Operacje wejścia/wyjścia

Pickle

Adam Chyła
chyla.org

Pickle

Moduł **pickle** pozwala na serializację i deserializację obiektów Pythona.

Serializacją (inaczej *marshalling*, *pickling* lub *flattening*) to proces przekształcania obiektu w ciąg bajtów. Deserializacja (inaczej *demarshalling* lub *unpicking*) to proces przeciwny - zamiana kolejnych bajtów na obiekt.[1]

Uzyskany w ten sposób ciąg bajtów można zapisać do pliku lub przesłać przez sieć. Dane zapisane w pliku mogą później posłużyć do odtworzenia stanu programu przy jego kolejnym uruchomieniu.

1. https://docs.python.org/3/library/pickle.html

Pickle

Za pomocą modułu `pickle` należy deserializować dane pochodzące wyłącznie z zaufanego źródła.

Deserializacja niezaufanych danych może wiązać się z poważnym zagrożeniem bezpieczeństwa dla systemu, na którym działa nasz program.

Proces deserializacji danych umożliwia wykonanie dowolnego kodu. Kontrola danych przeznaczonych do deserializacji za pomocą modułu pickle to silny atut w rękach atakującego.

Pickle - serializacja danych

Funkcja `dumps()` pozwala na serializację obiektu do ciągu bajtów.

Ciąg bajtów można zapisać do pliku lub przesłać przez sieć.

```
import pickle

phone_book = {"Jonna": "542124",
              "Maciej": "542323",
             }

bytes = pickle.dumps(phone_book)
```

Pickle - zapisywanie danych do pliku

Zapisanie danych do pliku możemy zrealizować za pomocą funkcji `dump()`.

Plik, w którym chcemy zapisać dane, musi zostać otworzony w trybie binarnym. Należy również pamiętać o jego zamknięciu.

```
import pickle

phone_book = {"Jonna": "542124",
              "Maciej": "542323",
             }

with open('app_data.pickle', 'wb') as file:
    pickle.dump(phone_book, file)
```

Pickle - deserializacja danych

Funkcja `loads()` pozwala zamienić ciąg bajtów na obiekt.

```
import pickle

bytes = (
b'(dp0\nVJonna\np1\nV542124\np2\nsVMaciej\np3\nV54
2323\np4\ns.'
)

phone_book = pickle.loads(bytes)
```

Pickle - wczytywanie danych z pliku

Odczyt danych z pliku możemy zrealizować za pomocą funkcji `load()`.

Plik, z którego chcemy odczytać dane, musi zostać otworzony w trybie binarnym. Należy również pamiętać o jego zamknięciu.

```
import pickle

with open('app_data.pickle', 'rb') as file:
    phone_book = pickle.load(file)
```

Wykonanie dowolnego kodu

Za pomocą modułu `pickle` należy deserializować dane pochodzące wyłącznie z zaufanego źródła.

Powyższe stwierdzenie jest bardzo ważne. Dobrze obrazuje to przykład, który deserializuje ciąg bajtów podany przez złośliwego użytkownika (albo atakującego).

```
import pickle

bytes = b"cos\nsystem\n(S'echo Usuwanie plikow.'\ntR."

pickle.loads(bytes)
```

Wykonanie dowolnego kodu

Za pomocą modułu `pickle` należy deserializować dane pochodzące wyłącznie z zaufanego źródła.

W strumieniu danych znajduje się wywołanie funkcji `system()`, która uruchamia podane polecenie w konsoli.

```
import pickle

bytes = b"cos\nsystem\n(S'echo Usuwanie plikow.'\ntR."

pickle.loads(bytes)
```

Pytania

1. Czym jest serializacja i deserializacja?
2. Dlaczego użycie modułu pickle może być niebezpieczne?

Literatura

1. pickle — Python object serialization, https://docs.python.org/3/library/pickle.html
2. Don't Pickle Your Data, https://www.benfrederickson.com/dont-pickle-your-data/