

Simulácia útoku hrubou silou na šifrovaný súbor

Autori:

Andrej Pecník (257151)

Katarína Kúdelová (257128)

David Gregora (257104)

Popis projektu:

Projekt Simulácia útoku hrubou silou na šifrovaný súbor je napísaný v jazyku C a demonštruje šifrovanie súboru pomocou XOR kľúča. Následne program prevedie brute-force útok a pokúsi sa nájsť šifrovací kľúč na základe známej časti pôvodného textu.

Celý projekt je ošetrovaný jednotkovými testami, aby bola zabezpečená jeho správna funkcionálna a bez chybovostí.

Princíp XOR operácie:

XOR (exclusive OR) je logická operácia, ktorá vracia pravdu, pokiaľ sú dva vstupy rôzne:

$0 \text{ XOR } 0 = 0$ $1 \text{ XOR } 0 = 1$ $0 \text{ XOR } 1 = 1$ $1 \text{ XOR } 1 = 0$.

V projekte je využitá pri šifrovaní a dešifrovaní, kde sa dáta kombinujú s kľúčom. Ak je kľúč kratší ako vstup, používa sa cyklicky. Šifrovanie a dešifrovanie je realizované v súbore xor.c funkciami xor_encrypt() a xor_decrypt() – pričom dešifrovanie využíva rovnaký algoritmus ako šifrovanie.

Princíp Brute-force útoku:

Brute-force útok (útok hrubou silou) spočíva v systematickom skúšaní všetkých možných kombinácií znakov (z množiny a-z a 0-9) s dĺžkou od 3 do maximálnej povolenej dĺžky. Každý takto vygenerovaný kľúč je použitý na dešifrovanie šifrovaného súboru. Ak dešifrovaný text obsahuje preddefinovaný reťazec (napr. tajne), predpokladáme, že sme našli správny kľúč. Táto časť kódu zároveň eviduje počet pokusov a meria čas výpočtu. Tento spôsob je síce jednoduchý, ale časovo náročný.

Princíp jednotkových testov:

Jednotkové testy (unit tests) overujú správne chovanie jednotlivých funkcií programu na základe predom definovaných vstupov a očakávaných výstupov.

Vytvorené sú dva hlavné testy:

XOR Encrypt/Decrypt Test: overuje, že dáta po zašifrovaní a následnom dešifrovaní zostanú nezmenené.

Brute Force Attack Success Test: testuje, že brute-force útok dokáže nájsť správny kľúč, ak poznáme časť pôvodného textu.

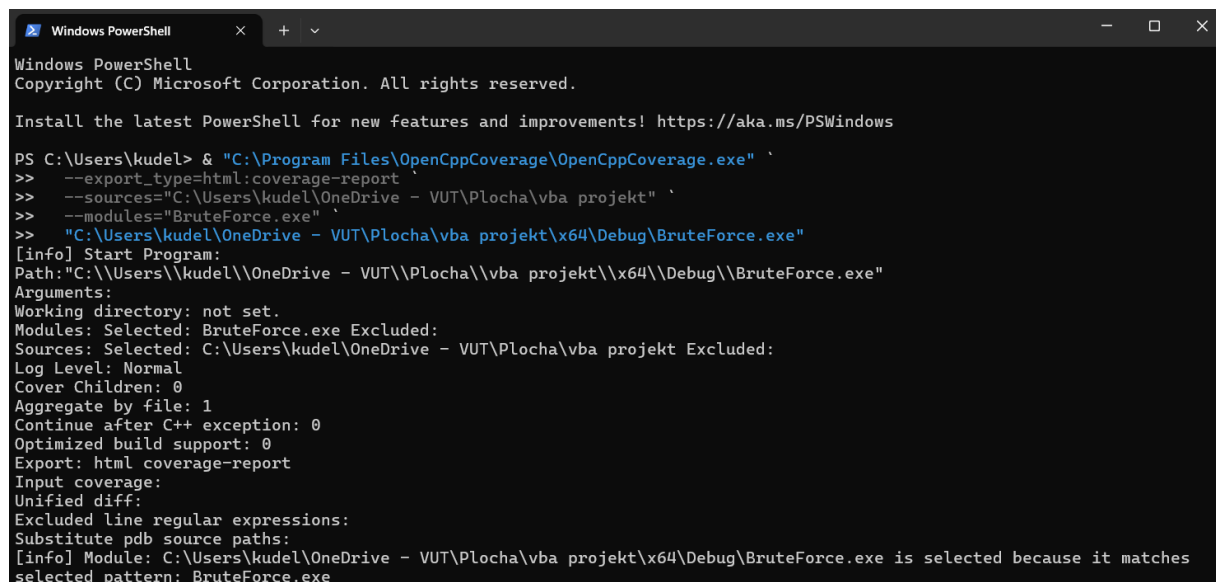
Testy využívajú testovací framework CUnit a sú spúšťané pomocou CUnit registrácie, vytvorenej sady testov a režimu výpisu výsledkov CU_basic_run_tests(). Cieľom je rýchle odhalenie chýb v implementácii a zaistenie stability chodu programu.

Meranie pokrytia kódu:

Na overenie kvality jednotkových testov používame nástroj OpenCppCoverage, ktorý nám umožňuje zistiť, ktoré časti zdrojového kódu boli počas testovania skutočne vykonané a ktoré zostali netestované. Vďaka tomu dokážeme lepšie vyhodnotiť úplnosť a efektivitu napísaných testov.

Program najskôr zostavujeme vo vývojovom prostredí Visual Studio 2022 v režime Debug x64, aby bolo možné sledovať priebeh jednotlivých riadkov pri vykonávaní. Následne spúšťame testovací spustiteľný súbor BruteForce.exe pomocou nástroja OpenCppCoverage cez príkazový riadok (PowerShell). Používame príkaz v nasledovnom tvare:

```
& "C:\Program Files\OpenCppCoverage\OpenCppCoverage.exe" `
--export_type=html:coverage-report `
--sources="cesta k testovaným .c súborom" `
--modules="BruteForce.exe" `
"cesta k BruteForce.exe"
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\kudel> & "C:\Program Files\OpenCppCoverage\OpenCppCoverage.exe" `
>> --export_type=html:coverage-report
>> --sources="C:\Users\kudel\OneDrive - VUT\Placha\vba projekt" `
>> --modules="BruteForce.exe" `
>> "C:\Users\kudel\OneDrive - VUT\Placha\vba projekt\x64\Debug\BruteForce.exe"
[info] Start Program:
Path: "C:\\Users\\kudel\\OneDrive - VUT\\Placha\\vba projekt\\x64\\Debug\\BruteForce.exe"
Arguments:
Working directory: not set.
Modules: Selected: BruteForce.exe Excluded:
Sources: Selected: C:\Users\kudel\OneDrive - VUT\Placha\vba projekt Excluded:
Log Level: Normal
Cover Children: 0
Aggregate by file: 1
Continue after C++ exception: 0
Optimized build support: 0
Export: html coverage-report
Input coverage:
Unified diff:
Excluded line regular expressions:
Substitute pdb source paths:
[info] Module: C:\Users\kudel\OneDrive - VUT\Placha\vba projekt\x64\Debug\BruteForce.exe is selected because it matches
selected pattern: BruteForce.exe
```

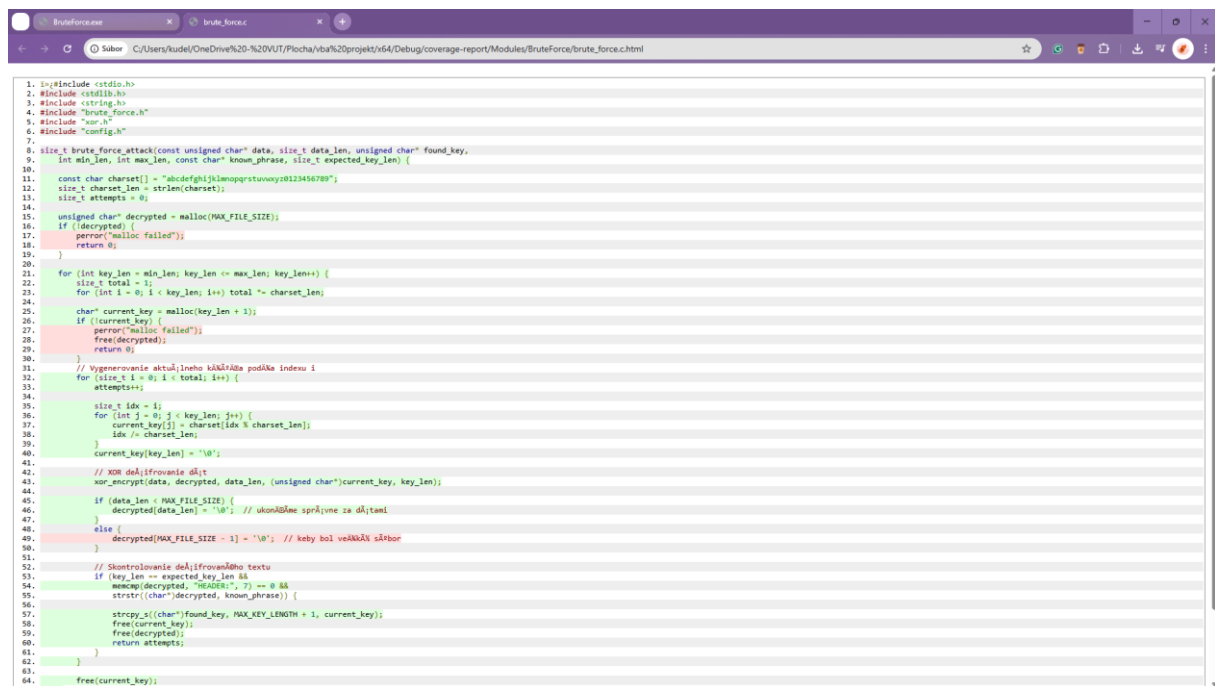
Výsledkom je prehľadný HTML report, ktorý zobrazuje, ktoré riadky v jednotlivých súboroch boli pokryté testami. Report zvýrazňuje pokryté riadky zelenou a nepokryté červenou farbou. Tento výstup nám umožňuje identifikovať miesta v kóde, ktoré si vyžadujú doplnenie testov, a tým zvyšovať spoľahlivosť celého programu.



BruteForce.exe

Coverage	Total lines	Items
<div><div>Uncover 19%</div><div><div></div><div></div><div></div></div><div>Cover 81%</div></div>	99	C:\Users\kudel\OneDrive - VUT\Placha\uba_projekt\ub4\Debug\BruteForce.exe
<div><div>Uncover 38%</div><div><div></div><div></div><div></div></div><div>Cover 62%</div></div>	8	C:\Users\kudel\OneDrive - VUT\Placha\uba_projekt\ub4\c
<div><div>Uncover 20%</div><div><div></div><div></div><div></div></div><div>Cover 80%</div></div>	40	C:\Users\kudel\OneDrive - VUT\Placha\uba_projekt\brute_force.c
<div><div>Uncover 14%</div><div><div></div><div></div><div></div></div><div>Cover 86%</div></div>	51	C:\Users\kudel\OneDrive - VUT\Placha\uba_projekt\main.c

Generated by [OpenCppCoverage](#) (Version: 0.9.9.0)



Funkcionalita:

1. Program požiada užívateľa o cestu k súboru, ktorý má byť zašifrovaný.
2. Užívateľ zadá šifrovací kľúč.
3. Súbor je zašifrovaný XOR operáciou prostredníctvom zadaného kľúča a je uložený ako encrypted.bin.
4. Program sa pokúsi o útok hrubou silou prostredníctvom znalosti časti obsahu súboru a o spätné získanie kľúča.
5. Po úspešnom nájdení kľúča je súbor opäť dešifrovaný a uložený ako decrypted.txt.

```
Enter path to file to encrypt:
> C:\Users\grego\source\repos\BruteForce\vba projekt\tajny_dokument.txt
Enter encryption key (max 8 characters):
> 123
Enter known phrase to search for in decrypted text:
> tajna
File encrypted to 'encrypted.bin'
Key found: '123'
Found after 38620 attempts.
Decrypted output saved to 'decrypted.txt'
Brute-force took 0.018 seconds.
```

```
CUnit - A unit testing framework for C - Version 2.1-3
http://cunit.sourceforge.net/
```

```
Suite: Basic Tests
Test: XOR Encrypt/Decrypt Test ...passed
Test: Brute Force Attack Success Test ...passed
```

Run Summary:	Type	Total	Ran	Passed	Failed	Inactive
	suites	1	1	n/a	0	0
	tests	2	2	2	0	0
	asserts	3	3	3	0	n/a

```
Elapsed time = 0.001 seconds
```

Hlavné súbory:

- main.c: Obsahuje hlavnú funkciu main(), riadi tok programu, volá funkcie a riadi vstupy od užívateľa.
- brute_force.c: Implementuje funkciu pre útok hrubou silou s využitím znalosti časti textu v súbore.
- xor.c: Obsahuje funciu pre šifrovanie a dešifrovanie dát XOR operáciou.
- test_project.c: Spúšťa jednotkové testy a kontroluje správne fungovanie programu.

K jednotlivým implementačným súborom patria zodpovedajúce hlavičkové súbory (xor.h, brute_force.h, config.h), ktoré obsahujú deklarácie funkcií, konštánt a potrebných dátových typov.

Záver

Tento projekt predstavuje jednoduchý, ale funkčný príklad použitia XOR šifrovania spolu s možnosťou spätného získania kľúča pomocou brute-force útoku. Ukazuje, ako aj zdanlivo jednoduchý algoritmus môže byť prakticky využitý – a zároveň, ako ľahko môže byť prelomený, ak nie je použitý správne.

Program je navrhnutý prehľadne a modulárne. Používateľ si môže jednoducho zvoliť vlastný súbor, kľúč a známu frázu, ktorú systém použije pri pokuse o zistenie kľúča. Správnosť šifrovania aj útoku sme overili pomocou jednotkových testov, ktoré pokrývajú najdôležitejšie časti logiky programu. Tiež sme využili nástroj OpenCppCoverage, vďaka ktorému vieme, ktoré časti kódu sú pokryté testami a ktoré nie.

Projekt nám pomohol lepšie porozumieť práci s pamäťou, testovaniu, ale aj základným princípom šifrovania a ich zraniteľnostiam. Zároveň ukazuje, že aj jednoduché nástroje môžu slúžiť na efektívne overenie bezpečnosti kódu.