CYBER SECURITY

3. forduló

Continental The Future in Motion

A kategória támogatója: Continental Automotive Hungary Kft.

Ismertető a feladathoz

A 3.forduló feladatait a hosszú hétvége miatt kivételesen szerda (11.02.) éjfélig tudod megoldani!

Érdemes ebben a fordulóban is játszanod, mert a következő forduló kezdetekor, 11.03-án 18 órától kiosztjuk az 1.-2.-3. fordulóban megszerzett badgeket!

A verseny közben az alábbi teljesítményeket díjazzuk:

- fordulógyőztes
- átlagnál jobb időeredmény
- átlag feletti pontszám
- hibátlan forduló

Szeretnénk rá felhívni figyelmedet, hogy az egyszer megkapott badge-eket nem vonjuk vissza, akkor sem, ha esetleg az adott fordulóban a visszajelzések alapján változások vannak.

Jó játékot!

3.forduló

Beágyazott rendszereken egy vezérlőegységre több szoftver egység (SW unit) is telepítésre kerülhet. Ezek meglétének ellenőrzésére találták ki az úgynevezett Chain of Trust rendszert, melynek lényege, hogy egy, a hardver által indított SW-től kezdve egy láncolt listára fűzve szerepelnek az egyes SW unitok. Minden unit tartalmaz (több más info mellett) egy pointert a következő unitra, valamint egy CMAC-et (Cipher-based Message Authentication Code), mellyel bármikor ellenőrizhető az adat valódisága.

Minden unit tartalmaz (több más info mellett) egy pointert a következő unitra, valamint egy CMAC-et (Cipher-based Message Authentication Code), mellyel bármikor ellenőrizhető az adat valódisága.

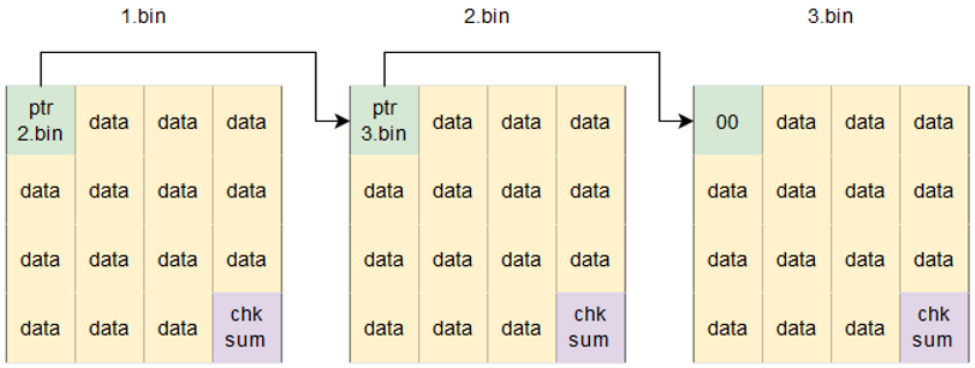
Ebben a fordulóban a feladat egy Chain of Trust rendszeren történő végigiterálás lesz, integritási hibát keresve.

Felhasznált idő: 00:00/25:00 **Elért pontszám:** 0/10

Indítás utáni csatolmányok

1. feladat 0/10 pont

Mellékelten találtok 26 darab bináris fájlt az angol ábécé összes betűjével elnevezve. Ezek a fájlok reprezentálják az egyes SW egységeket.



Minden fájl első bájtja tartalmazza a pointert a következő fájlra (ASCII betű).

A fájl utolsó bájtja egy checksum értéket tartalmaz, amivel validálni lehet fájl tartalmát.

A checksum számolása a következő módszerrel történik:

a fájl bájtjait (a pointert is beleértve) egyenként összeadogatjuk egészen a checksum előtti bájtig. Az eredményt csonkoljuk 1 bájt hosszúságúra (bitentkénti ÉS művelet 255-tel, kód: checksum=checksum&255).

Kezdőfájl: A.bin (feladat_3.zip-ben megtalálható az összes fájl)

A kezdőfájltól (A) kiindulva lépkedjetek végig a láncolt listán egészen addig, amíg nem találtok egy hibás checksumot tartalmazó fájlt. A megtalált, helyes checksumot tartalmazó fájlok betűit írjátok fel a láncolt lista sorrendjében. Ez a karaktersorozat lesz a megoldás.

Fontos: az első helytelen checksumot tartalmazó fájl neve már nem része a megoldásnak!

Megjegyzés: a láncolt lista sorrendjében első hibás fájl után következő fájlok esetében sem a checksum, sem a pointer nem konzekvens, előfordulhatnak hibás pointerek, illetve jó és rossz checksumok is.

Válasz

A helyes válasz:

AEBIWKTHYSQD

Magyarázat

Minta megoldás pythonban:

```python
import string
import random
import os

next_file = 65

result = ""

while True:
    filename = chr(next_file)+".bin"
    file = open(filename, "rb")
    content = file.read(128)
    checksum = 0
    for i in content[0:-1]:
        checksum += i
    checksum = checksum & 0xFF
    if checksum == content[127]:
        result += chr(next_file)
    else:
        break
    next_file = content[0]

print(result)
```