



## RESEARCH ARTICLE

# Detection and Neutralization Method of Network Worms Malicious Software Impact on the User's Computer System

Larissa Vladimirovna Cherckesova<sup>1</sup>, Safaryan O.A.<sup>2</sup>, Buryakova O.S.<sup>3</sup>, Akishin B.A.<sup>4</sup><sup>1,2,3,4</sup>Don State Technical University**ARTICLE INFO****ABSTRACT**

Received: Mar 17, 2025

Accepted: May 1, 2025

**Keywords**

Network worm  
Computer virus  
Worm peer-to-peer (P2P)  
Computer Espionage  
Server  
Software  
Cyber attack  
Spywares  
Key loggers  
Backdoors  
Trojans

Destruction, modification, accidental or intentional alteration or even fraudulent redirection of information has existed for long time. However, the automated information processing provided to specialists by knowledge of programming methods has given rise to the computer crimes of new type. Some of these crimes involve alteration or destruction of information, or disruption of the computer system. These features are the part of main characteristics of computer attacks by the new type – network worms. This malicious software reproduces itself on several computers using computer network. It has the ability to duplicate itself after execution. Unlike other viruses, network worm spreads without the need to be tied to other executable programs. This article proposes the way to detect and neutralize the threat of attacks by network worms on user's computer system.

**\*Corresponding Author:**

safari\_2006@mail.ru

**INTRODUCTION**

Expanding connectivity options and the development of digital infrastructures have increased the incentives and opportunities of intruders for cyberattacks. Although the significant progress has been made in developing and implementing the defense strategies, much of this effort has focused on developing of the new solutions rather than evaluating and promoting them. It is therefore important for governments, companies, and individuals to identify the models and means of collaboration to identify and evaluate strategies to reduce the risk associated with cyber threats such as network worms [1].

For this purpose, the field of information systems security could benefit from lessons learned and methods used in the field of public health. In particular, we believe that adopting public health perspective would provide comprehensive framework for identifying factors affecting information systems security and understanding their root causes, developing and evaluating effective strategies to improve information systems security, and implementing and disseminating the developed strategies to the public [2].

The following objectives are set within the framework of this article: research of the network worm viruses operating principles; classification of its varieties; protection against its attacks; development of software in the Kotlin programming language for detection and neutralization of P2P network worms in the Windows OS.

**1. Network Worms and their Action Mechanism**

Network worms are the programs replicating themselves in the different places on computer. It is subclass of viruses, with which worms sharing the common characteristics. The aim of this type

of malware is to saturate computers and networks to prevent their operating. This malware differs from other viruses on the several directions.

Network worm is the stand-alone program that can reside on the hard disk, unlike other viruses, which, like parasites, hide in files or in executable code contained in boot sector of hard disk [3]. Some rare worms do not write to the disk and are stored in memory.

Network worm can enter through the network directly by using any open port, but classic way is to present itself as *attachment* attached to E-mail. Some worms are launched directly by reading an E-mail (especially if computer system has not been updated). However, in the most cases, user needs to click on attachment for the worm to launch [4].

Network worm does not replicate locally, unlike other viruses, but its most common method of propagation is by sending automatically generated E-mails, which are sent without the user's knowledge to various addresses. These addresses are taken by the worm from the disk files (in particular, from address book), or are addresses constructed semi-randomly.

Network worms install usually on the attacked computer next malicious programs: spyware, key loggers, backdoors, and Trojans. These programs can be used to surveillance on the user's activity, intercept passwords or credit card numbers, or remotely controlling the computer to turn it into zombie-computer, which can be used as repeater for *Denial-of-service* (DDoS) carrying out attacks or for sending the mass spam. General public uses the term *virus* mistakenly to refer to this type of malicious programs [5].

Network worms written as scripts can be embedded in E-mail or HTML page (Trojans). These worms are activated by user's actions who believe that he is gaining access to information intended for him. The worm can be programmed in C, C++, Delphi, Assembler, Python or another programming language also.

In most cases, network worms exploit vulnerabilities in software to spread. Software vendors usually fix these flaws as soon as worms appear. By downloading the latest versions of antiviruses programs as soon as they appear, it can reduce significantly the likelihood of worms being infected. Data damaged or destroyed by the worm is unrecoverable usually.

Worm often gets onto computers through various routes, such as E-mail, unknown source programs, forum sites, pirated DVDs and CDs with games, USB drives, etc.

A worm is designed to automatically copy itself from one computer to another. It takes control of the properties that transfer files or information to the computer. Once worm gets into computer system, it may disappear on its own. It can lead to intensive network traffic due to domino effect that slows down networks and entire Internet operation.

New network worms spread very quickly when they appear. It spreads usually without user intervention and spreads complete (possibly modified) copies of itself from network to network. Such worm can consume the memory or network bandwidth, which can cause the computer crash [5]. Since the worms do not need the host program or file to spread, it can infiltrate your system and allow another person to control your computer remotely. Examples of recently appeared worms include the *Sasser* and *Blaster* worms.

## 2. Classification of network worms

Network worms *Peer-to-peer* (P2P) – is peer-to-peer network. It is the form of local area network, which connects computers in the serverless network, i.e. it establishes the direct connection between individual users (Fig. 1). Most file sharing networks on the Internet, such as Kazaa, Morpheus or BitTorrent systems, use the P2P technology.



**Figure 1 – Peer-to-peer network (P2P)**

### **Network Worm can Spread in the File-Sharing Network in Three Ways:**

First method is for network worm to copy itself to the shared folder from which other users can download files. For this worm type, it is important to name correctly the file containing it, because more users download the files with interesting name than files with randomly generated names. This type of distribution in file-sharing networks is simple, but not very effective, since file-sharing services, exchange usually quite large files and almost all file-sharing services have now effective filters to exclude suspicious files of certain formats [6].

*Second method* of spreading the network worm uses the P2P protocol to offer the infected file as the search result (e.g. torrent file) to other users in the P2P network every time they perform the search. Then user copies the network worm to computer and infects the own computer system when they open the file. This method of spreading is more effective if the worm file size is approximately equal to the file being searched for.

*Third method* is for the network worm to attack the vulnerability of its neighbors in the P2P network. This method can provide very high rate of propagation, when no user action is required (such as downloading the file and running it on the computer). Then the network worm infects these systems completely automatically.

Once the network worm can see the neighbor list of each infected client in the P2P network, it can target it that allows the network worm to avoid the detection because it does not have to make excessive number of connections to other systems on the Internet, which is considered abnormal behavior by security systems. The P2P network is based on the fact that each user establishes multiple connections to other participants, which make the network worm much more difficult to detect, based on the traffic it generates [6].

### **3. Instant Messaging Worms**

Instant messaging programs such as What Sapp, Telegram, Windows Live Messenger, ICQ, or Skype are also vulnerable to malware due to their connection to the Internet. This type of network worm spreads by sending the user the link to Web page containing the worm. If the user clicks on the link, the worm is installed then and executed on the user's computer, as instant messaging applications typically do not have their own HTML parser (analyzer) and instead use the user's default browser. Once installed, the network worm spreads by sending the links to all contacts registered on the infected computer.

#### **3.1. Removable multimedia worms**

These network worms automatically copy themselves onto removable computer storage media, such as USB sticks, to spread from one computer to another. Unlike the other types of network worms discussed before, this worm does not use the network to spread. It can take advantage of the automatic launch of the storage media [7].

#### **3.2. IRC (Internet Relay Chat) network worms**

IRC clients are the programs that allow any user to exchange text messages with other users in near real time via Internet Relay Chat. Most IRC programs use the special script to connect to IRC server (see Fig. 2), which is executed when the program is started. This scenario (script) contains the commands that the IRC program executes. These commands include the joining to the channel, writing the messages, and sending the files.

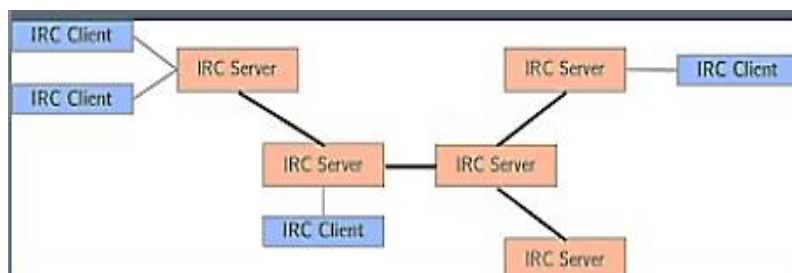


Figure 2 – IRC Server

IRC network worm that has infected computer searches for IRC programs that it can use to spread: when it finds one, it modifies script that is downloaded automatically.

The next time starting the program, the network worm automatically sends the scenario (script) to all users in the chat. If the user accepts the download and opens the downloaded file, the entire process is repeated. There are IRC network worms for at least five IRC programs (*mIRC*, *pIRCh*, *vIRC*, *dIRC*, and *Xircon*) take place currently [7].

### Example of IRC worm detection:

Fig. 3 shows network traffic dump infected with an IRC network worm. The host 192.168.45.130 made DNS request to 192.168.45.2 to resolve *irc.accesox.net*. It is assumed that the host at 192.168.45.130 may have been infected with IRC Trojan, which initiated TCP connection to *irc.accesox.net*, entered pre-defined IRC channel, and based on commands issued by the master bot, spread further infection by infecting the *autorun.inf* of connected USB devices.

No.	Time	Source	Destination	Protocol	Length	Info
1	8.889888	192.168.45.130	192.168.45.2	DNS	75	Standard query 0x4083 A irc.accesox.net
2	8.887976	192.168.45.2	192.168.45.130	DNS	107	Standard query response 0x4083 A irc.accesox.net A 91.121.130.60 A 91.121.56.162
3	8.893589	192.168.45.130	91.121.130.60	TCP	62	1038 → 5540 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM
4	1.598449	91.121.130.60	192.168.45.130	TCP	58	5540 → 1038 [SYN, ACK] Seq=9 Ack=1 Win=64240 Len=0 MSS=1460
5	1.597441	192.168.45.130	91.121.130.60	TCP	54	1038 → 5540 [ACK] Seq=1 Ack=1 Win=64240 Len=0
6	1.599000	192.168.45.130	91.121.130.60	TCP	69	1038 → 5540 [PSH, ACK] Seq=1 Ack=1 Win=64240 Len=14
7	1.569647	91.121.130.60	192.168.45.130	TCP	54	5540 → 1038 [ACK] Seq=1 Ack=15 Win=64240 Len=0
8	1.570812	192.168.45.130	91.121.130.60	TCP	77	1038 → 5540 [PSH, ACK] Seq=15 Ack=1 Win=64240 Len=23
9	1.570835	91.121.130.60	192.168.45.130	TCP	54	5540 → 1038 [ACK] Seq=1 Ack=38 Win=64240 Len=0
10	1.571154	192.168.45.130	91.121.130.60	TCP	82	1038 → 5540 [PSH, ACK] Seq=38 Ack=1 Win=64240 Len=38
11	1.571172	91.121.130.60	192.168.45.130	TCP	54	5540 → 1038 [ACK] Seq=1 Ack=78 Win=64240 Len=0
12	2.389970	91.121.130.60	192.168.45.130	TCP	249	5540 → 1038 [PSH, ACK] Seq=1 Ack=76 Win=64240 Len=195
13	2.400153	91.121.130.60	192.168.45.130	TCP	248	TCP RST (RST) Seq=1038 → 5540 [FIN, ACK] Seq=1 Ack=76 Win=64240 Len=195
14	2.400455	192.168.45.130	91.121.130.60	TCP	54	1038 → 5540 [ACK] Seq=76 Ack=196 Win=64045 Len=0
15	3.681576	91.121.130.60	192.168.45.130	TCP	107	5540 → 1038 [PSH, ACK] Seq=196 Ack=76 Win=64240 Len=113
16	3.692071	192.168.45.130	91.121.130.60	TCP	69	1038 → 5540 [PSH, ACK] Seq=76 Ack=309 Win=65932 Len=15
17	3.692114	91.121.130.60	192.168.45.130	TCP	54	5540 → 1038 [ACK] Seq=309 Ack=91 Win=64240 Len=0
18	3.816144	91.121.130.60	192.168.45.130	TCP	118	5540 → 1038 [PSH, ACK] Seq=309 Ack=91 Win=64240 Len=102
19	3.874526	192.168.45.130	91.121.130.60	TCP	81	1038 → 5540 [PSH, ACK] Seq=91 Ack=1371 Win=62870 Len=27
20	3.874582	91.121.130.60	192.168.45.130	TCP	54	5540 → 1038 [ACK] Seq=1371 Ack=118 Win=64240 Len=0
21	3.874703	192.168.45.130	91.121.130.60	TCP	71	1038 → 5540 [PSH, ACK] Seq=118 Ack=1371 Win=62870 Len=17
22	3.874802	91.121.130.60	192.168.45.130	TCP	54	5540 → 1038 [ACK] Seq=1371 Ack=135 Win=64240 Len=0

Frame 1: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface 0 Ethernet II, Src: VMware_84:fe:f9 (08:0c:29:84:fe:f9), Dst: VMware_eb:eb:eb:eb:eb:eb (08:00:50:56:eb:eb:eb) Internet Protocol Version 4, Src: 192.168.45.130, Dst: 192.168.45.2 User Datagram Protocol, Src Port: 1037, Dst Port: 53 Domain Name System (query) Transaction ID: 0x4083 Flags: 0x0100 Standard query Questions: 1 Answer RRs: 0 Authority RRs: 0 Additional RRs: 0 Queries * irc.accesox.net: type A, class IN Name: irc.accesox.net (Name Length: 15) (Label Count: 3) Type: A (Host Address) (1) Class: IN (0x0001) [Response: 3]	<pre> 0000  00 50 56 eb eb 00 00 0c 29 84 fe f9 08 00 50 56 eb eb eb 0010  00 3d 00 20 00 00 00 11 5e b0 c8 eb 2d 02 c8 00 00 0020  2d 02 c8 00 00 00 00 2d 07 01 46 03 01 00 00 01 0030  00 00 00 00 00 00 00 00 72 63 97 65 83 63 65 71 0040  ff 78 83 8e 05 74 00 00 01 00 01       </pre>
--	---

Figure 3 – Network traffic dump

The process of detection by IRC Trojan on the host at 192.168.45.130 was described in detail. It was demonstrated on the Fig. 4 – Fig.11:

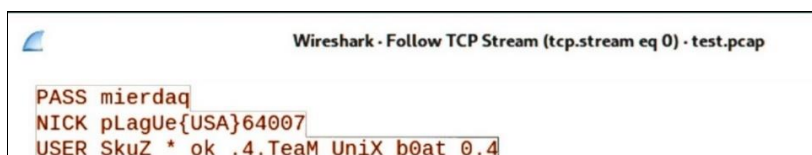
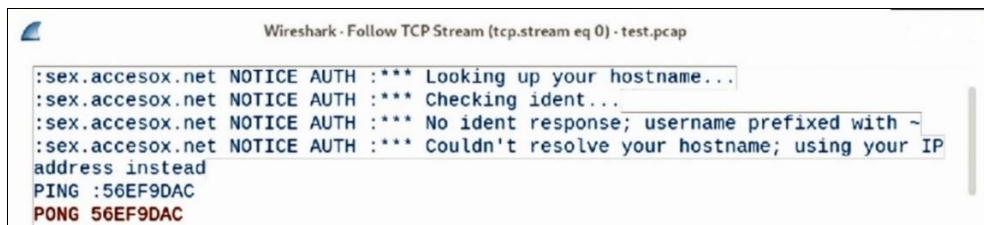


Figure 4 – Password and nickname given by host bot for connection, set by the host bot

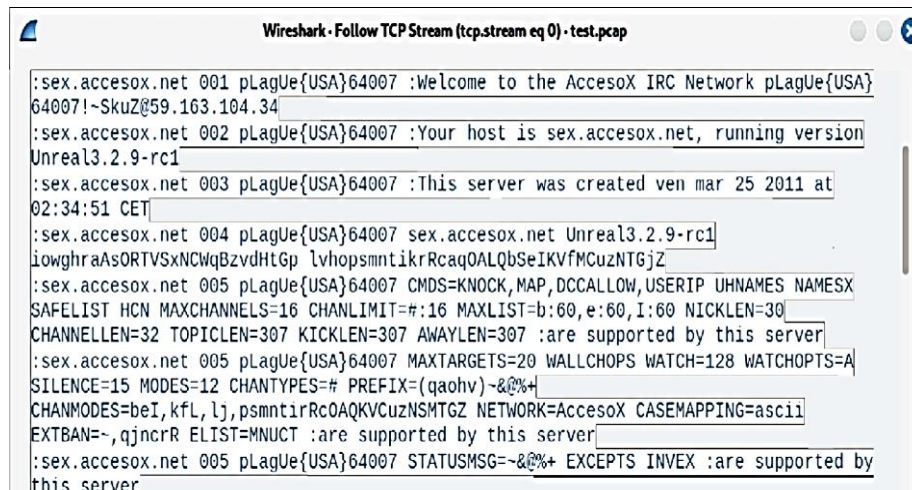


```

:sex.accesox.net NOTICE AUTH :*** Looking up your hostname...
:sex.accesox.net NOTICE AUTH :*** Checking ident...
:sex.accesox.net NOTICE AUTH :*** No ident response; username prefixed with ~
:sex.accesox.net NOTICE AUTH :*** Couldn't resolve your hostname; using your IP
address instead
PING :56EF9DAC
PONG 56EF9DAC

```

**Figure 5 – Notification from IRC server sex.accesox.net and 5-PING command from IRC server and response PONG from bot**



```

:sex.accesox.net 001 pLagUe{USA}64007 :Welcome to the Accesox IRC Network pLagUe{USA}
64007!~SkuZ@59.163.104.34
:sex.accesox.net 002 pLagUe{USA}64007 :Your host is sex.accesox.net, running version
Unreal3.2.9-rc1
:sex.accesox.net 003 pLagUe{USA}64007 :This server was created ven mar 25 2011 at
02:34:51 CET
:sex.accesox.net 004 pLagUe{USA}64007 sex.accesox.net Unreal3.2.9-rc1
iowghraAsORTVSxNCWqBzvdHtGp lvhopsmtikrRcaqOALQbSeIKVfMCuzNT6jZ
:sex.accesox.net 005 pLagUe{USA}64007 CMDS=KNOCK,MAP,DCCALLOW,USERIP UHNAMES NAMESX
SAFELIST HCN MAXCHANNELS=16 CHANLIMIT=#:16 MAXLIST=b:60,e:60,I:60 NICKLEN=30
CHANNELLEN=32 TOPICLEN=307 KICKLEN=307 AWAYLEN=307 :are supported by this server
:sex.accesox.net 005 pLagUe{USA}64007 MAXTARGETS=20 WALLCHOPS WATCH=128 WATCHOPTS=A
SILENCE=15 MODES=12 CHANTYPES=# PREFIX=(qaohv)~&@%+
CHANMODES=beI,kfL,lj,psmtirRcOaQKVCuzNSMTGZ NETWORK=Accesox CASEMAPPING=asciI
EXTBAN=~,qjncrR ELIST=MNUCT :are supported by this server
:sex.accesox.net 005 pLagUe{USA}64007 STATUSMSG=~&@%+ EXCEPTS INVEX :are supported by
this server

```

**Figure 6 – IRC information about network**

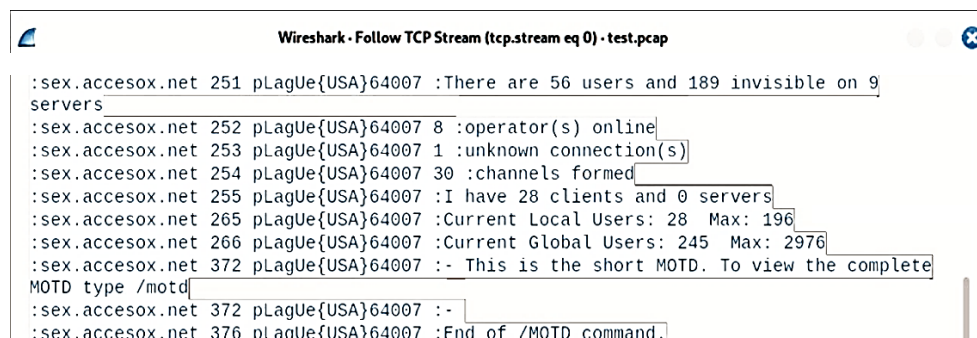


```

MODE pLagUe{USA}64007 -ix
JOIN ##verga##
JOIN ##verga##
PRIVMSG ##verga## :.4.NueVo PuTo InfeCcIoN.

```

**Figure 7 – Client bot with request to listen to the specific channel – “verga”**

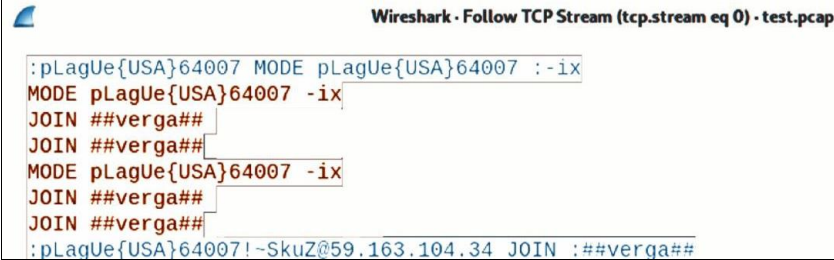


```

:sex.accesox.net 251 pLagUe{USA}64007 :There are 56 users and 189 invisible on 9
servers
:sex.accesox.net 252 pLagUe{USA}64007 8 :operator(s) online
:sex.accesox.net 253 pLagUe{USA}64007 1 :unknown connection(s)
:sex.accesox.net 254 pLagUe{USA}64007 30 :channels formed
:sex.accesox.net 255 pLagUe{USA}64007 :I have 28 clients and 0 servers
:sex.accesox.net 265 pLagUe{USA}64007 :Current Local Users: 28 Max: 196
:sex.accesox.net 266 pLagUe{USA}64007 :Current Global Users: 245 Max: 2976
:sex.accesox.net 372 pLagUe{USA}64007 :- This is the short MOTD. To view the complete
MOTD type /motd
:sex.accesox.net 372 pLagUe{USA}64007 :-
:sex.accesox.net 376 pLagUe{USA}64007 :End of /MOTD command

```

**Figure 8 – Additional message indicating the number of active channels, local users, online operators, etc. on the IRC server**



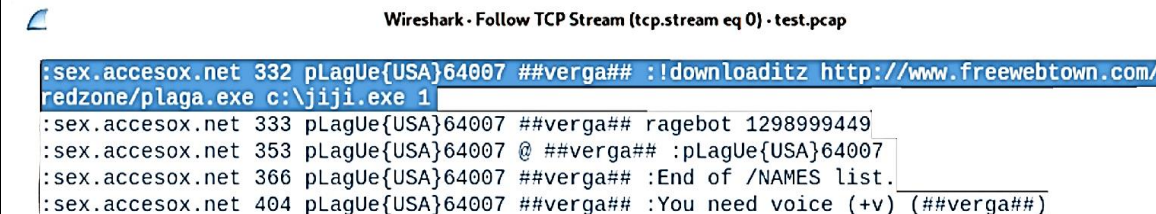
```

Wireshark - Follow TCP Stream (tcp.stream eq 0) - test.pcap

:pLagUe{USA}64007 MODE pLagUe{USA}64007 :-ix
MODE pLagUe{USA}64007 -ix
JOIN ##varga##
JOIN ##varga##
MODE pLagUe{USA}64007 -ix
JOIN ##varga##
JOIN ##varga##
:pLagUe{USA}64007!~Skuz@59.163.104.34 JOIN :##varga##

```

Figure 9 – Request message indicating on attempt to re-register client bot to listen the “Verga” channel



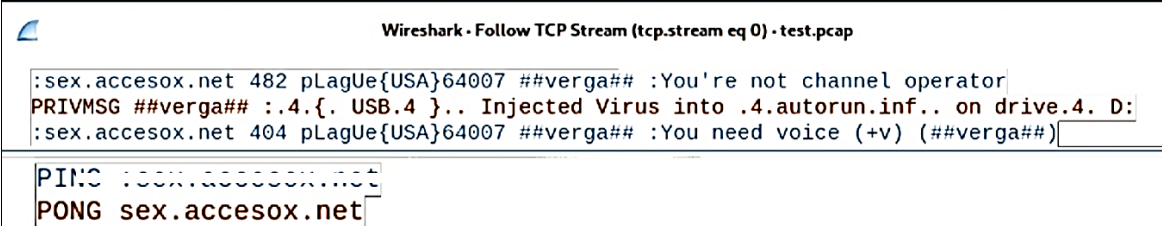
```

Wireshark - Follow TCP Stream (tcp.stream eq 0) - test.pcap

:sex.accesox.net 332 pLagUe{USA}64007 ##varga## :!downloaditz http://www.freewebtown.com/redzone/plaga.exe c:\jiji.exe 1
:sex.accesox.net 333 pLagUe{USA}64007 ##varga## ragebot 1298999449
:sex.accesox.net 353 pLagUe{USA}64007 @ ##varga## :pLagUe{USA}64007
:sex.accesox.net 366 pLagUe{USA}64007 ##varga## :End of /NAMES list.
:sex.accesox.net 404 pLagUe{USA}64007 ##varga## :You need voice (+v) (##varga##)

```

Figure 10 – Message from botnet – pLagUe download exe (plaga.exe) from <http://www.freewebtown.com/redzone/>



```

Wireshark - Follow TCP Stream (tcp.stream eq 0) - test.pcap

:sex.accesox.net 482 pLagUe{USA}64007 ##varga## :You're not channel operator
PRIVMSG ##varga## :.4.{ . USB.4 }.. Injected Virus into .4.autorun.inf.. on drive.4. D:
:sex.accesox.net 404 pLagUe{USA}64007 ##varga## :You need voice (+v) (##varga##)

PING :sex.accesox.net
PONG sex.accesox.net

```

Figure 11 – Private message from client bot indicating on successful injection of exe into autorun.inf of four USB devices

#### 4. Protection against Network Worms

The technical means reach their limits when it comes to protecting the user against psychological influence (social engineering), for example, with legitimate-looking E-mail asking to download some infected fragment. However, it is advisable to inform the users of the risks, for example during anti-phishing campaigns. User training makes it difficult for the worm's sender to convince user to open compromised file, such as E-mail attachment.

It is recommended not to open illegitimate files from E-mail attachments or other sources, even if they come from sender known to the recipient. Indeed, the fact that the sender is known is no guarantee of security since an attacker can use it [8]:

The sender may have very similar E-mail address (within one character, sometimes even relying on the similarity between certain characters: «o/0», «l/1», «-/\_» etc.), when trying to impersonate the known sender (the spoofing attack);

The attacker can compromise E-mail address of well-known sender, for example, by stealing his identifiers, or in some another way;

The well-known sender may become the victim of the network worm himself or herself that uses their mailbox as means of distribution.

Files can be pre-scanned for common malware, section about antivirus scanners.

This files intended for specific application (such as music files with *.mp3* extension or image files with *.jpg* and *.jpeg* extensions) should not be viewed simply by using "open" option, but by using "open with" option in pop-up menu by selecting the appropriate program.

This preventive measure is intended to avoid the misleading double file extension that uses the hiding of known file extensions (enabled by default in Microsoft operating systems after Windows 95) to trick the user into thinking that the infected file *Vacation.jpeg.exe*, which is actually executable file, is simply JPEG photo, since it appears to him as simply *vacation.jpeg*. In particular, Microsoft Office documents (including files *.doc/.docx*, *.xls/.xlsx*, *.ppt/.pptx*, and *.rtf*) obtained from external sources should not be opened with MS Office installed for simple viewing. There is the risk that the macros (small program with potentially destructive content) stored in the MS Office document will be executed. It is best to use some program application that can view and print such files without offering the option to run such suspicious macros.

#### 4.1. Software protection

Purpose of virus scanning is to detect the known viruses, network worms and Trojans, and attempt to block and destroy them. Antivirus solution is most effective when malware is detected by virus scanning before the malicious file is executed for the first time on the computer system it is protecting [9]. Therefore, it is recommended to scan any new file from external source (removable storage media, flash drive, disk, Web page, E-mail attachment etc.) with updated antivirus software before it executing or reading. To exclude other infection routes, it is recommended also to scan files from the general network if it is not possible to exclude the intermediate infection via one of the other ways.

Antivirus software uses "signature comparison" search method to detect malware: it analyzes the files given to it to look for known virus signatures. In reality, it cannot detect malware that it does not (yet) know about and therefore antivirus software cannot determine with certainty that a file is virus-free. Antivirus only detects viruses that it knows about.

From this point, there is the "race" between the network worm developers, who try to hide the network worm as much as it is possible, or change the known variant of worm so that it is no longer detected, and the antivirus software developer, who tries to keep its signature database in the actual current state [10].

Because antivirus software by nature is inherently always lagging behind threats, it also contains the components that monitoring the processes used on the computer system to detect suspicious activity that could potentially reveal the presence of malware, even if it has passed the antivirus software scan successfully. Thus, the methods for detecting and concealing malicious activity of viruses and worms are the subject of the similar "race".

Once the malware starts running, it can disable antivirus software or manipulate the computer system so that the antivirus scanner no longer detects the malware [10]. In this case, it need to install and configure the firewall: check the presence and correctness of the firewall settings on the computer or network. It need to make sure that access to all *Internet Relay Chat* (IRC) servers and ports is controlled and limited. Antivirus software must be used: install and update regularly the antivirus software on all computers in the network. The antivirus program must detect and block all known IRC network worms.

*It needs to limit user rights:* operate the system with minimal privileges, use limited user accounts for everyday tasks and grant administrative rights only when necessary [11].

*Software should be updated:* regularly update the operating system, browsers, plugins, and other software. Updates often contain fixes for vulnerabilities that network worms can exploit to intrude. *It need to be careful when opening attachments and links:* do not open attachments or click on links from unfamiliar or suspicious sources. IRC network worms are spread often through the malicious attachments or links.

*Regular malware scans* are required: regularly scan computers on the network with antivirus software to detect and remove possible malware, including IRC network worms.

*Network monitoring is required:* put in place the process for monitoring network traffic and system logs to detect abnormal activity and intrusion attempts via IRC worms.

*Passwords should be updated:* change passwords for computer system access regularly, including IRC accounts, use complex and unique passwords to prevent guessing or hacking.

*Back up important data regularly.* If computer system or data is infected with the IRC network worm, it is possible to restore the information from the last working backup.

*Intrusion Detection Systems (IDS)* analyze network traffic and system logs to detect anomalous activity consistent with network worms. It can use the rules and algorithms to identify potential threats and alert the system administrator.

*Network traffic monitoring* can detect unusual activity such as bulk packet sending or port scanning attempts that may be associated with the spread of network worms [12].

## 4.2. Detecting P2P Network Worms on Windows

Network worm was developed in Python programming language with following principle: malicious program called "Network worm" gets onto "Windows" operating system, is fixed in the registry (Fig.12), in the service (Fig. 13) and in the "AppData" package (Fig. 14). Then it receives instructions in the "instruction.txt" file and performs malicious actions if there is such task and spreads further via E-mail (Fig. 15).

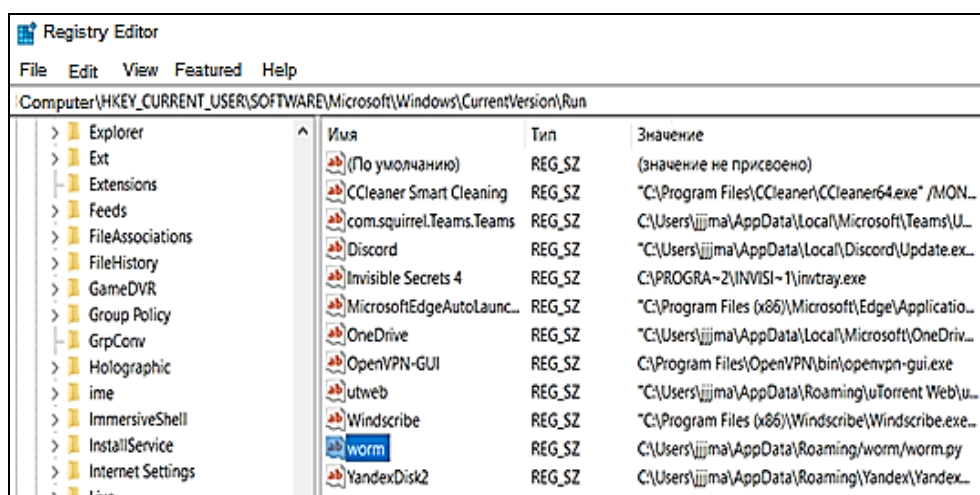


Figure 12 – Pinning "Network Worm" to the registry

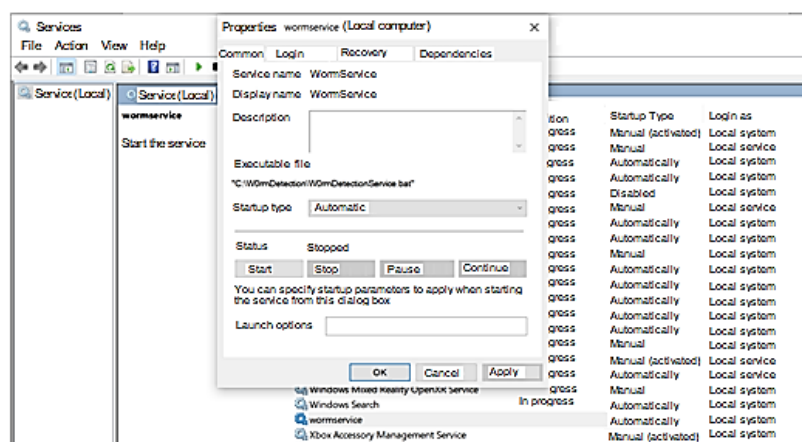
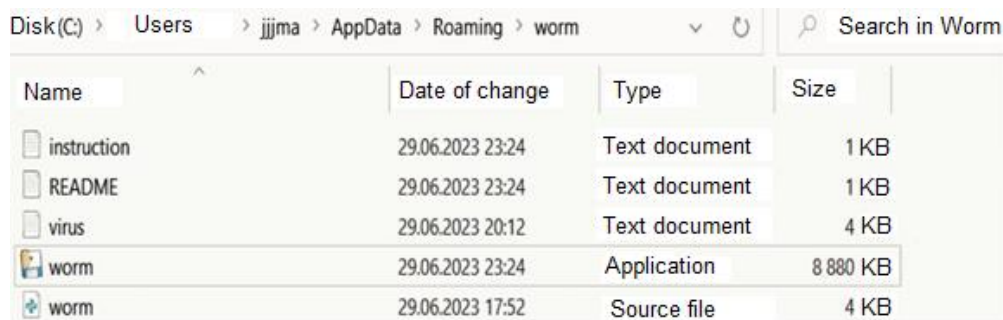


Figure 13 – «Network Worm» Service



Name	Date of change	Type	Size
instruction	29.06.2023 23:24	Text document	1 KB
README	29.06.2023 23:24	Text document	1 KB
virus	29.06.2023 20:12	Text document	4 KB
worm	29.06.2023 23:24	Application	8 880 KB
worm	29.06.2023 17:52	Source file	4 KB

Figure 14 – «Network Worm» in the Folder «AppData»

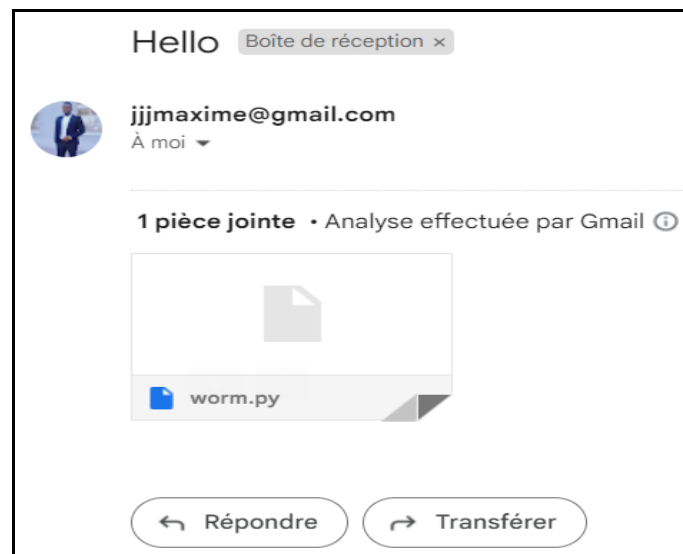


Figure 15 – Spreading the “Network Worm” by E-mail

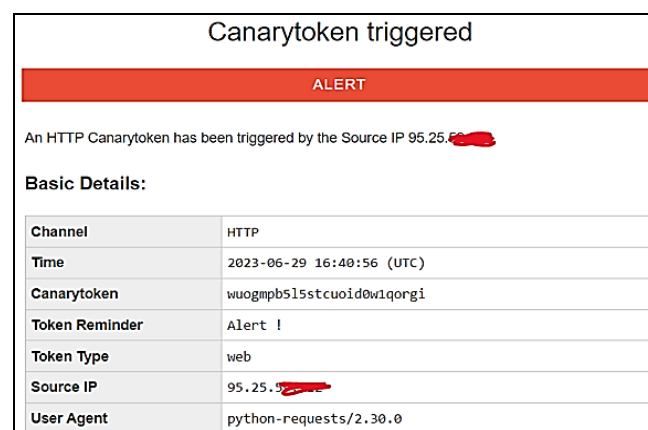
### The main tasks of the network worm after infecting the device with Windows OS:

Create itself copy on the infected device and write itself to startup via the registry.

Trigger (activate) the *canary token* (so that the notification is sent to E-mail that the network worm has settled on the next device (is demonstrated in the Fig. 16).

Contact server and receive from it some data (instructions). In this case, network worm will receive E-mail list addresses for the further sending of its copies to them (see Fig. 17).

Perform some actions on the infected device and begin self-propagation by sending its copies to previously received E-mails, as shown in Fig. 18



Canarytoken triggered	
ALERT	
An HTTP Canarytoken has been triggered by the Source IP 95.25.55.100	
Basic Details:	
Channel	HTTP
Time	2023-06-29 16:40:56 (UTC)
Canarytoken	wuogmpb515stcuoid0w1qorgi
Token Reminder	Alert !
Token Type	web
Source IP	95.25.55.100
User Agent	python-requests/2.30.0

Figure 16 – Canary token notification

```

(kali@kali)-[~]
$ python -m http.server 4444
Serving HTTP on 0.0.0.0 port 4444 (http://0.0.0.0:4444/) ...
10.10.0.1 - - [29/Jun/2023 10:52:50] "GET /instructions.txt HTTP/1.1" 200 -
10.10.0.1 - - [29/Jun/2023 10:58:02] "GET /instructions.txt HTTP/1.1" 200 -
10.10.0.1 - - [29/Jun/2023 11:08:15] "GET /instructions.txt HTTP/1.1" 200 -
10.10.0.1 - - [29/Jun/2023 11:52:59] "GET /instructions.txt HTTP/1.1" 200 -
10.10.0.1 - - [29/Jun/2023 11:59:54] "GET /instructions.txt HTTP/1.1" 200 -
10.10.0.1 - - [29/Jun/2023 12:05:06] "GET /instructions.txt HTTP/1.1" 200 -
10.10.0.1 - - [29/Jun/2023 12:11:27] "GET /instructions.txt HTTP/1.1" 200 -
10.10.0.1 - - [29/Jun/2023 12:12:13] "GET /instructions.txt HTTP/1.1" 200 -
10.10.0.1 - - [29/Jun/2023 12:14:40] "GET /instructions.txt HTTP/1.1" 200 -
10.10.0.1 - - [29/Jun/2023 12:16:27] "GET /instructions.txt HTTP/1.1" 200 -
10.10.0.1 - - [29/Jun/2023 12:23:52] "GET /instructions.txt HTTP/1.1" 200 -
10.10.0.1 - - [29/Jun/2023 12:24:46] "GET /instructions.txt HTTP/1.1" 200 -
10.10.0.1 - - [29/Jun/2023 12:25:21] "GET /instructions.txt HTTP/1.1" 200 -
10.10.0.1 - - [29/Jun/2023 12:26:26] "GET /instructions.txt HTTP/1.1" 200 -
10.10.0.1 - - [29/Jun/2023 12:27:01] "GET /instructions.txt HTTP/1.1" 200 -
10.10.0.1 - - [29/Jun/2023 12:27:18] "GET /instructions.txt HTTP/1.1" 200 -
10.10.0.1 - - [29/Jun/2023 12:30:19] "GET /instructions.txt HTTP/1.1" 200 -
10.10.0.1 - - [29/Jun/2023 12:31:23] "GET /instructions.txt HTTP/1.1" 200 -

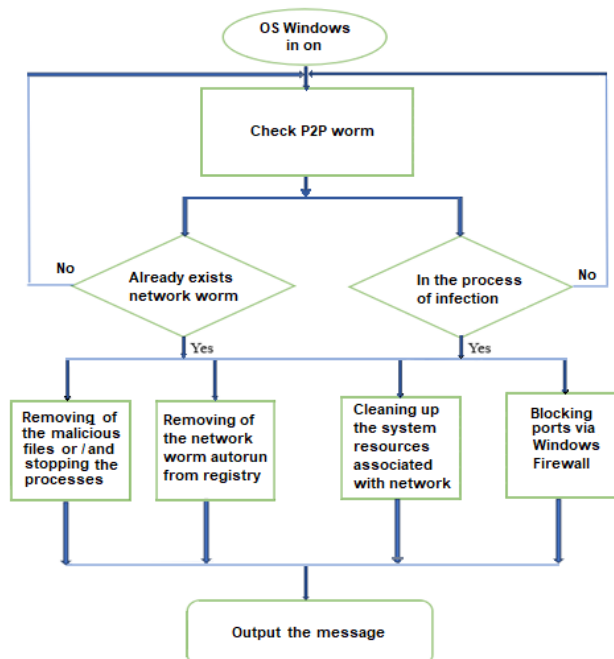
```

**Figure 17 – Contacting the Server and Receiving Instructions**

To detect such types of network worms, software was developed in programming language *Kotlin*, which has the following approach and block diagram (see Fig. 18):

When starting the Windows OS, the following steps must be performed:

1. Check the computer system for existing network worms.
2. Delete the malicious files and/or stop the processes.
3. Delete the process of network worm auto run from the registry.
4. Clean the system resources associated with the network worm.
5. Search for and neutralize the P2P network worm during infection.
6. Block the ports of computer system through Windows Firewall.



**Figure 18 – Software Block Diagram**

This software does not require information security and programming extensive knowledge, since the “bat” script (Fig. 19) and the service (Fig. 20) for auto starting the software have been created. There is algorithm description and main functions of software.

1. Run the developed program.

2. Create worm IPs empty list to store IP addresses of detected P2P network worms.
3. Call the get LocalIPandPort function to get the local IP address and port.
4. Extract the subnet from the local IP address.
5. Set the timeout value for the connection attempts.
6. Perform the following steps for each IP address in the subnet:
7. Check if IP address is P2P network worm by calling the *WormP2P* function with the IP address, local port, and timeout.
8. If it is P2P network worm, add the IP address to the *wormIPs* list.
9. For each network worm IP address in *wormIPs*, perform the following steps:
10. Call *neutralizeWormP2P* function; delete malicious files or/and stop processes; remove worm autorun from registry; clean system resources associated with worm; block ports via Windows Firewall; with the worm IP address to neutralize the worm.
11. If action is successful, print message indicating that network worm is neutralized.
12. Terminate the program.

Name	Date of change	Type	Size
w0rm	29.06.2023 17:52	Source file	4 KB
W0rmDetectionApp.kt	29.06.2023 16:50	File "KT"	9 KB
W0rmDetectionRun.kt	29.06.2023 16:52	File "KT"	2 KB
W0rmDetectionService	29.06.2023 18:37	Batch file	1 KB
worm	29.06.2023 17:52	Source file	4 KB
worm.spec	29.06.2023 23:24	File "SPEC"	1 KB

Figure 19 –Script “bat” for Launching Software

Name	State	Startup type	Login as
Shared PC Account Manager	Turned off	Local system	Network service
СМР дисковых пространств (Майкрософт)	Manually	Local system	Local system
System	In progress	Automatically	Local system
vagrant-vmware-utility	In progress	Automatically	Local system
VirtualBox system service	Manually	Local system	Local system
VMware Authorization Service	In progress	Automatically	Local system
VMware DHCP Service	In progress	Automatically	Local system
VMware NAT Service	In progress	Automatically	Local system
VMware USB Arbitration Service	In progress	Automatically	Local system
W0rmDetectionService	Automatically	Local system	

Figure 20 – Service for Auto start Software

The result of the software operation is shown in Fig. 21:

```

ristC:\Users\jjjma\AppData\Roaming\Microsoft\Windows\Recent\worm.lnkdelete file
file ristC:\Users\jjjma\AppData\Roaming\worm\worm.pydelete file ristC:\Users\jjjma\Desktop\ALL\worm.pydelete file
file ristC:\W0rmDetection\build\worm\worm.exe.manifestdelete file ristC:\W0rmDetection\build\worm\worm.pkgdelete file
file ristC:\W0rmDetection\dist\worm.exe.delete file ristC:\W0rmDetection\worm.pydelete file ristC:\W0rmDetection\worm.specdelete file rist

```

Figure 21 – Removing the Network Worm from the Folders and Registry

## CONCLUSION

Detecting and neutralizing network worms in Windows OS are critical task for information security specialists. Combination of antivirus software, intrusion detection systems, network monitoring of traffic, software patching and updating, as well as user training allows to effectively combating these threats. The information security should be priority for every organization and individual in order to minimize risks and maintain confidentiality, integrity and availability of data.

## REFERENCE

Bowden M. "Worm: The First Digital World War" 2019. – 140 c. Szor P. "The Art of Computer Virus Research and Defense", 2017. – 210 c.

- Skoudis E. and Zeltser L. "Malware: Fighting Malicious Code", 2015. – 41 c. Singh S. "Code Book: Science of Secrecy from Ancient Egypt to Quantum Cryptography", 2009. 200 p.
- Stuttard D. and Pinto M. "The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws", 2020. – 207 c.
- Aycock J. "Computer Viruses and Malware", 2013. – 127 c. Kislitsin N. Why Malware Crypting Services Deserve More Scrutiny. [Electronic resource]. URL: <https://krebsonsecurity.com/2023/06/why-malware-crypting-services-deserve-more-scrutiny/> (дата обращения: 21.06.2021).
- Chen C.J.Z. "Self-Learning Worm Using Importance Scanning" Proc. ACM CCS Wksp. Rapid Malcode (WORM '05), 2005.
- Salimi H., Jalili R. "Survey of Worm Detection and Containment Methodologies in Online Social Networks". [Electronic resource]. URL: [https://www.researchgate.net/publication/3454688\\_A\\_survey\\_of\\_Internet\\_worm\\_detection\\_and\\_containment](https://www.researchgate.net/publication/3454688_A_survey_of_Internet_worm_detection_and_containment)
- Gupta A., Gupta A.K., "Survey: Detection and Prevention of Wormhole Attack in Wireless Sensor Networks", Global Journal of Computer Science and Technology: E- Network, Web & Security. V. 14, 2014. 23– 31p.
- Biradar S., Raja K. "Worm Detection and Containment in Wireless Sensor Networks: A Survey". [Electronic resource]. URL: <https://www.sciencedirect.com/science/article/abs/pii/S138912860600243X>
- Marchette D.J., Pottenger W.M., Pottenger M.L. "Defending Against Internet Worms: Signature- Based Approach Using Distributed Sensors". URL: <https://www.cise.ufl.edu/~sgchen/Publications/TCa.05.pdf>