ISSN 1512-1232

AN APPROACH TO CREATE AND USE TEST (ECHO) SERVERS BASED ON TCL/TK

Boneva Ani[1], Ivanova Veronika[2] and Boneva Yordanka[1]

[1] *Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Bulgaria, Sofia 1113, Acad. G. Bonchev str., bl.2,*
ani.boneva@iict.bas.bg, *yordanka.boneva@iict.bas.bg*

[2]*Institute of Robotics - Bulgarian Academy of Sciences, Bulgaria, Sofia 1113, Acad. G. Bonchev str., bl.1,*
iwanowa.w@abv.bg

ABSTRACT

This article presents some of the methods used to control traffic in the operation of a corporate network. They are based on the capabilities provided to the developer by the software environment implemented on the basis of Tcl/Tk. The issues related to the creation and use of test (ECHO) servers, the exchange of encrypted secure information, the interaction with the servers included in the network (PROXY and HTTP) are described in more detail. Software examples for building client-server objects are presented. An algorithm for information block exchange between addressee and recipient with a variable encryption key within a corporate network is presented.

***Keywords*:** *ECHO server, ECHO-client, Tcl/Tk, Client-Server,* TCP/IP, *HTTP*

1 INTRODUCTION

The corporate network is a multi-component system, including a large and diverse number of components - from personal computers to powerful storage systems, system software and applications, network adapters, switching and routing equipment, cable systems and more. In each part of this diverse and complex infrastructure there are different opportunities to optimize and increase the efficiency of their work, to reduce maintenance costs and ensure the integrity and security of the transmitted information. Such networks can be used in industry, medicine, science, etc. [1, 2, 3, 4].

E-health, including the volume, diversity, speed and reliability of health data, is becoming an integral part of our healthcare. The reliability and security of electronic health networks are key factors in both the patient's diagnosis and the patient's medical history. With the development of technology, new and new problems of theoretical and practical nature for e-health arise, which must be solved. However, the main efforts are aimed at solving the problem of data security and reliability [5].

Some research focuses on detecting and thwarting the spread of counterfeit drugs through resilient electronic health networks by recording the logistics requirements for drugs from the patient's drug production on blockchain. A decentralized network of eleven computer nodes is used and its performance is compared with other existing methods in different network configurations. The method has been tested and its reliability has been proven [6].

Other authors are focused on their efforts on improving the Medical Telecommunications Information Systems Protocol (TMIS). The improvements are associated with the risk of offline guessing, replay and anonymity attacks. A biometric three-factor protocol with added security features has been proposed. There are also improvements in the new protocol for monitoring and verifying the health of patients, presented by Xu et al., Intended for WBAN environments. Researchers are working to address issues related to replay attacks and privacy issues.

ISSN 1512-1232

IMPROVING security is achieved by implementing BAN logic and an automated simulation tool [7].

The entry of corporate information networks as an active communication and service environment and means for business, services and production, brings to the fore the issue of security of information transmitted between local computer networks (such as transmission medium), as well as the protection of this information from destruction and unlawful encroachments on unfriendly outsiders (hackers).

Communication within the network is reduced to the exchange of packets in accordance with accepted protocols. Problems in the realization of this traffic arise due to objective and subjective reasons.

- Among the group of objective ones, one can point out the various failures of the connection between the participants in the network, as well as the influence of noise on the integrity of the data.
- The subjective ones arise from attempts to penetrate the network of unauthorized persons or destroy the connection by unscrupulous users.
- The events related to the occurrence of traffic failures are usually asynchronous and the fight against them requires conducting cyclic test procedures to clarify the correct functioning of the network. As a result of their implementation, the specific session is guaranteed, and information on the current status of the various participants (in the corporate network) is also updated.

In the present material are presented software tools (based on Tcl/Tk [8, 9, 10]), realizing secure data exchange in a corporate network, realized on the basis of TCP / IP. The possibilities for controlling the status of the individual participants in the network are considered, as well as the use of information transmission through blocks encoded with a variable key. Based on the concept of ECHO-SERVERS, implementations of a two-channel corporate protocol with a high degree of protection are proposed [11, 12, 13, 14].

The article is organized as follows: Section 1: Introduction; Section 2: Network communication; Section 3: Test servers (ECHO-SERVERS); Section 4: Application of ECHO servers; Section 5: Building a system for synchronizing data exchange and Section 6: Conclusion.

2 NETWORK COMMUNICATION

2.1. Discipline "Client Server"

At this section, the terms "client" and "server" refer to programs that implement these devices running Windows. They are written in Tcl / Tk and use the WINSOCK package [12, 15]. The information protocols are based on TCP / IP. By "computer address" of the corporate network we mean the corresponding IP address, presented as XX.XX.XX.XX - four-byte address. By "gate" we mean the channel number provided by WINSOCK for the implementation of the respective communication. These numbers are a number between 0 and 65535. A URL is the set of IP address and port number (possibly additional information) that identifies the information source / receiver. When using DNS, the URL can also be set symbolically in the form known from the INTERNET (eg http: //dev.corp.nt: 54000).

One or more devices of the following type can be installed on each computer on the corporate network: client or server. Each of these devices implements exactly one protocol. These protocols must include the corporate network communication protocol, hereinafter referred to as the corporate protocol.

ISSN 1512-1232

When it is claimed that a computer uses a protocol, it means that a server or client that works with the protocol is installed on it. Some corporate network participants support only one protocol, the corporate protocol, but others may support multiple protocols, such as HTTP and the test protocol described below. It is permissible for a computer to have both a server and a client installed at the same time, but running on different protocols.

There are several differences between server and client software devices [2, 8, 13]:

- **The first is functional -** the server receives a query, executes it by running program processes on its computer and returns results to the requester. This information is transferred through a special channel, which is maintained until the applicant releases it. The client forms a request to the server and expects the results from the latter on the implemented information channel.
- **The second** is related to the implementation of these logical devices. When creating a server, a channel is first opened on which a request from a potential client is expected. The server also registers a procedure that is executed immediately after the request occurs. This procedure automatically obtains the address and port number of the requesting client, as well as a unique identifier of the established channel between the server and the client (on the server side). After receiving the request, the server creates a copy of the gate pending a new request from another client on that port. The customer is informed about the opening of the channel. Then the information exchange begins.

2.2 Server and client software model.

The client takes the initiative to request a channel to the server, specifying the address of the server computer and the port number (to which the server is connected). It receives a channel identifier (from the client), which it then uses in communication. The client decides when to close the channel and this action is felt by the server, which also releases the local channel.

Formally, the server and client programs look like this:

- Server program.
  - Initialization

```
set serverchan [socket -server Execroutine 50000]
```

  - Execroutine processing procedure.

```
proc Execroutine { localchan address port } {
if { [catch { gets $localchan } buffer] {
if { [eof $localchan] } {
 close $localchan
}
return
}

// Analysis and processing of the application.

puts $localchan $result
}
```

  - Closing the server.

```
close $serverchan
```

ISSN 1512-1232

Here the server opens to port 50,000 and executes the Execroutine procedure when a client logs on. The server ID is the contents of the serverchan, which is used when closing. When a client appears, each of his messages is processed by the Execroutine procedure. The latter automatically receives as parameters the name of the local channel ID, address and port of the client. It analyzes the input and if necessary - if the client has closed the channel on his side, closes the channel on the server side.

- Client program.
  - Initialization.

```
set clientchan [socket 195.96.249.33  50000 ]
fconfigure $clientchan -buffering line
```

  - Processing procedure.

```
puts $clientchan $query
set result [gets $clientchan]
// Допълнителна обработка и диалог
```

  - Closing the channel on the client side.

```
close $clientchan
```

The client accesses a server at 195.96.249.33 and port 50000. The local identifier of the requested channel is written to the clientchan. The second initialization operator only specifies that the information will be displayed at once and will not be buffered. At this time, the server should be initialized (it starts first). The channel is opened on the client side after the client sends the first message (saved in the query variable) to the server. The client is waiting for a response from the server in the second processing operator. Then another similar dialogue is possible.

When the client closes its channel with the clientchan ID, the server senses this and closes its local channel.

In the previous examples, only the principles for building client and server objects are presented - in a real program, things are much more complex and other Tcl/Tk mechanisms are included.

3 TEST SERVERS (ECHO-SERVERS)

The main idea for creating tools for testing communication traffic within a distributed corporate network is based on the concept of ECHO-SERVERS [13]. These are ordinary servers, which together with their clients implement a mechanism for exchanging messages between the participants in the network. These messages are service in nature and precede any other transaction. Below is a simplified software implementation of such ECHO-SERVER devices. and ECHO-CLIENT. These programs are built on the model discussed in 2.2, but use more Tcl/Tk tools and are better structured. A global echo structure is created in the server program, which stores all the information related to a specific communication (client). When the channel is closed (by the client), the information about it is deleted. Information is read and processed in the background (BACK GROUND) thanks to the file event operator directed to this channel.

```
Proc Echo_Server {port}
    global echo
    set echo(main) [socket -server EchoAccept $port]
    }
```

ISSN 1512-1232

```
proc EchoAccept {sock addr port } {
global echo
set echo(addr,$sock) [list  $addr $port]
fconfigure $sock -buffering line
fileevent $sock readable [list Echo $sock]
}
proc Echo {sock} {
global echo
if{ [eof $sock] [catch {gets $sock line] } {
close $sock
unset echo(addr,$sock)
} else {
if { [string compare $line “quit”]==0} {
close $echo(main)
}

// Query processing stored in the line variable.

puts $sock $line
}
}
proc Echo_Client { host port } {
set s [socket $host $port]
fconfigure $s -buffering line
return $s
}
```

The server program is activated by running:

```
Echo_Server 50000
vwait forever
```

It must be started first (on the server computer) and use port 50,000 to wait for a request.

The client program is activated by running (on the client computer):

```
set s [Echo_Client 194.95.249.33 50000]
puts $s “TEST”
gets $s
```

It addresses the server's IP address: 194.95.249.33 and port 50000. The client sends a TEST request to the server (in our case). He then expects a response through his channel.

4 APPLICATION OF ECHO-SERVERS

The ECHO-SERVER test can perform various tests on your computer. It then creates an information block that he sends to the client. The idea of testing traffic comes down to associating such a server and client with every computer connected to the corporate network. All these server and client programs use the same port number and form a specific test protocol, which is used as a service. Each request to any of the computers according to the corporate protocol is preceded by a request according to the test protocol. If the request from the request server is accepted, the client can execute a request to the same server according to the corporate protocol.

ISSN 1512-1232

The test requests contain encrypted information and access passwords known to the servers. In case of coincidence of the verification information and functional suitability on the part of the server, the latter forms a confirmation message for permission of access. If one of the computers fails or an intrusion attempt is obtained from an external unauthorized source, the server does not receive a correct test protocol request or the server does not respond to the request. In both cases, an application under the corporate protocol is not accepted/sent.

As a side effect, the use of the test protocol allows the construction of a graph for the current state of the network. Local area networks include specialized (controlling) servers known as routers. The main task of the latter is to maintain information about the current status of the participants in the network, together with the times for distribution of requests to the various branches. The use of the ECHO-servers mechanism allows the expansion of these tools within the corporate network (it consists of more than one local area network).

5. BUILDING A SYSTEM FOR SYNCHRONIZATION OF DATA EXCHANGE

The corporate network is characterized by the use of several exchange protocols, the most important of which is the corporate protocol, allowing access to information arrays from corporate databases. This information is strictly confidential and is often of interest to unauthorized users. Ensuring the security of information exchange is achieved through the use of specific software network tools, the most important of which are [9, 21]:

- coding of information blocks by using different cryptographic algorithms [16, 17, 18, 19, 20];
- construction of a parallel synchronizing communication highway, using the mechanism of ECHO-servers;
- change of the coding keys during the transactions within the framework of a specialized security test protocol;
- use of connection validity information.

The proposed program implementation is based on the rupture of the corporate protocol into separate segments, building the information block. Each of these segments is encoded with a specific key known to the sender and recipient.

The segment is sent only after verification of the connection performed according to the test protocol. In this procedure, the sender and recipient specify the key used. Upon successful completion of the test communication, the segment of the corporate protocol is sent.

The test protocol is also coded and contains identification markers. These tags have the character of passwords (perceived by the participants in the procedure). The key for encoding the information during the test procedure is the one that was valid at the last transaction of the participants in the communication.

This discipline creates serious difficulties for hackers due to the double coding of information and synchronization blocks and the ability to dynamically change the keys in communication. It is an essential part of designing the overall security system of a corporate network.

In Figure 1 an information block exchange between the addressee and the recipient with a variable encryption key is presented, within the corporate network illustrating the above.

ISSN 1512-1232

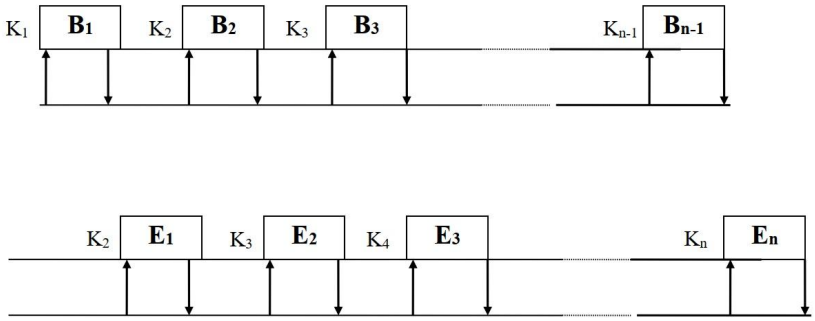

Figure 1. Block exchange with variable encryption key

The data block includes fields E_1 to E_n, each of which is encoded with an individual key K_2 to K_n. Each field is sent only after successful exchange of the corresponding synchronization information B_1 to B_{n-1}.

The synchronization blocks are coded with corresponding switches K_1 to K_{n-1}. Each block B_i contains a password for access to the addressee and an identifier of the key that will be used to encrypt the corresponding block E_i. The last of the information blocks E_n indicates the end of the exchange and contains control information guaranteeing the integrity of the packet.

The synchronization blocks Bi are transferred to the port (implementing the ECHO server), while the information E_i is transferred to the port implementing the corporate server.

The chosen approach guarantees the security of the information and its reliable protection from external access, due to the double-secured security scheme of exchange.

6 CONCLUSION

The methods and means for realization of the communication traffic within the distributed corporate network presented in this article are the basis for building specialized communication protocols, compliant with the security requirements (assigned to the objects of this class). They are an upgrade over the network capabilities offered in Tcl/Tk.

The tools provided by the Tcl/Tk software package offer a convenient approach for the implementation of secure information exchange, both within an individual company (having outsourced branches) and within large corporate networks. The built-in mechanism of information arrays processing in the libraries of Tcl/Tk together with possibilities for realization of the communication in background mode, make possible the construction of complex logical protocols without significant increase of the complexity of the projects.

Global research shows that breakthroughs in e-security have been increasing at a faster pace than measures to protect it. This fact once again shows that research in this area must continue.

ACKNOWLEDGMENTS

This research is supported by the Bulgarian FNI fund through the project “Modeling and Research of Intelligent Educational Systems and Sensor Networks (ISOSeM)”, contract КП-06-Н47/4 from 26.11.2020.

REFERENCES

[1] Kocjan W., Beltowski P., Tcl 8.5 Network Programming: Build network-aware applications using Tcl, a powerful dynamic programming language, Packt Publishing, ISBN 978-1-849510-96-7, pp. 1-589, (2010), www.packtpub.com

ISSN 1512-1232

[2] Ivanova V., A. Boneva, Y. Doshev, S. Ivanov, P. Vasilev, Multifunctional Operating Station Based on Tcl/Tk and its Applications, Proc. of the 6th IEEE International Conference BdKCSE'2019, Sofia, Bulgaria, IEEE, Electronic ISBN: 978-1-7281-6481-6, DOI: 10.1109/BdKCSE48644.2019.9010662, pp. 1-7, (2020)

[3] Vasilev P., ANSI/ISA-95 Segment Dependency usage in Finite Capacity Scheduling, International Scientific Journal "INDUSTRY 4.0", WEB ISSN 2534-997X; PRINT ISSN 2534-8582, Year V, ISSUE 4, Scientific Technical Union of Mechanical Engineering "Industry 4.0, pp. 160-163, (2020), https://stumejournals.com/journals/i4/2020/4/160.full.pdf

[4] Ilchev, S., Andreev, R., Ilcheva, Z., Otsetova-Dudin, E., Software for laser projection of CAD files for the clothing industry, International Conference on Technics, Technologies and Education (ICTTE) 2020, November, 4-6, 2020, Yambol, Bulgaria. Published in IOP Conference Series: Materials Science and Engineering, Vol. 1031, Art. 012040, IOP Publishing, ISSN: 1757-899X, DOI: 10.1088/1757-899X/1031/1/012040, pp. 1-8, (2021), https://iopscience.iop.org/article/10.1088/1757-899X/1031/1/012040/pdf

[5] Toral Cruz Homero , Debiao , He Albena , Mihovska D., Kim- Kwang Raymond Choo, Muhammad Khurram Khan, Reliable and Secure e- Health Networks, Wireless Personal Communications, 117, https://doi.org/10.1007/s11277-021-08104-z, pp. 1–6, (2021) https://link.springer.com/content/pdf/10.1007/s11277-021-08104-z.pdf

[6] Pandey, P., Litoriya, R., Securing E-health Networks from Counterfeit Medicine Penetration Using Blockchain, Wireless Personal Communications, 117, pp. 7–25, (2021). https://doi.org/10.1007/s11277-020-07041-7

[7] Kumari, S., Renuka, K., Design of a Password Authentication and Key Agreement Scheme to Access e-Healthcare Services, Wireless Personal Communications, 117, https://doi.org/10.1007/s11277-019-06755-7, pp. 27–45, (2021).

[8] Welch B, (1998 - 2000), Practical Programming in Tcl and TK, part3 - TclHttpd Web Server, Ajuba Solutions.

[9] Hipp R., Mktclapp A Toll For Mixing C/C^{++} with Tcl/Tk, Charlotte, NC, (1999).

[10] Roseman M., Meta Kit: Quick and Easy Storage for your Tcl Application, Equi4 Software - Draft, (2002)

[11] Abelson H., Greenspun Ph., Sandon L., Tcl for Web Nerds, USA, (2000).

[12] Ousterhout J., Tcl/Tk Engineering Manual, Sun Microsystems Inc., (1994).

[13] Jean-Claude Wippler, Scripted Documents, 7th USENIX Tcl/Tk Conference – Tcl/Tk, Austin, Texas, USA, (February 14-18, 2000)

[14] Tcl/Tk program, https://www.tcl.tk/ (last visited 30.05.2021)

[15] Ousterhout J., J. Levy, B. Welch, The Save-Tcl Security Model, Sun Microsystems Inc., (March, 1997).

[16] Denis R., Madhubala P., Hybrid data encryption model integrating multi-objective adaptive genetic algorithm for secure medical data communication over cloud-based healthcare systems, Multimedia Tools and Applications, 80, pp. 21165–21202, (2021), https://doi.org/10.1007/s11042-021-10723-4

[17] Setyaningsih E., Wardoyo R, Anny Kartika Sari, Securing color image transmission using compression-encryption model with dynamic key generator and efficient symmetric key distribution, Digital Communications and Networks, Vol. 6, Issue 4, https://doi.org/10.1016/j.dcan.2020.02.001, pp. 486-503, (2020) https://www.sciencedirect.com/science/article/pii/S2352864819301063?via%3Dihub

[18] Boneva A., Krasteva R., Batchvarov D., Zamanov A., Stanishev K., Software tools for enrypted data transfer in corporate networks, Proceedings "Scientific reports" October 2002, ISSN1310-3946, CLMI-BAS, Conference "ROBOTICS and MECHATRONICS'2002", Drjanovo, Bulgaria, pp. 4.11-4.16, (2002), (in Bulgarian).

ISSN 1512-1232

[19] Gaydarski, I., Minchev, Z., Andreev, R., Model Driven Architectural Design of Information Security System. Advances in Intelligent Systems and Computing, Madureira A., Abraham A., Gandhi N., Silva C., Antunes M. (eds) Proceedings of the Tenth International Conference on Soft Computing and Pattern Recognition (SoCPaR 2018)., 492, Springer, ISBN:978-3-030-17064-6, ISSN:2194-5357, DOI:10.1007/978-3-030-17065-3_35, pp. 349-359, (2019).
[20] Hambali Moshood Abiola, Gbolagade Morufat Damola, Olasupo Yinusa Ademola, Cloud Security using Least Significant Bit Steganography and Data Encryption Standard Algorithm, GESJ: Computer Science and Telecommunications, No.1 (58), pp. 17-29, (2020)
[21] https://www.tcl.tk/software/tclhttpd/TCLHTTPD.html - (last visited 30.05.2021)

Article received: 2021-05-30