Proiect de atestat profesional la Informatică

Password Cracking

Profesor coordonator:

Florea Delilah

Numele și prenumele candidatului:

Mărginean Miruna Elena

Sibiu, 2023

Cuprins

Password Cracking

Tema

Proiectul demonstrează problemele de securitate cibernetică, testând trei algoritmi diferiți de aflare a parolei utilizați de hackeri. În funcție de eficiența de care dau dovadă cei trei algoritmi se poate stabili cât de sigură este o parolă și ce ar mai trebui modificat pentru ca parola să fie destul de puternică pentru a face față unui atac cibernetic.

Alegerea temei

Odată cu dezvoltarea rapidă a tehnologiei apar și probleme de securitate cibernetică pe care utilizatorii trebuie să le ia în calcul înainte să dezvăluie date personale confidențiale. De aceea, alegerea unei parole puternice este importantă. Desigur, o parolă trebuie să fie ușor de reținut, însă cu cât este mai simplă, cu atât utilizatorul este mai expus în fața unui atac cibernetic. Pentru a percepe aceste riscuri și din punct de vedere practic, am implementat trei algoritmi cunoscuți de decodificare a parolei: Brute Force Attack, Dictionary Attack și Rainbow Attack. Astfel, utilizatorul își poate testa parola proprie, aflând astfel cât de utilă este pentru a-i proteja datele.

Aspecte teoretice

Programul este implementat în Visual Studio 2022, Windows Forms App (.Net Framework), în limbajul C#. După o scurtă introducere, utilizatorul își poate introduce numele și parola, apoi utilizatorul poate alege care dintre algoritmi îi va testa parola.

1. Brute Force Attack: Este cel mai simplu, dar și cel mai ineficient dintre algoritmi. După cum spune și numele, algoritmul folosește conceptul de brute force pentru a forma toate combinările posibile de caractere (litere, cifre, simboluri) cu scopul de a nimeri, într-un final, parola introdusă de utilizator. Ineficiența acestui algoritm apare din cauza posibilităților multiple: se pot folosi atât litere mari, cât și mici, cifre, dar și diferite simboluri, formând un total de 74 de caractere. În acest program am optat pentru un algoritm de Backtracking care formează toate posibilitățile și o afișează doar pe cea care este corectă. Din cauza ineficienței, algoritmul are nevoie de prea mult timp pentru a afișa o parola cu mai mult de 4 caractere, astfel dacă parola e mai lungă de atât nu este afișată.

2. Dictioary Attack: Este mai eficient decât Brute Force, dar nu cel mai util în realitate. După cum spune și numele, algoritmul se folosește de o listă de cuvinte în limba engleză, un dicționar. Cuvintele din această listă sunt comparate cu parola, iar dacă nu sunt găsite, parola este construită din mai multe cuvinte (ex. waterfall = water + fall). Timpul de executare este mult mai scurt decât cel al algoritmului precedent, îar acest algoritm poate găsi orice parolă, indiferent de lungime sau complexitate. Cu toate acestea, nu este așa periculos cum pare. Dacă cineva ar dori să introducă parola pentru a accesa un cont, ar avea nevoie de mai mult de 3 încercări pentru a o nimeri, fapt care de cele mai multe ori nu este permis. Astfel, această modalitate este la fel de utilă ca și Brute Force.
3. Rainbow Attack: Deși poate în primă instanță nu pare a fi la fel de complex, acest algoritm poate deveni periculos dacă este folosit cum trebuie. Pentru a evita piedicile puse de o interfață de Log in (nu se permite introducerea parolei de mai mult de câteva ori), hackerii preferă să ajungă la datele salvate în baza de date. Acolo, parolele sunt salvate sub forma unor valori hash, iar procesul este ireversibil. Astfel, se preferă transformarea unor cuvinte, păstrate într-un tabel, Rainbow Table, în valori hash, iar apoi valoarea hash obținută din baza de date este căutată în tabel. Pentru Rainbow Attack am folosit algoritmul de indexare criptografică prin hashing SHA-256 (Secure Hash Algorithm 256-bit), a cărui premisă este ireversibilitatea și unicitatea hashului. În cazul de față, algoritmul nu funcționează pentru toate parolele, însă într-o situație reală, Rainbow Attack poate deveni periculos.

În urma rulării acestor algoritmi se pot trage anumite concluzii despre parola aleasă de utilizator. Parola nu trebuie să fie prea scurtă, să aibă cel puțin 8 caractere, altfel algoritmi ca Brute Force Attack vor putea accesa parola prea ușor. Parola trebuie să conțină cât mai multe simboluri și cifre pentru a deruta algoritmul Dictioary Attack, iar utilizarea literelor mari ajută la creerea unei parole complexe. Se recomandă utilizarea unei parola care să nu conțină cuvinte obișnuite pentru a fi mai greu de descoperit cu ajutorul algoritmului Rainbow Attack. De asemenea, utilizarea datelor personale în parole poate face parola mai ușor de descoperit.

Descrierea aplicației

Programul începe cu o scurtă introducere în care utilizatorul trebuie să-și introducă numele și parola.

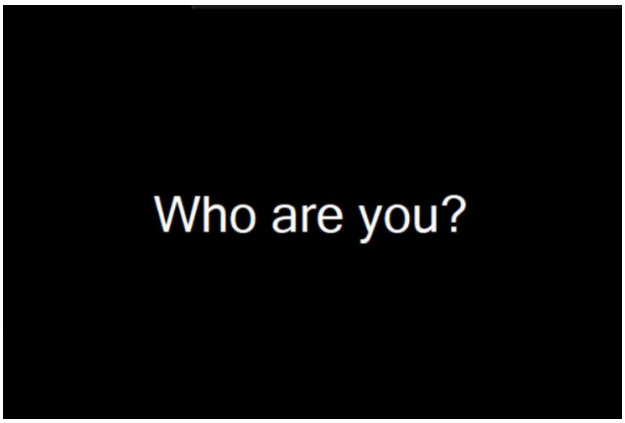

Figure 1

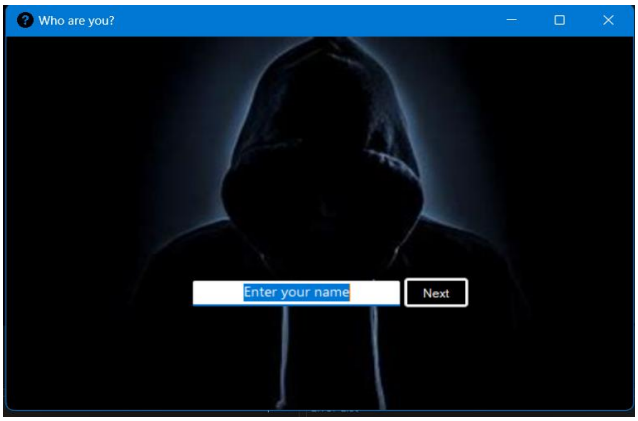

Figure 2

După ce urmează toți pașii ajunge la Menu, forma principală, de unde pot fi accesați cei 3 algoritmi.

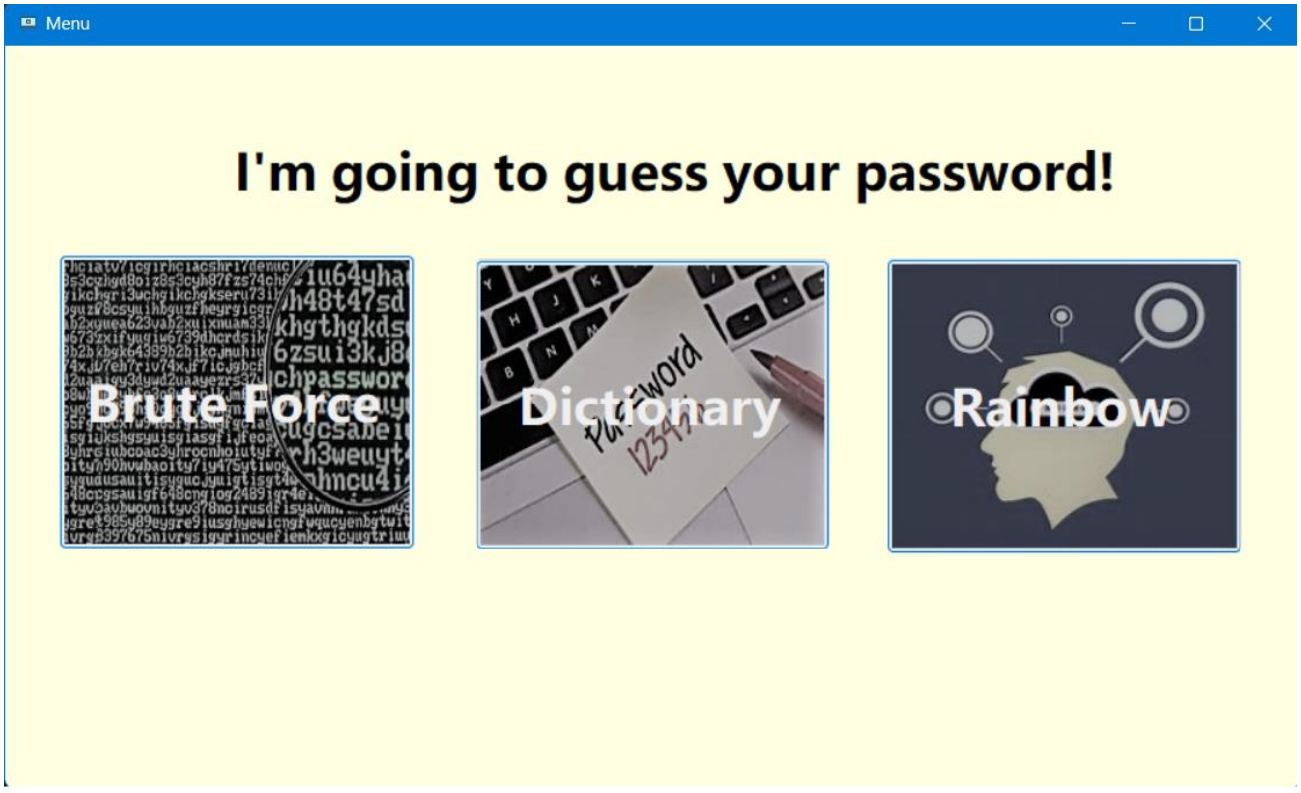

Figure 3

Fiecare reprezintă un buton ce conduce spre forma de pe care se poate rula algoritmul corespunzător. De pe fiecare dintre aceste forme se poate ieși din program (Quit) sau se poate reveni la forma principală (Menu).

Figure 4

Figure 5

Figure 6

Brute Force Attack (*Figure 4*) afișează parola găsită numai dacă aceasta este mai scurtă de 5 caractere și informează utilizatorul câte încercări au fost necesare pentru a descoperi parola.

Dictionary Attack (*Figure 5*) afișează parola găsită în orice caz și dezvăluie câte încercări au fost necesare pentru a descoperi această parolă.

Rainbow Attack (*Figure 6*) afișează parola găsită doar dacă aceasta există în lista de cuvinte transformate în hash și afișează și valoarea hash corespunzătoare parolei introduse.

După ce au fost testați toți cei trei algoritmi, utilizatorul poate găsi pe forma Menu un nou buton, Bonus. Acest buton duce spre o nouă formă unde inițial utilizatorului i se cere permisiunea de a afișa parola așa cum a fost introdusă de la tastatură. Dacă se acceptă atunci parola este afișată și analizată. Se afișează concluziile, iar utilizatorul poate afla dacă parola introdusă poate fi utilizată într-o situație din realitate. Dacă există recomandări pentru a îmbunătăți parola introdusă, acestea vor fi afișate.

Figure 7

Concluzii

În concluzie, scopul principal al proiectului este de a ajuta utilizatorii să-și îmbunătățească sistemul de securitate, observând cum reacționează diferiți algoritmi asupra parolei. De asemenea, programul încearcă să explice prin exemple practice importanța de a avea o parolă puternică, dezvăluind modul prin care hackerii se folosesc de algoritmi asemănători pentru atacuri cibernetice.

Bibliografie

SHA-256 Algorithm

Password-cracking-techniques