Autor: Ruxandra F. Olimid
Departamentul de Informatică, Universitatea din București

- Laboratorul 6 -
Generatoare de numere pseudo-aleatoare
(PRNG)

Disclaimer: Pe parcursul acestui curs/laborator vi se vor prezenta diverse noțiuni de securitate informatică, cu scopul de a învăța cum să securizați sistemele. Toate noțiunile și exercițiile sunt prezentate în scop didactic, chiar dacă uneori se presupune să gândiți ca un adversar. Nu folosiți aceste tehnici în scopuri malițioase! Acestea pot avea consecințe legale în cazul comiterii unor infracțiuni, pentru care **deveniți pe deplin răspunzători**!

1. Noțiuni introductive

 Vizualizați [1].

 De ce următoarele secvențe de cod *Candidate 1 – Candidate 3* nu definesc un PRNG?

```
1  seed = int(input("Introduceti seed "))
2
3  #Candidate 1
4  try:
5      while True:
6          print(seed)
7          seed=seed^seed
8  except KeyboardInterrupt:
9      pass
10
11 #Candidate 2
12 try:
13     while True:
14         print(seed)
15         seed=int(seed+seed/2)
16 except KeyboardInterrupt:
17     pass
18
19 #Candidate 3
20 print(seed>>2)
```

2. Secrets.py

 Citiți despre modulul *secrets.py* [2].

 Folosind acest modul, creați o secvență de cod / o mică aplicație care să aibă următoarele funcționalități:

- Generează o parolă de minim 10 caractere care conține cel puțin o literă mare, o literă mică, o cifră și un caracter special (.!$@).
  La ce poate să folosească într-o aplicație informatică această funcționalitate? Dați exemplu de un scenariu de utilizare.
- Generează un string URL-safe de (cel puțin) 32 caractere.
  La ce poate să folosească într-o aplicație informatică această funcționalitate? Dați exemplu de un scenariu de utilizare.
- Generează un token hexazecimal de (cel puțin) 32 cifre hexazecimale.
  La ce poate să folosească într-o aplicație informatică această funcționalitate? Dați exemplu de un scenariu de utilizare (diferit de scenariul anterior).
- Verifică dacă 2 secvențe sunt identice sau nu, minimizând riscul unui atac de timp (*timing attack*).
- Generează o cheie fluidă binară care ulterior să poată fi folosită pentru criptarea unui mesaj de 100 caractere.
- Stochează parole folosind un modul / o librărie care să ofere un nivel suficient de securitate. Ce ați folosit? De ce?

 Amintiți-vă aceste aspecte atunci când aveți de dezvoltat o aplicație, spre exemplu lucrarea de licență (pentru managementul parolelor, randomizarea URL-urilor, generarea token-urilor, etc.).

3. CVE, CWE, CAPEC

 Accesați paginile web CVE, CWE și CAPEC [3-6].

 Răspundeți la următoarele cerințe:

- Ce problemă identificați în următoarele secvențe de cod?

Example Language: **Java**

```
private static final long SEED = 1234567890;
public int generateAccountID() {
  Random random = new Random(SEED);
  return random.nextInt();
}
```

Figura 1. Generarea unui AccountID [7]

Example Language: **PHP**

```
function generateSessionID($userID){
  srand($userID);
  return rand();
}
```

Figura 2. Generarea unui SessionID [7]

- Care este CWE ID asociat scenariilor de mai sus si problemei pe care acestea o ridică?
- Ce se întâmplă dacă nu se folosește același seed de fiecare dată, dar spațiul seed-urilor posibile este mic? Puteți găsi un CWE ID corespunzător acestui caz?
- Căutați atacul identificat la punctul precedent în [5]. Identificați și aici o mențiune la seed?
- Găsiți alte utilizări defectuoase ale PRNG explicate în alte CWE-uri. Există CVE-uri corespunzătoare acestora?
- Căutați înregistrări CVE care se referă la vulnerabilități în legătură cu PRNG. Câte ați identificat ca fiind definite în acest an?

 Amintiți-vă aceste lucruri atunci când veți utiliza PRNG-uri în diferite sisteme informatice.

 Folosiți aceste resurse și pentru alte aspecte care vă interesează, securitatea PRNG este doar un exemplu.

Referințe bibliografice

1. Coding Math. *Episode 51 - Pseudo Random Number Generators Part I.* Accesibil la: https://youtu.be/4sYawx70iP4 Ultima accesare: noiembrie 2021.
2. Python secrets.py. Accesibil la: https://docs.python.org/3/library/secrets.html Ultima accesare: noiembrie 2021.
3. CVE. Accesibil la: https://www.cve.org/ Ultima accesare: noiembrie 2021.
4. CWE. Accesibil la: https://cwe.mitre.org/index.html Ultima accesare: noiembrie 2021.

5. CAPEC. Accesibil la: https://capec.mitre.org/index.html Ultima accesare: noiembrie 2021.
6. CVE. CVE-CWE-CAPEC Relationships. Accesibil la: https://cve.mitre.org/cve_cwe_capec_relationships Ultima accesare: octombrie 2021.
7. CWE. CWE-336: Same seed in Pseudo-Random Number Generator (PRNG). Accesibil la: https://cwe.mitre.org/data/definitions/336.html Ultima accesare: noiembrie 2021.