**BSI EXAM 2**

**[Linux] Logs analysis**

1. Log file statistics

On the server, in the file */var/log/fontconfig.log* there is a record of events (logs) that occurred in the system.

The task solution is in the file *codebook.txt* (a file in your home directory) in the line of the number equal to the number of lines of the file */var/log/fontconfig.log*.

Server address: *10.0.51.236*

Username: *alice*

Password: *cyberskiller\_ssh*

Solution:

ssh -l alice 10.0.51.236

yes -->password from task→ ls → cd /var/log → wc fontconfig.log→(first number of previous command in place of 23  in cat command )

cd → cat codebook.txt | head -23 | tail -1

1. Log analysis by date

Solution:

Log in as in previous task-->cd */var/log → then cat audit.log and look for specified in task log*

*2nd option :* cat audit.log | grep 10:10:31 (here paste date from task without days)

1. Log analysis by length

On the server, in the file */var/log/audit.log* there is a record of events (logs) that occurred in the system. The task solution is in the entry (line) which has more than 150, but less than 180 characters (at the end of the line).

Server address: *10.0.28.38*

Username: *alice*

Password: *cyberskiller\_ssh*

Solution:

log as in previous → go to same folder as in previous task → cat audit.log | grep -E "^.{150,180}"

1. Log analysis by time frame

On the server, in the file */var/log/audit.log* there is a record of events (logs) that occurred in the system. The task solution is in the entry (line) added between 8:00 and 8:02 (at the beginning of the line).

Server address: *10.0.107.250*

Username: *alice*

Password: *cyberskiller\_ssh*

Solution:

Log in→ cd /var/log → cat audit.log | grep -e "08:0[0-2]:[0-5][0-9]" → correct anserw is one of these flag in my case second form the end

1. Log analysis by a particular service

On the server, in the file */var/log/audit.log* there is a record of events which occurred in the system. The task solution is in the entry (line) which concerns a VSFTPD, FTP, or XINETD service.

Server address: *10.0.116.220*

Username: *alice*

Password: *cyberskiller\_ssh*

Solution: log in → cd /var/log → cat audit.log | grep "vsftpd\|ftp\|xinted"

1. Failed SSH login attempts analysis

On the server, in the file */var/log/audit.log* there is a record of events regarding logging in through SSH. One of them includes a failed login attempt entry.

The task solution is the name of the user on whose account an attempt was made to log in.

Server address: *10.0.80.64*

Username: *alice*

Password: *cyberskiller\_ssh*

Solution: log in → cd /var/log → cat → cat audit.log | grep -E 'sshd.\*failed'   or

cat audit.log | grep -E 'sshd.\*invalid'

1. Successful SSH login attempts analysis

On the server, in the file */var/log/audit.log* there is a record of events regarding logging in through SSH. One of them is a successful login attempt, which means that the intruder gained access to the server.

The task solution is the name of the user on whose account an attempt was made to log in.

Server address: *10.0.11.152*

Username: *alice*

Password: *cyberskiller\_ssh*

Solution: same first 2 steps as in previosu tasks → cat audit.log | grep opened | grep -v root

1. Multiple failed SSH login attempts analysis

On the server, in the file */var/log/audit.log* there is a record of events regarding logging in through SSH. A part of them are unsuccessful login attempts to accounts of users *user\_X*, where *X* assumes various values.

The task solution can be constructed from the values of *X* ordered by the time of login attempt – from the oldest to the latest one.

Server address: *10.0.16.139*

Username: *alice*

Password: *cyberskiller\_ssh*

Solution:

log in → cd /var/log -->cat audit.log | grep user\_

**[Linux] Bypassing basic mechanisms of user block**

1. **Automatic logout 2**

On the server, in the user’s home directory, a file *secret.txt* was placed which includes the task solution. However, as a result of the file *.bash\_rc* modification, after logging in, the user is automatically logged out. Find a way and read the file content and provide the task solution.

Server address: *10.0.38.17*

Username: *alice*

Password: *cyberskiller\_ssh*

ssh -l alice 10.0.38.17 "ls"

sh -l alice 10.0.38.17 -t  "sh -i"

alice@10.0.38.17's password:

$ ls

secret.txt

$ car secret.txt

sh: 2: car: not found

$ cat secret.txt

CS4997605949

1. **Access to files in the /root folder 2**

On the server there is a file *secret.txt* in the folder */root* whose owner is the *root* account. The task is to read the content of the file which includes the solution. To solve this exercise, you can use one of the user’s home directory files.

Server address: *10.0.63.78*

Username: *alice*

Password: *cyberskiller\_ssh*

sh -l alice 10.0.38.17 -t  "sh -i"

alice@10.0.38.17's password:

*$ ls*

*archive*

*$ cat archive*

*cat: archive: Is a directory*

*$ cd archive*

*$ ls*

*file  file.c  file.sh*

*$ cd ../root*

*sh: 5: cd: can't cd to ../root*

*$ cd ../../root*

*sh: 6: cd: can't cd to ../../root*

*$ ls*

*file  file.c  file.sh*

*$ ./file*

*Please Select File to read:*

*/root/secret.txt*

*CS7421629098*

*Speedrun version (same thing as above, but less lines):*

*ssh -l alice 10.0.110.77 -t  "sh -i"*

*$ cd archive*

*$ ls -la*

*-rwsr-xr-x 1 root  root  16744 Jun 17 18:21 file*

*-rwxrwxr-x 1 root  root    174 Jun 17 18:21 file.c*

*-rwsrwxr-x 1 root  root    153 Sep  1  2019 file.sh*

*$ ./file*

*Please Select File to read:*

*/root/secret.txt*

*CS0331205871*

1. **Access to files belonging to the root user 2**

On the server there is a file *secret.txt* in the folder */root* whose owner is *root*. The task is to read the content of the file with the solution.

To solve the task you can use one of the files in the user’s home directory which allows to read files from the folder */tmp* and is prone to a Path Traversal attack.

Server address: *10.0.76.78*

Username: *alice*

Password: *cyberskiller\_ssh*

ssh -l alice 10.0.76.78  -t  "sh -i"     255 ⨯

Are you sure you want to continue connecting (yes/no/[fingerprint])? yes

Warning: Permanently added '10.0.76.78' (ECDSA) to the list of known hosts.

alice@10.0.76.78's password:

$ ls

archive

$ cd archive

$ ls

file  file.c  file.sh

$ ./file

Please Select File to read:

../root/secret.txt

CS2494287228

1. **User privilege escalation 3**

On the server there is a file *secret.txt* in the folder */root* whose owner is *root*. The task is to read the content of the file including the solution.

In order to solve the task, use one of the programs with SUID permissions to escalate privileges to the *root* account.

Server address: *10.0.42.229*

Username: *alice*

Password: *cyberskiller\_ssh*

LOG IN

php7.4 -f /root/secret.txt

CS0114047994

1. **Shell other than Bash 3**

After logging in through SSH you need to log in to the account of user *bob*, however, after logging in to *bob's* account, shell other than *Bash* gets launched.

Identify what shell this is, exit it to Bash shell, and read the file from the home directory of user bob with the task solution.

Server address: *10.0.70.252*

Username: *alice*

Password: *cyberskiller\_ssh*

*log → cat etc/passwd → cd /usr/bin→ cat bob →*

cd ~ -- > ll → cd .ssh → ll → ssh -l bob localhost -i bob\_key.pem → man more → make terminal smaller (like 1/12 of screen) login again → press v key →  :e secret.txt

*short/similar version:*

*cd .ssh  
make terminal window very small*

*ssh -l bob localhost -i bob\_key.pem*

*press v*

*write “:e secret.txt” then hit enter*

1. Deleted file

In your home directory there is a program *file.py* (written in Python) which reads the file with the solution to the task. Unfortunately, it has been deleted, but it is still running in the background.

Your task is to read the content of the file containing the task solution.

Server address: *10.0.3.44*

Username: *alice*

Password: *cyberskiller\_ssh*

Solution:

login

cd archive→ ll → cat file.py → ps -aux → ps -x → check PID of file.py -- >cd /proc/

→cd PID → ls → cd fd → cat  ./3

**[Cryptography]** Hash **functions**

1. **The MD5 hash function 1**

A TCP service was started on port 1337 on the server. Having established the connection, you will receive a string of characters.

Generate the MD5 hash for this string of characters and send it back in hexadecimal format to obtain the task solution.

Server IP address: *10.0.13.218*

You can use the netcut command to connect to the server: *nc 10.0.13.218 1337* .

**nc 10.0.13.218 1337**

Encode the string: **"GrY6WNAbNWmnPmLTAgf7"** (no quote characters) using the MD5 algorithm, and then send the result to this service to get the answer for the task.

Send MD5 sum in hexadecimal format.

q

┌──(hubert㉿kali)-[~]

└─$ **echo -n "GrY6WNAbNWmnPmLTAgf7" | md5sum**

c282289a1759482d2b96df186f33102a  -

┌──(hubert㉿kali)-[~]

└─$ **nc 10.0.13.218 1337**

Encode the string: "GrY6WNAbNWmnPmLTAgf7" (no quote characters) using the MD5 algorithm, and then send the result to this service to get the answer for the task.

Send MD5 sum in hexadecimal format.

c282289a1759482d2b96df186f33102a

Correct answer, secret for exercise is: CS3445861132

1. **The SHA1 hash function 1**

A TCP service was started on port 1337 on the server. Having established the connection, you will receive a string of characters.

Generate the SHA1 hash for this string of characters and send it back in hexadecimal format to obtain the task solution.

Server IP address: *10.0.26.141*

You can use the netcat command to connect to the server: *nc 10.0.26.141 1337* .

nc **10.0.26.141 1337**

Encode the string: "z**IKOGmnRiKX4bLeo59Wo**" (no quote characters) using the SHA1 algorithm, and then send the result to this service to get the answer to the task.

Send SHA1 sum in hexadecimal format.

q

Invalid answer, please try again

┌──(hubert㉿kali)-[~]

└─$ **echo -n "zIKOGmnRiKX4bLeo59Wo" | sha1sum**

**2e53bfd09dda26dbe1c440ad033f053930db2b40**  -

┌──(hubert㉿kali)-[~]

└─$

┌──(hubert㉿kali)-[~]

└─$ **nc 10.0.26.141 1337**                                                                         130 ⨯

Encode the string: "zIKOGmnRiKX4bLeo59Wo" (no quote characters) using the SHA1 algorithm, and then send the result to this service to get the answer to the task.

Send SHA1 sum in hexadecimal format.

**2e53bfd09dda26dbe1c440ad033f053930db2b40**

Correct answer, secret for exercise is: CS2370748797

1. **The SHA3 hash function 1**

A TCP service was started on port 1337 on the server. Having established the connection, you will receive a string of characters.

Generate the SHA3 hash for this string of characters and send it back in hexadecimal format to obtain the task solution.

Server IP address: *10.0.87.163*

You can use the netcat command to connect to the server: *nc 10.0.87.163 1337* .

**python3**

Python 3.9.1+ (default, Feb  5 2021, 13:46:56)

[GCC 10.2.1 20210110] on linux

Type "help", "copyright", "credits" or "license" for more information.

>>> **import hashlib**

>>> **h = hashlib.sha3\_512()**

>>> **h.update("YP4VqDce654sFGorQhUx".encode("utf-8"))**

>>> **h.digest()**

b' \x0cxx\x9d\x8c\xa2\xc2\xb4\xa0T2\xf9v\x9e\x96\xcbC\xf8o\xdc\xbc\xc6c\x18\xdd5\x07\xb3\x9c\xddBm\xce\xe0L\x97\xec\x01\x1ei\x82v\xbf\x1eQ;\xa6\x08\xeej\_\x88\xbd4\x0f\x07E=\x93\r;\xe7k'

>>> **h.hexdigest()**

'200c78789d8ca2c2b4a05432f9769e96cb43f86fdcbcc66318dd3507b39cdd426dcee04c97ec011e698276bf1e513ba608ee6a5f88bd340f07453d930d3be76b'

>>>

copy result of h.hexdigest() and paste into nc url

1. **Breaking the MD5 hash with the dictionary method 1**

A TCP service was started on port 1337 on the server. Having established the connection, you will receive the *MD5* hash.

Using the John The Ripper program and the *rockyou.txt* dictionary of most popular passwords, find a string of characters matching the MD5 hash.

Return the string of characters to the website in order to obtain the task solution.

Server IP address: *10.0.51.196*

You can use the netcat command to connect to the server: *nc 10.0.51.196 1337* .

Connect →

copy hash

rockyou → download in raw from git

save your hash in file  echo “hash here” > pass.txt

john --wordlist=rockyou.txt --format=raw-md5 pass.txt    in terminal

nc url port

paste passwd form john The Ripper

1. **Hash function identification 1**

A TCP service was started on port 1337 on the server. Having established the connection, you will receive a string of bytes and the result of one of the hash functions for these bytes (in hexadecimal format).

Identify which hash function has been used and return its name (written in small characters and without special characters, e.g., sha256) to the server in order to receive the task solution.

**Note:** The identified algorithm can be sent only once. If the answer is incorrect, the task will need to be restarted.

Server IP address: *10.0.112.196*

You can use the netcat command to connect to the server: *nc 10.0.112.196 1337* .

nc url port → new terminal → python3

python3

Python 3.9.1+ (default, Feb  5 2021, 13:46:56)

[GCC 10.2.1 20210110] on linux

Type "help", "copyright", "credits" or "license" for more information.

>>> import hashlib

>>> hashlib.algorithms\_available

{'sha3\_256', 'sha3\_224', 'sha512\_256', 'sha512\_224', 'blake2b', 'sha3\_384', 'shake\_256', 'md5', 'shake\_128', 'sha512', 'sha224', 'sha1', 'sha384', 'blake2s', 'sm3', 'ripemd160', 'whirlpool', 'sha256', 'md4', 'sha3\_512', 'md5-sha1'}

>>> s = "A77x69BPSPeHHMfTNRuD".encode("utf-8")

>>> hh = "99150d6f27bfda009f9a077147f5f924ac77355c01209e98a49b007ba13533e9"

>>> for alg in hashlib.algorithms\_available:

... try:

...         h = hashlib.new(alg)

...         h.update(s)

...         if h.hexdigest() == hh:

...                 print("Alg: ", alg)

... except:

...                 print("Exception: ", alg)

 run it → ctrl + r → paste alg into nc url port to get a flag

1. **The HMAC authentication code 1**

A TCP service was started on port 1337 on the server. Having established the connection, you will receive a string of characters and a password.

Using the*OpenSSL* library, generate authentication code for this string of characters and password with the use of the*HMAC* algorithm and the *SHA512* function. Send back the authentication code in hexadecimal format to obtain the task solution.

Server IP address: *10.0.34.24*

You can use the netcat command to connect to the server: *nc 10.0.34.24 1337* .

nc url port

openssl dgst -sha512 -hmac “hash” → enter → then paste message from nc  ctrl + d

→ echo -n “message form nc” | openssl dgst -sha512 -hmac “hash”  → copy paste into nc url port

1. **MD5 collision 1**

A TCP service was started on port 1337 on the server. Having established the connection, send to the service two strings of bytes (in hexadecimal format, e.g., *aabbccdd11223344*) separated by the newline character, which generate the same MD5 hash function. After sending the correct blocks, you will receive the solution to the task.

Server IP address: *10.0.76.82*

You can use the netcat command to connect to the server: *nc 10.0.76.82 1337* .

log in → copy to file

d131dd02c5e6eec4693d9a0698aff95c2fcab58712467eab4004583eb8fb7f89

55ad340609f4b30283e488832571415a085125e8f7cdc99fd91dbdf280373c5b

d8823e3156348f5bae6dacd436c919c6dd53e2b487da03fd02396306d248cda0

e99f33420f577ee8ce54b67080a80d1ec69821bcb6a8839396f9652b6ff72a70

d131dd02c5e6eec4693d9a0698aff95c2fcab50712467eab4004583eb8fb7f89

55ad340609f4b30283e4888325f1415a085125e8f7cdc99fd91dbd7280373c5b

d8823e3156348f5bae6dacd436c919c6dd53e23487da03fd02396306d248cda0

e99f33420f577ee8ce54b67080280d1ec69821bcb6a8839396f965ab6ff72a70

 place it in 2 lines → save file

nc url port << your file with text from above

**[Cryptography] Symmetric encryption**

1. **Data decryption with AES 1**

A TCP service was started on port 1337 on the server. Having established the connection, you will receive a string of encrypted characters (coded in *BASE64*) and the key used for encryption.

Decrypt the string of characters with the use of the *AES-256-ECB* algorithm and the given key in order to get the task solution.

Server IP address: *10.0.126.219*

You can use the netcat command to connect to the server: *nc 10.0.126.219 1337* .

nc url port → echo “ message” > data.enc → openssl enc -aes-256-ecb -d -a -in data.enc -K key from nc

1. **File decryption with AES 1**

There is a website on a server with an encrypted file and a file with the encryption key.

Using the received key, decrypt the encrypted file in order to get the task solution, being aware that:

* The algorithm used is AES-256-CBC,
* The key generating function is PBKDF2.

Website URL:<http://10.0.115.255>

download 2 files → go to downloads in terminal → openssl enc -aes-256-cbc -d -in cs.out -kfile pwd.pass -pbkdf2

1. **Data decryption with the 3DES algorithm 1**

A TCP service was started on port 1337 on the server. Having established the connection, you will receive a string of encrypted characters (encoded in *BASE64*) and the key used for encryption.

Decrypt the string of characters with the *3DES* algorithm and using the given password and the *PBKDF2* key generating algorithm in order to get the task solution.

Server IP address: *10.0.24.197*

You can use the netcat command to connect to the server: *nc 10.0.24.197 1337* .

gedit data1.enc → paste there message → openssl enc -d -a -in data1.enc -k **c48fe623-1e3c-432e-8cc5-a2e7868213e9** -pbkdf2 -des-ede3-ecb

→ above code is password from nc url port

**if error file input save message usign command cho “ message” > data.enc**

1. **Decryption using PBKDF1 algorithm 1**

A TCP service was started on port 1337 on the server. Having established the connection, you will receive a string of encrypted characters (encoded in *BASE64*), the password used for encryption, key generation algorithm, and the number of iterations of the algorithm.

Decrypt the string of characters with the *AES-256-ECB* algorithm and using the given password and the key generation algorithm *PBKDF1* as well as the indicated number of iterations of the algorithm in order to obtain the task solution.

Server IP address: *10.0.123.23*

You can use the netcat command to connect to the server: *nc 10.0.123.23 1337*

*Copy message paste it into file ex data.enc → openssl enc -d -aes-256-ecb -k KEY -in data.enc -a -iter num\_of\_iterations*

**if error file input save message usign command cho “ message” > data.enc**

1. **PBKDF2 with non standard number of iterations 1**

A TCP service was started on port 1337 on the server. Having established the connection, you will receive a string of encrypted characters (encoded in *BASE64*), the password used for encryption, key generation algorithm, and the number of iterations of the algorithm.

Decrypt the string of characters with the *AES-256-CBC* algorithm and using the given password, the key generation algorithm *PBKDF2*, as well as the indicated number of iterations of the algorithm in order to obtain the task solution.

Server IP address: *10.0.9.96*

You can use the netcat command to connect to the server: *nc 10.0.9.96 1337* .

**if error file input save message usign command cho “ message” > data.enc**

*openssl enc -d -aes-256-cbc -k KEY -in data.enc -a -iter num\_of\_iterations*

openssl enc -d -aes-256-cbc -k 6c2a1ff1-cf53-4150-908a-364964dddf94 -in data.enc -a -iter 1625792

1. **Identification of the encryption algorithm 1**

A TCP service was started on port 1337 on the server. Having established the connection, you will receive a string of bytes and the result of one of the encryption functions for these bytes (in hexadecimal format).

Identify which encryption function has been used and send back its name (written is small characters and without special characters) to the server to obtain the task solution.

Server IP address: *10.0.38.239*

You can use the netcat command to connect to the server: *nc 10.0.38.239 1337* .

nc address port → open new terminal → cp message paste it into data.enc → copy passwd → create pwd.pass file with passwd → openssl enc -ciphers → copy ciphers with end like -ecb → paste into file

bash

save as ciphers.txt → create bash script

while read cipher;

do

    echo "----"

    echo $cipher

    openssl enc -d -a $cipher -in data.enc -kfile pwd.pass -pbkdf2

    echo " "

done;

save as name.sh → run it → bash name.sh < ciphers.txt → find with no errror ex aes-256-ecb → pastre it into nc url port

1. **Encrypted Zip file**

On a server there is a website with a *.zip* archive, encrypted in one of the popular passwords and containing the task solution.

In order to get the solution to the task, unpack the *.zip* archive, protected with password.

Hint:To break the password, use a tool like *fcrackzip* or *John The Ripper* and the file with most popular passwords *rockyou.txt*.

Website URL:<http://10.0.116.155>

Solution:

Open site → download file → unzip name of file in terminal→ download rockyou wordlist -> fcrackzip -u -D -p rockyou.txt name\_of\_file → unzip name\_of\_File → founded passwd → cat secret.txt

sudo apt-get install fcrackzip  → if u dont have fcrack installed yet

**[Cryptography] Asymmetric encryption**

1. **RSA keys generation 1**

A TCP service was started on port 1337 on the server. Generate a 4096 byte pair of *RSA* keys and send to the server in PEM format to get the task solution.

The data necessary to connect will be found below after launching the exercise with the Start button.

Server IP address: *10.0.10.201*

You can use the netcat command to connect to the server: *nc 10.0.10.201 1337* .

netcat command

openssl genrsa -out priv.pem 4096

cat priv.pem

openssl rsa -in priv.pem -text

openssl rsa -in priv.pem -pubout -out pub.pem

cat pub.pem

openssl rsa -in pub.pem -pubin -text

mosepad keys.pem

copy public key

space

copy paste

private key

nc url port

paste value

1. **Key generation on elliptic curves 1**

TCP service was started on port 1337 on the server. Generate a pair of keys based on the *NIST P-256 prime256v1* elliptic curve and send them to the server in *PEM* format to get the task solution.

The data necessary to connect will be found below after launching the exercise with the Start button.

Server IP address: *10.0.110.194*

You can use the netcat command to connect to the server: *nc 10.0.110.194 1337*

*openssl ecparam -list\_curves*

*openssl ecparam -name prime256v1 -genkey -out priv.pem*

*cat priv.pem*

*openssl ec -in priv.pem -out pub.pem -pubout*

*cat pub.pem*

*cat pub.pem > keys.pem*

*echo “” >> keys.pem*

*cat priv.pem >> keys.pem*

*cat keys.pem*

*copy paste after nc address port*

1. **Encryption with the public key 1**

A TCP service was started on port 1337 on the server. Having established the connection, you will get an *RSA* server public key in *PEM* format and a string of characters.

Encrypt the string of characters with the received *RSA* server public key and return the encrypted data, encoded with algorithm *BASE64*, to the server in order to obtain the task solution.

Server IP address: *10.0.23.142*

You can use the netcat command to connect to the server: *nc 10.0.23.142 1337* .

nc url port

copy key

mousepad pub.pem paste key

copy string and save it in data.txt

openssl rsautl -encrypt -in data.txt -inkey pub.pem -pubin -out data.enc -oaep

base64  -w0 data.enc

copy paste into nc url port

1. **Certificate with a given Common Name field 1**

A TCP service was started on port 1337 on the server. Having established the connection, you will get a certain string of characters.

Create a Certificate Authority and then generate a certificate which will contain in its *CN (Common Name)* field a previously set string of characters. Send the certificate in *PEM* format to the server in order to obtain the task solution.

Server IP address: *10.0.86.227*

You can use the netcat command to connect to the server: *nc 10.0.86.227 1337*

*mkdir ca*

*cd ca/*

*openssl genrsa -out ca.key 2048*

*openssl  req -new -x509 -key ca.key -out ca.crt*

*fill certificade*

*Name CyberSkiller, Unit name CA*

*String from nc url port goes to common name*

*email field may stay clean*

*cat ca.crt*

*paste into nc*

1. **Certificate based on request 1**

A TCP service was started on port 1337 on the server. Having established the connection, you will receive a Certificate Signing Request in *PEM* format.

Create a Certificate Authority and then generate the certificate on the basis of the received request. Send the certificate in *PEM* format to the server in order to obtain the task solution.

Server IP address: *10.0.115.58*

You can use the netcat command to connect to the server: *nc 10.0.115.58 1337* .

cd ca/

mousepad client.csr

paste there certificate from nc

//cat /usr/lib/ssl/openssl.cnf

openssl x509 -req -CAkey ca.key -CA ca.crt -in cient.csr -CAcreateserial -out client.crt

cat client.crt

1. **The PGP key generation 1**

A TCP service was started on port 1337 on the server. Having established the connection, you will receive a string of characters.

Generate a pair of *PGP* keys which will contain in its name a previously given string of characters. Send the public key in *PGP* to the server to obtain the task solution.

Server IP address: *10.0.69.91*

You can use the netcat command to connect to the server: *nc 10.0.69.91 1337* .

nc

copy string

gpg --gen-key

paste string

Adress [test@cs.com](mailto:test@cs.com)

D

input chosen passwd

gpg --list-key

gpg --export --armor from uid string before email

1. **Encryption with the PGP key 1**

A TCP service was started on port 1337 on the server. Having established the connection, you will receive a string of characters and a *PGP* public key.

Using the *GPG* program, encrypt the string of characters with the use of the given *PGP* key and send the encrypted message (in *PGP* format) to the server in order to obtain the task solution.

Server IP address: *10.0.28.91*

You can use the netcat command to connect to the server: *nc 10.0.28.91 1337* .

nc url port

copy key

mousepad pub.pem

paste there key

gpg --show-key pub.pem

gpg --import pub.pem

copy string from nc

mousepad data.txt

save there string

//gpg --encrypt --recipient cyberskiller data.txt

//yes

//cat data.txt.gpg

//gpg --encrypt --armor --recipient cyberskiller data.txt

yes

echo -n string from nc | gpg --encrypt --armor -recipient cyberskiller

cat data.txt.asc

copy

paste into nc url port

echo -n string from nc | gpg --encrypt --armor -recipient cyberskiller

1. **Decryption with PGP key**

Using the *GPG* program, generate a pair of keys (public-private) for a any user.

Next, connect to the server on port 1337 (TCP) and send the *PGP* public key in order to get from the server an encrypted *PGP* message.

Decrypt the message using the private *PGP* key to obtain the task solution.

Server IP address: *10.0.49.227*

You can use the netcat command to connect to the server: *nc 10.0.49.227 1337* .

nc

gpg --gen-key

Test CyberSkiller

[test@cs.com](mailto:test@cs.com)

password generate

gpg --export --armor test

copy

nc

paste it there

mousepad data.enc

copy output message

save

gpg --decrypt data.enc

enter passwd which u specified before