Обзор избранных уязвимостей в безопасности python программ.

1 Введение.

1.1 Вступление.

...

1.2 Постановка проблемы (что хотел сказать автор, какую проблему он затрагивает).

...

1.3 Обрисовка проблемы, её проблемы и актуальность).

...

1.4 Обрисовка проблемы, её аспекты и актуальность.

...

2 Основная часть.

2.1 [CWE-78].

...

2.2 [CWE-377]

...

2.3 Избыточные права доступа для критического ресурса[CWE-732].

Если ресурс имеет права доступа больше, чем нужно для нормального исполнения программы, то это может привести к раскрытию конфиденциальной информации или несанкционированному изменению этого ресурса. Это особенно опасно, когда ресурс связан с конфигурацией программы, ее выполнением или конфиденциальными данными пользователей.

2.3.1 Пример 1.

Приведённый ниже код устанавливает маску режима создания пользовательских файлов (umask) процесса равной нулю, создаёт файл и записывает в него строку “Hello, world!”.

```
# umask.py
import os


os.umask(0)

with open('hello.out', 'w') as f:
   f.write('Hello, world!');
```

После его исполнения на UNIX системе, результат использования команды ‘ls -l’ может быть следующим:

```
-rw-rw-rw- 1 <name> <name> 13 Sep 22 11:39 hello.out
```

Строка “rw-rw-rw-” указывет на то, что владелец, группа и все пользователи могут читать и редактировать этот файл.

2.3.2 Пример 2.

Рассмотрим стандартный процесс создания файла.

```
# without-chmod.py
with open('hello.out', 'w') as f:
```

```
    f.write('Hello, world!');
```

Файл ‘hello.out’ будет иметь параметры доступа “rw-rw-r–”. Это значит, что сторонние пользователи не могут его редактировать.

Теперь добавим несколько дополнительных строк кода.

```
# with-chmod.py
with open('hello.out', 'w') as f:
    f.write('Hello, world!');

from os import chmod
chmod('hello.out', 0o666)
```

После исполнения данного кода на UNIX системе, результат команды ‘ls -l’ будет другим: “rw-rw-rw-”. Это приводит нас к той же проблеме, что и в 2.3.1.