

POLITECHNIKA WROCŁAWSKA  Wydział Informatyki i Telekomunikacji	Wydział: Informatyki i Telekomunikacji Kierunek: Cyber bezpieczeństwo Rok Akademicki: 2024/2025 Rok studiów, semestr: 2, 4 Grupa: 1 Termin: pon., 7:30
--	---

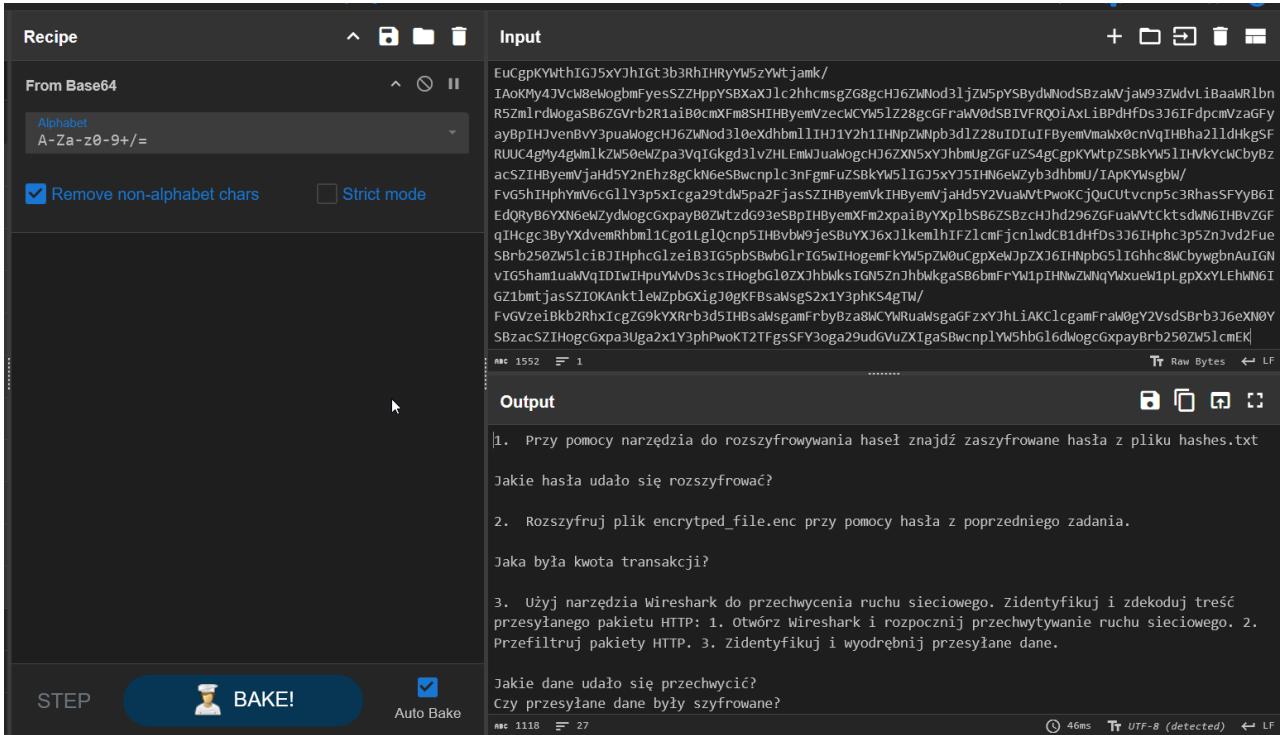
CBESI0053G Informatyka śledcza – Laboratorium 3

Prowadzący: mgr inż. Adrian Florek	Autor: 1. Gerard Błaszczyk
Data wykonania ćwiczenia: 17.03.2025	
Data oddania sprawozdania: 23.03.2025	

1. Cel ćwiczenia

Szyfrowanie i deszyfrowanie plików oraz analiza ruchu sieciowego.

2. Odszyfrowanie instrukcji laboratoryjnej



Rysunek 1: Odszyfrowanie instrukcji za pomocą narzędzia CyberChef

Treść instrukcji:

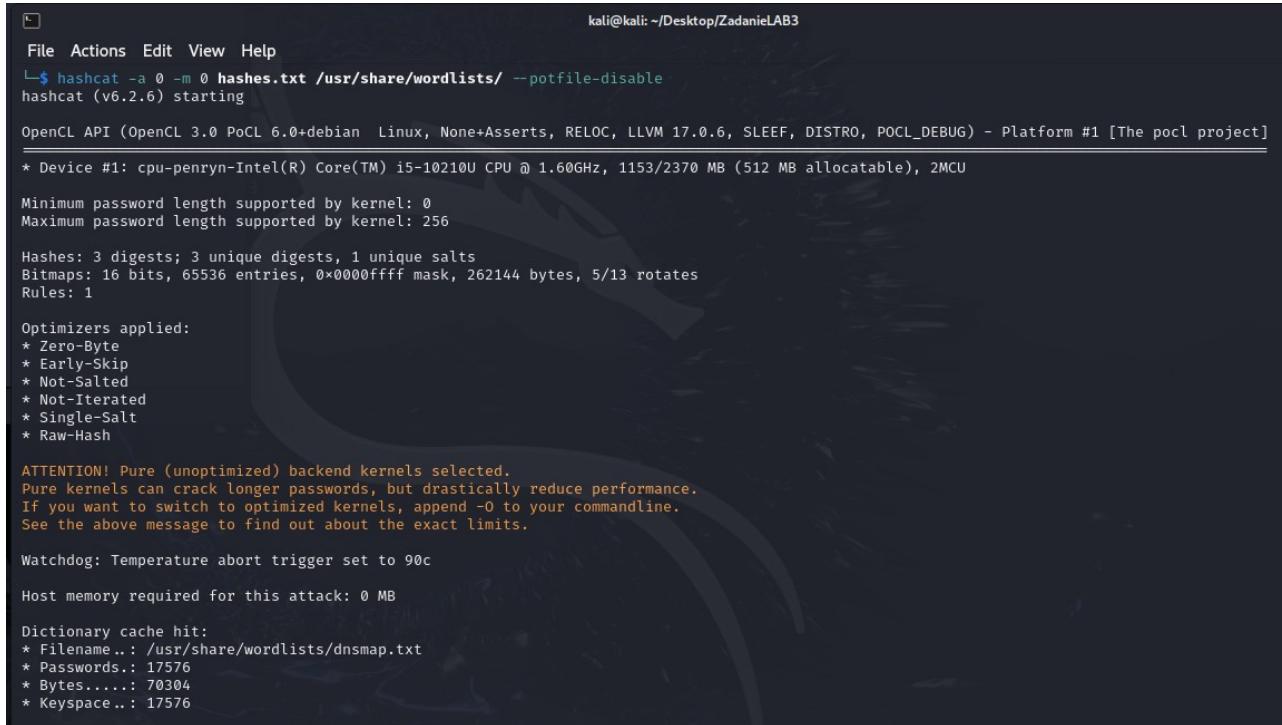
- Przy pomocy narzędzia do rozszyfrowywania haseł znajdź zaszyfrowane hasła z pliku hashes.txt
Jakie hasła udało się rozszyfrować?
- Rozszyfruj plik encrytped_file.enc przy pomocy hasła z poprzedniego zadania.
Jaka była kwota transakcji?
- Użyj narzędzia Wireshark do przechwycenia ruchu sieciowego. Zidentyfikuj i zdekoduj treść przesyłanego pakietu HTTP: 1. Otwórz Wireshark i rozpocznij przechwytywanie ruchu sieciowego. 2. Przefiltruj pakiety HTTP. 3. Zidentyfikuj i wyodrębnij przesyłane dane.
Jakie dane udało się przechwycić?
Czy przesyłane dane były szyfrowane?
Jak można zabezpieczyć komunikację przed przechwyceniem?
- Korzystając z GPG zaszyfruj plik tekstowy i prześlij razem ze sprawozdaniem
Klucz podaj w sprawozdaniu
- Przy pomocy narzędzia Veracrypt utwórz zaszyfrowany kontener I zapisz w nim plik np z zadaniem.
Wybierz silne hasło, np. co najmniej 20 znaków, z literami, cyframi i znakami specjalnymi. Włącz funkcję „Keyfile” (Plik Klucza). Możesz dodać dodatkowy plik jako składnik hasła.
W jakim celu korzysta się z pliku klucza?
Odłącz kontener i przeanalizuj plik kontenera

Rysunek 2: odkodowana treść instrukcji

3. Realizacja zadań

Zadanie 1. Przy pomocy narzędzia do rozszyfrowywania haseł znajdź zaszyfrowane hasła z pliku hashes.txt

Hipoteza: hashe w pliku hashes.txt mają po 32 znaki, więc najprawdopodobniej są hashami MD5.



The screenshot shows a terminal window on a Kali Linux desktop. The title bar says "kali@kali: ~/Desktop/ZadanieLAB3". The command entered is "hashcat -a 0 -m 0 hashes.txt /usr/share/wordlists/ --potfile-disable". The output shows the hashcat version (v6.2.6), system information (OpenCL API, CPU details), password length constraints (0-256), and configuration details (digests, salts, rules, optimizers). It also includes a warning about pure kernels and a note about watchdog settings. The dictionary cache hit section lists file name, passwords, bytes, and keyspace size.

```
kali@kali: ~/Desktop/ZadanieLAB3
File Actions Edit View Help
$ hashcat -a 0 -m 0 hashes.txt /usr/share/wordlists/ --potfile-disable
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEP, DISTRO, POCL_DEBUG) - Platform #1 [The pool project]
* Device #1: cpu-penryn-Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz, 1153/2370 MB (512 MB allocatable), 2MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 3 digests; 3 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Salt
* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 0 MB

Dictionary cache hit:
* Filename..: /usr/share/wordlists/dnsmap.txt
* Passwords.: 17576
* Bytes.....: 70304
* Keyspace ..: 17576
```

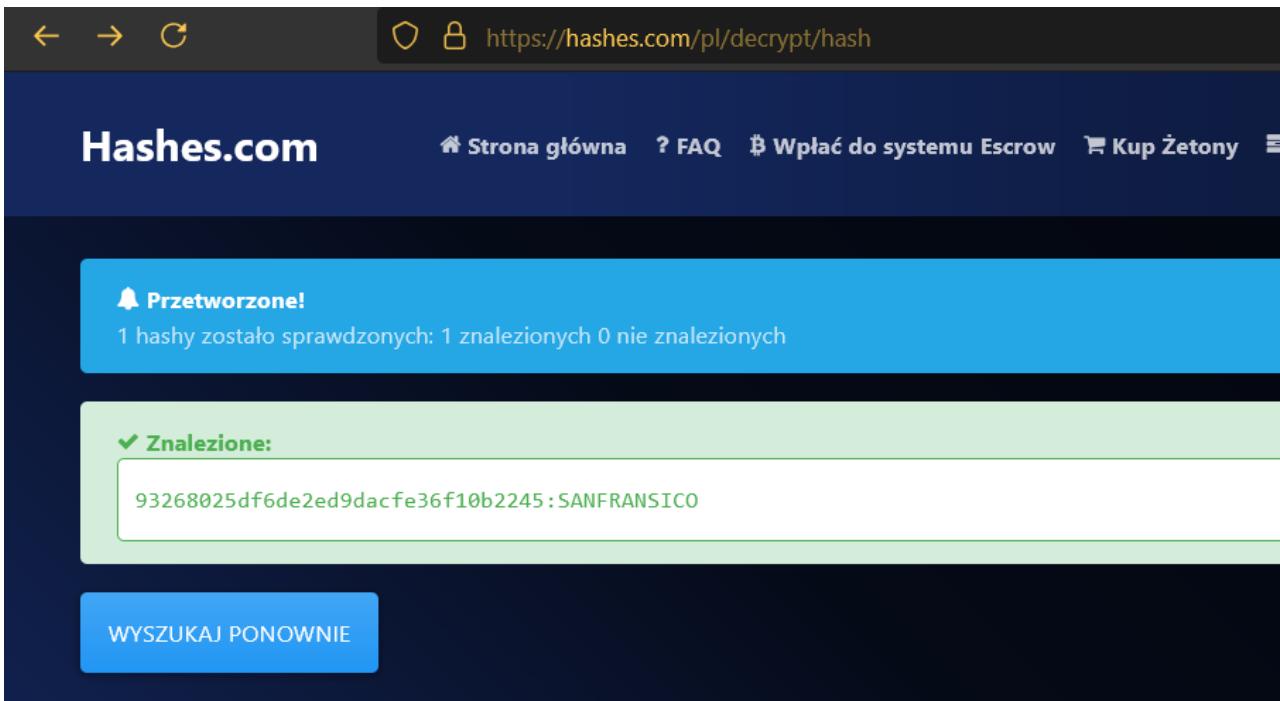
Rysunek 3: Atak słownikowy na plik hashes.txt

Atak słownikowy złamał dwa z trzech hashy w pliku:

79d8e340812f9db0bbfa508beb319dea:holiday

5cca9a53ece2fd6b499e035388db4171:rainb0ws

Ostatni hash został odkryty za pomocą narzędzia <https://hashes.com/pl/decrypt/hash>



Rysunek 4: Przeszukanie bazy hashy na hashes.com

Jakie hasła udało się rozszyfrować? – Udało się rozszyfrować hasła „holiday”, „rainb0ws”, „SANFRANCISCO”. Są to niebezpieczne hasła ponieważ są podatne na proste ataki słownikowe.

Analiza pliku przed odszyfrowaniem:

```
(kali㉿kali)-[~/Desktop]
$ exiftool ZadanieLAB3.zip
ExifTool Version Number : 13.00
File Name               : ZadanieLAB3.zip
Directory              :
File Size                : 598 bytes
File Modification Date/Time : 2025:03:17 03:38:28-04:00
File Access Date/Time   : 2025:03:17 03:38:58-04:00
File Inode Change Date/Time : 2025:03:17 03:38:29-04:00
File Permissions        : -rw-rw-rw-
File Type                : ZIP
File Type Extension     : zip
MIME Type                : application/zip
Zip Required Version    : 10
Zip Bit Flag             : 0
Zip Compression          : None
Zip Modify Date          : 2025:03:17 07:26:30
Zip CRC                  : 0xb5f0b4c2
Zip Compressed Size      : 224
Zip Uncompressed Size     : 224
Zip File Name            : encrypted_file.enc
Warning                 : [minor] Use the Duplicates option to extract tags for all 2 files

(kali㉿kali)-[~/Desktop]
$ binwalk ZadanieLAB3.zip
DECIMAL      HEXADECIMAL      DESCRIPTION
---          ---           ---
0            0x0           Zip archive data, at least v1.0 to extract, compressed size: 224, uncompressed size: 224, name: encrypted_file.enc
272          0x110          Zip archive data, at least v2.0 to extract, compressed size: 72, uncompressed size: 99, name: hashes.txt
576          0x240          End of Zip archive, footer length: 22

(kali㉿kali)-[~/Desktop]
$ file ZadanieLAB3.zip
ZadanieLAB3.zip: Zip archive data, at least v1.0 to extract, compression method=store

(kali㉿kali)-[~/Desktop]
$
```

Rysunek 5: Analiza archiwum

Komentarz: Plik to archiwum zip zawierający pliki encrypted_file.enc oraz hashes.txt

Zadanie 2. Rozszyfruj plik encrypted_file.enc przy pomocy hasła z poprzedniego zadania.

```

kali@kali: ~/Desktop/ZadanieLAB3

File Actions Edit View Help
└─(kali㉿kali)-[~/Desktop/ZadanieLAB3]
$ openssl enc -d -aes-256-cbc -in encrypted_file.enc -out out.txt -pass pass:holiday
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

└─(kali㉿kali)-[~/Desktop/ZadanieLAB3]
$ cat out.txt
ID transakcji: 20250315-001
Kwota: 15000 USD
Data: 2025-03-15
Numer konta nadawcy: 1234-5678-9101-1121
Numer konta odbiorcy: 9876-5432-1098-7654
Kod SWIFT: ABCDUS33
Status: WYKONANO

└─(kali㉿kali)-[~/Desktop/ZadanieLAB3]
$ 

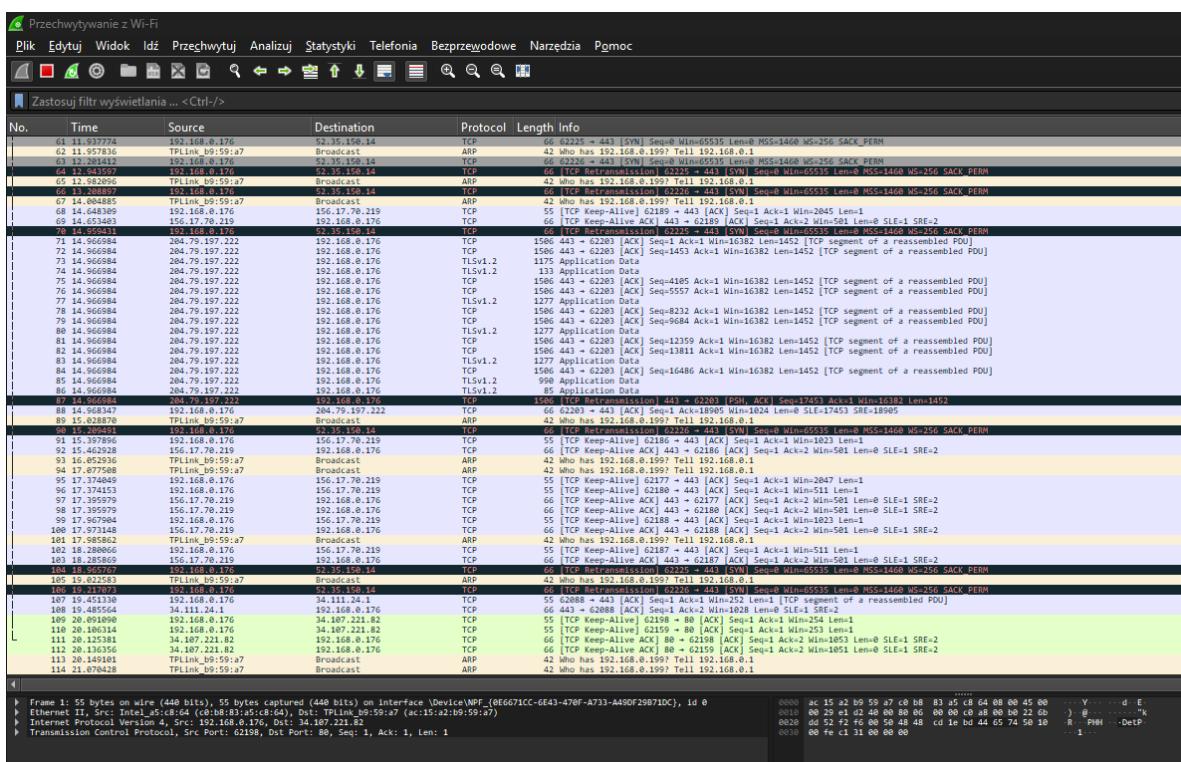
```

Rysunek 6: Rozszyfrowanie pliku encrypted_file.enc

Jaka była kwota transakcji? – 15000 USD

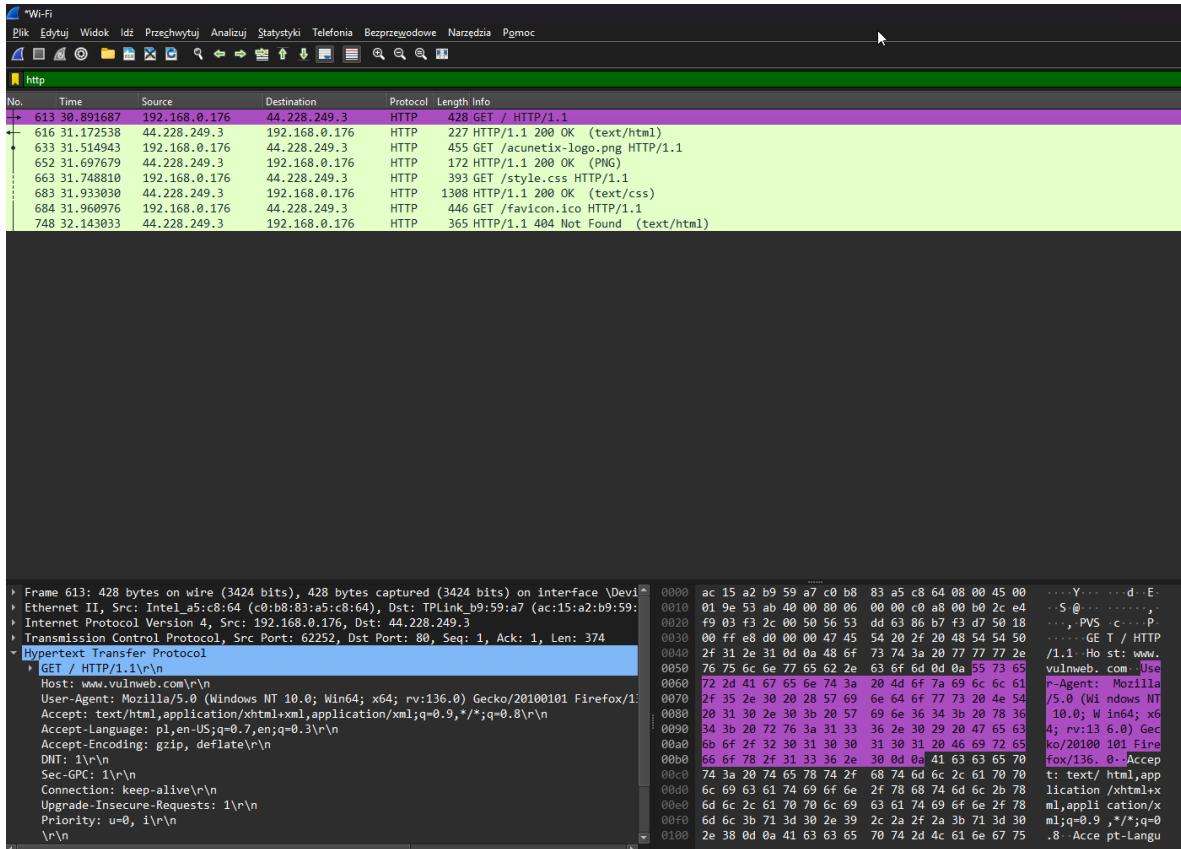
Zadanie 3. Użyj narzędzia Wireshark do przechwytycia ruchu sieciowego. Zidentyfikuj i zdekoduj treść przesyłanego pakietu HTTP:

1. Otwórz Wireshark i rozpoczęź przechwytywanie ruchu sieciowego.



Rysunek 7: Rozpoczęcie przechwytywania ruchu sieciowego w programie Wireshark

2. Przefiltruj pakiety HTTP.

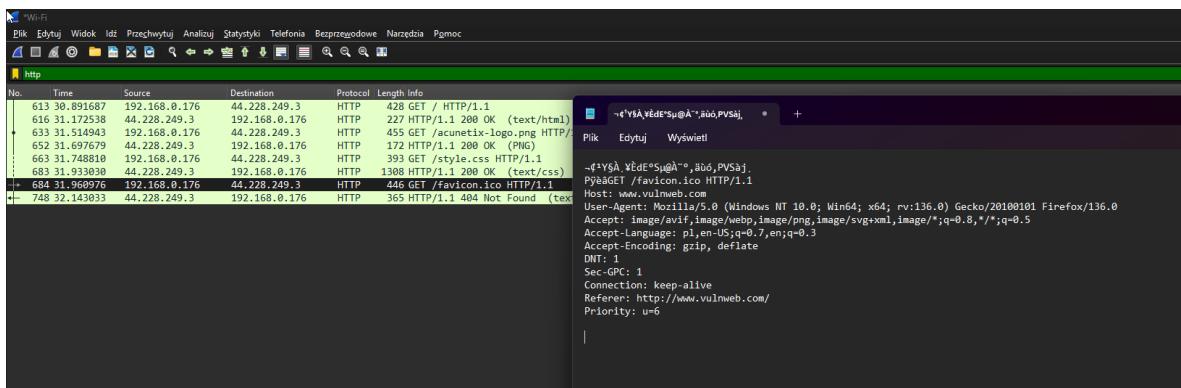


Rysunek 8: Filtrowanie pakietów http

Komentarz: Połączenie ze stroną <http://www.vulnweb.com/>.

3. Zidentyfikuj i wyodrębnij przesyłane dane.

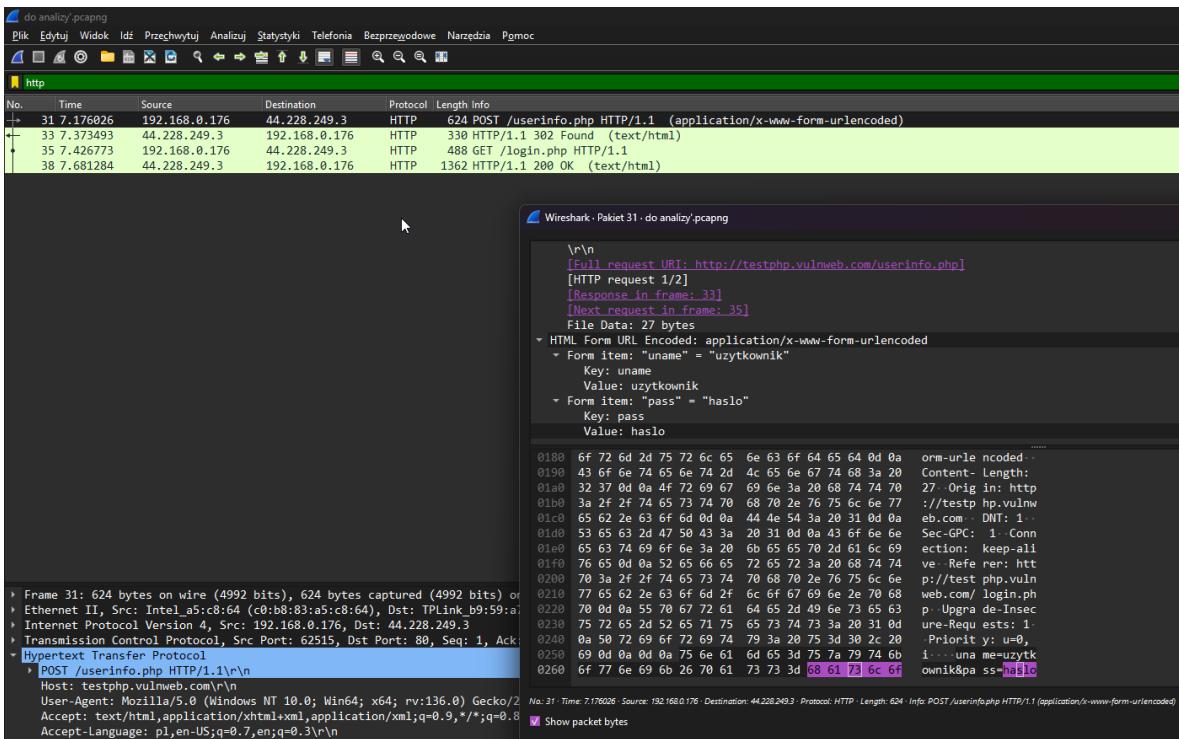
a. Strona bez logowania



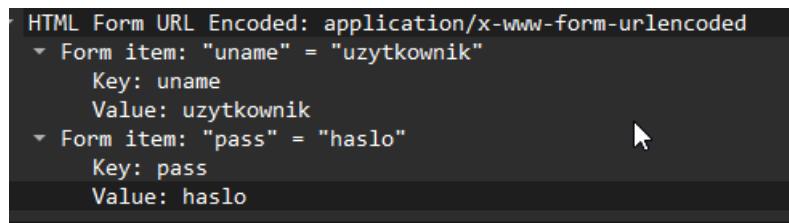
Rysunek 9: Wyodrębnienie danych z pakietu

Komentarz: W niezaszyfrowanym pakiecie HTTP można wyczytać szczegółowe informację o użytkowniku: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0 czy też typ żądania: GET /favicon.ico

b. Strona z logowaniem



Rysunek 10: Analiza pakietów przesyłanych z/do strony z logowaniem



Rysunek 11: Zbliżenie rys. 9

Komentarz: W protokole HTTP nawet dane logowania nie są w żaden sposób chronione. Są dostępne w tekście jawnym jak widać na rys 9.

Jakie dane udało się przechwycić? - Np. dane logowania.

Czy przesyłane dane były szyfrowane? – Nie.

Jak można zabezpieczyć komunikację przed przechwytem? – Korzystać z protokołu HTTPS.

Zadanie 4. Korzystając z GPG zaszyfruj plik tekstowy i prześlij razem ze sprawozdaniem. Klucz podaj w sprawozdaniu

```
(kali㉿kali)-[~/Desktop/lab_3]
└─$ gpg -r Gerard --sign zadania.txt
```

Rysunek 12: Szyfrowanie za pomocą klucza prywatnego (podpisanie)

```
(kali㉿kali)-[~/Desktop/lab_3]
$ gpg -a --export 279460@student.pwr.edu.pl > pub.key

(kali㉿kali)-[~/Desktop/lab_3]
$ cat pub.key
-----BEGIN PGP PUBLIC KEY BLOCK-----

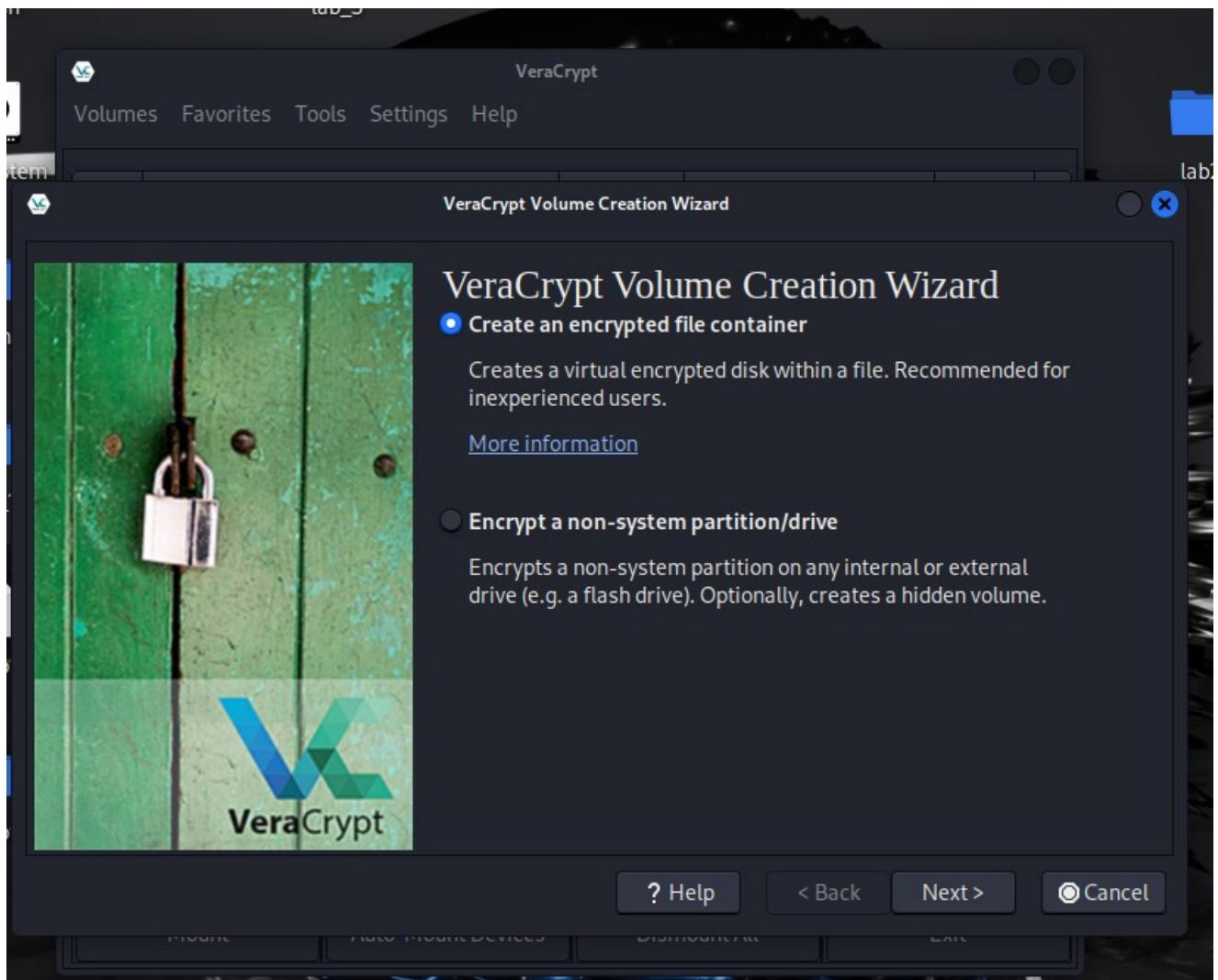
mQGNBGFQB6IBDACpygRpKeli6IXJwcot8kgCQlEJ3I5FdZd2SOVObNmyQsKM5np
3ne7PVcLeF3Zk+aa/ZFn3PHjIDc1SV84pLETFYTcmmapupJFEni9PCsrLvdcs5S2r/
a06h2bvIKV75MFb6k0qT1aGgCp7glRSut9RM3rwtCpJZco9NyX+uvj1vunn0IT8m
uixQciLZPMdYCd+QL+orj/9viub3HevA5p7Dr2hJqYBEbDxGGn6Ntm3CvltgYcLb
Uy3y+//6gq4Wq0z60VJmrWVTLbz3CHcRAcSBTLis8g82578csFV1RtiDIWMZicDd
3KWPcJsI/9uZqZ1N0wp2jle6woTX0qqPELy3E2obT0L5m++HZukPRCjYsL221vU5
UKOSD1i70E35GxlKxLHozw6Tv8kcm0z062ELIxwyW1F1H1m+afdBPyTmCmnMvb8zu
V22i7o5Pb1Xabg2B+u+5SE07akCs-iEWLAMM0C+1PTUWpbJaCgI/H5dEWivvQ
```

Rysunek 13: Eksport klucza publicznego

Komentarz: Klucz dołączony do sprawozdania

Zadanie 5. Przy pomocy narzędzia Veracrypt utwórz zaszyfrowany kontener i zapisz w nim plik np z zadaniem.

Wybierz silne hasło, np. co najmniej 20 znaków, z literami, cyframi i znakami specjalnymi.
Włącz funkcję „Keyfile” (Plik Klucza). Możesz dodać dodatkowy plik jako składnik hasła.
W jakim celu korzysta się z pliku klucza?
Odłącz kontener i przeanalizuj plik kontenera



Rysunek 14: Tworzenie kontenera



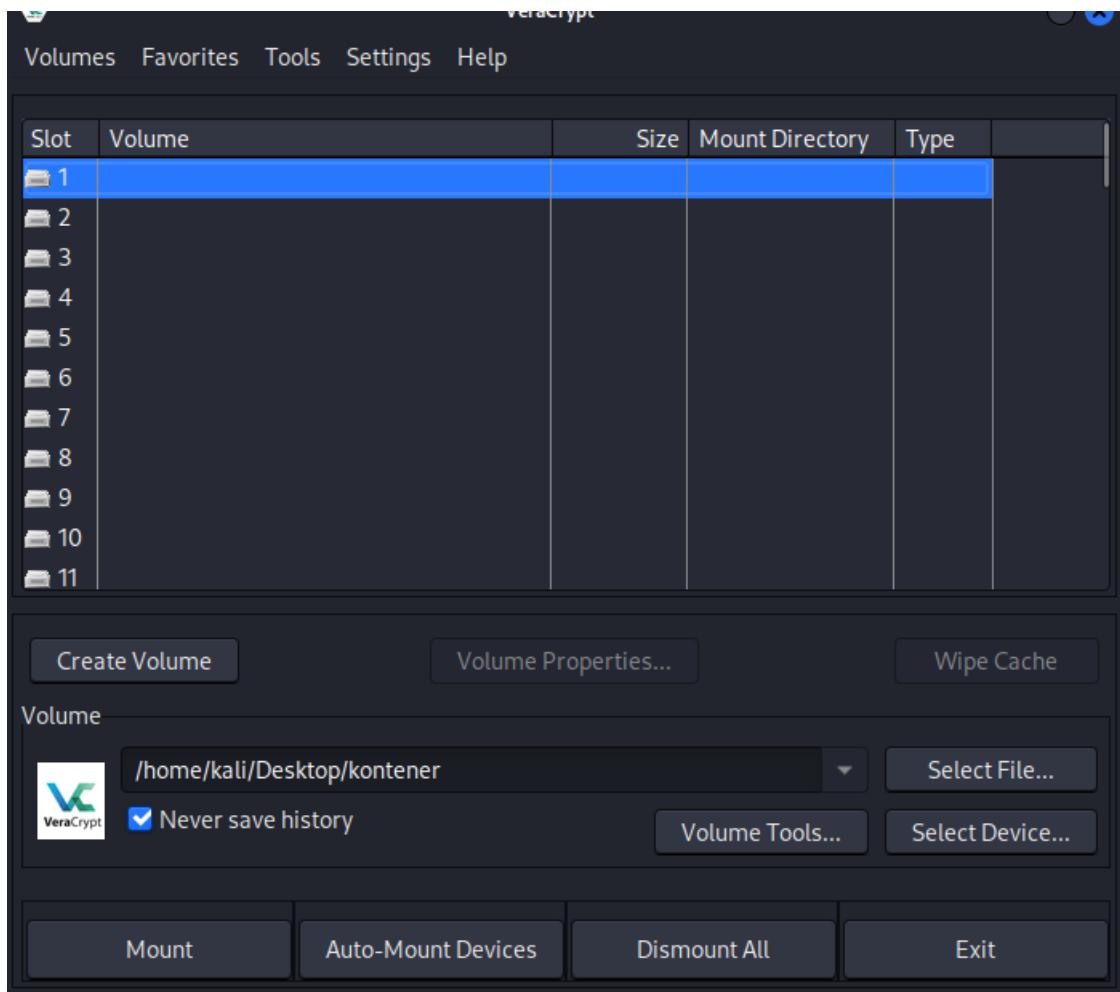
Rysunek 15: Ustawianie silnego hasła

Dodatkowo została zaznaczona opcja use keyfiles gdzie wybrany został plik tekstowy z pulpitu.

Po utworzeniu kontenera utworzył się jego plik, który przy użyciu VeraCrypt został zamontowany na pulpicie. Wymagało to podania hasła oraz odpowiedniego pliku tekstowego.

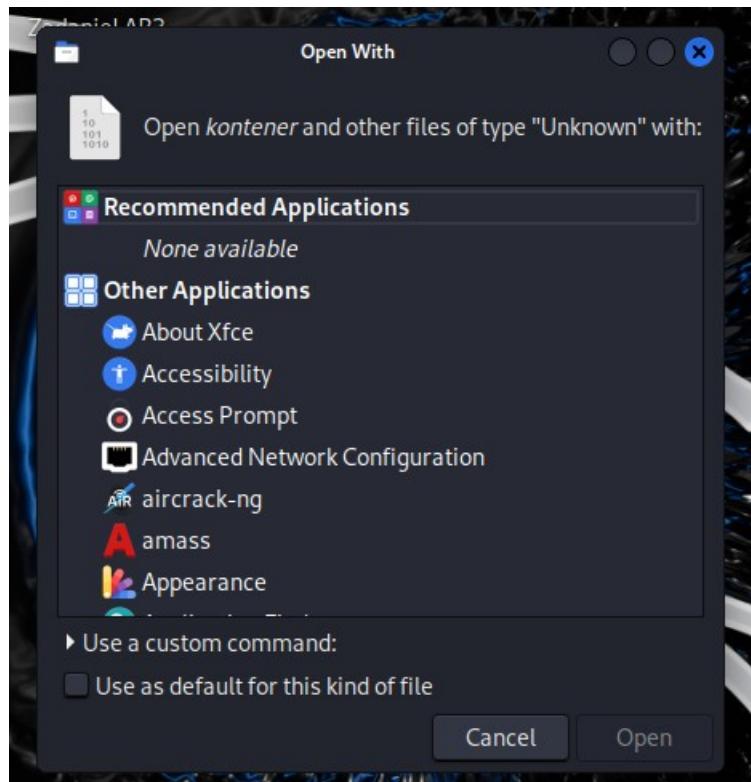
A terminal window showing a session on a Kali Linux system. The prompt is "(kali㉿kali)-[~/media/veracrypt1]". The user runs the command "cp /home/kali/Desktop/zadanie.txt /media/veracrypt1". Then they run "ls" to list the contents of the mounted volume, which shows a file named "zadanie.txt". Finally, they run "history" to show the command history.

Rysunek 16: zapisanie pliku do kontenera



Rysunek 17: Odmontowanie kontenera

Po odmontowaniu kontenera pozostaje tylko jego plik, niezrozumiały dla zwykłego systemu operacyjnego. Dostęp do plików zapisanych na kontenerze byłby możliwy tylko dzięki ponownym podaniu hasła i odpowiednich plików w aplikacji VeraCrypt.



Rysunek 18: plik kontenera

```
(kali㉿kali)-[~/Desktop] $ file kontener
kontener: data

(kali㉿kali)-[~/Desktop] $ exiftool kontener
ExifTool Version Number      : 13.00
File Name                   : kontener
Directory                  :
File Size                   : 52 MB
File Modification Date/Time : 2025:03:18 08:27:57-04:00
File Access Date/Time       : 2025:03:18 08:42:22-04:00
File Inode Change Date/Time : 2025:03:18 08:34:05-04:00
File Permissions            : -rw-
Error                      : Unknown file type

(kali㉿kali)-[~/Desktop] $ binwalk kontener
DECIMAL      HEXADECIMAL      DESCRIPTION
-----      -----      -----
4889100      0x4A9A0C      JBOOT STAG header, image id: 0, timestamp 0x80ACC879, image size: 3583886024 bytes, image
0T checksum: 0x935D
25407438     0x183AFCE     JBOOT STAG header, image id: 10, timestamp 0xF32CBB14, image size: 1583526662 bytes, image
0OT checksum: 0x6BDD
28776739     0x1B71923     PGP RSA encrypted session key - keyid: 6CF5F3F2 A10331B1 RSA (Encrypt or Sign) 1024b
40856892     0x26F6D3C     JBOOT STAG header, image id: 7, timestamp 0x3ADA91AF, image size: 253462665 bytes, image
T checksum: 0x5EEA

(kali㉿kali)-[~/Desktop] $
```

Rysunek 19: analiza pliku kontenera

W jakim celu korzysta się z pliku klucza? – zwiększenie bezpieczeństwa poprzez wymaganie dodatkowego pliku poza hasłem.

4. Opis wykonanej pracy przez autora