Odpowiedzi

1.1
```
docker run -dp 10001:10001 mazurkatarzyna/bsk-book-p1-ch1-ex001
curl -X GET http://127.0.0.1:10001/random
echo -ne "hello" |  openssl dgst -md5
curl -X GET http://127.0.0.1:10001/check_md5/XXX
```

1.2
```
docker run -dp 10002:10002 mazurkatarzyna/bsk-book-p1-ch1-ex002
curl -X GET http://127.0.0.1:10002/random
echo -ne "hello" |  openssl dgst -sha256
curl -X GET http://127.0.0.1:10002/check_sha256/XXX
```

1.3
```
docker run -dp 10003:10003 mazurkatarzyna/bsk-book-p1-ch1-ex003
curl -X GET http://127.0.0.1:10003/random
echo -ne "hello" |  openssl dgst -sha512
curl -X GET http://127.0.0.1:10003/check_sha512/XXX
```

1.4
```
docker run -dp 10004:10004 mazurkatarzyna/bsk-book-p1-ch1-ex004
curl -X GET http://127.0.0.1:10004/random

openssl kdf -keylen 24 -kdfopt pass:laveritz -kdfopt salt:NaCl2024 -kdfopt iter:1
-kdfopt memcost:8192 ARGON2D |  tr -d ':'  | tr '[:upper:]' '[:lower:]'

curl -X GET http://127.0.0.1:10004/check_argon2d/XXX
curl -X GET http://127.0.0.1:10004/check_argon2i/XXX
curl -X GET http://127.0.0.1:10004/check_argon2id/XXX
```

1.5
```
cat ex13.txt | openssl dgst -md5
```

1.6
```
cat ex14.txt | openssl dgst -md5
```

1.7
```
openssl rand -base64 4
openssl rand -base64 4 | openssl dgst -md5
```

1.8
```
openssl rand -base64 16 | openssl dgst -sha512
```

1.9 Generowanie hasła:
```
cat /dev/urandom | base64 | head -n 1 | tr -dc '[:alnum:]' | cut -c -16
```

Skrót MD5 hasła:
`cat /dev/urandom | base64 | head -n 1 | tr -dc '[:alnum:]' | cut -c -16 | openssl dgst -md5`

Kodowanie Base64:
`base64` lub `base64 -e`

Wypisz `n` pierwszych linii z pliku:
`head -n 1`

Usuń znaki (`-d`), użyj zbioru dopełnienia (`-c`), `[:alnum:]` - wszystkie litery i cyfry
`tr -dc '[:alnum:]'`

Pobierz określoną liczbę znaków:
`cut -c -16`

1.10 Generowanie haseł:
`crunch 3 3 1234567890 -o test.txt`

Haszowanie haseł:
`while read line; do echo -n ''$line'' | openssl dgst -sha1; done < test.txt`

1.11 Generowanie haseł:
`crunch 5 5 -t %a^b% -o test.txt`

Haszowanie haseł:
`while read line; do echo -n ''$line'' | openssl dgst -sha3-224; done < test.txt`

```
, for all uppercase letters
@ for all lowercase letters
% for all numeric characters
^ for all special characters
```

1.12 Generowanie haseł:
`crunch 3 3 abc + 468 ?%: -t @%^`

Haszowanie haseł:
`while read line; do echo -n ''$line'' | openssl dgst -sha3-224; done < test.txt`

1.13 `docker run -it mazurkatarzyna/hash-identifier:latest`. Hash to NTLM.

1.14 `docker run -it mazurkatarzyna/hash-identifier:latest`. Hash to JWT. Sprawdź `https://jwt.io/`.

1.15 Lokalizacja pliku ze słownikiem:
`/usr/share/wordlists/rockyou.txt.gz`
`gzip -d rockyou.txt.gz`

Złamanie hasha:
`john --format=raw-md5 --wordlist=''wordlist.txt'' hash.txt`
`john --show --format=Raw-MD5 hash.txt`

1.16 Lokalizacja pliku ze słownikiem:
`/usr/share/wordlists/rockyou.txt.gz`
`gzip -d rockyou.txt.gz`

Złamanie hasha:
`john --format=raw-SHA256 --wordlist=''wordlist.txt'' hash.txt`
`john --show --format=Raw-SHA256 hash.txt`

1.17 Polecenie:

`hashcat -a 0 -m 0 --force ex1.6.txt /usr/share/wordlists/rockyou.txt`
`hashcat --show ex1.6.txt`
`-a 0` - atak słownikowy, tu nazywany atakiem straight
`-m 0` - łamanie MD5
`--show` - pokaż wynik

Hasło: `8afa847f50a716e64932d995c8e7435a:princess`

1.18 Polecenie:
`sudo gedit /etc/john/john.conf`

```
[Incremental:z5shadow]
File = $JOHN/utf8.chr
MinLen = 3
```

```
MaxLen = 3
CharCount = 196

john --incremental:z5shadow z5.shadow
john z5.shadow --show
```

Hasło:

```
u9:123:16026:0:99999:7:::
u10:dhw:16026:0:99999:7:::

2 password hashes cracked, 0 left
```

1.19 Polecenie:

```
john z2.shadow --wordlist=/usr/share/wordlists/rockyou.txt
john z2.shadow --show
```

```
u1:google:16026:0:99999:7:::
u2:onelove:16026:0:99999:7:::

2 password hashes cracked, 0 left
```

1.20 Polecenie:

```
sudo john --make-charset=charset.chr
sudo gedit /etc/john/john.conf
```

```
[Incremental:z6shadow]
File = $JOHN/utf8.chr
MinLen = 5
MaxLen = 5
CharCount = 196

john --incremental:z6shadow z6.shadow
john z6.shadow --show
```

Hasło:

1.21 Polecenie:

```
sudo gedit /etc/john/john.conf
```

```
[List.Rules:z7shadow]
^[#]sa@se3si1$[#]
```

```
john --wordlist=/usr/share/wordlists/rockyou.txt z7.shadow --rules=z7shadow
john z7.shadow --show
```

Hasło to:

```
#p1n3@ppl3#
```

1.22 Polecenia:

```
cp z2.shadow z8.shadow
```

Pozostawienie w pliku jedynie hasha

```
hashcat -m 1800 -a 0 z8.shadow /usr/share/wordlists/rockyou.txt
```

```
hashcat -m 1800 -a 0 z8.shadow /usr/share/wordlists/rockyou.txt --show
$6$jTfZyjJr$xzqXw3CFldUMA3JESiGMyE2N2jr9YEO62otJsiwLSWn9yWc/n0J0UszKzia/3IFnPh6c7ZSUaahgnRP/cuAYJ.:google
$6$pmqtr7vg$j3NPrwFohrNYY3VTTVAlYdWja.pNnrce7nNbP.Uiq8WksCUkfFFtRJ3udehVjk8rVpanXxYFmlHHMzouP2Iyv.:onelove
```

1.23 Polecenie:

```
cp z5.shadow z9.shadow
```

Pozostawienie w pliku jedynie hasha

```
hashcat -m 1800 -a 3 z9.shadow -i --increment-min=3 --increment-max=3
hashcat -m 1800 -a 3 z9.shadow -i --increment-min=3 --increment-max=3 --show
```

Hasło:

```
$6$3tVyi5Or$TvxtOe7bNTtJE7QmYSWC7HTLlxxhaOXHgDiOfRceOBnsFppOCue/zkz21gO7wPEUimLCEhd33oWF7HD4JUns21:123
$6$FPMAFD62$GhrAXEUS359cBy3bh0.uY6VKqZx/Byx91M9wyFPLMYMTltSbfT9WU6sFtjUHMl8rTfijWeSLplFDkyY6YY9WE.:dhw
```

1.24 Polecenie:

```
cp z6.shadow z10.shadow
```

Pozostawienie w pliku jedynie hasha

```
hashcat -m 1800 -a 3 z10.shadow -i --increment-min=5 --increment-max=5
hashcat -m 1800 -a 3 z10.shadow -i --increment-min=5 --increment-max=5 --show
```

1.25 Polecenie:

```
cp z7.shadow z11.shadow
gedit z11.shadow
```

Pozostawienie w pliku tylko hashy

```
nano rules.txt
^# sa@ se3 si1
hashcat -m 1800 -a 0 z11.shadow /usr/share/wordlists/rockyou.txt -r rules.txt
hashcat -m 1800 -a 0 z11.shadow /usr/share/wordlists/rockyou.txt -r rules.txt --show
```

1.26 Skrypt:

```
#!/bin/env/python
import sys
import hashlib

h = hashlib.md5()
h.update(sys.argv[1].encode("utf-8"))
h.digest()
print(h.hexdigest())
```

1.27 Skrypt:

```
#!/bin/env python
import sys
import hashlib

fname = sys.argv[1]

with open(fname) as f:
    lines = f.readlines()
```

```
fcontent = " ".join(lines)

h = hashlib.sha1()
h.update(fcontent.encode("utf-8"))

print(h.hexdigest())
```

1.28 Skrypt:

```
#!/bin/env/python
import hashlib

# algs = hashlib.algorithms_available
# print(algs)

cleartxt = "R3iSrSNmgU9SFHxVekUD".encode("utf-8")
hashtxt = "48cab4b54bef42fddaa6353c68a20b369f40026e"

for alg in hashlib.algorithms_available :
        try:
                h = hashlib.new(alg)
                h.update(cleartxt)
                if h.hexdigest() == hashtxt :
                        print(alg)
        except:
                pass
```

1.29 Wygenerowanie hasha pliku i porównanie z hashem na stronie:
`openssl dgst -md5 gparted-live-1.3.1-1-amd64.iso`

1.30 Polecenie:

`md5sum a.txt`, `md5sum b.txt`

W pliku `a.txt` za pomocą steganografii został ukryty inny tekst, dlatego nie są one takie same, chociaż na pierwszy rzut oka (`cat a.txt b.txt`) tak wyglądają:

```
stegsnow -C -m "UMCS 2021"b.txt a.txt
stegsnow -C a.txt
```