International Research Journal of Multidisciplinary Scope (IRJMS), 2024; 5(2): 516-534

Original Article | ISSN (O): 2582-631X DOI: 10.47857/irjms.2024.v05i02.0558

Analysis of the Possibilities of Carrying Out Attacks on the Functions of Transferring Control to Operating System Console Using Active Intelligence Methods

Larisa Cherckesova, Elena Revyakina*, Olga Safaryan, Vitaliy Porksheyan, Maxim Kazaryan

Department of Cyber Security of Information Systems, Don State Technical University, Rostov-on-Don, Russia. *Corresponding Author's Email: elena.a.revyakina@gmail.com

Abstract

This paper researches the possibility of conducting attacks on the console control transfer functions using active intelligence methods. The research employs a comprehensive approach involving ports scanning, directories searching, parameters modifying, and credentials searching based on a user dictionary. Additionally, the study involves the development of a software tool designed to detect vulnerabilities in network nodes. The software developed within the framework of this study is delivered in the form of two modules, the first module contains the main program with the mechanisms implemented in it to indicate the possibility of attacks, with an emphasis on current web applications and services. Checking for the possibility of an attack for any network node is that first a list of requests is compiled, the purpose of which is to identify weaknesses of a web application running on the server, and a list of expected responses from the server is also compiled for these requests. After that request goes to the server, the program waits for responses from the server, and if the expected responses from the compiled list coincide with the actual ones, then this fact signals the possibility of an attack on the web application. The second module stores localization dictionaries responsible for the presence of two interface languages in the program – Russian and English. The result of this work is the MaxNetScanner2022 software, which identifies the possibility of an attack on the system control transfer functions using active intelligence tools.

Keywords: Active Intelligence Methods, Attack, Information Safety, Network Packets, Traffic Analysis.

Introduction

Traffic analysis is a necessary and demanded direction in the implementation of a secure network, which includes both the analysis of individual packets and the diagnosis of complex problems, taking into account the context of the work of the web service under study (1, 2). The results of such an analysis are useful in drawing up a network operation model, since today there is a real threat of massive hacker attacks (3, 4).

As an example, one of these attack vectors is CVE-2018-20062 – vulnerability of the ThinkPHP framework, which allows you to gain full control over the server by accessing its console. You can also select the CVE vector-2018-12536 – the vulnerability of the DefaultServlet component of the Jetty HTTP server is related to the shortcomings of error handling when using invalid requests. Exploiting the vulnerability may allow an intruder operating remotely to gain unauthorized access to protected information by displaying the InvalidPathException message included in the error report.

Traffic analysis can be considered as the main tool for intercepting confidential data of network users (5, 6). The analysis itself is performed using specialized software tools, which are also known as "sniffers" (7). This type of software performs two actions: collects all packets passing through the network node and selects from the collected packages those containing information about user credentials.

(Received 29th January 2024; Accepted 21st April 2024; Published 30th April 2024)

It is worth noting that at the moment the following protocols are quite common in public networks (Table 1). At this moment, there are many different vulnerabilities in existing systems and devices. Among them, we can single out particularly common gaps that pose the greatest danger to users (8-11).

Table 1: Most popular protocols to date

HTTP	web pages, documents, arbitrary data transmission
FTP	file transfer, working with the file system
TELNET	providing access to the functionality of the device
SMTP	sending and relaying mail messages between servers or client and server
POP3	receiving messages from the mail server (the message is completely saved on the client side and deleted from the server)
IMAP	receiving messages from the mail server (only the header of the letter is transmitted to client, by which it is possible to access the message lying on the mail server)
NNTP	receiving news material. Data exchange between news servers
IRC	sending and receiving messages in real time

HNAP (Home Network Administration Protocol) is a home network management protocol created by Pure Networks. Allows you to manage network devices. Serious security problems were found in this protocol in terms of access control to system objects, therefore, devices that entered the market after 2016 no longer support this protocol. SQLite Manager is an SQLite server management system implemented in PHP. It is vulnerable to executing arbitrary PHP code by adding the latter to the request parameters. Hudson Java is a Hudson Ci and Jenkins Ci family of servers written in Java. These systems are tools for continuous development. Support for this software was discontinued in 2016. Vulnerable to execution of arbitrary Java code. phpMyAdmin is a web application implemented in PHP, its main task is to administer a MySQL database. It is subject to XSS attacks, SQL injections and exploits that allow the execution of arbitrary code. ThinkPHP is a lightweight PHP web application development environment. At the end of 2018, a vulnerability was discovered related to the use of the invoke Function method in the body of a GET request, which allows you to run arbitrary code on the server. Despite the release of patches, this problem is still being identified in new versions of this framework (12).

All the protocols listed above transmit information in unencrypted form, or use weak encryption methods, so any information, starting from passwords for accessing network resources, and ending with trade secrets or personal information, will be compromised when using these protocols. Also, in the absence of encryption, there is a possibility of spoofing the request sent to the server, which, if the server is configured incorrectly, may allow the attacker to get the information necessary for him or full access to the victim's server.

In this matter, the purpose of the study was to research the possibility of conducting attacks on the console control transfer functions using active intelligence methods.

To achieve the purpose of the study, the following tasks were set:

- to show the development of approaches and methods of analysis from a historical point of view.
- highlight popular traffic analysis techniques that are used in most software solutions.
- conduct a review on software tools for traffic analysis.
- implement a software module that searches for vulnerabilities in network nodes.

In the next sections of the article, the basic algorithm of the operation of infrastructural methods of network traffic research, which is used in practice in a slightly modified form, will be demonstrated. The stages of the study will also be considered with a brief list of their characteristics.

Literature Review

There are exists only two main directions of development of network traffic analysis technologies:

- Quality growth of the researching network packet, in other words, an increasing level in the OSI model of the data which are analyzed.
- Getting comprehensive information at the output about the characteristics of the stream to which the packet belongs, and other data-related streams.

Depth of Network Packet Analysis

According to this model, the technological processes of studying traffic were formed alternately, any further process inherited a share of the previous elements, and also added its own. It is possible to note three degrees of formation of technological processes (13), which are shown in Figure 1.

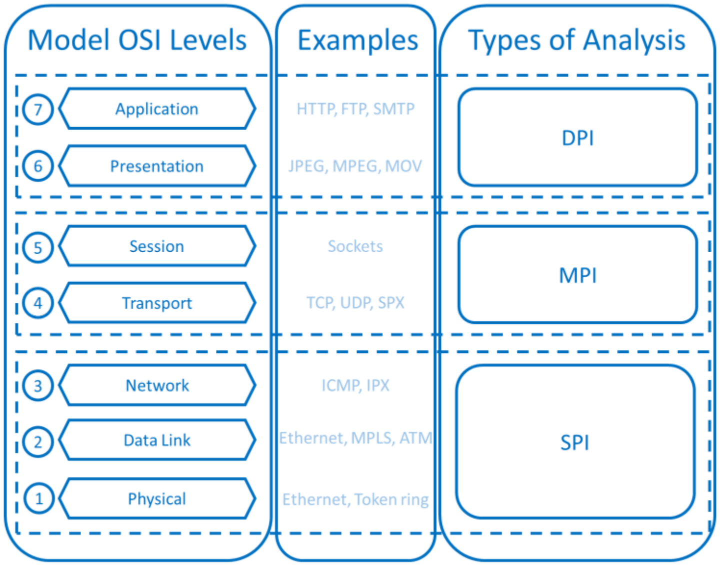


Figure 1: Levels of development of network traffic analysis technology by "depth"

SurfacePacketInspection (SPI)analysis

This technology checks packet headers (which are information placed at the beginning of a data block, such as the IP addresses of the sender and recipient), as opposed to the body or "payload" of the packet. This type of packet inspection allows messages to remain uncompressed, since the contents of the packets are not tracked, and the information in the header is used only for packet routing. SPI technologies manage (relatively) simplified firewalls running on Windows XP, Windows Vista, and OS X operating systems. These firewalls stand between a specific client computer and the network to which it is connected. They restrict the ability of the client computer to send or receive the content specified by the user. When the server sends a packet to the client computer, SPI technologies check the packet header information and compare it with the blacklist. These firewalls, in particular, focus on the source and destination IP addresses that the packet is trying to access. If the package contains a header whose parameters are blacklisted, then such a package will not be delivered. When SPI technology refuses to deliver a packet, it simply refuses to transmit it without notifying the source that the packet has been rejected (14).

MiddlePacketInspection (MPI) analysis

It works as a proxy server that is located between end-user computers and an Internet service provider or Internet gateways. These proxy servers can check the packet header information against their parsing checklist. When a packet arrives at a proxy server, it is parsed against a parsing list that system administrators can easily update. The parsing list allows you to allow or deny certain types of packets depending on their data format types and their respective location on the Internet, not just their IP address. MPI devices can read the payload representation layer of a packet and identify aspects of that layer. Using MPI devices, administrators can prevent client computers from receiving flash files from YouTube or image files from social networking sites. MPI technologies can prioritize some packages over others by examining application commands located at the application level and file formats at the presentation level. MPI devices suffer from poor scalability, which limits their usefulness for Internet service providers, where tens of thousands of applications can transmit packets at any time (15).

This method allows you to solve a wider range of tasks, in addition to configuring access rights, this technology also copes with the following tasks:

- Saving traffic data to the cache.
- Investigation of traffic that has been encrypted or compressed.
- The ability to impose a ban on the execution of individual commands.
- In the case of working in proxy server mode, it can serve as an internet connection optimizer.

A significant disadvantage of MPI technology is that each command and protocol require its own input-output port. In addition, the proxy mode consumes a lot of CPU time, which reduces the performance of the system as a whole. To optimize the work of the proxy server, the ICAP protocol was created, the idea of which was to shift the procedure for checking the security of packages to third-party servers. Such a system is organized in the ClamAV antivirus product, which can connect to the Squid and NetCache proxy servers (16).

These conditions significantly limit the possibility of using this technology at the provider level, given the large width of their communication channels and the multitude of protocols being processed.

DeepPacketInspection (DPI) analysis

Also known as DPP (Deep Packet Processing), this technology performs the following actions on the package:

- Modification
- Filtration
- Redirection

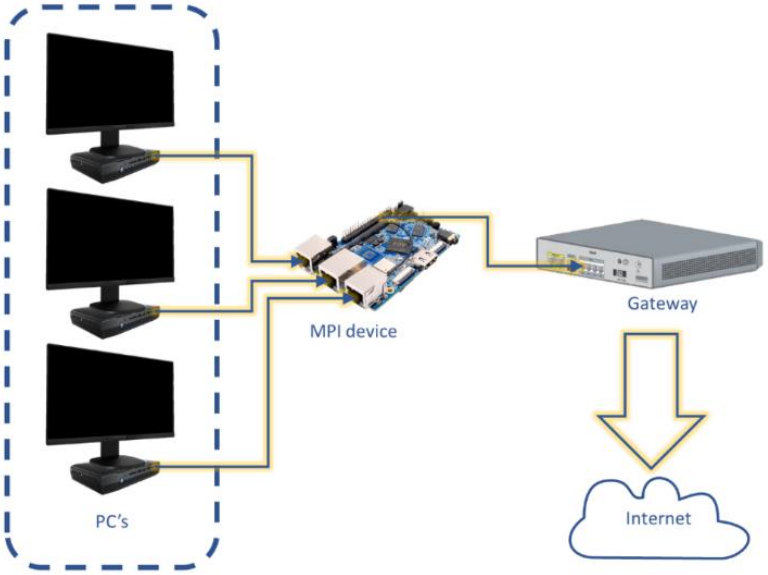


Figure 2: Scheme of application of analysis devices based on MPI technology

At this moment, both names have the same meaning. This type of analysis can be called a natural successor to MPI. In this system, the parser reads the contents of the package completely, the decision on the further fate of the package is made not only based on the data contained in the package, but also based on the specifics of network programs and protocols. Probabilistic data can be used for these purposes. For example, conducting an experiment to establish the number of encounters of certain characters, packet sizes, the delay time between the arrival of packets. The DPI method has become widespread due to a sharp increase in the computing power of processors, the speed of their memory and, of course, due to the high degree of accuracy of the analysis (17).

In contrast to MPI, this method was created for fast processing and identification of a variety of applications in real time. It follows from this that DPI has good abilities to expand over network channels, and thousands of well-known network applications are also contained in the database of this solution. In any implemented project, DPI acts as a module that establishes the correspondence of the package to the class of network protocols. It is also worth noting that the accuracy of this operation depends on the goals assigned to the system:

- type of protocol or program (e.g. Web, Peer-to-Peer, VoIP);
- specific application layer protocol (HTTP, BitTorrent, SIP telephony);
- application which uses a protocol (Google Chrome, Microtorrent, Skype).

It is important to note that the correspondence between classes of different levels of accuracy is not unambiguous, as shown in Figure 3.

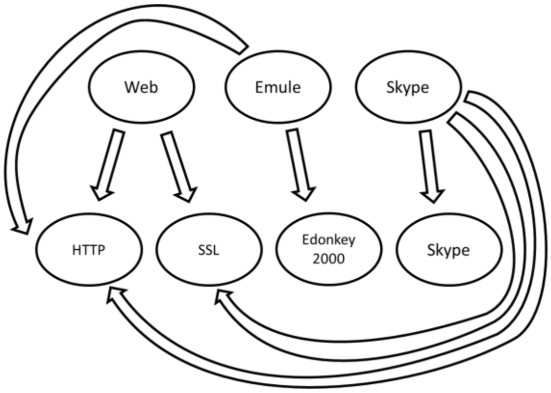


Figure 3: The difference between the identification of applications (on the top) and protocols (on the bottom)

Today, deep packet analysis is the standard for programs that scan traffic. This type of application belongs to the field of extremely important decisions in the field of network security, especially from the point of view of the letter of the law. So, over the past few years, many standards, specifications and recommendations have been approved. Despite the fact that this method has not found popularity in firewalls, there are solutions using this technology. For example, the Hogwash and Shield screens (16, 17).

As the next branch in the field of analysis, it is possible to single out the accounting of the behavior of the packet flow in the network, in this group there are two methods of traffic research:

- With accounting the state of the connection in the stream.
- Without accounting the state of the connection in the stream.

This applies only to protocols that connect using a transport protocol. That is, before setting up the data transmission channel, the process of "establishing a connection" takes place, where subscribers mutually transmit a special chain of packets, called a "handshake", at the end of the data transmission process, the connection is closed. This type of protocols includes:

- TransmissionControlProtocol (TCP)
- User Datagram Protocol (UDP), if connection-establishing protocol is organized on top of it, for example, the QUIC protocol.

Therefore, it is impossible to exclude the method of traffic investigation taking into account the connection establishment status for UDP packets (17). To give an accurate description of the above methods, it is necessary to disclose such a concept as a "packet stream". There are various interpretations of this phenomenon. Information about the most common of them is presented on the web page of the San Diego Supercomputing Center (SDSC). Consider the unidirectional flow of the transport layer. In the stream itself, the following characteristics can be distinguished:

- Source IP-address
- Source port
- Endpoint IP-address
- Endpoint port
- Name of protocol

According to this model, it is easy to establish what is the difference between the traffic research approach taking into account the state in the stream and without taking into account the state in the stream. In the first case, information about which stream the package under study belongs to is taken into account, and information obtained during the study of previously received packets belonging to the same stream is also taken into account. It is worth noting that the traffic analysis method, taking into account the state in the stream, has a very vague definition. Therefore, such a method from one implementation to another may have different accuracy, optimization quality, and performance (18). A possible gradation option in practice is demonstrated in Figure 4.

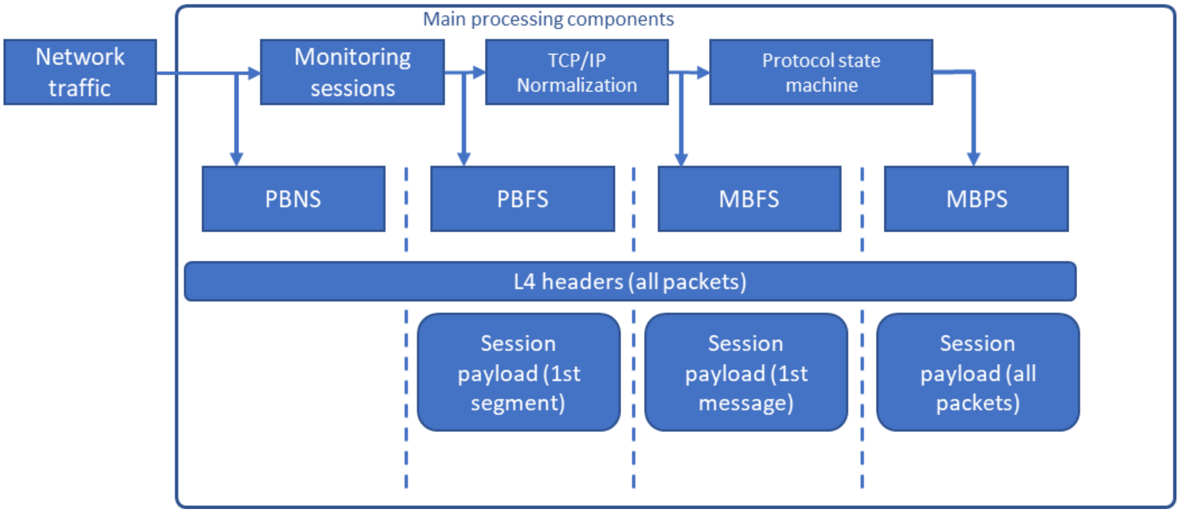


Figure 4: Gradations of completeness of state accounting in the flow

The list of degrees of state research in the stream is represented by the following levels:

- Exploring some packets, ignoring threads and states (Packet Based No State, PBNS).
- Investigation of packets by streams (Packet Based Per Flow State, PBFS).
- Data exploration within a single stream (Message Based Per Flow State, MBPS), i.e. The collection of IP fragments into IP packets (IP normalization) and the assembly of TCP segments into TCP sessions (TCP normalization)
- Data research within the protocol (Message Based Peer Protocol State, MBPS), i.e. The state of the protocol automaton (its ability to work with various types of data) is taken into account. An example of an HTTP protocol state machine is shown in Figure 5. Vertices correspond to states, edges correspond to transition conditions, which may include receiving and transmitting data, the results of their processing, and timeout expiration.

Usually, developments that use the DPI method are related to research without taking into account the state in the stream – the analysis is performed only on individual packages in one stream without saving the state between their studies. This approach is sufficient for most existing software products and is not wasteful of hardware resources. However, there is a class of problems that require a higher level of accuracy. And then it is necessary to turn to methods that use the method of research taking into account the state in the flow. This can include a stateful packet inspection (SPI) and a detailed study of the package composition (DCI) (19).

The meaning of the SPI method is that the application or hardware on which it is active, during the initialization of a new connection, checks it according to the established security policy, and stores information about this connection until it is closed. Thanks to such technical solutions, the legitimacy of the connection is checked. This method is often used in many modern routers, as an SPI firewall. In addition, this technology is used in CheckPoint organization firewalls and in a variety of IDS/IPS type systems that belong to the third generation. In such a system, not only the packets received and sent are monitored, but also the status of each active connection is taken into account, information about which is stored in special tables. Therefore, in this method, the following factors are taken into account when checking the package:

- Security System Policy.
- The state of the connection over which the investigated packet was sent.
- The state of previously committed connections that are somehow related to the package in question.

An exemplary example of the superiority of a firewall with the presence of packet inspection taking into account the state of the stream in comparison with similar screens, but without using this technology in them is the operation of the file transfer protocol. This protocol initializes a new thread for a specific command.

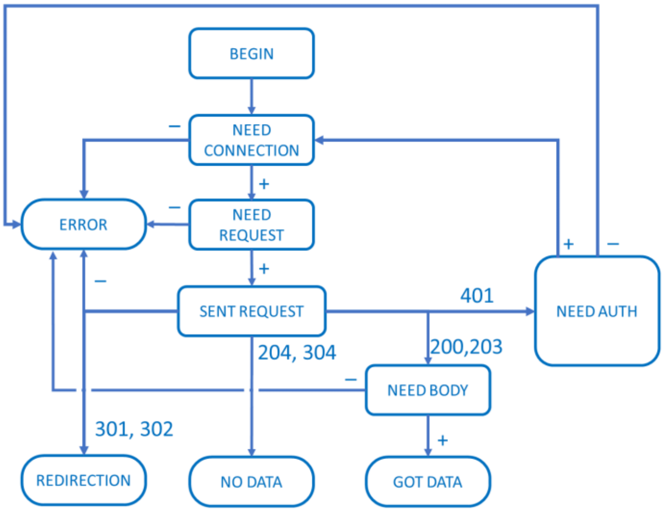


Figure 5: An example of an HTTP protocol state machine

The stream itself runs on an arbitrary port larger than 1024. The firewall will block such streams at the same time, due to the fact that it cannot determine whether these streams belong to the allowed protocol. However, if there is a technology for checking packets taking into account the state of the stream, such a case will be adequately handled, and information about the new stream will be entered in the table of allowed streams, thereby skipping the session to the network (20).

In the DCI method, in addition to defining the network stream protocol, the streams are distributed into groups related to certain services. The definition of the software product that uses this protocol is also performed, and the contents are assembled into a package, according to the format specified by the application (21). From the point of view of functionality, the main contribution of DCI in addition to the classification module (the main functionality of DPI) is a set of parsing modules for various application–level protocols and various types of data in various encodings (for example, MIME) that they contain. The functions of the parsing modules are reduced to two main statements:

1. Analysis of the information contained in the buffer (network packet or session data), based on the specification of the protocol format, for which there is a description in one of the special languages such as ASN.1 and P4.
2. Selection of active connections for protocols with connection establishment and their subsequent investigation.

Currently, a special place is occupied by the issue of the development of DPI and DCI tools in terms of the organization of a centralized and universal research system. Such an implementation is designated as a kind of service. Its idea is that if there are disparate security tools in the system that analyze traffic in various ways, it makes sense to transfer these protection modules to separate hardware. Such a solution will conduct an exhaustive study of network data and transmit its information to all network devices based on their specifics. Shifting priorities towards such a model is similar to the transition to software-Defined Networks (SDN) in terms of working with traffic, where all traffic decisions are made by SDN controllers, which positively affects the scalability of the system and makes it possible to expand the network functionality without debugging and reconfiguring nodes (22).

The idea of using DPI in the format of a separate hardware module is ideal for creating a standardized vulnerability management system and security organization. Such a solution is ideally combined with current trends, where the main aspect is the integration of various components into one hardware and software solution, which will solve the following number of problems:

- Network load balancing.
- The possibility of theft of important information.
- Disparity of components that investigates traffic.

The most common plan for the implementation of this system is shown in Figure 6, where the "external interface" is a hardware part with DPI running on it. Policy and Charging Rules Function is a database that contains information about security settings. "internal interface" is a database containing statistical data, information about package verification, as well as logs containing information about access to system resources.

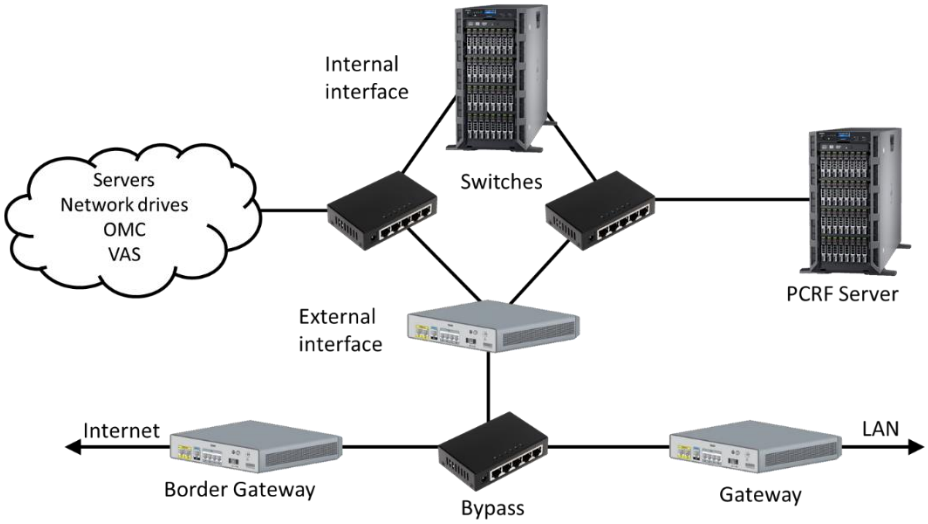


Figure 6: The scheme of using the DPI system to apply policies to network traffic

Also, DPI can be used as a means to collect useful information, a firewall, or as an intrusion detection and prevention system.

Basic Algorithm of Infrastructure Methods of Network Traffic Research

The basic algorithm for studying network packets is divided into the following stages, which are graded according to the degree of representation of the object of study:

Collection of data located in the network connections under study. The resul t of this stage is the received packages:

- Partial packet analysis is a type of analysis in which not all the contents of a package are considered, but only some part of it. The practice of using this type of research has shown its effectiveness in solving the problem of determining the packet protocol.
- The selection of packets according to the specified parameters implies the interception of packets that satisfy them, based on the security settings of the system. Over time, this technology has acquired a large number of selection methods (23).
- For systems where high accuracy of traffic research is required, it is necessary to read the data of traffic passing through the node completely (24).

1. Distribution of packets by threads, based on information about their source. Such a study is divided into two types:
   - With accounting the package data.
   - Without accounting the package data.
2. Their differences are shown in Figure 7. Analysis based on data stream is most common due to low hardware requirements, which is caused by a small amount of processed data. This method of analysis allows you to work with packages, both locally and remotely. Many protocols are used to transfer data from the collection point to the analysis site. The list of data recorded during analysis may vary from implementation to implementation, however, it is possible to identify the main data that are used most often (25):

- Source and destination IP addresses.
- Transport level protocol.
- In the case of TCP/UDP protocols, the source/destination port numbers (26)
- A set of counters: the number of transmitted packets and bytes, the time of creation and completion of the stream.

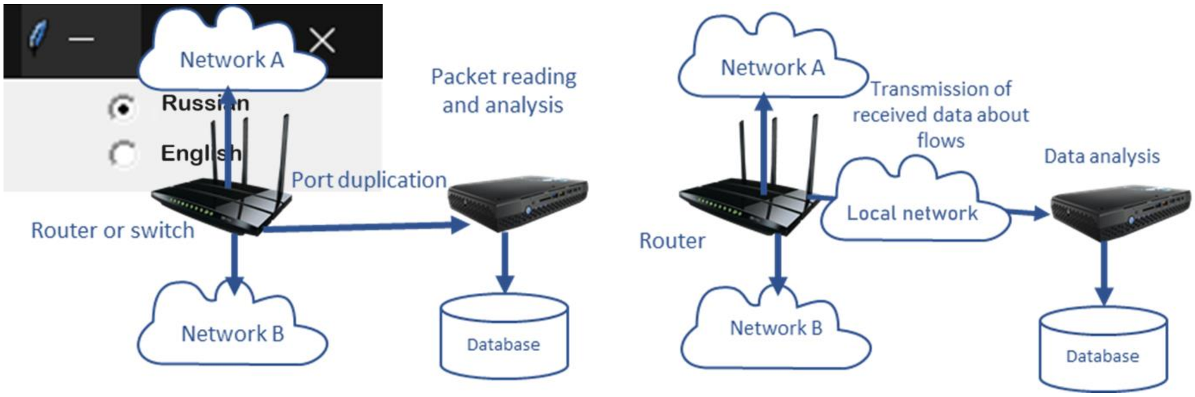


Figure 7: Differences between typical packet (on the left side) and flow-based (on the right side) analysis schemes

It is worth noting that this method rather economically manages the resources of the hardware, but it is not flexible enough – it does not allow you to set the number of received data. Also, in many practical tasks, the number of threads is not much less than the number of packets, due to the presence of many short-lived threads consisting of a small number of packets (27). In order to regulate the number of such streams, the technology of packet selection according to specified conditions was applied. However, due to limited memory, the device will not be able to collect data from a single stream for an infinite amount of time. Therefore, a parameter is introduced in this system that limits the maximum duration of the study.

3. Performing packet separation by application layer protocols or web application. The protocol or application network stream acts as an object for research. Further, additional transformations can be performed on the object, the nature of which is directly related to the applied problem being solved. In general, the following transformations can be distinguished (28):

- Parsing protocol fields.
- Collecting a protocol session for protocols with connection establishment.
- Extraction of application data (content extraction) – site pages (html), files of various types (executable, images, text documents, etc.), emails, audio-video streams, etc..
- Application content parsing.

Additionally, it is worth paying attention to the fact that in addition to these approaches, there is another source of information about network traffic – it is hierarchically organized information that is accessed using the network management protocol (29).

Applications for collecting, storing and transmitting data in MIB format are used in a variety of devices. Information is exchanged via the SNMP protocol. The information obtained by this method has a small size and is not associated with any protocol. As an example of such characteristics, we can give information about the number of packets and the total number of bytes that passed through a certain network interface on a certain device.

It is also worth noting that the reason for the growing popularity of MIB technologies and the distribution of packets across streams are disputes about the legality of deep traffic analysis, since traffic research with such accuracy encroaches on privacy rights. Therefore, in various scientific papers, traffic is encrypted using information security tools before conducting high-precision analysis (30).

Methodology

Development of a Software Tool for analyzing the Possibilities of Carrying Out Attacks on the Functions of Transferring Control to the Operating System Console using Active Intelligence Methods

In this study, we employed a multifaceted approach to evaluate the feasibility of attacks on operating system console control functions via active intelligence methods. Specifically, we developed the MaxNetScanner2022 software, a tool designed to simulate real-world cyber-attack scenarios by scanning for vulnerabilities across network nodes. This tool leverages a comprehensive database of known vulnerabilities, including CVE-2018-20062 and CVE-2018-12536, to perform targeted attacks on simulated network environments.

IDLE (Integrated Development and Learning Environment) was chosen as the development environment, which is an integrated development (and learning) environment that comes with Python. IDLE itself is written in Python using the Tkinter library, so it is a cross-platform application (can run on Windows, Mac OS, Linux). IDLE can also be presented as a text editor with syntax highlighting, autofill, smart indentation and other features for the convenience of writing Python code.

To develop the graphical part of the program, the tkinter library was chosen – a cross-platform library for developing a graphical interface in Python. It is included in the standard Python library package.

This library has the following advantages:

- Support for built-in Windows forms, windows and elements.
- The presence of three types of packers (Alignment of elements: on the edge, on the grid or forced assignment of coordinates).
- Conciseness of the code compared to WinAPI.

Description of the work and testing of the program

The developed software tool is a utility that checks hosts for vulnerabilities of control transfer functions in applications or web services running on them, by conducting active intelligence - checking for open ports, analyzing the server response to requests with distorted data pointers.

This product consists of the following modules, displayed in Table 2. The modules responsible for conducting a vulnerability search work within the framework of decision theory, in particular, a probabilistic problem is solved, which is expressed by the following conditions:

$$\{P\} = \left(\frac{1}{N}\right)^{k} P > \ x$$

where P is the probability that a vulnerability exists, k is the number of successful vulnerability checks, N is the total number of checks implemented in modules, x is the probability value, if exceeded, there is reason to believe that a vulnerability exists.

Each of these modules works according to the following scheme (Figure 8).

When the program starts, the main window becomes available to the user (Figure 9), where it is possible to go to the scanning settings, specify the list of modules to scan, as well as run the scan and view the results of its work.

In the scanning settings window, the user is given the opportunity to add IP addresses of nodes and port numbers for scanning, Figure 10.

Table 2: Modules of the program (Source: compiled by the authors)

ip_list	**stores the entered ip addresses or subnet ip masks;**
port_list list	stores the entered ports
scan_results_l list	stores scan results
variables of the main_ family	store the parameters of the GUI buttons and an instance of the main window object
langs variable	imports a list of available languages from the languages module
scan_sets list	stores settings for selecting active modules
progress variable	stores the percentage value of the scan execution to indicate the progress of the program
operations	calculated variable that stores the number of operations scheduled for execution
step	calculated variable that stores the step value to indicate the progress of the program
hnap_timer	variable that fixes the delay between sent requests
progr_add function	is responsible for updating information about the progress of the scan and the indicator readings
proxy_pump	a class prepared for running in multithreaded processing mode, is responsible for searching for unsafe proxy servers
proxy_scan	the function to start scanning proxy servers
hnap_pump	a class prepared for running in multithreaded processing mode, is responsible for searching for devices with vulnerabilities using the HNAP protocol
hnap_scan	the function to launch the HNAP vulnerability scan
ThinkPHP_pump	a class prepared to run in multithreaded processing mode, is responsible for searching for devices with vulnerabilities in the ThinkPHP web application
ThinkPHP_scan	the function to launch vulnerability scanning in ThinkPHP
MyAdmin_pump	a class prepared to run in multithreaded processing mode, is responsible for searching for devices with vulnerabilities in the phpMyAdmin web application
MyAdmin_scan	the function to run a vulnerability scan in phpMyAdmin
BruteForce_pump	a class prepared for running in multithreaded processing mode, is responsible for sorting credentials in login forms
BruteForce_scan	brute force start function
HJS_pump	class, prepared for running in multithreaded processing mode, is responsible for searching for unsafe Hudson Java servers
HJS_scan	the function to start scanning the Hudson Java server
SQLite_pump	a class prepared for running in multithreaded processing mode, is responsible for searching for unsafe SQLite servers
SQLite_scan	the function to start scanning SQLite servers
scan_sets_ind	a function that sets flags in the scanning module selection menu
is_port_valid	a function that checks whether the port number is entered correctly
is_ip_valid	a function that verifies the correctness of entering the ip address
value_input_wnd functions	displays various windows-prompts for data entry
refresh_ip_port function	saves updated information about the list of scanned ip addresses and ports

functions of the scan_ family	are responsible for the operation of the scanning settings windows, the selection of plug–ins
language_change function	is responsible for restarting the main window when changing the language
language_settings function	is responsible for the operation of the language settings window
main_window function	is responsible for the operation of the main menu window
LOCALIZED_STRINGS dictionary	contains localization strings for Russian and English

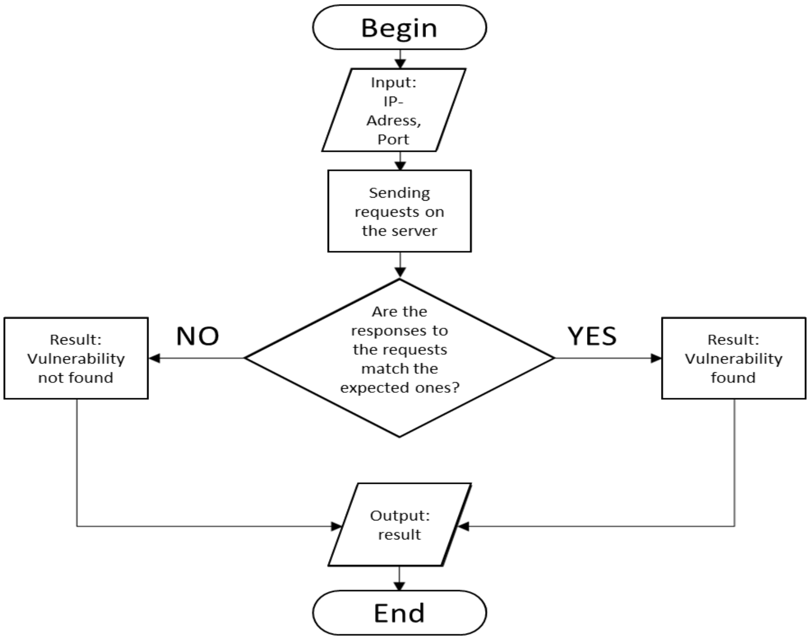

Begin
Input: IP-Adress, Port
Sending requests on the server
Are the responses to the requests match the expected ones?
NO
YES
Result: Vulnerability not found
Result: Vulnerability found
Output: result
End

Figure 8: General flow diagram of the vulnerability scanning module

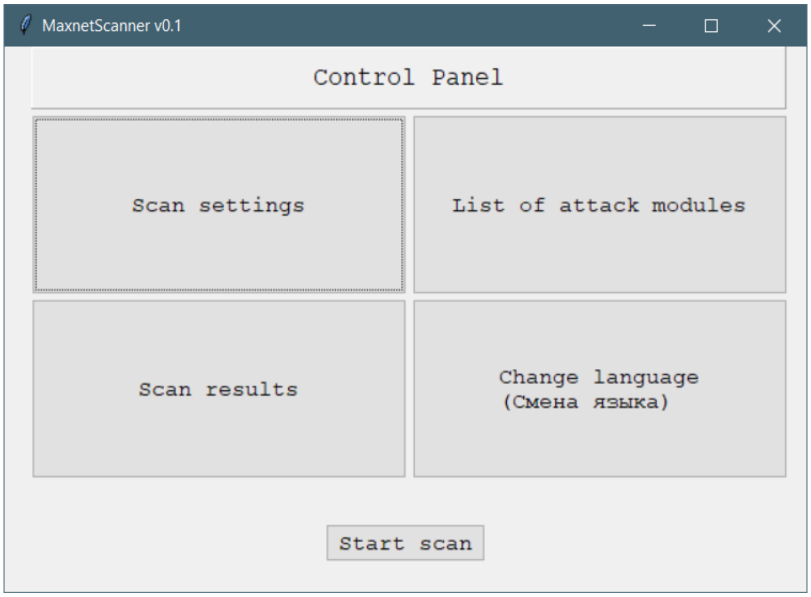
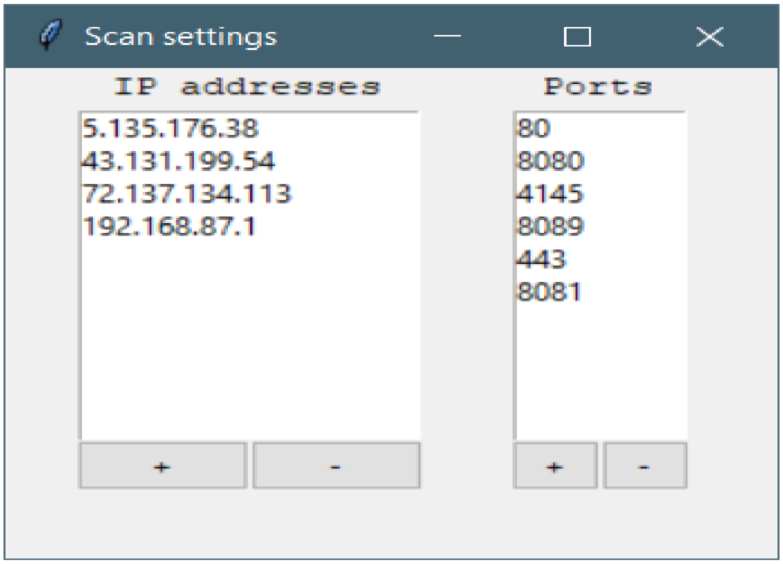


Figure 9: Main window of software



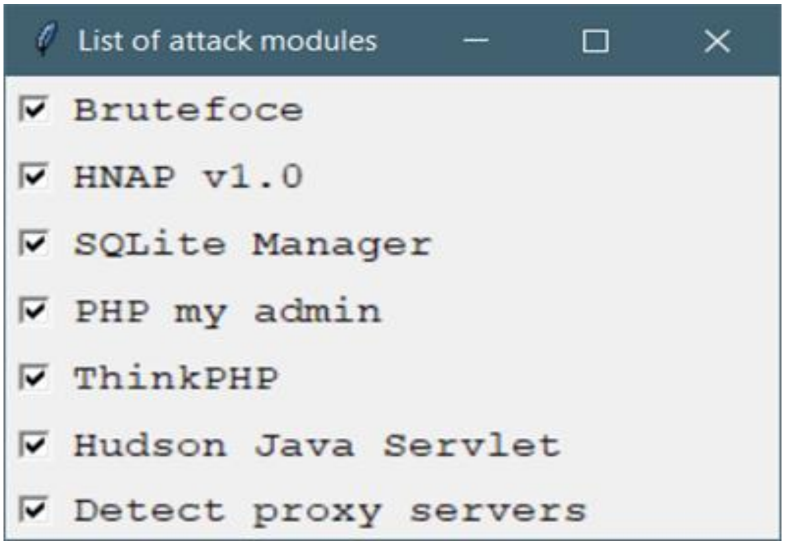
IP addresses

5.135.176.38
43.131.199.54
72.137.134.113
192.168.87.1

Ports

80
8080
4145
8089
443
8081

Figure 10: Scan Settings window



☑ Bruteface
☑ HNAP v1.0
☑ SQLite Manager
☑ PHP my admin
☑ ThinkPHP
☑ Hudson Java Servlet
☑ Detect proxy servers

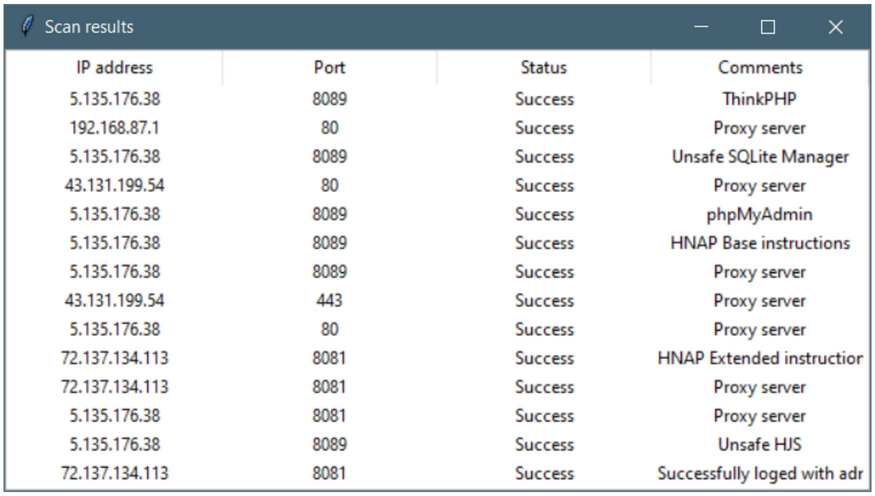
Figure 11: Module selection window

The choice of modules for scanning nodes in the network is possible using the module selection window, accessible from the main menu, item "List of modules", Figure 11.

It is worth noting that, unlike Router Scan, this product runs modules in parallel and outputs the results of the modules separately, which allows you to get a more detailed picture of the presence of possible vulnerabilities.

The scan results are recorded in a special table, which is accessible from the main menu by pressing the "Scan Results" button, Figure 12.

When the scan starts, a special window opens in which the user is shown the progress of the work by filling in the progress bar, according to the percentage of completion of the running scan, Figure 13.



IP address	Port	Status	Comments
5.135.176.38	8089	Success	ThinkPHP
192.168.87.1	80	Success	Proxy server
5.135.176.38	8089	Success	Unsafe SQLite Manager
43.131.199.54	80	Success	Proxy server
5.135.176.38	8089	Success	phpMyAdmin
5.135.176.38	8089	Success	HNAP Base instructions
5.135.176.38	8089	Success	Proxy server
43.131.199.54	443	Success	Proxy server
5.135.176.38	80	Success	Proxy server
72.137.134.113	8081	Success	HNAP Extended instructior
72.137.134.113	8081	Success	Proxy server
5.135.176.38	8081	Success	Proxy server
5.135.176.38	8089	Success	Unsafe HJS
72.137.134.113	8081	Success	Successfully loged with adr

Figure 12: Vulnerability Search Results demonstration window

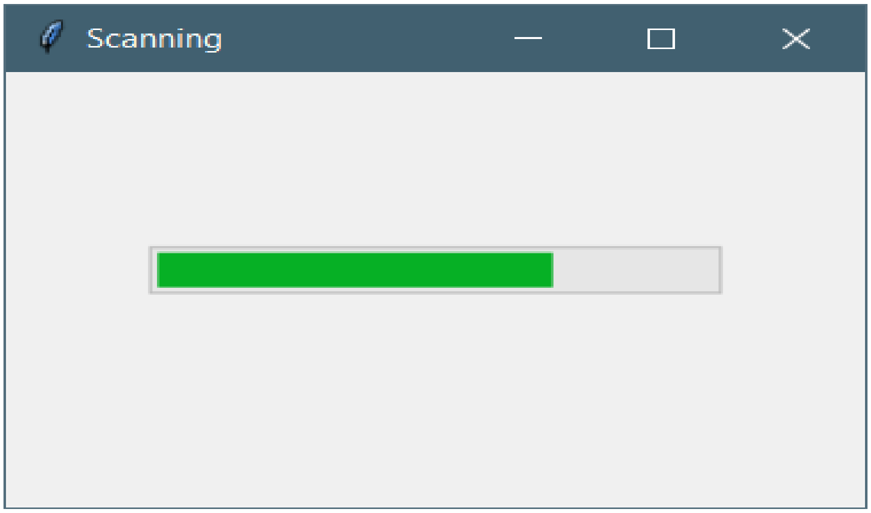
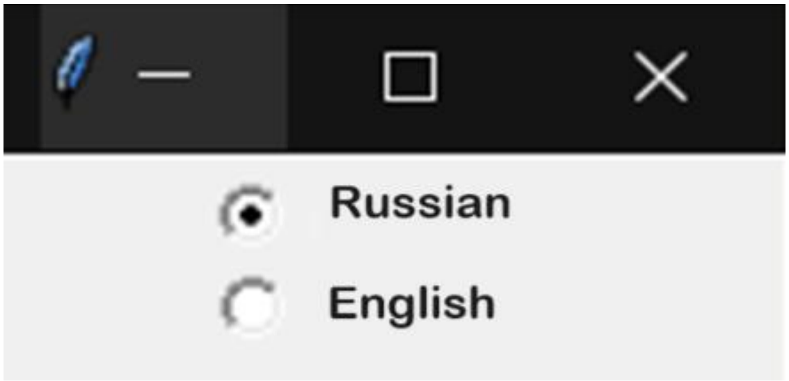


Figure 13: Scan Progress indicator window

The program also implements the possibility of changing the language, for this it is necessary to select the item "Change language" in the main window, the view of the language selection window is shown in Figure 14.



● Russian

○ English

Figure 14: Language Selection panel

When you select a language, the main window will restart to change the localization lines.

Comparison with analogues

The table shows a comparison of the capabilities of various attacks and performance on various systems of the developed software with existing software products:

Table 3: Comparative analysis of software tools

Programs **Explored possibilities**	**RouterScan**	**MaxnetScanner**	**Patrator**	**Retina**
Windows support	+	+	+	+
Linux support	+	+	+	+
Bruteforce	+	+	+	-
HNAP	+	+	-	+
PhpMyAdmin	+	+	+	+
ThinkPHP	-	+	+	-
HudsonJavaServlet	+	+	+	-
SQLite Manager	+	+	+	+
Proxy-serverdetecting	+	+	-	+

Practical tests were also conducted on the quality of vulnerability search and the time spent on their detection for each software product under consideration. All tests were conducted in a dedicated LAN for testing. In order to achieve maximum objectivity of the study, web applications were selected that may have vulnerabilities handled in the software products in question. Thus, the local network involved: 2 routers with support for the insecure version of HNAP 1, 5 servers with administration using phpMyAdmin, 2 servers with the ThinkPHP framework, 3 Java servers with the Hudson servlet extension, 2 SQLite servers and one proxy server that works without encryption of the transmitted data.

Results and Discussion

The results of the study are presented in Figure 15 and Figure 16.

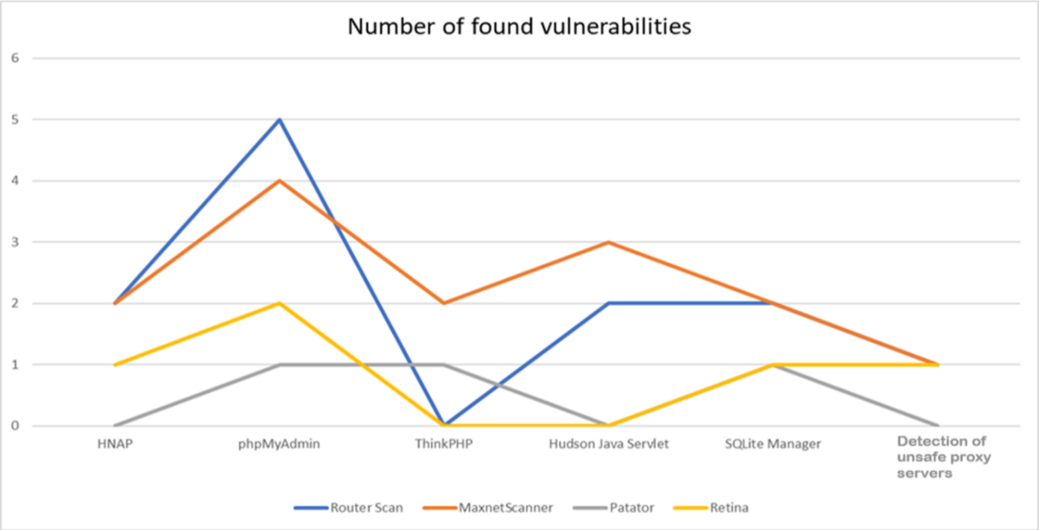


Figure 15: Diagram of the number of vulnerabilities found by programs

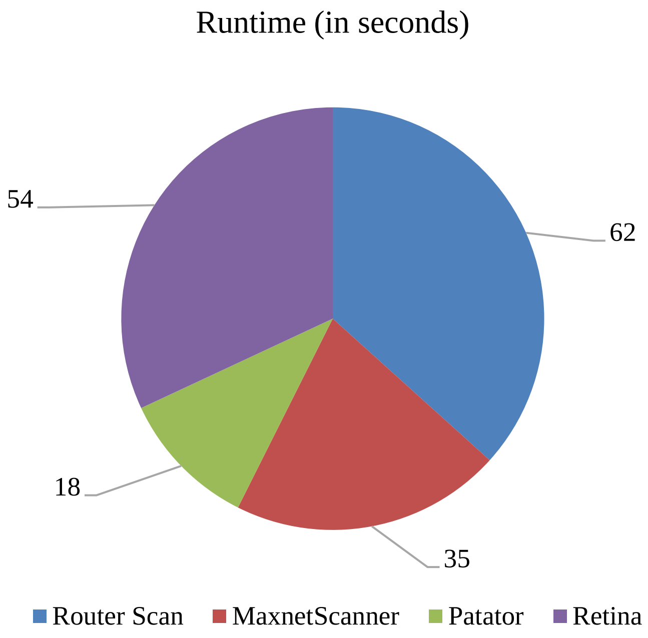


Figure 16: Diagram of the operating time of programs in the test LAN

As a result of the study, MaxnetScanner software tool obtained during the development process performed best in testing systems for the presence of vulnerabilities in the ThinkPHP framework and the Hudson Java Servlet server, according to other indicators, the developed program was only slightly inferior in detecting vulnerabilities in the phpMyAdmin web application. In terms of speed, the resulting software product took second place among the studied.

Thus, based on the information presented in Table 1, Figure 15 and Figure 16, the MaxnetScanner software product, which was developed as part of the current study, proved to be competitive, surpassing in some aspects existing software solutions on the market. It is worth noting that this type of attack is quite common on the network and includes such methods as: substitution of data in the request, transmission of characters to the request body that the server can read without escaping them, which can cause inappropriate behavior of the web application and allow the attacker to execute code on the remote server side (31, 32).

The authors of this study describe modern approaches to traffic analysis, they come to the conclusion that the choice of the analysis method should be based on the expediency of the means used, relative to a specific task, that is, it is necessary to take into account the amount of analysis costs and the adequacy of its depth in terms of the load on the system under study. The authors also note that attacks using active intelligence tools are the most dangerous due to the fact that they have a mass character, can spread malicious software and are capable of destabilizing the system by simply sending multiple requests that the server cannot leave unanswered and spends its processor time processing them. Having studied recent articles about attacks on the system control transfer functions, the authors concluded that modern web applications can be subject to such attacks using active intelligence tools, since today most companies and various communities pay insufficient attention to configuring and testing servers, web applications and network screens.

This research has successfully demonstrated the feasibility of exploiting vulnerabilities in console control transfer functions through active intelligence methods. The study's in-depth analysis of CVE-2018-20062 and CVE-2018-12536 vulnerabilities underscored the critical importance of network security in modern computing environments. By employing a combination of techniques like ports scanning, directories searching, modifying parameters, and credentials searching, the study highlighted significant security gaps that can be exploited by attackers.

The development and implementation of a specialized software tool proved to be a pivotal aspect of this study. This tool, designed to identify vulnerabilities in network nodes, serves as a significant advancement in cybersecurity practices. It not only aids in pinpointing potential security weaknesses but also offers a framework for developing more robust defense mechanisms against similar attacks in the future.

We've gone deeper into exploring different ways attackers might try to break into systems, looking at their techniques, why they might do it, and what could happen if they're successful.

Our findings advocate for a dynamic approach to cybersecurity, emphasizing the necessity of continually updating security measures and protocols to address the ever-evolving threat landscape. Moreover, the insights derived from this study serve as a valuable resource for policymakers, informing the creation of comprehensive cybersecurity frameworks that prioritize preemptive measures. Additionally, this research acts as a catalyst for innovation, encouraging the development of new security technologies and solutions that anticipate and neutralize potential cyber threats before they can exploit system vulnerabilities.

Furthermore, the findings of this study have broader implications for the field of cybersecurity. They emphasize the need for continuous monitoring and updating of security protocols to guard against evolving cyber threats. The study also opens avenues for further research, particularly in the area of developing more advanced tools and methods for detecting and mitigating such vulnerabilities. While the MaxNetScanner2022 software demonstrates promising capabilities in detecting vulnerabilities using active intelligence methods, future research should explore the integration of artificial intelligence (AI) and machine learning (ML) to enhance its adaptability to emerging threats. The dynamic landscape of cyber threats necessitates continuous advancements in detection technologies to maintain efficacy against new attack vectors.

Conclusion

Thus, the tasks set in the scientific work were fulfilled, and the goals were achieved. The experience gained during the research and the acquired skills will be useful in further training and professional activity.

Abbreviation

Common Vulnerabilities and Exposures (CVE); Hypertext Transfer Protocol (HTTP); Telecommunication Network (TELNET);BFile Transfer Protocol (FTP); Simple Mail Transfer Protocol (SMTP); Post Office Protocol 3 (POP3); Internet Message Access Protocol (IMAP); Network News Transfer Protocol (NNTP); Internet Relay Chat (IRC); Home Network Administration Protocol (HNAP); Structured Query Language (SQL); Hypertext Preprocessor (PHP); Continuous Integration (Ci); Cross-Site Scripting (XSS); Surface Packet Inspection (SPI); Middle Packet Inspection (MPI); Deep Packet Inspection (DPI); Packet Based No State (PBNS); Packet Based Per Flow State (PBFS); Message Based Per Flow State (MBPS); Detailed Study of the Package Composition (DCI); Software-Defined Networks (SDN); Intrusion Detection System (IDS); Intrusion Prevention System (IPS); Management Information Base (MIB); Simple Network Management Protocol (SNMP); Integrated Development and Learning Environment (IDLE); Graphical User Interface (GUI); Domain Name System (DNS); Transmission Control Protocol (TCP); User Datagram Protocol (UDP); Application Programming Interface (API); Local Area Network (LAN).

Acknowledgment

Nil

Author Contributions

Larisa Cherckesova wrote the final paper, participated in research and editing of the article. Elena Revyakina: Drafted, participated in writing and research for the article.Olga Safaryan participated in all stages of the research and writing, Is responsible for graphic design. Vitaliy Porksheyan participated in writing and editing the final version of the manuscript, aided with all the paperwork. Maxim Kazaryan was responsible was data collection

and management, aided with editing and cleaning up the manuscript.

Conflict of Interest

The authors declare no conflict of interest.

Ethics Approval

Ethical approval was obtained from ethics committee of the Don State Technical University

Funding

Nil

Reference

1. Sardjono W, Perdana WG. Interactive User Interfaces in the Digital World Make the Application Attractive and Easier for User Access. J Theor Appl Inf Technol. 2024; 101(1): 7266-7274.
2. Neza V, Llazo E. E-Service as a sustainable future: A case of e-Albania. Int J Sustain Dev Plan. 2023; 18(12): 3873-3881.
3. Hdidou R, El Alami M. Intrusion Detection Systems in Internet of Things: A Recent State of the Art. J Theor Appl Inf Technol. 2024; 101(1): 297-317.
4. Vanel AR, Innocent EJ, Coffi AC. Analysis of cascading effects on key urban networks during flooding in Brazzaville, Congo. Int J Sustain Dev Plan. 2023; 18(11): 3467-3475.
5. Sahib HR, Al-Kutubi HS. Evaluating parameters and survival function in the exponential distribution model: A contrast between complete and censored data. Math Model Eng Probl. 2023; 10(6): 2063-2068.
6. Fatman AN, Ahmad T, Jean De La Croix N, Hossen MS. Enhancing data hiding methods for improved cyber security through histogram shifting direction optimization. Math Model Eng Probl. 2023; 10(5): 1508-1514.
7. Network sniffing on switches (arp mac security sniffer ethernet). [Online]. [Accessed 1 June 2022]. Available from: https://www.opennet.ru/base/sec/arp_snif.txt.html.
8. Router Scan v2.60 Beta by Stas'M. [Online]. [Accessed 1 June 2022]. Available from: http://stascorp.com/load/1-1-0-56
9. Retina Network Security Scanner. [Online]. [Accessed 1 June 2022]. https://drweb.datasystem.ru/catalog/view/645/
10. BeyondTrust. Retina Network Security Scanner. [Online]. [Accessed 1 June 2022]. Available from: https://www.beyondtrust.com/products/retina-network-security-scanner/
11. Cherckesova LV, Safaryan OA, Beskopylny AN, Revyakina E. Development of Quantum Protocol Modification CSLOE–2022, Increasing the Cryptographic Strength of Classical Quantum Protocol BB84. Electronics. 2022; 11(23): 3954.
12. Korochentsev DA, Cherckesova LV, Revyakina EA, Goncharov RA. Investigation of the Application of Software Generator of the Speech-Like Interference to Protect Acoustic Information from Leakage through an Acoustic Channel. J. Phys. Conf. Ser. 2021; 2131(2): 022091.
13. Lin YD, Lin PC, Prasanna VK, Chao HJ, Lockwood JW. Guest Editorial Deep Packet Inspection: Algorithms, Hardware, and Applications. IEEE J. Sel. Areas Commun. 2014; 32(10): 1781-1783.
14. ITU-T Y.2770 recommendation "Requirements for deep packet inspection (DPI) in next generation networks (NGNs)". 2012. International Telecommunication Union. [Online]. [Accessed 1 June 2022]. Available: https://www.itu.int/rec/dologin_pub.asp?lang=f&id=T-REC-Y.2770-201211-I!!PDF-E&type=items
15. ITU-T Y.2771 recommendation "Framework for deep packet inspection". 2014. International Telecommunication Union. [Online]. [Accessed 1 June 2022]. Available: https://www.itu.int/rec/T-REC-Y.2771/_page.print
16. ITU-T Y.2772 recommendation "Mechanisms for the network elements with support of deep packet inspection". 2016. International Telecommunication Union. [Online]. [Accessed 1 June 2022]. Available: https://ouci.dntb.gov.ua/en/works/7qozOjQl/
17. ITU-T Y.2773 recommendation "Performance models and metrics for deep packet inspection". 2017. International Telecommunication Union. [Online]. [Accessed 1 June 2022]. Available: https://handle.itu.int/11.1002/1000/13015
18. Filimonov P, Ivanov M. Modern approaches to classifying the traffic of physical Internet channels. In: Distributed computer and communication networks: control, computing, communication (DCCN-2015). Proceedings 18th International Conference. 2015 Oct 19-22; Moscow, Russia. p. 466-474.
19. Aun Y, Manickam S, Karuppayah S. A temporal-aware signature extraction method using sliding-window mechanism for scalable, cost-effective and accurate traffic classification. In Proceedings - 7th IEEE International Conference on Control System, Computing and Engineering, 2017 Nov; Nee Jersey, USA. p. 156-161.
20. Bosshart P, Daly D, Gibb G, Izzard M, McKeown N, Rexford J, Schlesinger C, Talayco D, Vahdat A, Varghese G, Walker D. P4: Programming protocol-independent packet processors. Comput. Commun. Rev. 2014; 44(3): 87–95.
21. Bremler-Barr A, Harchol Y, Hay D, Koral, Y. (2014). Deep Packet inspection as a service. In CoNEXT 2014 - Proceedings of the 2014 Conference on Emerging Networking Experiments and Technologies. 2014 Dec 2-5; Sydney, Australia. p. 271–282.
22. Shandilya SK, Ganguli C, Izonin I, Nagar PAK. Cyber-attack evaluation dataset for deep packet inspection and analysis. Data in Brief. 2023, 46.
23. Srisuresh P, Holdrege M. IP Network Address Translator (NAT) Terminology and Considerations. [Online]. IETF Informational. 1999; 53: 1689–1699. [Accessed 1 June 2022]. Available: https://tools.ietf.org/html/rfc2663.html

24. Software NAT. [Online]. [Accessed 1 June 2022]. Available: http://www.nat32.com/v2/
25. Kotey SD, Tchao ET, Gadze JD. On Distributed Denial of Service Current Defense Schemes. Technologies. 2019; 7(1): 19.
26. Ribeiro RH, Jacobs AS, Zembruzki L, Parizotto R, Scheid EJ, Schaeffer-Filho AE, Granville LZ, Stiller, B. A deterministic approach for extracting network security intents. Computer Networks. 2022; 214: 109109.
27. Afek Y, Bremler-Barr A, Harchol Y, Hay D, Koral Y. (2012). MCA2: Multi-core architecture for mitigating complexity attacks. In ANCS 2012 - Proceedings of the 8th ACM/IEEE Symposium on Architectures for Networking and Communications Systems. 2012 Oct 29-30; Austin, TX USA. p. 235–246.
28. Nguyen XH, Nguyen XD, Huynh HH, Le KH. Realguard: A Lightweight Network Intrusion Detection System for IoT Gateways. Sensors. 2022; 22(2): 432.
29. Abro GEM, Zulkifli SABM, Masood RJ, Asirvadam, VS, Laouiti A. Comprehensive Review of UAV Detection, Security, and Communication Advancements to Prevent Threats. Drones, 2022; 6: 284.
30. Oruc A, Amro A, Gkioulos V. Assessing Cyber Risks of an INS Using the MITRE ATT&CK Framework. Sensors. 2022; 22(22): 8745.
31. Sirotkin A, Sharypov S. Construction of optimization model of system of the object control and monitoring. Int. Res. J. 2016; 6(48): 126-129.
32. Albalawi M, Aloufi R, Alamrani N, Albalawi N, Aljaedi A, Alharbi AR. Website Defacement Detection and Monitoring Methods: A Review. Electronics. 2022; 11: 3573.