Programmation pour sécurité informatique

1er octobre 2020

Question : pourquoi savoir programmer ?

Pourquoi savoir programmer ?

- Ouvre la porte à beaucoup de possibilités en informatique
- Utile afin de résoudre des problèmes
- Ça paraît bien sur un CV
- Utile lors de la participation à des CTF et lors de la résolution de challenges en sécurité

Et beaucoup d'autres raisons…

Programmation en sécurité

- Challenges de programmation dans des CTF
- Écrire des scripts pour résoudre des challenges (ex. pwn)
- Bonne façon pour les débutants de contribuer à leur équipe dans les compétitions

Présentation du langage Python

- Langage de programmation de haut niveau
- Programmation fonctionnelle et orientée objet
- Typage dynamique
- Syntaxe simple
- Efficace et facile à apprendre, utile pour l'écriture de scripts

```python
from pwn import *

# this is a comment
for i in range(10):
    if i % 2 == 0:
        print("even:" + str(i))
    else:
        print("odd:" + str(i))

message = "this is a message"
print(message)
```

Installation (linux) :

```
sudo apt-get install python3 python3-pip
```

Outil utilisé: pwntools

PWNTOOLS

- Librairie en python
  - Contient un grand nombre de fonctions utiles pour résoudre des problèmes de sécurité informatique
  - Est destiné à rendre l'écriture d'exploits plus rapide et efficace
  - Contient, entre autres, des fonctions pour ouvrir et gérer des sockets (on y reviendra plus tard…)

Installation (linux) : `python3 -m pip install --upgrade pwntools`

Parenthèse : les langages ésotériques

- Relativement populaires dans les CTF
- Définition la plus simple : des langages qui « expérimentent »…
- Parfois dans le but d'étudier sérieusement le design des langages de programmation, parfois simplement en blague
- Ne sont de manière générale pas conçus pour être utilisés
- Souvent plutôt étranges, ne s'apparentant pas à du vrai code

Langages ésotériques : quelques exemples

- print(“Hello World!”) en brainfuck :

```
++++++++[>++++[>++>+++>+++>+<<<<-]>+>+>->>+[<]<-]>>.>---.+++
++++..+++.>>.<-.<.+++.------.--------.>>+.>++.
```

Source : https://en.wikipedia.org/wiki/Brainfuck

Langages ésotériques : quelques exemples

- print(“Hello World!”) en SPL (Shakespeare Programming Language) :

The Infamous Hello World Program.

Romeo, a young man with a remarkable patience.
Juliet, a likewise young woman of remarkable grace.
Ophelia, a remarkable woman much in dispute with Hamlet.
Hamlet, the flatterer of Andersen Insulting A/S.

Act I: Hamlet's insults and flattery.

Scene I: The insulting of Romeo.

[Enter Hamlet and Romeo]

Hamlet:
You lying stupid fatherless big smelly half-witted coward!
You are as stupid as the difference between a handsome rich brave hero and thyself! Speak your mind!

You are as brave as the sum of your fat little stuffed misused dusty old rotten codpiece and a beautiful fair warm peaceful sunny summer's day. You are as healthy as the difference between the sum of the sweetest reddest rose and my father and yourself! Speak your mind!

You are as cowardly as the sum of yourself and the difference between a big mighty proud kingdom and a horse. Speak your mind.

Speak your mind!

[Exit Romeo]

Scene II: The praising of Juliet.

[Enter Juliet]

Hamlet:
Thou art as sweet as the sum of the sum of Romeo and his horse and his black cat! Speak thy mind!

[Exit Juliet]

Scene III: The praising of Ophelia.

[Enter Ophelia]

Hamlet:
Thou art as lovely as the product of a large rural town and my amazing bottomless embroidered purse. Speak thy mind!

Thou art as loving as the product of the bluest clearest sweetest sky and the sum of a squirrel and a white horse. Thou art as beautiful as the difference between Juliet and thyself. Speak thy mind!

[Exeunt Ophelia and Hamlet]

Act II: Behind Hamlet's back.

Scene I: Romeo and Juliet's conversation.

[Enter Romeo and Juliet]

Romeo:
Speak your mind. You are as worried as the sum of yourself and the difference between my small smooth hamster and my nose. Speak your mind!

Juliet:
Speak YOUR mind! You are as bad as Hamlet! You are as small as the difference between the square of the sum between my little pony and your big hairy hound and the cube of your sorry little codpiece. Speak your mind!

[Exit Romeo]

Scene II: Juliet and Ophelia's conversation.

[Enter Ophelia]

Juliet:
Thou art as good as the quotient between Romeo and the sum of a small furry animal and a leech. Speak your mind!

Ophelia:
Thou art as disgusting as the quotient between Romeo and twice the difference between a mistletoe and an oozing infected blister! Speak your mind!

[Exeunt]

Source : https://en.wikipedia.org/wiki/Shakespeare_Programming_Language

Langages ésotériques : quelques exemples

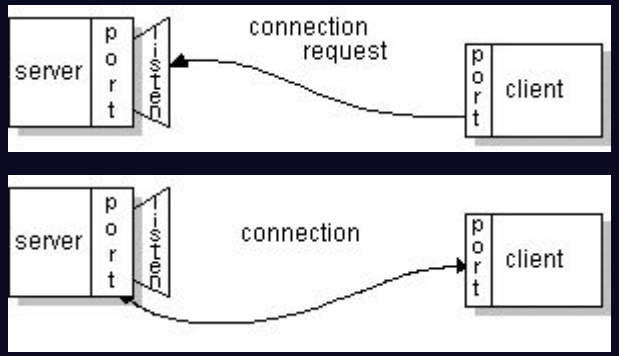
- print(“Hello World!”) en Whitespace :

Source : https://github.com/rdebath/whitespace/blob/master/tests/rdebath/helloworld.ws

Résolution de challenges

Qu’est-ce qu’un socket

- Connection client-serveur
- Dans les challenges de ce soir, le but sera de recevoir un message du serveur et de lui renvoyer une certaine réponse, tout ça en respectant une **limite de temps**.



Les sockets avec pwntools

Commandes utiles :

`from pwn import *`	Importer la librairie pwntools
`socket = remote("127.0.0.1", 3000)`	Ouvrir une connexion vers une certaine adresse IP
`socket = remote("localhost", 3000)`	Ouvrir une connexion vers un certain hôte
`message = socket.recv(1024)`	Réception d'au plus 1024 octets du serveur
`message = socket.recvline()`	Lecture de données jusqu'à l'obtention d'un "\n"
`socket.send("message")`	Envoi d'un message vers le serveur
`socket.sendline("message")`	Envoi d'un message vers le serveur en ajoutant "\n" à la fin
`socket.interactive()`	Interagir avec la connexion dans la console

Challenges de ce soir

- UnitedCTF
- Root-Me

Challenge #1 : Initiation aux sockets

- But : Renvoyer le message reçu directement au serveur
- Description :

  Le but du challenge est simplement d'ouvrir un socket vers le serveur et d'envoyer exactement le même texte qu'il vous envoie. Ce défi de programmation a pour but de vous initier à la programmation de sockets.

Les objets JSON

- JSON : JavaScript Object Notation
- Représentation d'objets dans un format facilement lisible par les humains et les programmes
- Exemple pour la représentation d'humains avec leurs enfants (provient du prochain challenge) :

```
{
	"name": "Michael Dixon",
	"age": 68,
	"child": {
		"name": "Candy Garcia",
		"age": 28,
		"child": {
			"name": "Helen Schmidt",
			"age": 2,
			"favorite-color": "orange"
		}
	}
}
```

Challenge #2 : Initiation au JSON

- But : Manipulation d'objets JSON
- Description

Dans ce défi, le serveur vous envoie un objet JSON décrivant une personne, son enfant et son petit-enfant. Vous devez:

- Aller chercher l'objet du petit-enfant
- Ajouter la propriété grandparent avec comme valeur le nom de son grand-parent
- Envoyer cet objet avec sa nouvelle propriété au serveur

```
{
  "name": "Grand-parent",
  "child": {
    "name": "Parent",
    // ...
    "child": {
      "name": "Petit-enfant"
      // ...
    }
  }
}
```

```
{
  "name": "Petit-enfant",
  // ...
  // Autres propriétés de l'enfant envoyées par le serveur
  // ...
  "grandparent": "Grand-parent"
}
```

Challenge #3 : Solve math

- But : Résoudre une série d’équations mathématiques simples
- Description :

  Vous devez calculer le résultat de dix équations mathématiques. Trouver dix bons résultats de suite et obtenez le flag.

  Vous n'avez que deux secondes pour calculer le résultat et l'envoyer au serveur. On vous suggère de faire de la reconnaissance au début afin de voir quelles sortes d'équations le serveur vous envoie, avec quels opérateurs et combien de termes.

Challenge #4 : Hash breaker

- But : Briser une série de textes hashés
- Description :

  Es-tu capable de briser les hash? Brise dix hash de suite et obtient le flag!

  La limite de temps pour briser un hash et envoyer le texte clair correspondant est fixée à deux secondes. On vous suggère d'essayer de briser les hash avec un "password hash cracker" en ligne afin de figurer le format des textes hashés avant de commencer à implémenter une solution.

Challenge #5 : Labyrinthe

- But : Résoudre un labyrinthe envoyé à l'aide de caractères ASCII par le serveur
- Description :

  Dans ce défi, un défi classique de programmation: résoudre un labyrinthe! Le serveur génère des labyrinthes aléatoires. Il vous donne une case de départ, une case d'arrivée et vous devez trouver un chemin qui se rend à la destination.

Exemple :

```
8
########
#D#000##
#0#F#00#
#00##0##
#0#0000#
#0####0#
#000000#
########
```

Format de la réponse:

```
Nombre de cases du chemin
Case #1
Case #2
...
```

Exemple :

```
7
1 0
1 1
1 2
2 2
3 2
4 2
4 3
```

Challenge #6 : IRC

- Serveur : irc.root-me.org port 6667
- But : Effectuer des calculs mathématiques et communiquer avec le protocole IRC
- Lien : https://www.root-me.org/en/Challenges/Programming/IRC-Go-back-to-college

Des questions ?