Instructies

Geef een duidelijk antwoord op alle vragen die hieronder worden gesteld.
Geef een toelichting als dat wordt gevraagd !
Maak van je antwoorden een verslag in **pdf**-vorm en lever dat in op de **DLO**.

Opgave 3b

We gaan hier een analyse uitvoeren op de executable '**`crackme`**'.
Deze kun je vinden in de file : `crackme.zip` .

Het programma vraagt om een wachtwoord.
Dit wachtwoord wordt in 2 methoden gecontroleerd.
Als het voldoet aan de eisen dan zal het programma een zinvolle tekst op het console tonen.
Hierin staat ook het wachtwoord voor de `crackme_src.zip` file. (deze bevat de source code!)

Maar helaas, je kent het wachtwoord niet en je hebt ook geen informatie om dit te bepalen. Je zou de executable code in detail kunnen analyseren. Maar dat is lastig en kost erg veel tijd.

Dus willen we graag een 'crack' toepassen op deze exe.

Actie 1

Analyseer de main functie en bepaal in welke 2 methoden het wachtwoord wordt gecontroleerd.
Je kunt hiervoor het bekende ***gdb*** commando gebruiken, maar ook ***IDA freeware*** en/of ***Ghidra***.

Vraag 1 : geef de namen van de beide methoden !!

Actie 2

Analyseer de 1e controle methode.

Vraag 2 : wat is **het adres** van deze 1e methode ?
Vraag 2a : wat zijn de return waarden van deze methode (er zijn 2 waarden mogelijk!)
(*bedenk dat een methode een (return) waarde in het EAX register plaatst vlak voor de return instructie!)*
Vraag 2b : analyseer de programmastructuren in deze methode en geef deze in Pseudo-C code weer. (Zoals gedaan in de voorbeeldopgave.)
Vraag 2c : In deze methode wordt ergens (??) een **globale variabele** aangepast.
Is de naam van deze globale variabele ?
Wat is het adres van deze variabele ?
Welke waarde krijgt deze variabele ?

Actie 3

Analyseer de 2e controle methode.

Vraag 3 : wat is **het adres** van deze 2e methode ?
Vraag 3a : wat zijn de return waarden van deze methode (er zijn 2 waarden mogelijk!)
(*bedenk dat een methode een (return) waarde in het EAX register plaatst vlak voor de return instructie!)*

Vraag 3b : analyseer de programmastructuren in deze methode en geef deze in Pseudo-C code weer. (Zoals gedaan in de voorbeeldopgave.)
Vraag 3c : In deze methode wordt ergens (??) een **globale variabele** aangepast.
Is de naam van deze globale variabele ?
Wat is het adres van deze variabele ?
Welke waarde krijgt deze variabele ?

Actie 4

Pas de executable aan mbv de hex-editor of met een of meer commandline instructies zoals de *python3* regel uit de demo.
Het resultaat moet zijn dat je de melding in een zinvolle (=leesbare) vorm op het scherm krijgt !!

Je kunt dit op veel verschillende manieren doen !

Vraag 4.1 : Wat is het juiste wachtwoord, volgens de gegeven melding van het programma ?
Vraag 4.2 : beschrijf welke aanpassingen je hebt aangebracht in de executable en hoe deze zijn gerealiseerd.