3-бөлім	Раздел 3	Section 3
Информатика	Информатика	Computer Science

IRSTI 20.15.05 DOI: https://doi.org/10.26577/JMMCS2024121110

A. Makulova[1], **B. Sharipova**[2], **M. Othman**[3], **A. Pyrkova**[4], **G. Ordabayeva**[4*]

[1]Narxoz university, Kazakhstan, Almaty
[2]Almaty technological university, Kazakhstan, Almaty
[3]Universiti Putra Malaysia, Malaysia, Selangor D.E.
[4]Al-Farabi Kazakh national university, Kazakhstan, Almaty
*e-mail: gulzi200988@mail.ru

DETECTION OF OPERATING SYSTEM VULNERABILITIES AND NETWORK TRAFFIC ANALYSIS METHODS

Researchers and experts on information protection develop antivirus programs and applications to improve the security of operating systems and security policies. Threats will be relevant to organizations that do not consider security policies and regular software updates. This paper discusses applications for scanning and analyzing network traffic, such as Netdiscover, Wireshark, and Nmap. The model network is based on a virtual machine. This research aims to determine methods for scanning and analyzing network traffic and detecting network vulnerabilities. This study conducted a penetration test for Windows 10 using the Kali Purple operating system and identified the vulnerability of the operating system. The calculation of network traffic is performed with (1) the determination of the arithmetic means of network traffic, (2) the calculation of the variance, and (3) the determination of the magnitude of fluctuations relative to the average M, the range of maximum and minimum values of D, and the Hurst coefficient. As a result of the conducted research on students enrolled in the educational program 6B06301 – Information Security Systems at Farabi University, the proficiency in MS Excel and C# skills amounted to 77.11%. The research results can be used in the field of information security systems.

Key words: network traffic, penetration, analysis, vulnerability, exploit, attack, Kali Linux, Windows.

А. Макулова[1], Б. Шарипова[2], M. Othman[3], А. Пыркова[4], Г. Ордабаева[4*]
[1]Narxoz университеті, Қазақстан, Алматы қ.
[2]Алматы технологиялық университеті, Қазақстан, Алматы қ.
[3]Putra университеті, Малайзия, Селангор Д.Е.
[4]Әл-Фараби атындағы Қазақ ұлттық университеті, Қазақстан, Алматы қ.
*e-mail: gulzi200988@mail.ru

Операциялық жүйе осалдығын анықтау және трафикті талдау әдістері

Зерттеушілер мен ақпаратты қорғау жөніндегі сарапшылар операциялық жүйелердің қауіпсіздігін және қауіпсіздік саясатын арттыру үшін вирусқа қарсы бағдарламалар мен қосымшалар әзірлейді. Мақалада Netdiscover, Wireshark және Nmap сияқты желілік трафикті сканерлеуге және талдауға арналған қосымшалар қарастырылды. Желілік трафикті талдай білу – киберқорғаудың алғашқы желісі. Виртуалды кеңістік – деректерді қорғау технологиясы саласындағы оқыту сценарийлерін іске асыруға арналған орын. Осы зерттеудің мақсаты желілік трафикті талдау әдістерін және желінің осалдығын анықтау. Зерттеуде Kali Purple операциялық жүйесінің көмегімен Windows 10 жүйесіне ену тесті жүргізілді және операциялық жүйенің осалдығы анықталды. Сондай-ақ, желілік трафиктің орташа арифметикалық мәнін (1), дисперсияны есептеу (2), орташа M (3) қатысты ауытқу мәнін анықтау, D максималды және минималды мәнінің диапазоны және Херст коэффициентіне талдау жүргізілді.

Желілік трафикті талдаудың және осалдықтарды анықтаудың ұсынылған әдістемесі Ethernet жергілікті желісіне шабуылдарды неғұрлым жоғары дәлдікпен және толықтықпен анықтауға және бұғаттауға мүмкіндік берді. Нәтижеде анықталған Херст коэффициентінің мәні $((\leq 0,5))$ өзіндік ұқсастығы жоқ эргодикалық қатар екені айқындалды. Сонымен қатар, орындалған зертханалық жұмыс нәтижесінде Farabi университетінің 6B06301 – Ақпараттық қауіпсіздік жүйелері білім беру бағдарламасы бойынша студенттердің MS Excel және C# бойынша біліктілігі 77,11% тең болды. Алынған зерттеу нәтижелері ақпараттық қауіпсіздік жүйесі саласында пайдаланылуы мүмкін.

Түйін сөздер: желілік трафик, ену, талдау, осалдық, эксплоит, шабуыл, Kali Linux, Windows.

А. Макулова[1], Б. Шарипова[2], M. Othman[3], А. Пыркова[4], Г. Ордабаева[4]*
[1]Университет Narxoz, Казахстан, г. Алматы
[2]Алматинский технологический университет, Казахстан, г. Алматы
[3]Университет Putra, Малайзия, Селангор Д.Е.
[4]Казахский национальный университет имени аль-Фараби, Казахстан, г. Алматы
*e-mail: gulzi200988@mail.ru

Обнаружения уязвимости операционной системы и методы анализа сетевого трафика

Исследователи и эксперты по защите информации разрабатывают антивирусные программы и приложения для повышения безопасности операционных систем и политик безопасности. В данной статье рассматриваются приложения для сканирования и анализа сетевого трафика, такие как Netdiscover, Wireshark и Nmap. Умение анализировать сетевой трафик – первая линия защиты от киберугроз. Виртуальное пространство – место для реализации сценариев обучения в области технологии защиты данных. Цель данного исследования определить методы анализа сетевого трафика и обнаружение уязвимости сети. В данном исследовании с помощью операционной системы Kali Purple проведен тест на проникновение в Windows 10 и определена уязвимость операционной системы. Также, проведен расчет сетевого трафика с определением: среднеарифметическое значение сетевого трафика (1), вычисление дисперсии (2), определение значения колебаний относительно среднего M (3), диапазон максимального и минимального значения D и коэффициент Херста. Предложенная методика анализа сетевого трафика и обнаружения уязвимостей позволила с более высокой точностью и полнотой выявить и блокировать атаки на локальную сеть Ethernet. По результатам показателя Херста $(\leq 0,5)$ определен эргодический ряд, который не обладает самоподобием. В результате проведенного исследования студентов по образовательной программе 6B06301 – Системы информационной безопасности университета Farabi навыки работы с MS Excel и C# составили 77,11%. Полученные результаты исследования могут быть использованы в области системы информационной безопасности.

Ключевые слова: сетевой трафик, проникновение, анализ, уязвимость, эксплоит, атака, Kali Linux, Windows.

1 Introduction

Today, data transmission is developing rapidly. This means the availability of the local network and easy connection of users. The local data transmission environment also creates conditions for listening to network traffic and connecting attackers to the network. Unregistered port numbers make network traffic monitoring and intrusion detection difficult.

Klenilmar L. Dias et al. [7] consider a module for classifying video traffic based on machine learning. The proposed naive Bayes algorithm is used to relax the independence hypothesis and in quality assurance schemes for computer networks. The results of this module are applied in real-time scenarios.

Some other authors [4] propose an effective statistical approach to attack detection based on traffic characteristics and an algorithm for dynamic detection of threshold values. The data from the MIT Lincoln Laboratory DARPA and developments of the university laboratory using this algorithm were used to derive attributes based on distributed denial-of-service characteristics.

Markus Ring et al. [13] proposed a new methodology for generating real network traffic based on the flow for evaluating network-based intrusion detection systems (NIDS). The data is based on Generative Adversarial Networks (GANs), which are used to generate images. The new method proposed for estimating generated network traffic based on the flow of the CIDDS-001 dataset has shown the ability to generate high-quality data.

In this paper, the algorithms for modeling network graphs are considered, and applications for network analysis are used. The experimental part shows the analysis of network traffic based on a virtual machine and the use of network traffic filtering. In order to protect the information, an exploit of the Windows operating system was identified, and a vulnerability scan module was searched. Based on the results of the network traffic calculation, the Hurst index ($\leq 0,5$) is obtained.

2 Methods and materials

In the research by N. Clarke et al. [1], traffic metadata is used to identify users. The results of the experiments conducted to investigate the relationship between user actions and network signals are shown in Table 1.

F. Pacheco et al. [11] have studied the methods of machine learning and Deep Learning (DL) for the classification of Internet traffic. The platform under study is satellite communications, where encrypted, unencrypted, and tunnel communications are considered.

Table 1: Services and interaction with users [1].

Services	User interaction
Bbc	Watching video clips, TV programs, listening to audio clips, commenting, sharing news
Dropbox	Uploading files, general viewing of files, folders
Facebook, Twitter	Posting, commenting, sharing, finding friends, attaching files, chatting, messaging
Google	Keyword searching, creating, editing, deleting online documents
Hotmail	Downloading and uploading file attachments, composing an email, deleting content, replying to email
Skype	Sending text messages, transferring files, changing online presence
YouTube	Searching, watching videos, downloading songs and videos, writing comments
Wikipedia	Searching, reading articles

Y. Kawasaki et al. [6] propose a state-space model that estimates the traffic state over a two-dimensional network with alternative routes. This method also employs sequential Bayesian filtering with a cell transmission model GTM for the flow model.

The article by Makarenko S.I. et al. [9] presents a comparative analysis of foreign and Russian penetration testing methodologies and standards, such as The Open Source Security Testing Methodology (OSSTMM), Information System Security Assessment Framework

(ISSAF), Open Web Application Security Project (OWASP), Penetration Testing Execution Standard (PTES), Technical Guide to Information Security Testing and Assessment (NIST SP 800-115), Study a Penetration Testing Model (BSI), Methodology of Information Systems Security Penetration Testing (PETA), Penetration Testing Framework (PTF), and Positive Technologies. Additionally, definitions of basic terminology are provided.

Ethical hacking (pentest, pentesting) involves authorized simulated attacks on computer systems to identify weaknesses in the security system. It is also used to assess the security of operating systems, network security, web applications, and wireless systems. To protect the system, professional pentesters utilize various tools and methods that malicious actors use for hacking. Stages of penetration testing:

1) Information gathering – detecting network hosts, open ports, etc.;

2) Vulnerability analysis – checking for unpatched systems, misconfigurations, etc.;

3) Exploitation – gaining access using discovered vulnerabilities;

4) Post-exploitation – maintaining access using backdoors, rootkits, etc.;

5) Reporting – presenting results and recommendations for preventing identified vulnerabilities.

A research study was conducted at the Faculty of Information Technology of Farabi University to identify vulnerabilities in the Windows 10 operating system. The research consisted of two parts: conducting a penetration test and calculating the Hurst exponent using Wireshark.

For the study, students specializing in information security systems at the Faculty of Information Technology were selected. A survey was conducted among the students, which included the following questions: 1) participation in CTF competitions; 2) knowledge of scanning tools; 3) practical experience in identifying OS vulnerabilities. Data analysis involved observing 86 participants. According to the survey results, 21% of students participated in CTF competitions, 62% possessed practical skills in scanning tools, and 17% had practical experience in identifying OS vulnerabilities.

3 Results and discussion

The experimental part of the work analyzes network traffic based on a virtual machine. Data connection type is a network bridge. In order to analyze network traffic, we use the *Kali Purple* operating system.

The *Netdiscover* utility discovering network interfaces without an IP address configuration was used to determine the nodes available on the network.

One of the key features of the Wireshark utility is traffic interception. The Wireshark utility fixes the problem with the network, debugging of web applications, network programs, and sites and allows viewing the packet data at all OSI levels.

The Wireshark window consists of panels: Packet List, Packet Details, and Packet Bytes. In the window, one can see the traffic related to the wireless access point and which protocols are used.

In the *Filter* menu, entering the command *ip. src==192.168.137.136* and pressing *Enter* make it possible to get only those packets that came from the specified IP address and the results of filtering.

The *Statistics-Capture File Properties* command shows the average number of packets per second, the average packet size, and the traffic intensity.

Traffic results are as follows: packet intensity $\lambda = 2.5$ packets/s, average packet size $L = 159$ bytes, and traffic intensity $a = 3159$ kb/s.

An open-source network scanner used in Windows and Kali Purple operating systems is Network Mapper (*Nmap*). This utility determines the devices connected to the network, installed programs, the type of operating system, and the types of filters applied. *Nmap* opens a port on a computer and uses incoming connections to connect to another program.

Yu.V.Markin [10] considers a table with the results of network analyzers (Table 2):

Table 2: Summary table of the overview of network analyzers [10].

Objectives	Wireshark	Snort	Bro	ntopng
Thread recovery	+/-	-/+	+	-
Analysis of encrypted data	+	-	-	-
Analysis of nested tunnels	+	-	+	-
Adding protocol support	-/+	+/-	+/-	+/-

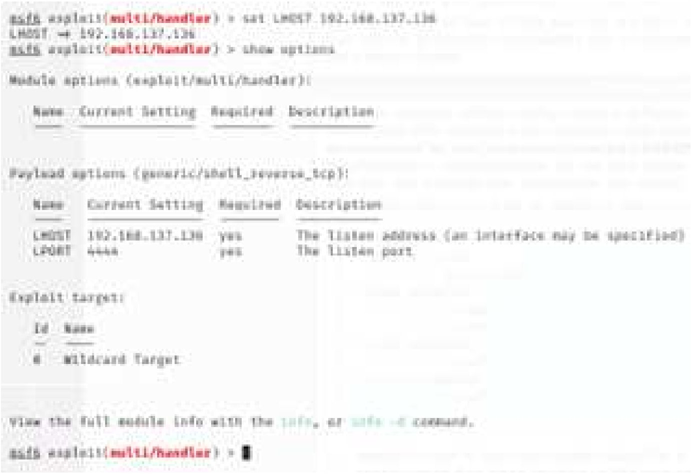
In order to achieve the set goals, the following requirements have been developed:

1. Difference in data flows when sending and recovering;

2. Supported format of archived and classified traffic;

3. Supported tunnel protocols with arbitrary stack configuration.

Based on the analysis performed to protect the information, exploits for the Windows 7 operating system were identified, and a search for the exploit/multi/handler vulnerability scanning module was performed (Fig. 1).

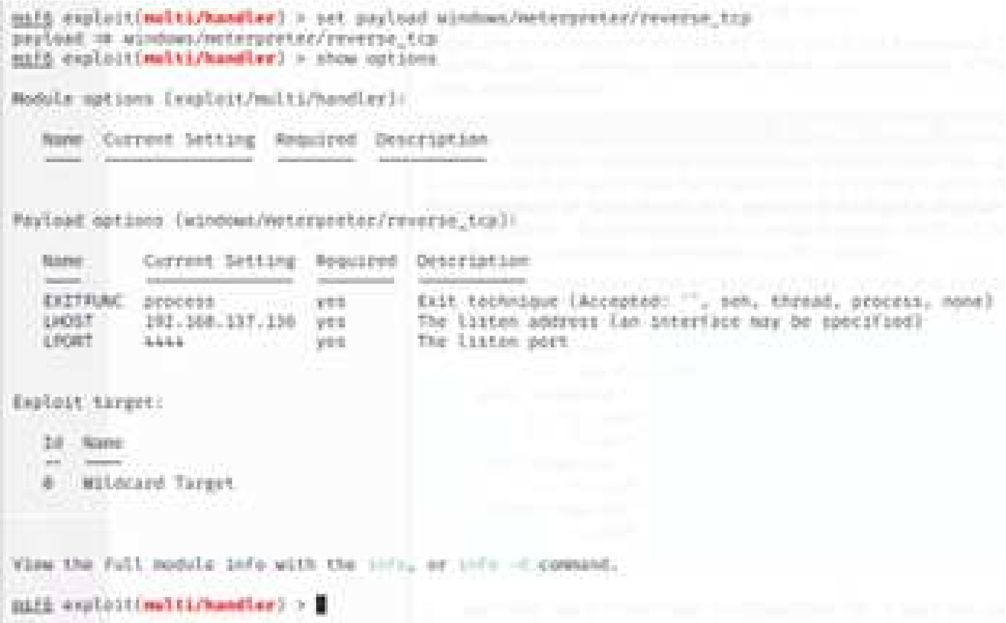


```
msf6 exploit(multi/handler) > set LHOST 192.168.137.136
LHOST => 192.168.137.136
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (generic/shell_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.137.136  yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target



View the full module info with the info, or info -d command.

msf6 exploit(multi/handler) > 
```

Figure 1: Defining exploits

The use of certain network layer vulnerabilities is based on the IEEE 802.11 standard. For the experiment, a test local private wireless network Broadcom 802.11 n Network Adapter was used. The attack generation environment uses a virtual machine with the installed Kali Purple distribution version kali-linux-2024.1-installer-purple-amd64.iso with a set of special utilities for testing for network penetration. A virtual machine with Windows 7 OS was used to analyze the attacks. Metasploit Exploitation Framework tool was used to test for penetration.

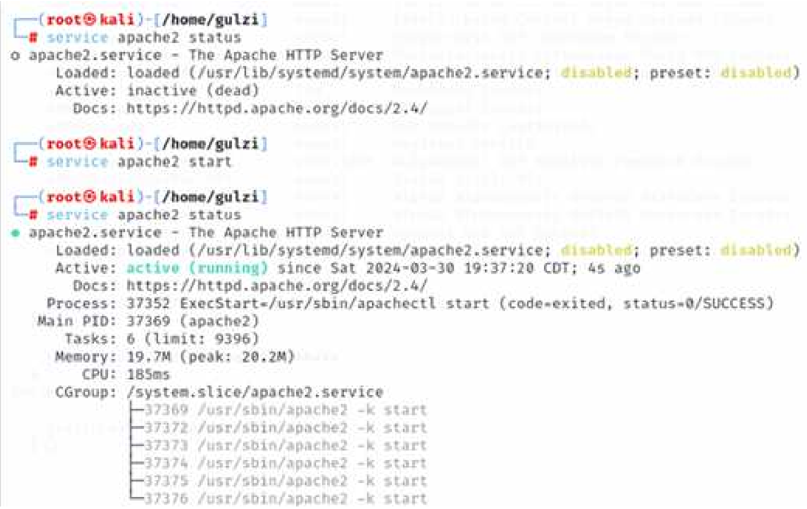
The results of exploits that can be applied in the tested are shown in Fig. 2.



```
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.137.130  yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target



View the full module info with the info, or info -d command.

msf6 exploit(multi/handler) > █
```

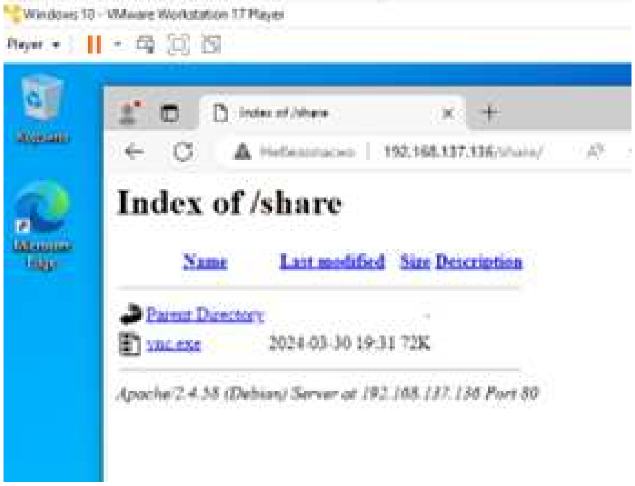
Figure 2: Results of the vulnerability definition

On Windows 7, a hack was detected using the Servis apache2 (Fig. 3-4)



```
┌──(root㉿kali)-[/home/gulzi]
└─# service apache2 status
○ apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled; preset: disabled)
     Active: inactive (dead)
       Docs: https://httpd.apache.org/docs/2.4/

┌──(root㉿kali)-[/home/gulzi]
└─# service apache2 start

┌──(root㉿kali)-[/home/gulzi]
└─# service apache2 status
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled; preset: disabled)
     Active: active (running) since Sat 2024-03-30 19:37:20 CDT; 4s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 37352 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 37369 (apache2)
      Tasks: 6 (limit: 9396)
     Memory: 19.7M (peak: 20.2M)
        CPU: 185ms
     CGroup: /system.slice/apache2.service
             ├─37369 /usr/sbin/apache2 -k start
             ├─37372 /usr/sbin/apache2 -k start
             ├─37373 /usr/sbin/apache2 -k start
             ├─37374 /usr/sbin/apache2 -k start
             ├─37375 /usr/sbin/apache2 -k start
             └─37376 /usr/sbin/apache2 -k start
```

Figure 3: Loading the Servis apache2



Index of /share

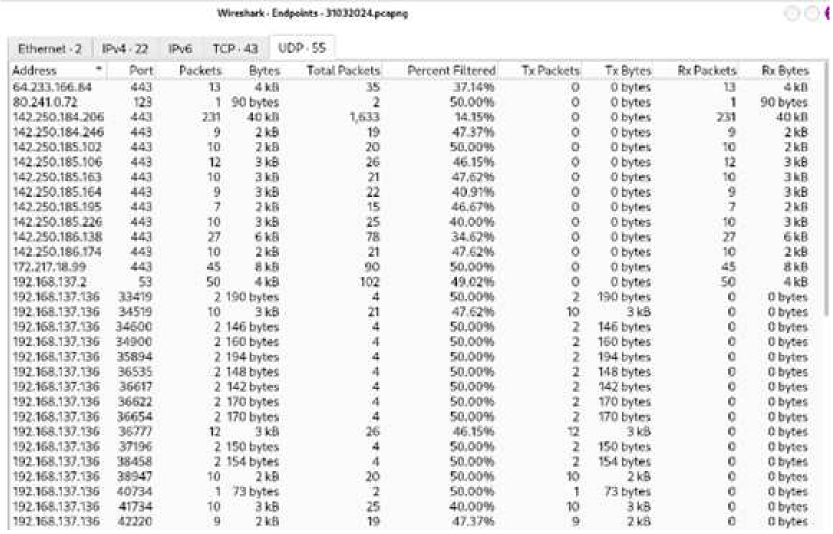
Name Last modified Size Description

Parent Directory -
vnc.exe 2024-03-30 19:31 72K

Apache/2.4.58 (Debian) Server at 192.168.137.136 Port 80

Figure 4: Open apache2 in the Windows 10

In our research paper, we used the Hurst definition of R/S statistics to calculate network traffic per second. The traffic load results are shown in Fig. 5. The simulation duration is 330 seconds, and the number of packets is 3093. This value can be changed to study the nature of the traffic.



Address	Port	Packets	Bytes	Total Packets	Percent Filtered	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
64.233.166.84	443	13	4 kB	35	37.14%	0	0 bytes	13	4 kB
80.241.0.72	123	1	90 bytes	2	50.00%	0	0 bytes	1	90 bytes
142.250.184.206	443	231	40 kB	1,633	14.15%	0	0 bytes	231	40 kB
142.250.184.246	443	9	2 kB	19	47.37%	0	0 bytes	9	2 kB
142.250.185.102	443	10	2 kB	20	50.00%	0	0 bytes	10	2 kB
142.250.185.106	443	12	3 kB	26	46.15%	0	0 bytes	12	3 kB
142.250.185.163	443	10	3 kB	21	47.62%	0	0 bytes	10	3 kB
142.250.185.164	443	9	3 kB	22	40.91%	0	0 bytes	9	3 kB
142.250.185.195	443	7	2 kB	15	46.67%	0	0 bytes	7	2 kB
142.250.185.226	443	10	3 kB	25	40.00%	0	0 bytes	10	3 kB
142.250.186.138	443	27	6 kB	78	34.62%	0	0 bytes	27	6 kB
142.250.186.174	443	10	2 kB	21	47.62%	0	0 bytes	10	2 kB
172.217.18.99	443	45	8 kB	90	50.00%	0	0 bytes	45	8 kB
192.168.137.2	53	50	4 kB	102	49.02%	0	0 bytes	50	4 kB
192.168.137.136	33419	2	190 bytes	4	50.00%	2	190 bytes	0	0 bytes
192.168.137.136	34519	10	3 kB	21	47.62%	10	3 kB	0	0 bytes
192.168.137.136	34600	2	146 bytes	4	50.00%	2	146 bytes	0	0 bytes
192.168.137.136	34900	2	160 bytes	4	50.00%	2	160 bytes	0	0 bytes
192.168.137.136	35894	2	194 bytes	4	50.00%	2	194 bytes	0	0 bytes
192.168.137.136	36535	2	148 bytes	4	50.00%	2	148 bytes	0	0 bytes
192.168.137.136	36617	2	142 bytes	4	50.00%	2	142 bytes	0	0 bytes
192.168.137.136	36622	2	170 bytes	4	50.00%	2	170 bytes	0	0 bytes
192.168.137.136	36654	2	170 bytes	4	50.00%	2	170 bytes	0	0 bytes
192.168.137.136	36777	12	3 kB	26	46.15%	12	3 kB	0	0 bytes
192.168.137.136	37196	2	150 bytes	4	50.00%	2	150 bytes	0	0 bytes
192.168.137.136	38458	2	154 bytes	4	50.00%	2	154 bytes	0	0 bytes
192.168.137.136	38947	10	2 kB	20	50.00%	10	2 kB	0	0 bytes
192.168.137.136	40734	1	73 bytes	2	50.00%	1	73 bytes	0	0 bytes
192.168.137.136	41734	10	3 kB	25	40.00%	10	3 kB	0	0 bytes
192.168.137.136	42220	9	2 kB	19	47.37%	9	2 kB	0	0 bytes

Figure 5: Traffic under study for calculation

The following packets were received according to the analysis results: Ethernet – 2, TCP – 43, IPv4 – 22, and UDP – 55. The calculations of the results obtained are shown in Table 3.

Traffic duration $m = 330$ seconds, $N = 330/(3093/330) \approx 35$

Where, N is the number of blocks, and 3093 is the number of packets.

$i = 1$, time $=$ 150 s – 31, packet $=$ 345

$P1 = 345/(150 * 31) = 0.074$

$i = 2$, time $=$ 210 s – 46, packet $=$ 417

$P2 = 417/(210 * 6) = 0.043$
$i = 3$, time = 270 s - 113, packet = 2961
$P3 = 2961/(270 * 113) = 0.097$
$i = 4$, time = 330 s - 141, packet = 3093
$P4 = 3093/(330 * 14) = 0.066$

Table 3: Summary table of Network Analyzer overview.

Xi	1	2	3	4
Pi	0.074	0.043	0.097	0.066

The arithmetic mean of network traffic is determined by the following formula:

$$M = \frac{1}{N} \sum_{i=1}^{N} X_i \qquad M = 0.07 \tag{1}$$

Calculating the variance:

$$S^2 = \frac{1}{N} \sum_{i=1}^{N} (X_i - M)^2 \qquad S = 0.019 \tag{2}$$

Determination of the value of the oscillations relative to the mean M :

$$\begin{aligned} &D_j = \textstyle\sum_k^j X_k - j \cdot M \\ &D1 = X1 - N \cdot M = 0.004 \\ &D2 = X1 + X2 - N \cdot M = -0.023 \\ &D3 = X1 + X2 + X3 - N \cdot M = 0.004 \\ &D4 = X1 + X2 + X3 + X4 - N \cdot M = 0 \end{aligned} \tag{3}$$

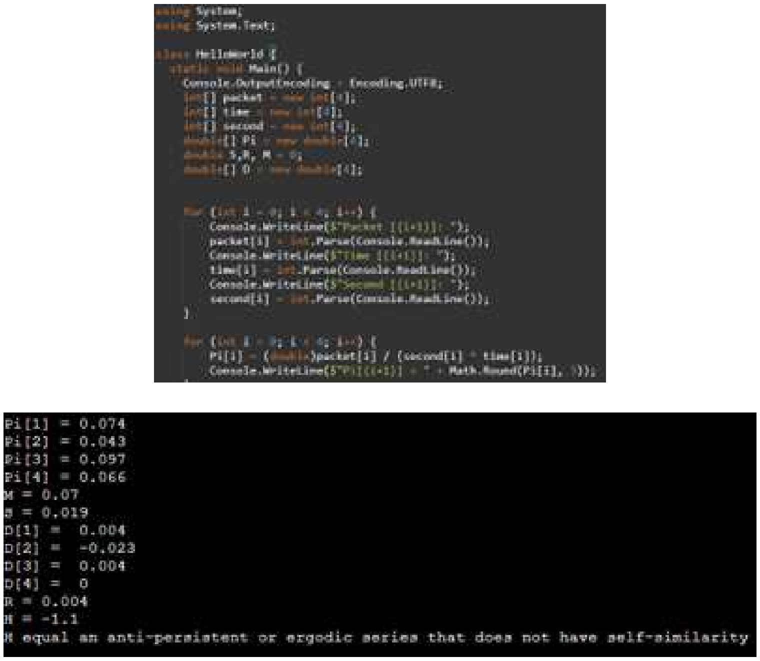
Maximum and minimum D value range:

$$R = max\left\{D_j\right\} - min\left\{D_j\right\} \qquad R = 0.004 \tag{4}$$

The Hurst coefficient is determined by the following formula:

$$H = \frac{ln\left(\frac{R}{S}\right)}{lnN} \qquad H = -1.1 \tag{5}$$

This traffic calculation can be performed using the C# compiler (Fig. 6).



```
using System;
using System.Text;

class HelloWorld {
  static void Main() {
    Console.OutputEncoding = Encoding.UTF8;
    int[] packet = new int[4];
    int[] time = new int[4];
    int[] second = new int[4];
    double[] Pi = new double[4];
    double S,R, M = 0;
    double[] D = new double[4];

    for (int i = 0; i < 4; i++) {
      Console.WriteLine($"Packet [{i+1}]: ");
      packet[i] = int.Parse(Console.ReadLine());
      Console.WriteLine($"Time [{i+1}]: ");
      time[i] = int.Parse(Console.ReadLine());
      Console.WriteLine($"Second [{i+1}]: ");
      second[i] = int.Parse(Console.ReadLine());
    }

    for (int i = 0; i < 4; i++) {
      Pi[i] = (double)packet[i] / (second[i] * time[i]);
      Console.WriteLine($"Pi[{i+1}] = " + Math.Round(Pi[i], 3));
```

```
Pi[1] = 0.074
Pi[2] = 0.043
Pi[3] = 0.097
Pi[4] = 0.066
M = 0.07
S = 0.019
D[1] =  0.004
D[2] =  -0.023
D[3] =  0.004
D[4] =  0
R = 0.004
H = -1.1
H equal an anti-persistent or ergodic series that does not have self-similarity
```

Figure 6: Results C#

Based on the results of the Hurst indicator (H), the following processes are determined:

- $H \leq 0.5$ – an anti-persistent or ergodic series that does not have self-similarity;
- $H = 0.5$ – complete random series with particle displacement in classical Brownian motion;
- $H \geq 0.5$ – a persistent (self-sustaining) process that has a long memory and is self-similar [12].

According to the results, we have $H \leq 0.5$, and this process is anti-persistent and does not have self-similarity.

In the following studies, the vulnerability analysis of the modules of the biometric voice identification system is conducted, and a block diagram of the system of voice identification of the user by voice with enhanced protection against attacks is proposed. This scheme for the use of elementary speech units in the developed identification systems allows improving computational performance, reducing subjective decisions in biometric systems, and increasing security against attacks on voice biometric identification systems with a probability of the first and second errors of the kind of 0.025 and 0.005 [15].

Using the MS Excel and C# compiler, the calculation was carried out by 77.11% of students (Fig. 7).

Computer Emergency Response Team (KZ-CERT) is a center that collects and analyzes information on computer incidents and provides advisory and technical support to users in preventing computer security threats. Together with Nitro Team LLP, more than 170 potentially vulnerable Microsoft Exchange IP addresses were discovered in the republican segment. Attackers can gain access to any Microsoft Exchange Server email account using these vulnerabilities. KZ-CERT experts sent instructions on identifying vulnerable software to all government agencies and operational information security centers [8].

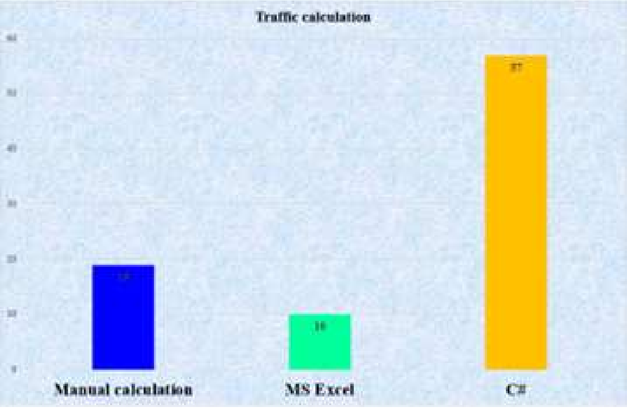


Figure 7: Methods of traffic calculation

Emrah Yasasin et al. [16] examine the vulnerability of software packages and the impact of exploits. Based on the National Vulnerability Database (NVD), the authors used mean absolute error (MAE) and root-mean-square error (RMSE) to measure prediction accuracy using single, double, and triple exponential smoothing, Croston's methodology, ARIMA, and the neural network approach. The results have shown that the optimal forecasting methodology depends on the software, and forecasting accuracy is reliable within the two applied forecasting error metrics.

Darshana Upadhyay et al. [14] investigate the vulnerability of the Supervisory Control and Data Acquisition (SCADA) network. In the scientific work, real incidents of SCADA vulnerabilities registered in standard databases are considered, and recommendations for SCADA security are given.

4 Conclusion

The goal of this research was to identify effective utilities for analyzing network traffic and detecting network vulnerabilities. Based on the analysis, the following threats were identified: broadcast scanning; interception of network traffic; modification and implementation of network traffic; getting information about the device; changing the ARP-spoofing table; implementation of a false DHCP server.

Identification of vulnerabilities in Windows 10 is done using the Kali Purple distribution. The following categories of attacks were identified: violation of the network perimeter; violation of integrity; violation of confidentiality; accessibility violation; link layer attack; Ethernet network layer attacks.

The analysis revealed the lack of full protection against harmful network activity. The developed methodology for detecting vulnerabilities and cracking the OS is based on the IDEF0 and IDEF1X methodology.

The proposed method of analyzing network traffic and detecting vulnerabilities allows you to identify and block attacks on the local Ethernet network with higher accuracy and completeness. The obtained results of the Hurst exponent ($H \leq 0.5$) determined an ergodic series that does not have self-similarity.

The vulnerability database grows every year. Organizations are undergoing changes that are associated with a security risk. Information security management automates the inventory of resources and the identification of vulnerabilities using modern security tools. Each vulnerability must be verified. Thus, the proposed methods of analyzing and scanning the vulnerability of network resources are the first step towards security. This is a cyclical process, and the regularity of the process allows minimizing the risk of attacks on private, corporate infrastructure.

Further research will continue to study new types of attacks in local networks and improve the architecture of the information security system.

References

[1] Clarke N., Li F., Furnell S., "A novel privacy preserving user identification approach for network traffic" , *Computers & Security,* 70 (2017): 335–350.

[2] Gorodnichev M.G., et al, "Machine learning in the tasks of identifying unwanted content" , *In: Wave electronics and its application in information and telecommunication systems (WECONF),* Saint-Petersburg (2019).

[3] Gubareva O.Yu., Bourdine A.V., Evtushenko A.S., et al, *Secure data transmission channel protected by special fiber optic link based on optical crypto-fibers.* (2018). DOI:10.1117/12.2318579.

[4] Jisa D., Ciza T., "Efficient DDoS flood attack detection using dynamic thresholding on flow-based network traffic" , *Computers & Security,* 82 (2019): 284–295.

[5] Kali Linux, Penetration Testing and Ethical Hacking Linux Distribution. [Electronic resource]. URL: https://www.kali.org/ (Date: 25.02.2024).

[6] Kawasaki Y., Hara Y., Kuwahara M., "Traffic state estimation on a two-dimensional network by a state-space model" , *Transportation Research Part C: Emerging Technologies,* 113 (2020): 176–192.

[7] Klenilmar L., Dias M.A., Pongelupe W.M., "An innovative approach for real-time network traffic classification" , *Computer Networks,* 158 (2019): 143–157.

[8] KZ-CERT (2021) 170 IP addresses of potentially vulnerable Microsoft Exchange mail servers found in Kaznet. https://cert.gov.kz/news/11/1441.

[9] Makarenko S.I., Smirnov G.E., "Analysis of penetration testing standards and methodologies" , *Systems of Control, Communication and Security,* 4 (2020): 44–72 (in Russian). DOI: 10.24411/2410-9916-2020-10402.

[10] Markin Yu.V., "Methods and means of in-depth analysis of network traffic" , *Dissertation, V.P. Ivannikov Institute of System Programming of the Russian Academy of Sciences,* (2017).

[11] Pacheco F., Exposito E., Gineste M., "A framework to classify heterogeneous Internet traffic with machine learning and deep learning techniques for satellite communications" , *Computer Networks,* 173 (2020): 107213.

[12] Paramonov A.I., "Development and research of a complex of traffic models for public communication networks" , *Dissertation, St. Petersburg State University of Telecommunications named after Professor M. A. Bonch-Bruevichm,* (2014).

[13] Ring M., Schlör D., Landes D., et al, "Flow-based network traffic generation using. Generative Adversarial Networks" , *Computers & Security,* 82 (2019): 156–172

[14] Upadhyay D., Sampalli S., "SCADA (Supervisory Control and Data Acquisition) systems: Vulnerability assessment and security recommendations" , *Computers & Security,* 89 (2020): 101666.

[15] Vanyushina A.V., "Classification of IP traffic in a computer network using machine learning algorithms" , *Dissertation, Moscow Technical University of Communications and Informatics,* (2019).

[16] Yasasin E., Prester J., Wagner G., et al, "Forecasting IT security vulnerabilities – An empirical analysis" , *Computers & Security,* 88 (2020): 101610.