<table>
<thead>
<tr>
<th>Book Title</th>
<th>Modern Development Paths of Agricultural Production</th>
</tr>
</thead>
<tbody>
<tr>
<td>Series Title</td>
<td></td>
</tr>
<tr>
<td>Chapter Title</td>
<td>Software Development for the Security of TCP-Connections</td>
</tr>
<tr>
<td>Copyright Year</td>
<td>2019</td>
</tr>
<tr>
<td>Copyright HolderName</td>
<td>© Springer Nature Switzerland AG</td>
</tr>
<tr>
<td>Corresponding Author</td>
<td></td>
</tr>
<tr>
<td>Family Name</td>
<td>Lubko</td>
</tr>
<tr>
<td>Particle</td>
<td></td>
</tr>
<tr>
<td>Given Name</td>
<td>Dmytro</td>
</tr>
<tr>
<td>Prefix</td>
<td></td>
</tr>
<tr>
<td>Suffix</td>
<td></td>
</tr>
<tr>
<td>Role</td>
<td></td>
</tr>
<tr>
<td>Division</td>
<td></td>
</tr>
<tr>
<td>Organization</td>
<td>Tavria State Agrotechnological University</td>
</tr>
<tr>
<td>Address</td>
<td>B. Khmelnitsky ave. 18, Melitopol, 72310, Ukraine</td>
</tr>
<tr>
<td>Email</td>
<td><a href="mailto:dmytro.lubko@tsatu.edu.ua">dmytro.lubko@tsatu.edu.ua</a></td>
</tr>
<tr>
<td></td>
<td><a href="mailto:di75ma@gmail.com">di75ma@gmail.com</a></td>
</tr>
<tr>
<td>ORCID</td>
<td><a href="http://orcid.org/0000-0002-2506-4145">http://orcid.org/0000-0002-2506-4145</a></td>
</tr>
<tr>
<td>Author</td>
<td></td>
</tr>
<tr>
<td>Family Name</td>
<td>Sharov</td>
</tr>
<tr>
<td>Particle</td>
<td></td>
</tr>
<tr>
<td>Given Name</td>
<td>Sergii</td>
</tr>
<tr>
<td>Prefix</td>
<td></td>
</tr>
<tr>
<td>Suffix</td>
<td></td>
</tr>
<tr>
<td>Role</td>
<td></td>
</tr>
<tr>
<td>Division</td>
<td></td>
</tr>
<tr>
<td>Organization</td>
<td>Bohdan Khmelnytsky Melitopol State Pedagogical University</td>
</tr>
<tr>
<td>Address</td>
<td>Hetmanska St, 20, Melitopol, 72312, Ukraine</td>
</tr>
<tr>
<td>Email</td>
<td></td>
</tr>
<tr>
<td>ORCID</td>
<td><a href="http://orcid.org/0000-0001-5732-9980">http://orcid.org/0000-0001-5732-9980</a></td>
</tr>
<tr>
<td>Author</td>
<td></td>
</tr>
<tr>
<td>Family Name</td>
<td>Strokan</td>
</tr>
<tr>
<td>Particle</td>
<td></td>
</tr>
<tr>
<td>Given Name</td>
<td>Oksana</td>
</tr>
<tr>
<td>Prefix</td>
<td></td>
</tr>
<tr>
<td>Suffix</td>
<td></td>
</tr>
<tr>
<td>Role</td>
<td></td>
</tr>
<tr>
<td>Division</td>
<td></td>
</tr>
<tr>
<td>Organization</td>
<td>Tavria State Agrotechnological University</td>
</tr>
<tr>
<td>Address</td>
<td>B. Khmelnitsky ave. 18, Melitopol, 72310, Ukraine</td>
</tr>
<tr>
<td>Email</td>
<td></td>
</tr>
<tr>
<td>ORCID</td>
<td><a href="http://orcid.org/0000-0002-6937-3548">http://orcid.org/0000-0002-6937-3548</a></td>
</tr>
</tbody>
</table>
Abstract
Nowadays, cybersecurity can be considered as an important aspect of the policy of any state in the conditions of the existence of a global information space, widespread communication and interaction through the Internet. For its adequate provision, they develop appropriate technologies of information protection, legislation at the state level, hardware and software, etc. The article focuses on the development of software tool for intercepting and analyzing outbound TCP-connections (the sniffer); there is a description of the stages of its development and the requirements for use; there is also a short manual for a user. Functional capabilities of the developed software tool allow to automate some of the system administrator’s functional duties. Among them, we should mention the interception, interpretation and storage of data transmitted over the network with 32-bit programs. Analysis of the received data packets will allow to analyze the suspicious activity of some programs or to identify individual network problems. The access to the software functions is implemented with the main, context menu and hot keys. The main language for the development was chosen the C++. To develop a user interface, a Qt library was used. The programming language Lua was used as a tool for writing and executing scripts.

Keywords
TCP-connections - Sniffer - Cybersecurity - Software - Protection
Software Development for the Security of TCP-Connections

Dmytro Lubko, Sergii Sharov and Oksana Strokan

1 Introduction

Nowadays, cybersecurity can be considered as an important aspect of the policy of any state in the conditions of the existence of a global information space, widespread communication and interaction through the Internet. Despite constant development of new methods of protecting information and computer systems from unauthorized access as well as their updating, cybercriminals continue finding the ways of bypassing the information security systems and carrying out their devastating activities. In turn, the vulnerability of information systems to certain actions (conscious or unconscious) of end users also raises a significant issue in protecting personal data and information flows. The protection of information involves the achievement and saving the security properties in user resources aimed at preventing relevant cyberthreats. As a result, the development of qualitative software products and digital equipment will give a possibility to increase the level of information security for commercial enterprises, government agencies and the state. A number of system and application software designed to provide information security can be found in the software market today. The development and dissemination of the Internet led to a common use of special programs, called sniffers, which are intended for interception and analysis of the network traffic.
2 Analysis of Recent Research and Publications

Issues of computer networks, their monitoring and analysis, as well as the development of appropriate specialized network traffic analyzers, were studied by such well-known scientists, researchers and programmers: Michael W. Lucas, Oliver M. Heckmann, Ed Wilson, N. Mendkovich, G. Konakhovich, V. Chuprin and other scholars. M. Doinea researched software products for analyzing open-source network traffic [1], M. A. Qadeer studied the packet sniffer principles [2], and Z. Xiao, L. Guo and J. Tracey used sniffers to analyze the traffic of instant messaging systems [3]. Among Ukrainian scientists, we can mention D. Dubov, O. Baranov and others who deal with the problem of cybersecurity. At the same time, the problem of developing software products for analyzing network traffic that perform specific functions is still considered to be relevant.

3 Results and Discussion

There is a tendency today toward the active use of information resources in all spheres of economic, social, educational, political and other activities. Along with the rapid growth of hardware capacity, cases of unauthorized data collection, use, distribution of personal data and important information, fraud in the Internet are spreading. It is due to its widespread and massive nature that cybercrime has become transnational, capable of causing a severe harm to the interests of the individual, society and the state [4]. The mentioned circumstances explain the relevance of effective mechanism development and use to ensure the cybersecurity of the state and resolving of the related problems [5].

Currently, there are several issues of information security and protection of personal data on the Internet. Primarily, it concerns the unification of the cybersecurity notion. Today, there are different definitions of this notion, namely: It is a condition of security of the vital interests of the individual, society and the state in the use of computer systems and/or telecommunication networks [6]; it is a status of a person’s ability, society and the state to prevent and avoid the directed, first of all, unconscious, negative influence of information [7]. Since national and international documents differ significantly in cybersecurity determination, then the approaches to cybersecurity are different.

As Bellovin and Cheswick [8] point out, the situation with the information security of a computer connected to the network is complicated by several factors at once:

1. The mail program, the network file system and the distributed database are potential sources of danger. In addition, the authentication used by some protocols may be inadequate. However, they must be loaded into RAM to ensure that users are properly served.
2. There are many entry points from which you can start an attack. If computer users are limited to one task, it will be difficult for the outsider to try to penetrate
the security system. On the other hand, a computer connected to the network can be attacked from anywhere in the world, connected to the Internet.

3. Connecting to hazardous computers connected to the Internet causes problems of transitive trust, which also require some actions to prevent them.

Another problem related to providing information security at least at a minimum level is due to the lack of awareness of users about cyberthreats. The constant use of the Internet by ordinary users, who often do not know about the potential risks of losing personal information and cyberattacks by unauthorized third parties, often leads to cases of Internet fraud, theft of passwords and files, loss of money. In this case, the solution to the problem is to inform users of the existing risks regarding the use of the Internet and the importance of security of personal information [9].

Effective cybersecurity is associated with the development of appropriate mechanisms, hardware and software tools to protect against unauthorized access to information (firewall, sniffer, cryptographic protocols, etc.).

The main solution for a bulk of information protection issues is in use of firewall that monitors and analyzes the connection of your computer to the Internet, which is the basis for deciding whether to allow this connection on the basis of the settings of the program. That is, the network filter passes the traffic of only those programs that the user allows. As a result, the effectiveness of the firewall depends on the experience of the user or the system administrator who installs the software setup. There are two types of firewall: personal, installed on a personal computer, and corporate software installed on the gateway between the Internet and the local network. In the first case, the setting is done by the user and the other by the system administrator. In both cases, it is recommended to allow the traffic to only trusted applications and only through the ports on which they work. By all means, in order to prohibit access to certain programs, you need to know how they harm the computer or the corporate network.

One of the tools for analyzing network traffic is sniffers, which means a program or hardware and software device designed for interception, storage and analysis of network traffic. This type of software is often used by system administrators in a legitimate way to address certain problems when transmitting data over a network [2].

Sniffers are designed for a specific type of network, usually for the Ethernet; that is why it is often called network analyzer or the Ethernet Sniffer. This program passively intercepts data addressed to other computers at the level of the network adapter NIC (OSI network interface card). Trapping is carried out using several methods:

- Usual «interception» of the network interface. This method is effective when used in a segment of hubs instead of commutators;
- Connection of a sniffer to a channel gap;
- Branching (software or hardware) of traffic and directing its copy to the sniffer;
- Analysis of side electromagnetic emissions and traffic restoration.
Analysis of network traffic that has passed through a sniffer allows:

- to identify viral and/or looped traffic that increases the download of network equipment and communication channels;
- to identify malicious and unauthorized software, such as network scanners, flooders, Trojans, peer network clients and others;
- to capture any unencrypted (sometimes encrypted) network traffic in order to obtain passwords or other important information;
- to locate a network failure or network agent configuration error.

All the sniffers can be divided into two categories: sniffers that support download and work from the command line and sniffer with graphical interface. Some hybrid variants combine both modes of operation. There are plenty of examples of existing network traffic analyzers for various operating systems: Wireshark, Iris, WinDump, Sniffit, Ultra Network, sniffer, analyzer, Packetizer, IPDump2, Ferret, LanGrabber Ethereal Network Analyzer Wireshark and others. All these software features are similar in function, but differ in the protocols they support, the depth of the analysis of intercepted packets, filtering capabilities, as well as compatibility with other software products, the user interface and the ability to generate statistical reports. It should be noted that since in the “classic” sniffer traffic analysis is done manually, using only simple automation tools (protocol analysis, TCP stream restoration), it is suitable for analysis of only small volumes of data.

To work with a sniffer, you need to know several recommendations:

- If the sniffer is used on the corporate network, then the rules and procedure for its use must be described [1].
- It is desirable to use filters when intercepting data packets in order to increase the efficiency of analytics.
- The use of sniffer does not make any sense if the data intercepted by it are not used for further analysis.
- Make sure that a specific sniffer is able to intercept network traffic (incompatibility of protocols, operating systems, applications).

According to Androschuk [4], Ukrainian users are highly vulnerable to infections owing to refusal to update software or use of pirated copies of programs. The situation is aggravated by the fact that some users use outdated operating systems, such as Windows XP, where protection from modern cyberattacks is almost absent.

In view of this, we set ourselves the task of developing a software tool to intercept and analyze outbound TCP-connections in 32-bit programs. A software tool allows you to perform tasks that often have to be solved by system administrators, for example, to identify specific causes of client network software failures, to test programs for suspicious activity and intercept the transmitted information. It also provides the ability to control the need for the entire data transfer process in arbitrary software.

The designed sniffer should be fast, reliable, free and stable, and store the data received for further monitoring. Its development was carried out in the following stages:
1. Analysis of the operation of the protocol TCP/IP, specifics of TCP-connections, review of the functional capabilities of the existing network traffic analyzers;
2. Selecting the format of data storage that will be needed to run the software;
3. Selecting the tool environment for the development of the sniffer;
4. Creating software modules;
5. Testing, validation of the software in real conditions, error correction and retesting.

Let us consider the basic requirements for the developed software.

Requirements for interface design.

Graphical user interface software should be easy to use, flexible and intuitively comprehensible, zoomable, looking the same on computers with different localizations.

The user interface language is English. The main window should have a minimum number of controls.

Access to basic operations should be implemented through the main menu. Transactions over data packets must be implemented through the context menu, and the ability to view the contents of the package in text as well as in the 16-year form should be implemented.

The download window of the new process should be able to select the executable file by using the dialog or manually entering the path to it. You must also be given the option to load the process parameters and delete them. The connection window to the active process should contain a list of processes and a block of control buttons.

The list of processes needs to be implemented in the form of a table with two columns. The first column contains the digital ID of the program and the icon. If the program does not have icons, you should use the default icon. The second column contains the name of the executable process file.

The manual window of the data packet must contain packet contents and a data management block. The function of sending a package several times in a row with a given interval must be implemented.

The search box should have an intuitive interface with the ability to select a search method (text, using a regular expression or 16-bit sequence).

Requirements for software and hardware.

The main language of software development was the C++ programming language.

To develop the graphical user interface, a cross-platform Qt software development library was used. The programming language Lua [10] was used as a tool for writing and executing scripts.

Software functionality requires the following software and minimum hardware:
operating system: 32-bit Windows XP, 7; installed Microsoft Visual C++ 2010 Redistributable Package (x86) software; RAM—2 GB or more; hard drive—500 GB or more. Software Restriction: The sniffer only works with 32-bit software.

Software aspects of sniffer development.

The main software module consists of five main parts. The general scheme of the software is shown in Fig. 1.
Developing a software tool for analyzing outbound TCP-connections involves the development of a dll-module that redirects all new TCP-connections of the analyzed software to the main module. Before starting to develop this module, the following issues should be solved:

1. How to implement the method of intercepting ws2_32.connect and ws2_32.WSAconnect? There are four basic methods of intercepting the target function call located in the dll. The first one is to replace the function pointer in the library export table. The second involves replacing the pointer to the function in the import table of the main module of the program.

The essence of the third method is to replace the first few instructions (or one if its size is ≥ 5 bytes) to the instructions for the unconditional jump to the address (jmp long addr), the parameter of which is the address of the handler’s function. The fourth method is most often used in cases where the function code structure does not allow to replace its first instructions with the five-byte instruction jmp long. When developing the software, the third variant of the interception was used, as the most versatile, and has the least number of shortcomings. The scheme of the method is shown in Fig. 2.

2. How to implement the method of loading a working library into the target process? The loading of the working library to the address space was as follows: The target process opens with kernel32.OpenProcess; it allocates the memory area with access rights PAGE_EXECUTE_READ, where using the kernel32.WriteProcessMemory function writes the loading code of the library with the dll extension; a stream that executes this code is loaded.

3. How to implement the protocol of information exchange with the main module of the software? SOCKS version 5 was chosen to exchange information with the main software module, but only part of the protocol that was needed to process the output TCP-connections was implemented.
This allowed intercepting traffic from part of client network applications (those that support work through an intermediate SOCKS server) without the use of a work library.

4. How to integrate the Lua scripting language into a software tool? This task was divided into several stages.

The first stage involved creating an interpreter control class that runs in a separate flow. Due to the fact that the Qt platform provides the ability to implement multi-threaded operation and has a built-in functionality for transmitting data between them using signals, this stage was executed fairly quickly.

To achieve this, two classes were implemented—luaEngineWorker, which encapsulates the entire low-level interpreter control code, and the luaEngine control class, which implements the interface between flows and manages the execution of the interpreter’s flow.

The second stage consisted of writing classes that implement the interface between the main code and the interpreter of the Lua programming language. In this step, classes were written that provide full access to the data repository and allow you to manage intercepted connections. For the possibility of full control of the data transmission process, a system of events was implemented, the handlers of which can arbitrarily control the process of data transmission in intercepted connections.

The final stage involved the integration of the Lua interpreter with the software using the SWIG toolkit. This approach frees the programmer from writing a huge amount of code for the interface and significantly reduces the time of development and adjustment of the software.
Let us consider the most significant moments in the development of the main module.

Access to most functions was implemented through the main menu of the program, but functions that directly work with a particular package (e.g., the “Close connection” function for forced connection failure) were resolved to be brought to the context menu that can be called in the data tree.

When developing a software tool, it became clear that the Qt Designer software used to design forms lacked the ability to create and edit a context menu. In view of that fact that the structure and appearance of the menu do not change during the work of the sniff, it was decided to create objects of the context menu when initializing the main window of the software using the procedure initContextMenu. Then we developed, the onPacketListContextMenu (Q Point p) function handler was created for the customContextMenuRequested signal, which is automatically generated when the right mouse button is pressed. This function in the process of running a software tool displays a screen built-in window initialization context menu. The binding of the handler function to the signal was done using the Qt Designer slot editor.

The event log is implemented using the QTextBrowser widget, which has the following benefits: fast work, small overhead for saving text, no flicker when you add a large number of entries in a small amount of time. The function of adding new entries to the event log was to be implemented similarly to the function of the standard printf library. This approach has made it easy to format event log entries without creating any excessive software code.

It was decided to implement the content of the intercepted data packet in text and 16-bit views. The first format of the data presentation was implemented using the standard QPlainText widget, which allows you to view a data packet as text, which is very useful in analyzing text data protocols. The second data viewer mode was implemented using the QHexEdit2 widget, which is freely licensed under the GNU Lesser General Public License.

To have the developed software tool for analysis of network traffic properly used, we will give a brief user’s manual.

Work with the program tool (executable file ReqPacketTool.exe) is carried out using the main menu items, the context menu of the data tree and the controls. In addition, for the automation of executable actions, the sniffer developed can accept command line parameters that can be combined arbitrarily. Figure 3 shows the analysis of network traffic on the home page of Wikipedia at https://uk.wikipedia.org.

The main menu contains several sections, which, in turn, consists of sub-items. The section “Main” includes the following items:

- The “Start process” item allows the user to download an arbitrary software tool for further analysis. To do this, manually or by using the dialog (button «…»), select the main executable file of the target software, set all necessary parameters for loading it, if necessary, and click on the “Start” button.
- The “Inject to process” item allows the user to intercept all new connections of the working process. After this, a process selection window appears that contains
a list of all processes in the operating system. The user must select the desired process and click on the “Inject” button. It should be noted that with the help of the developed software tool it is impossible to intercept connections of processes that are started by users with a higher level of rights than that of the user who launched this sniffer.

- «Pause sniffing/Start sniffing» allows the user to pause, if necessary, the accumulation of intercepted data and to continue this process at any time.

- The «Clear captured packet list» option allows you to clear the tree of the intercepted data if necessary.

- The “Find” item allows the user to search for the requested information in the intercepted data. The search box has three main modes of operation, namely: searching for text data (which is sensitive to the case), searching through regular expressions and searching for 16-year-old sequences. For the convenience of the user in the search mode of 16-bit sequences, the search according to the template is realized. To do this, all data packets that have an undefined value are marked with a “?”. For example, the query “FF??” will find all two-byte sequences that start with the FF code.

- The “Load packets queue”/“Save packets queue” items allow you to download previously exported data/export intercepted data.
The manual sending of a data packet that appears after you select the «Send custom packet» context menu item in the context menu of the data can send a manually generated data packet. In this window, the user can change the contents and size of the sent packet, as well as its direction. To send a package multiple times, you need to specify the number of data packets and the interval between sending them.

The «Configure» item of the «Options» section allows you to modify the configuration of the software, namely to switch on/off logging events, disable/enable the receipt of domain names of remote servers, change the number of columns in the 16-bit data viewer, change the port number of the embedded SOCKS server.

The «Lua script» section contains the following items:

- Item «Start script» allows the user to execute arbitrarily selected script or choose one of the scripts used by the latter.
- Item «Stop script» allows the user to forcefully terminate the script that is being executed at the moment.

Items «Save»/«Clear» in the «Log» section allow you to save the contents of the event log to the file and clear the contents of the event log.

In order to speed up the operation and for the user convenience of the software for the main operations of the main menu, shortcut keys were assigned.

4 Conclusion

Consequently, in the conditions of the existence of a global information space cybersecurity takes a significant place among the important issues of many states. For its provision, appropriate technologies of information security are being developed, and state-level legislative documents, hardware and software are being worked out. Consequently, sniffers are designed to intercept traffic for further analysis.

In the process of research based on a technical specification, a software tool was developed for interception and analysis of output TCP-connections and a user’s manual was written. The object of automation of the developed sniffer is the system administrators’ work, who often have to solve the tasks of providing information security and the correct operation of the internal and external network. Among such tasks, one can distinguish interception, interpretation and storage of data transmitted through the network by various programs. Further analysis of the received data packages helps to analyze suspicious activity of some programs or to identify certain problems in the early stages when it is possible to find the ways of their solution faster and to avoid negative consequences.

The prospects for further research lie in increasing the functionality of the software tool, specifically in modifying it for use in networked software with 64-bit architecture.
References

7. Furashev, V.: Key aspects of the draft law of Ukraine «On information security». Veche 6, 29–30 (2012)