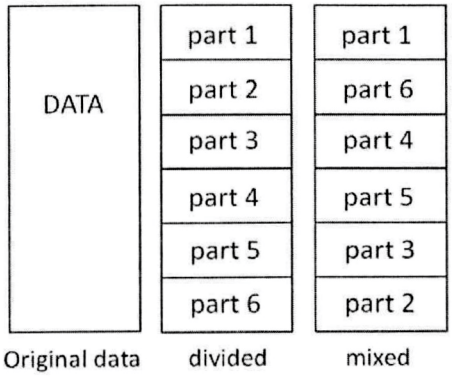

Fig. 10. Mixing data process

There could be used any of encryption algorithm which is well known and widely used. Process of sending data using different links is something which is given by simply using MPTCP. The only thing which should be implemented is a process of dividing data into block in random sequence. This process must use random sequence and have to be made on sender side while information about right sequence should be passed to receiver side. There are a few possibility:

- setting some code book which will be used during transmission and will describe data sequence,
- draw order of mixing data in secure manner.

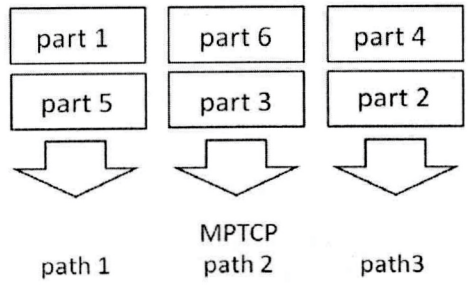

Fig. 11. Transmission process

The second option should be better because it will provide more secure level. The worst thing is that it will require some additional implementation on MPTCP layer but this process should not be so difficult. It is possible to use some well known protocols used for establishing secure crypto key for example. Such protocol is used in IPSec implementation. It is possible also to establish secure connection and share information about data sequence using shared key. Of course such solution is less secure while it could be implemented in short time. The first test which was taken was to check data transfer with MPTCP implemented in part of the network to check data mixing process over different path. Two long distance point over Internet network was chosen. Data was downloaded from Kosice in Slovak Republic to Bydgoszcz in Poland. Data test was made during one hour in ordinary day which should give the same condition of Internet to all data transfer. There was two file to download: 118 and 512 MB. The test was made in four conditions: a) without MPTCP implemented in the network, b) with MPTCP implemented on two interfaces and the faster one was the primary, c) with MPTCP implemented and only the faster one was working, d) with MPTCP implemented and the slowest one was the primary. The faster interface was a Gigabit Ethernet network connection, while the slowest one was a 3G connection which was limited to 512 kbit/s transfer speed.

TABLE I. DATA TRANSFER RESULTS

Condition	Download time	
	File 118 MB	*File 512 MB*
a) without MPTCP	2 min 34 sec	9 min 39 sec
b) MPTCP and faster interface as primary	2 min 27 sec	9 min 03 sec
c) MPTCP and only faster interface	2 min 24 sec	8 min 42 sec
d) MPTCP and slower interface as primary	4 min 53 sec	15 min 11 sec

As it was noticed, using slow interface in the network increases time required to download files. Also using slow interface as primary provided the same situation.

IV. CONCLUSIONS

In this article, a MPTCP concept was presented in security aspects. MPTCP provides such functionality which could increase security level of transmitted data in easy way. Hacker needs to sniff data on many links, and needs to know possible way of transmission and more than one transmission technology, which is not so easy. Using MPTCP user will get not only faster data transmission but also more secure. The authors are working right now on algorithm for randomization on data block sequence. This is very important task in presented algorithm but it should not have impact on data transfer throughput. The first test made in cooperation between Technical University of Kosice and Kazimierz Wielki University showed that there is a requirements to appropriate network configuration, because setting slower interface as primary one will increase downloading time.

REFERENCES

[1] Ł. Apiecionek, M. Romantowski, Secure IP Network Model. Computational Method in Science and Technology 19(4) 209-213 (2013), DOI:10.12921/cmst.2013.19.4.209-216,

[2] Łukasz Apiecionek, Michał Romantowski, Security solution for Cloud Computing, Journal of Information, Control and Management Systems, Vol. 11, 2013, No. 2, ISSN 1336-1716,

[3] L. Vokorokos, M. Ennert, M. Hartinger, J. Radušovský, A Survey of Parallel Intrusion Detection on Graphical Processors, Proceedings of International Scientific Conference INFORMATICS 2013, November 5-7, 2013, SpišskáNováVes, Slovakia,

[4] L. Apiecionek, J. M. Czerniak, QoS solution for network resource protection, Proceedings of International Scientific Conference INFORMATICS 2013, November 5-7, 2013, SpišskáNováVes, Slovakia,

[5] Ł. Apiecionek, J. M. Czerniak, W. T. Dobrosielski, Quality of Services Method as a DDoS Protection Tool, Intelligent

Systems' 2014, Proceedings of the 7th IEEE International Conference Intelligent Systems IS'2014, September 24-26, 2014, Warsaw, Poland, Volume 2: Tools, Architectures, Systems, Applications, pp. 225-234

Springer International Publishing Switzerland 2015, D. Filev et al. (eds.), Intelligent Systems'2014, Advances in Intelligent Systems and Computing 323, DOI: 10.1007/978-3-319-11310-4_20,

[6] W. R. Cheswick, S. M. Bellovin, Firewalls and Internet Security: Repelling the Wily Hacker, Addison-Wesley Publishing Company, 1994, ISBN 0-201-63357,

[7] B. Chapman, E. D. Zwicky, Building Internet Firewalls, O'Reilly & Associates, Inc., 1995, ISBN 1-56592-124-0,

[8] J. Postel, Transmission Control Protocol, RFC 793, IETF, September 1981.

[9] U. Krieger, Evolution of Transport Protocols in high-Speed Networks, Lectures materials Multimedia-Kommunikation in Hochgeschwindigkeitsnetzen (KTR-MMK-M), 2014,

[10] A. Ford, C. Raiciu, M. Handley, TCP Extensions for Multipath Operation with Multiple Addresses, RFC 6824, January 2013

[11] K. M. Schneider, K. Mast, U. R. Krieger, Adapting Content-Centric Networking to the Characteristics of Multihomed Mobile Terminals, 2014.

Firewall rule with token bucket as a DDoS protection tool

Łukasz Apiecionek
Institute of Technology
Kazimierz Wielki University
Bydgoszcz, Poland
lukasz.apiecionek@ukw.edu.pl

Wojciech Makowski
TELDAT Sp. z o.o. sp.k.
Bydgoszcz, Poland
wmakowski@teldat.com.pl

***Abstract*— The subject of this article are the security problems of network resources in computer networks. Presently the main problem of computer networks are Distributed Denial of Service attacks which can block them. The methods suggested by the literature that mostly base on using firewall and IDS/IPS mechanisms to fight the attacks are not sufficient enough. In this article the author presents a new method for counteracting DDoS attacks - firewall rule with token bucket implementation from Quality of Services method. This new concept is different than previous one, because it gives possibility for user to finish they work which was started before the DDoS attack occurs and they do not suffer from DDoS attacks. The proposed method has already been tested. The results presented in this article suggest that the method could be applied in practice. This article is a part of authors papers focused on IT security.**

***Index Terms*— DoS, security, network, firewall, protection**

I. Introduction

IT systems are nowadays omnipresent. The Users need a fast access to information from every part of the network. Denial of Service attacks, or rather Distributed Denial of Service attacks lately, have become a problem as they cause network unavailability by blocking services via seizing system resources in computers in the network until they stop working. A user who has already started working in the system loses the connection and cannot even log out of the system, which has to do it for him after the connection timeout is reached or when a broken connection is detected. DDoS attacks are nowadays a serious obstacle for IT systems' efficient functioning and they have to be eliminated. Common methods of fighting the DDoS attack problems [1][2][4][5][6] are usually limited to using the Intrusion Detection System and Intrusion Prevention System (IDS/IPS in short) solutions. Such systems are efficient provided that they have a description of well known attacks or some kind of Artificial Inteligence solution which could learn the actions in some specific scenarios of attack. Other solutions suggest using a firewall mounted on the network edge. However, this firewall will only block the incoming traffic on specific ports or IP address ranges, which is not sufficient. This paper presents a new firewall method which implements some of the Quality of Services mechanisms to eliminate the DDoS attacks.

The structure of this paper is as follows. Chapter II shortly describes the issue of the DDoS attacks and introduces the proposed method for fighting them. Chapter III presents the results of the implementation of the described method. Chapter IV provides a conclusion and discussion over the developed method.

II. Enhanced QoS method

A. Description of the Distributed Denial of Service attacks

The DDoS attacks are widely described in the literature [1] [2]. These attacks can be performed on various system resources: TCP/IP sockets [2] [3] or DNS servers. Regardless of the method, the main principle is to simulate so many correct user connections that their number exceeds the actual system performance and drives it to abnormal operation. Papers [1][2][4][5][6] describe methods for dealing with the DDoS attacks by their global detection and the necessity of cooperation between network providers. The transmission of the attackers' packets is done through the provider's network and if it cannot be blocked, it leads to data link saturation. Such saturation results in lack of connection to the server. The proposed solutions to prevent such situations are not specific and their implementation is associated with many problems. The most common concern is the limited performance of network devices. However, it is possible to limit the incoming traffic on a firewall and allow the servers to deal with the already established connection. This will let the users finish their work and the new users will be able to connect to the server.

978-1-4673-9868-8/15/$31.00

B. Firewall rule with token bucket

The role of the input firewall is to control the incoming traffic on the network edge. When the network is to give access to the server to the external users, a specific type of traffic has to be allowed by the incoming rules. For instance, in the case of a http server, usually the TCP port 80 has to be opened for the incoming connections. When an attack on the server occurs, this port is still open. This situation leads in turn to the server overload. Thus, a special firewall module was developed, the role of which is to filter the traffic on the server's open port and to limit it according to the determined policy. This rule functions as follows:

- during the server's regular work all packets are passed through, the network is not under any attack,
- when a given allowable number of packets *packet_limit* is exceeded in a time slot *t1*, a filtration process is launched, as a DDoS attack has been detected,
- at the beginning of the filtration a list of the IP addresses which communicate with the server correctly, i.e. which are not a part of the DDoS attack – *listIP* – is read from the server,
- during the filtration each packet is checked whether it is on the list of the valid IP addresses *listIP*; if so, the packet is sent to the network, if no, a counter of the passed packets *packet_counter* is checked whether its value is greater than the allowable packet limit *packet_limit* in a time slot *t1*,
- if the limit of the packets is exceeded, the packet is dropped - *DROP*,
- in the following time slot the number of current packets *packet_counter* is zeroed and the above mentioned filtration process is restarted,
- when in a given number of the subsequent time slots *some_limit* the limit of the packets is exceeded, we are facing a large attack on the server, and in order to give the server some time to regain efficiency the limit of packets *packet_limit* is decreased,
- if in the following time slots the packet limit is not exceeded, the *packet_limit* is increased to the determined limit value.

Changing the packet limit allows the server to handle the incoming connections which may be potentially correct or to release the resources used incorrectly by the attacker. Despite the attack, the server is still accessible to the users who were working on it when the attack was detected.

The process of decreasing the packet limit can depend on the server type, its needs and kind of work. Moreover, the limit values may require experimental determination or setting them basing on the server's resources, its operating system, the amount of memory and processors type.

The operation algorithm was developed by analogy to the methods for traffic quality assurance (Quality of Services), where for various types of traffic there are given amounts of packets or data by using the so-called token bucket mechanisms. In this case the mechanism is implemented by setting the limit of packets allowed to pass in a given time slot *time_slots*.

Pseudocode of the main part of the algorithm responsible for passing the packets as well as narrowing the limits is shown below:

```
packet_counter:=packet_counter + 1
if packet_counter < packet_limit then
 packet pass
else
 begin
  if IP address in listIP then
   packet pass;
  else
   packet drop;
 end;
if times_slots ends then
 begin
  if packet_counter>packet_limit then
   overdrop_times=overdrop_times + 1;
  packet_counter=0;
  if overdrop_times>some_limit then
   packet_limit=packet_limit/2;
   overdrop_times=0;
  else
   packet_limit=packet_limit*2;
   overdrop_times=0;
 end;
```

The *packet_counter* variable contains the number of packets which are passed through in a certain time slot. When its value does not exceed the permissible limit *packet_limit*, the packet is sent, but when the limit is exceeded, further tests are performed. If the packet is present in the database of the known IP addresses *listIP*, the packet is passed through, otherwise it is dropped. When the *time_slots* timeout expires, a verification is performed whether the limit of packet was exceeded in this time slot. In this case in the implemented method the counter of the limit was increased in the subsequent time slots. When the limit was exceeded in the following two time slots, the allowable limit of packets was decreased (*packet_limit*/2). If in the next time slots the limit had not been exceeded, the limit of packets was increased (*packet_limit**2).

III. Implementation Results

In order to verify if the server will indeed work continuously, the method was implemented and tested. The implementation consisted of a module for the firewall IPTables module on Debian Linux system based on kernel 2.6.32. The tests simulating the most common types of attacks on the servers were performed on a simple network, which was built for this aim according to well now structures and routing protocol requirements [7] [8][9] . A http server from Asterix FreePBX distribution, working under CentOS with kernel 2.6.32 and an Apache 2.2.15 server, equipped with 1GB of RAM was used as a receiver. In order to perform the attack, a

Sender machine was used – based on Debian Linux with kernel 2.6.32 (equipped with 512 MB of RAM) and DDOSIM software (Layer 7 DDoS Simulator v0.2). The structure of this network is shown on the figure below. The firewall with IP Tables module was running on Linux Debian with kernel 2.6.32 with 512 MB of RAM.

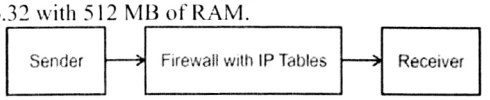

Fig. 1. The test network.

During the test, the memory usage and http server response times were observed. The attack consisted in sending a large amount of HTTP Get requests to the server. Without the implemented module, the server's memory usage raised up to 100% and the server stopped responding. After launching the module, RAM memory usage was observed.

The module was configured with the following limits:

- time slot for analyzing the amount of transmitted data time_slots = 1 second;
- allowable limit of packets packet_limit = 30;
- minimal limit of packets = 10;

Five tests of the attack on the server were performed, each lasting one hour and the conditions were as follows:

- 100 HTTP GET requests sent every 30 seconds;
- 1000 HTTP GET requests sent every 30 seconds;
- 2000 HTTP GET requests sent every 10 seconds;
- 10000 HTTP GET requests sent every 10 seconds;
- 50000 HTTP GET requests sent every 30 seconds.

None of the cases resulted in server overload. Prior to the attack a connection with the server through the firewall was established and it remained active because it was started before the attack and the computer was recognized as allowed to communicate with the server.

During the test the amount of free RAM memory was observed on the http server and on the firewall. On the http server the memory remained mostly on a constant level.

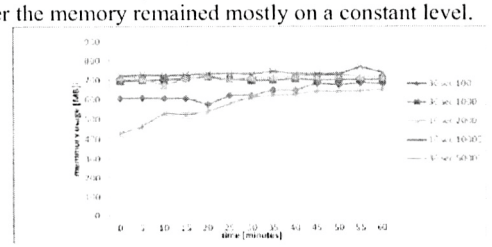

Fig. 2. RAM usage on http server.

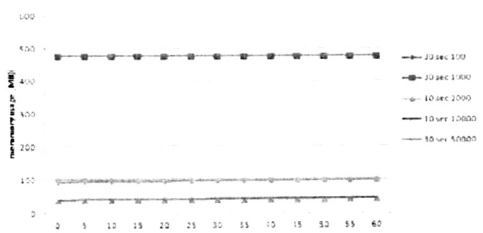

Fig. 3. RAM usage on firewall.

On the firewall the RAM usage remained on a constant level. During the test which consisted of sending 1000 packets every 10 seconds, RAM memory usage was higher, which was probably a result of other operations of the device.

RAM memory usage test was also performed in a network built without using the proposed method of fighting the attacks. The result is shown in figure 4. Without the active method the memory usage rose up to 100% several times (the server was equipped with 1GB of RAM).

In the description of the method a mechanism for narrowing the amount of packets in a single time slot was mentioned. Figure 5 shows a fragment of the operation of the implemented mechanism. It regulates the number of packets which can be transmitted in a time slot dynamically, depending on the load. This mechanism may require some refining depending on the server used. In this implementation, a simple step change of the packet limit was used.

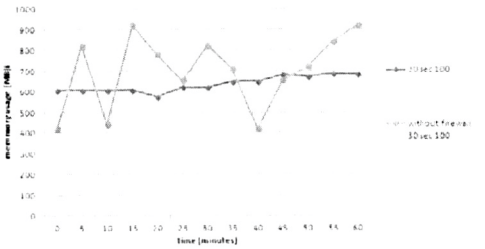

Fig. 4. RAM usage on the server with and without the DDoS protection tool.

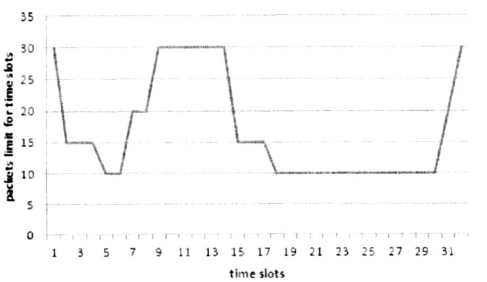

Fig. 5. Packet limit on time slots.

IV. Conclusions

In this article a new concept of eliminating DDoS attacks was introduced.

The methods suggested in the literature can block the access to the resources when the attack occurs, by using a firewall along with IDS/IPS mechanisms. During the time of the blockage no user from an external network can connect to the desired resources. Moreover, such solution does not allow to complete the work started by the users who were already connected. The users who worked with the server lose their connection.

The method described in this article allows the users to continue their work. It is possible for the users who connected

to the server prior to the attack and the server informed the method about this fact.

The method was implemented and tested in practice. It does not cause an increase of the firewall load but prevents the server from overload by keeping the server's load at a stable level during the attack. The proposed method may be successfully implemented on any firewall-type devices. Only the mechanism of regulating the number of packets in a single time slot may require some adjusting. The mechanism should be selected for a particular server, depending on its capability and this can be a subject of further research.

The author is ready to provide the sources of the described method for further analysis.

REFERENCES

[1] Rocky K., Chang C., Defending against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial, IEEE Communications Magazine, October 2002, pp. 42-51,

[2] Moor D., Shannon C., Brown D.J., Voelker G. M., Savage S. Inferring Internet Denial-of-Service Activity, ACM Transactions on Computer Systems (TOCS) 24 (2), 115-139, 2006

[3] Schuba C. L., Krsul, I., Huhn M. G., Spafford E. H., Sundaram A., Analysis of a Denial of Service Attack on TCP (1996). Computer Science Technical Reports. Paper 1327. http://docs.lib.purdue.edu/cstech/1327,

[4] CERT Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks, November 2000, http://www.cert.org/advisories/CA-1996-21.html,

[5] CERT Advisory CA-1996-01 UDP Port Denial-of-Service Attack, September 1997, http://www.cert.org/advisories/CA-1996-01.html,

[6] Apiecionek Ł., Czerniak J. M., Zarzycki H., Protection Tool for Distributed Denial of Services Attack, "Beyond Databases, Architectures, and Structures", 405-414, 2014, Springer International Publishing,

[7] Apiecionek Ł., Romantowski M., Secure IP Network Model, Computational Method in Science and Technology 19(4) 209-213 (2013), DOI:10.12921/cmst.2013.19.4.209-216,

[8] Piechowiak M., Zwierzykowski P.: "The Evaluation of Unconstrained Multicast Routing Algorithms in Ad-hoc Networks", The International Science Conference: Computer Networks CN2012, Szczyrk, Poland, 2012,

[9] Piechowiak M., Zwierzykowski P.: "The Evaluation of Multicast Routing Algorithms with Delay Constraints in Mesh Networks", 8th IEEE, IET Int. Symposium on Communication Systems, Networks and Digital Signal Processing CSNSDP 2012, Poznań, Poland, 2012.

i'15

SSAKI KPI

KATEDRA POČÍTAČOV A INFORMATIKY FEI TU V KOŠICIACH KPI

TECHNICKÁ UNIVERZITA V KOŠICIACH FEI

ZSVTS

IEEE

ISBN 978-1-4673-9867-1