
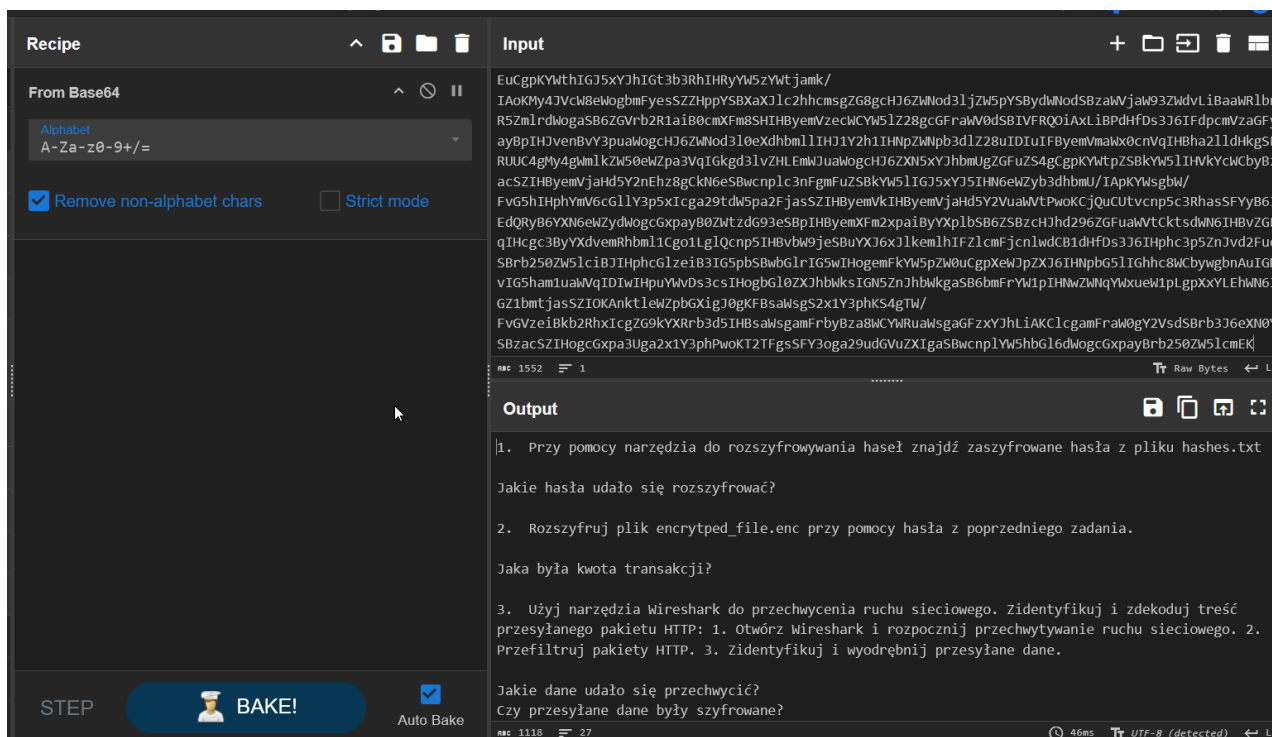


<b>POLITECHNIKA WROCŁAWSKA</b>  Wydział Informatyki i Telekomunikacji	Wydział: Informatyki i Telekomunikacji Kierunek: Cyberbezpieczeństwo Rok Akademicki: 2024/2025 Rok studiów, semestr: 2, 4 Grupa: 1 Termin: pon., 7:30
<b>CBESI0053G Informatyka śledcza – Laboratorium 3</b>	
Prowadzący: mgr inż. Adrian Florek	Autor: 1. Gerard Błaszczuk
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:

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

Jakie hasła udało się rozszyfrować?

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

Jaka była kwota transakcji?

3. 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?

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

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?

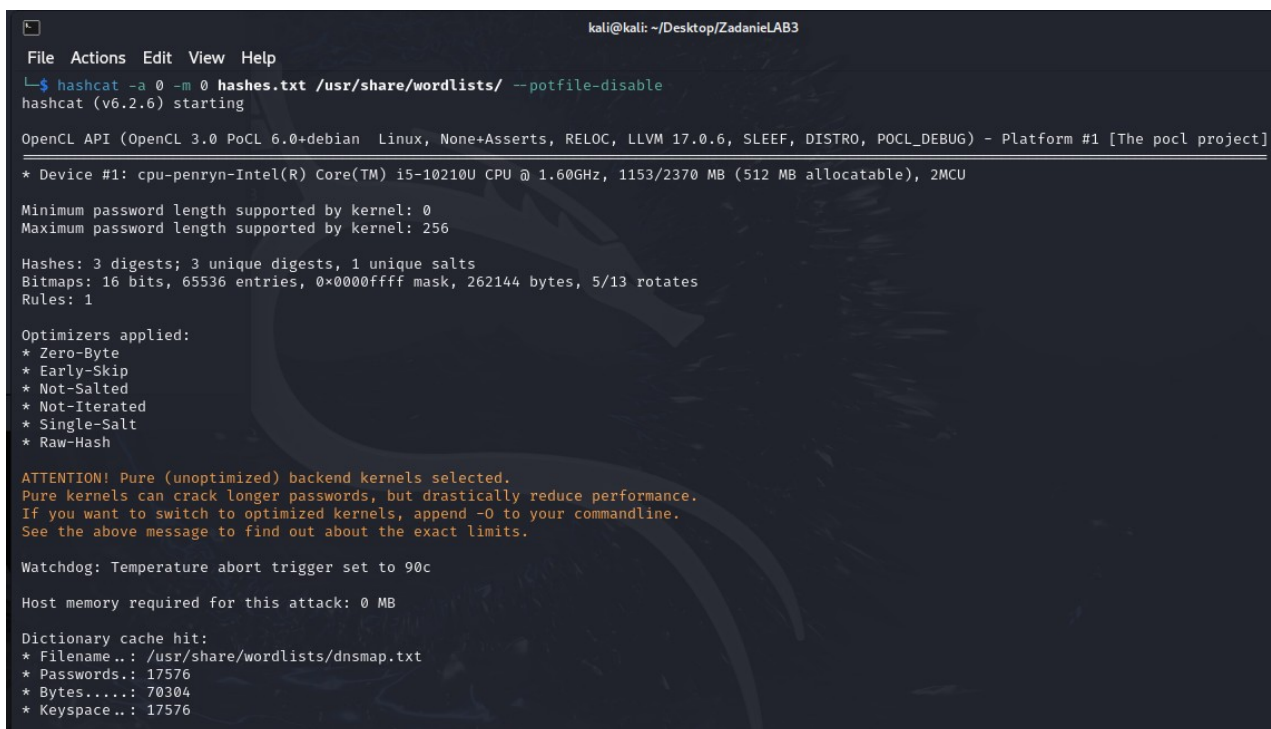
Odłącz kontener i przeanalizuj plik kontenera
```

Rysunek 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.



```
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, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl 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
```

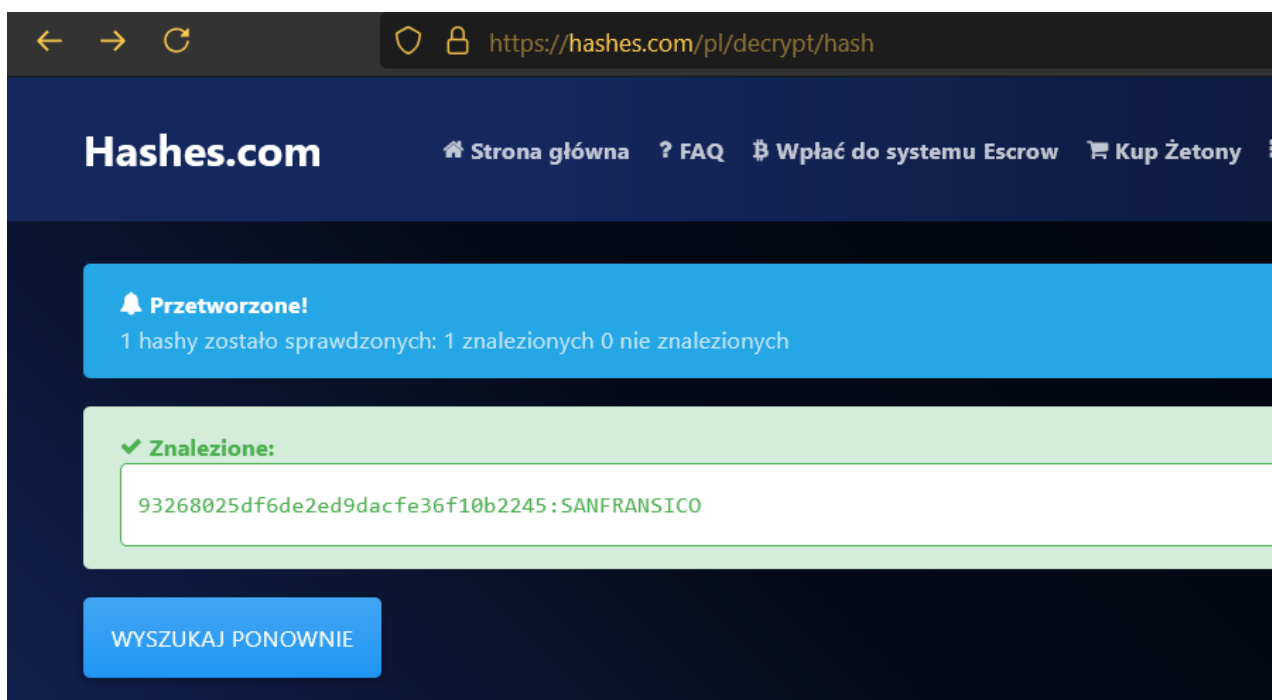
Rysunek 3: Atak słownikowy na plik hashes.txt

Atak słownikowy złamał dwa z trzech hashy w pliku:

79d8e340812f9db0bbfa508beb319dea:holiday

5cca9a53ece2fd6b499e035388db4171:rainb0ws

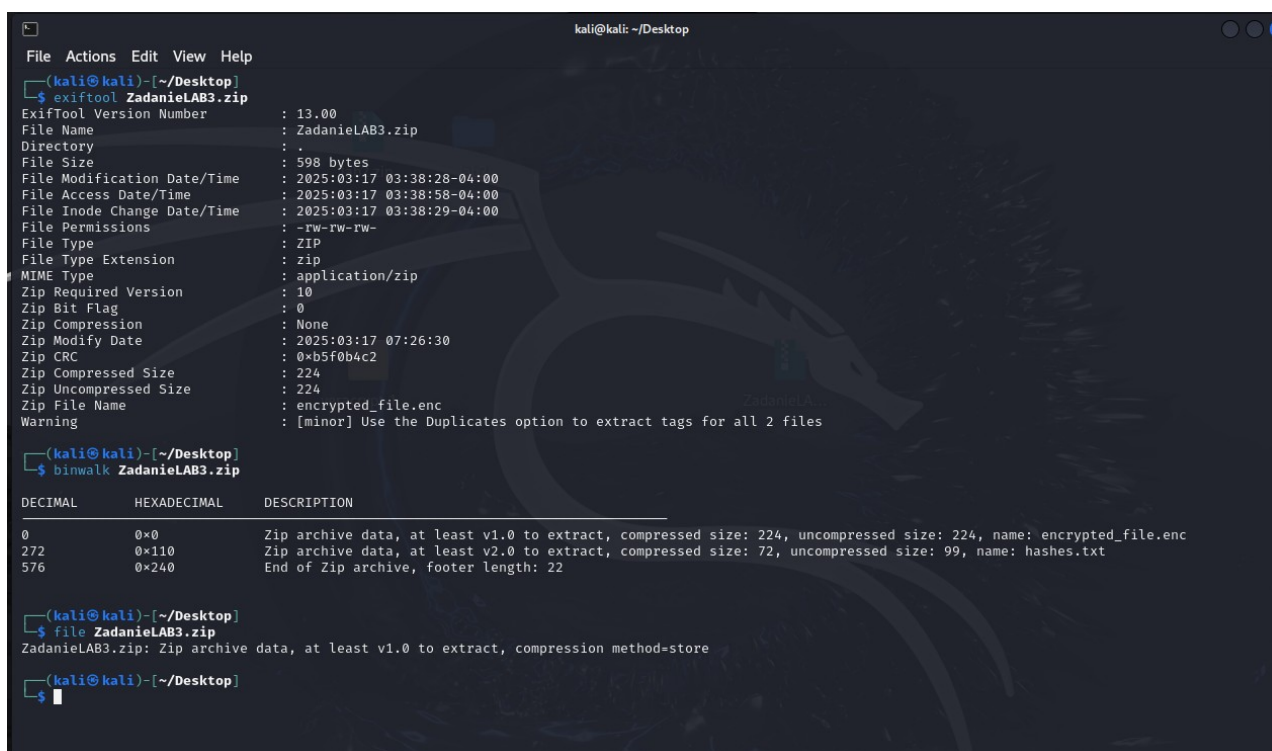
Ostatni hash został odkryty za pomocą narzędzia <https://hashes.com/pl/decrypt/hash>



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:



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

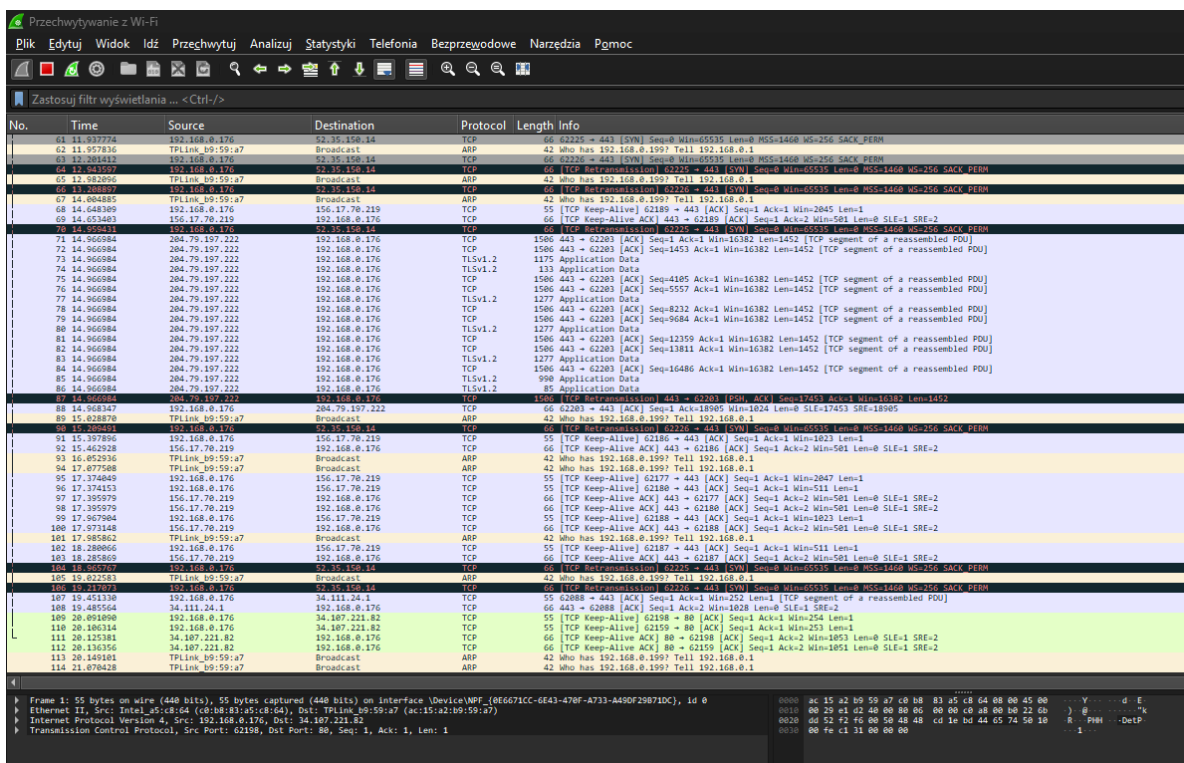
(kali@kali)-[~/Desktop/ZadanieLAB3]
$
```

Rysunek 6: Rozszyfrowanie pliku encrypted\_file.enc

Jaka była kwota transakcji? – 15000 USD

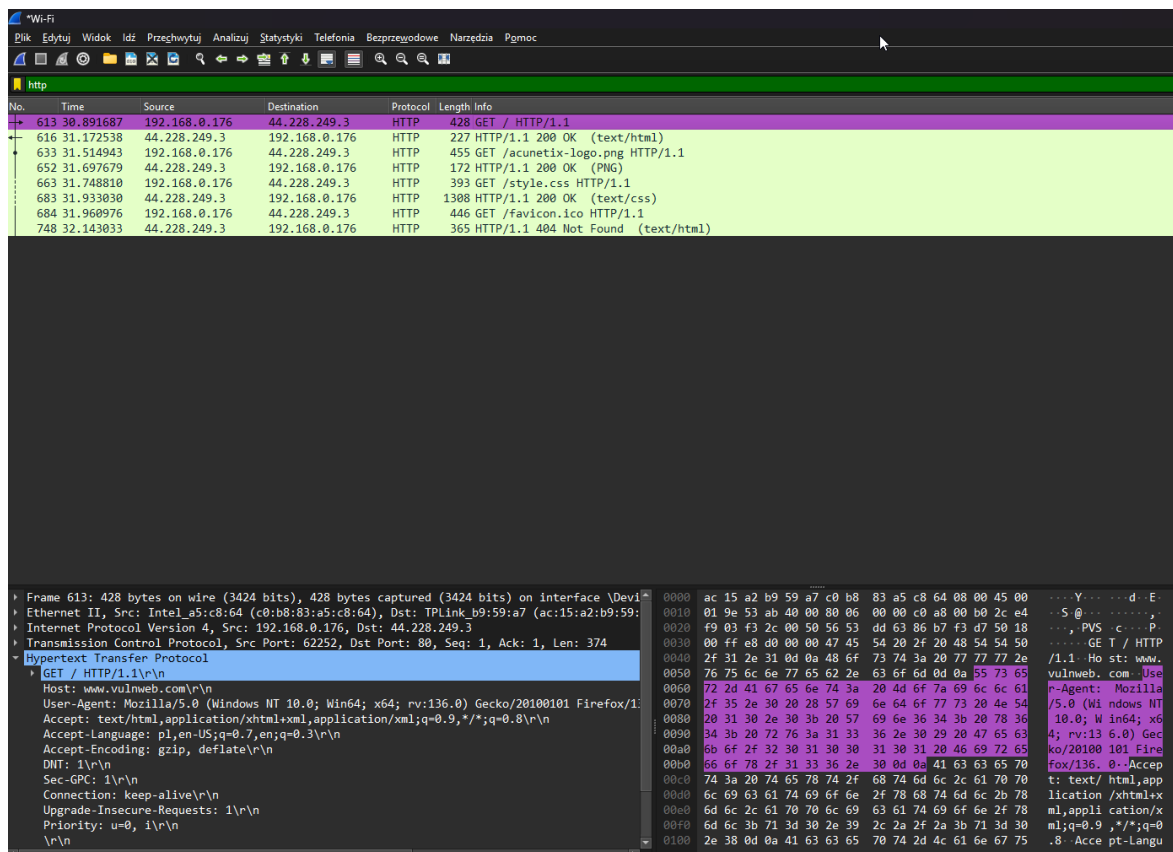
**Zadanie 3.** Użyj narzędzia Wireshark do przechwycenia ruchu sieciowego. Zidentyfikuj i zdekoduj treść przesyłanego pakietu HTTP:

1. Otwórz Wireshark i rozpocznij przechwytywanie ruchu sieciowego.



Rysunek 7: Rozpoczęcie przechwytywania ruchu sieciowego w programie Wireshark

## 2. Przefiltruj pakiety HTTP.

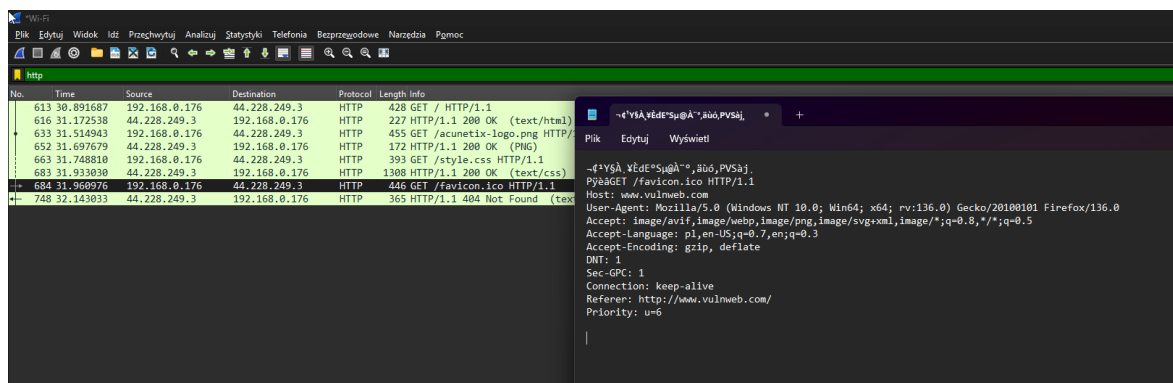


Rysunek 8: Filtrowanie pakietów http

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

## 3. Zidentyfikuj i wyodrębnij przesyłane dane.

### a. Strona bez logowania

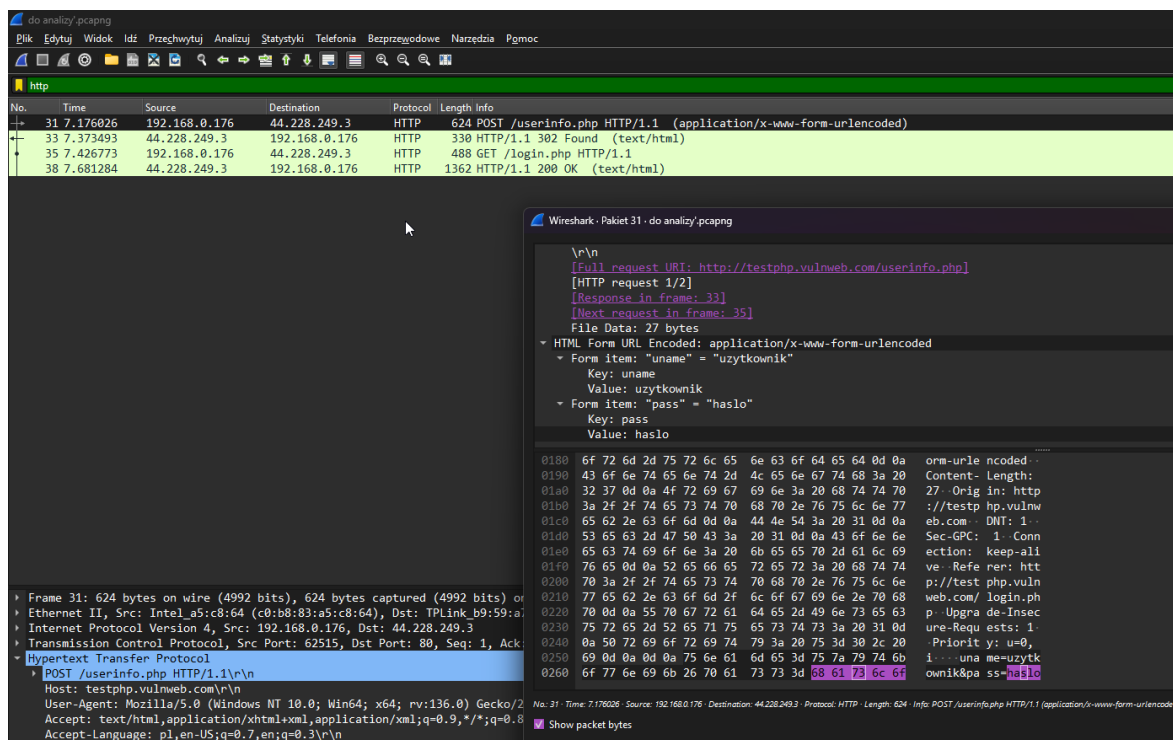


Rysunek 9: Wyodrębnienie danych z pakietu

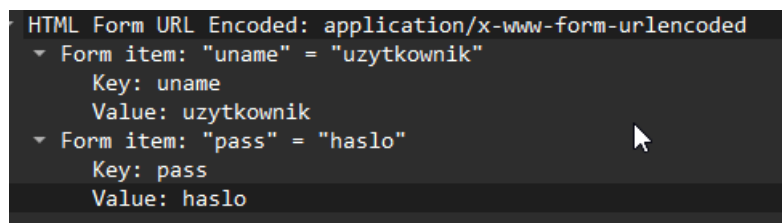
Komentarz: W niezaszyfrowanym pakiecie HTTP można wyczytać szczegółowe informacje o użytkowniku: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0 czy też typ żądania: GET /favicon.ico

### b. Strona z logowaniem





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



Rysunek 11: Zbliżenie rys. 9

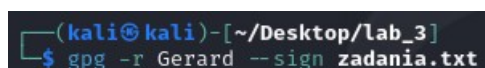
Komentarz: W protokole HTTP nawet dane logowania nie są w żaden sposób chronione. Są dostępne w tekście jawnym jak widać na rys 9.

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

Czy przesyłane dane były szyfrowane? – Nie.

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

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



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-----

mQGNBGfQB6IBDACpygRpKelI6IXJwcoT8kgCQlEJ3I5FdxZd2SOVObNmyQsKM5np
3ne7PVcLeF3Zk+aa/ZFn3PHjIDc1SV84p1ETfYTcmapupJFEni9PCsrLvdc5S2r/
a06h2bvIKV75MFB6k0qT1aGgCp7g1RSut9RM3rwtCpJZco9NyX+uvj1vunn0IT8m
uixQciLZPMdYCd+QL+orj/9viub3HevA5p7Dr2hJqYBEbDxGGn6Ntm3CvltgYcLb
Uy3y+//6gq4Wq0z60VJmrWVTLbz3CHcRacSBTLis8g82578csFV1RtiDIWMZicDd
3KWPcJsI/9uZqZ1N0wp2jIe6woTX0qqPELy3E2obT0L5m++HZukPRCjYsL221vU5
UKOSD1i70E35GxlKxLHozw6Tv8kcmOzO62ELIxywW1F1H1m+afdPyTmCmnMvb8zu
VQ22Tas30fP3jYasb32R7Hs55503akG6s3FVUAMW0Fv1DTHVPh3eGdL4U5d5W5V4Q
```

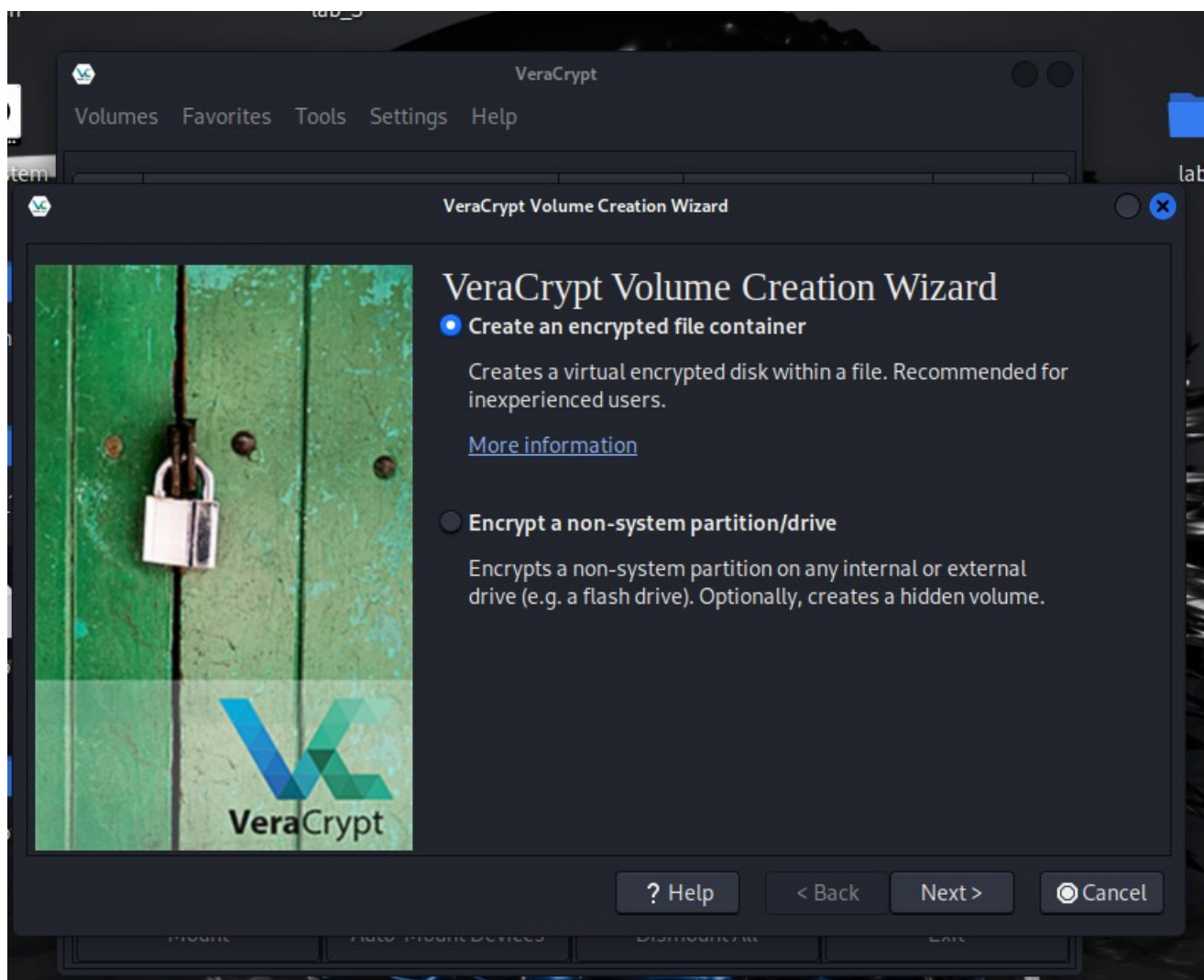
*Rysunek 13: Eksport klucza publicznego*

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.

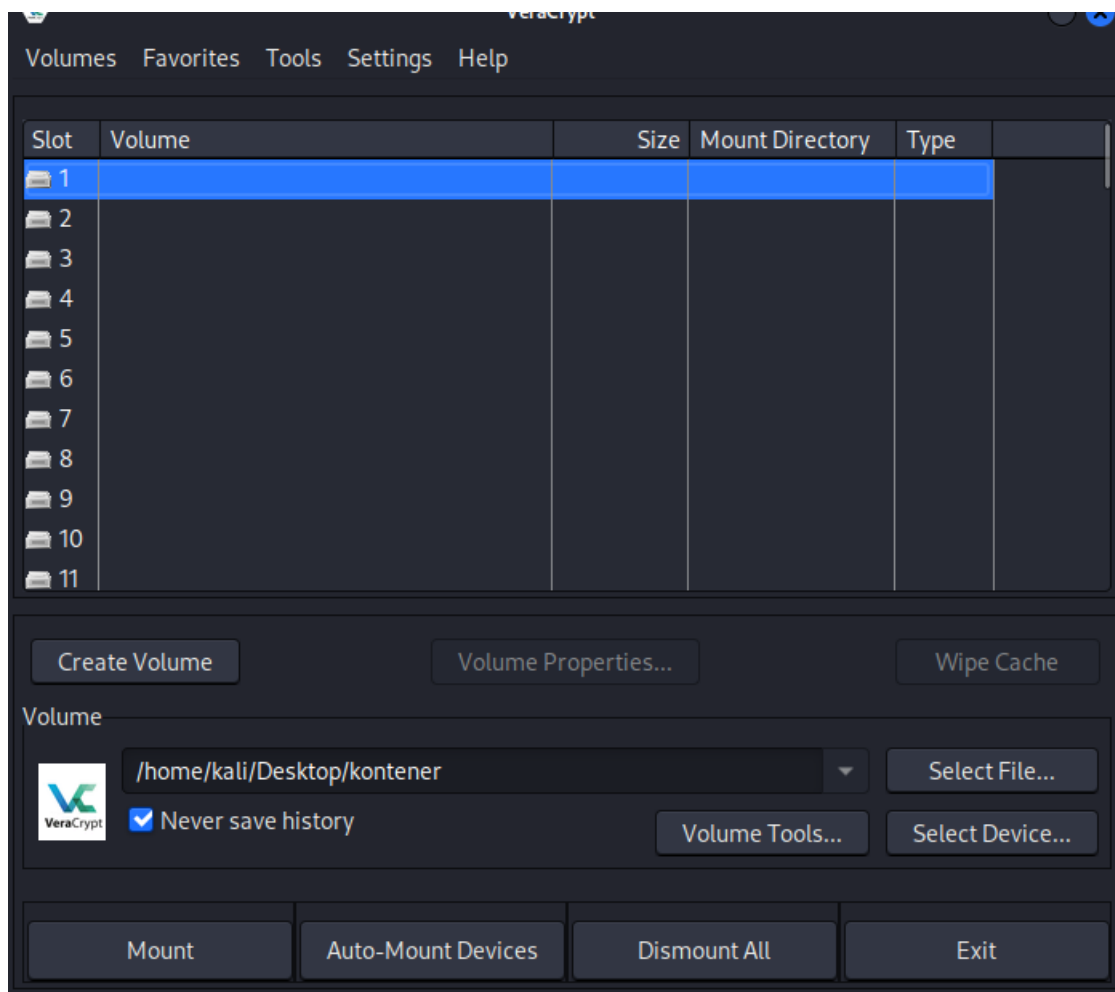
Po utworzeniu kontenera utworzył się jego plik, który przy użyciu VeraCrypt został zamontowany na pulpicie. Wymagało to podania hasła oraz odpowiedniego pliku tekstowego.

```
(kali@kali)-[/media/veracrypt1]
$ cp /home/kali/Desktop/zadanie.txt /media/veracrypt1

(kali@kali)-[/media/veracrypt1]
$ ls
zadanie.txt

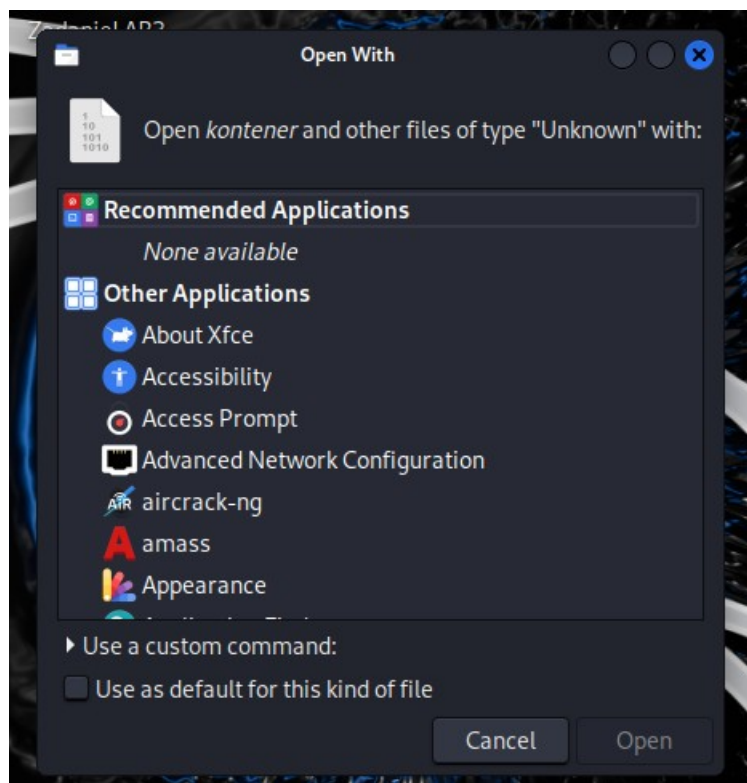
(kali@kali)-[/media/veracrypt1]
$
```

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

```

File Actions Edit View Help
(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       JB00T STAG header, image id: 0, timestamp 0x80ACC879, image size: 3583886024 bytes, image
OT checksum: 0x935D
25407438     0x183AFCE      JB00T STAG header, image id: 10, timestamp 0xF32CBB14, image size: 1583526662 bytes, imag
OOT checksum: 0x6BDD
28776739     0x1B71923      PGP RSA encrypted session key - keyid: 6CF5F3F2 A10331B1 RSA (Encrypt or Sign) 1024b
40856892     0x26F6D3C      JB00T STAG header, image id: 7, timestamp 0x3ADA91AF, image size: 253462665 bytes, image
T checksum: 0x5EEA

```

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**