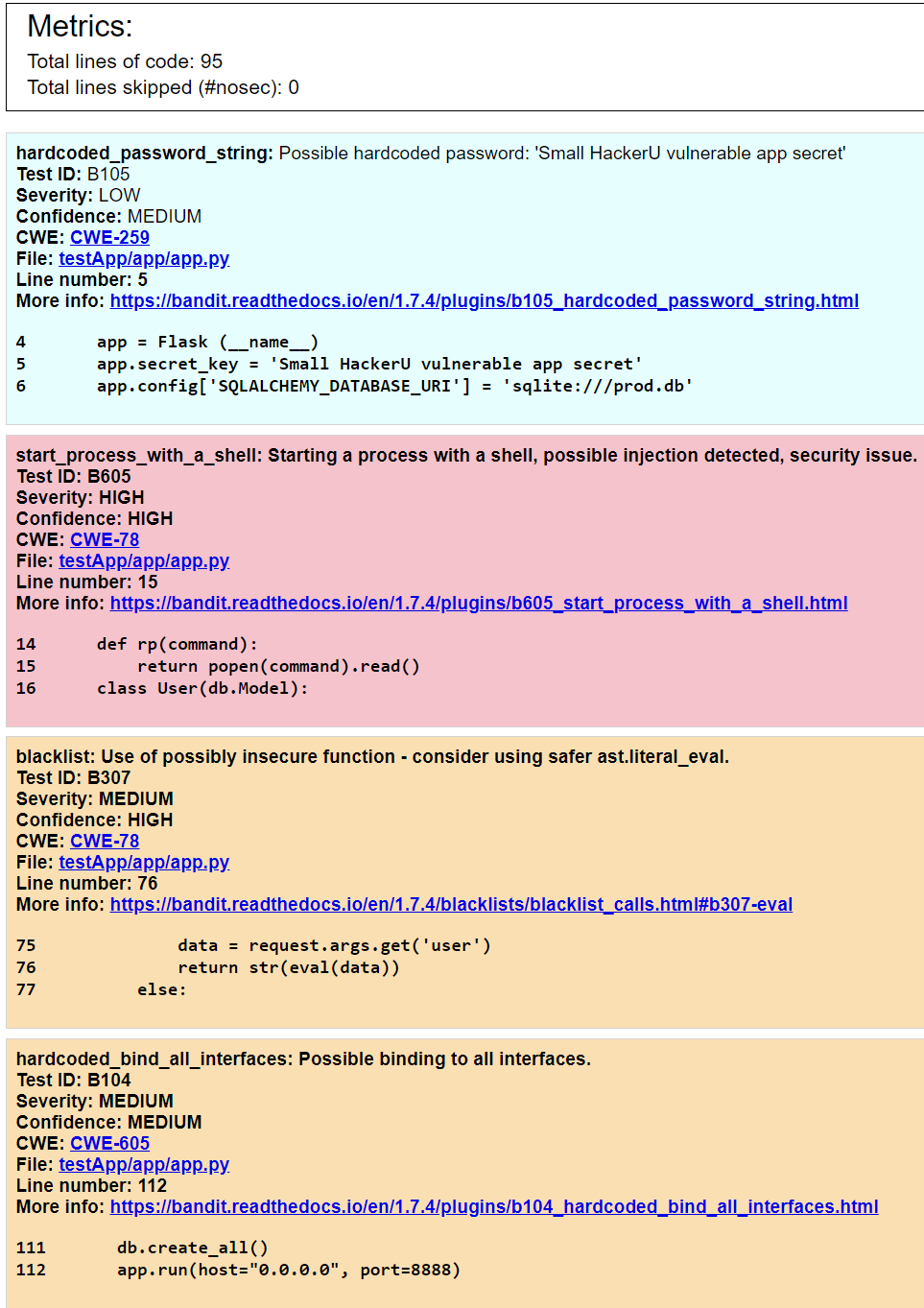
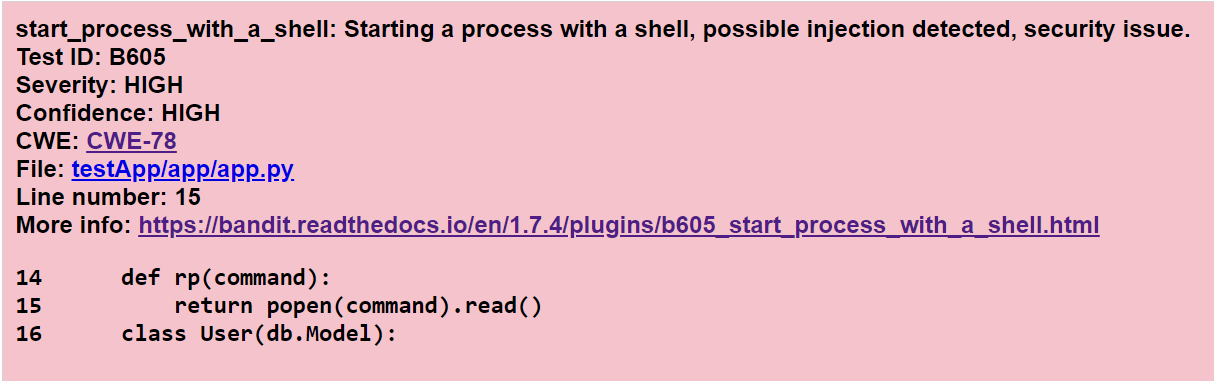
Анализ CWE, выявленных Bandit в Small HackerU vulnerable app



В результате анализа исходного кода выявлено 4 CWE: 1 высокого уровня критичности, 2 – среднего и 1 – низкого.

# CWE-78: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')



Уязвимость выявлена в файле testApp/app/app.py.

В приложении выполняется запуск команды:

def rp(command):  
 return popen(command).read()

На вход метода подается следующее значение:

rp("nslookup " + address)

Метод предназначен для выполнения команды nslookup для введенного пользователем значения IP адреса / имени хоста. При этом, вводимое значение никак не валидируется и не очищается, что позволяет ввести произвольную команду, которая выполнится на сервере, например, передав в GET запрос вида «/what\_ip?address=123123; cat /etc/passwd», или подставив «123123; cat /etc/passwd» в форму ввода адреса, можно прочитать содержимое файла passwd. При этом, вывод команды осуществляется обратно на страницу в браузере пользователя.

Для предотвращения уязвимости необходимо:

1. Запретить обращение к данному эндпойнту методом GET:

@app.route('/what\_ip', methods = ['POST'])

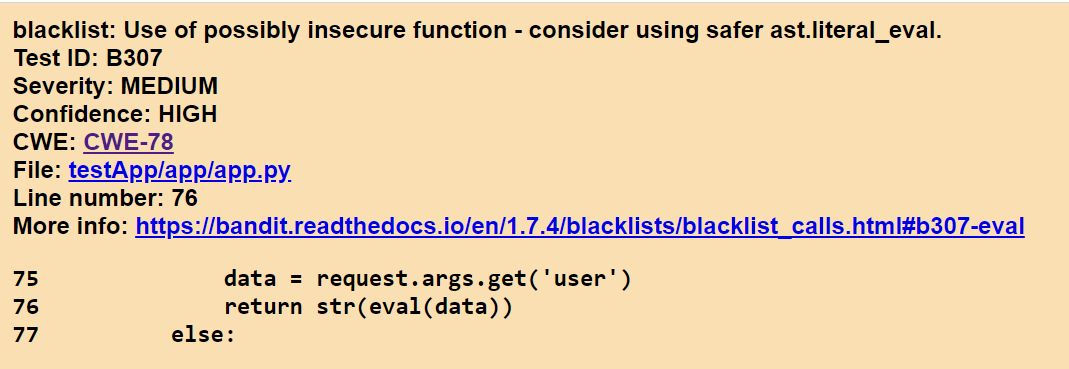
1. Использовать subprocess API, передавая команду в виде списка аргументов

def rp(command):  
 return subprocess.Popen(command, shell=False)

*Python не знаю, поэтому могу ошибаться в нужном методе. Есть еще subprocess.call, subprocess.run и т.п.*

1. Производить валидацию и санитизацию пользовательского ввода. В данном случае, можно использовать regexp для IP адреса / hostname.

# CWE-78 в строке 76 (CWE-94: Improper Control of Generation of Code ('Code Injection'))



Уязвимость выявлена в файле testApp/app/app.py.

В приложении выполняется запуск команды:

data = request.args.get('user')  
return str(eval(data))

Функция eval() оценивает строковое значение как выражение Python. Если значение data может быть введено извне программы, это может привести к выполнению произвольного кода.

В MITRE данная уязвимость классифицируется как CWE-94: Improper Control of Generation of Code ('Code Injection').

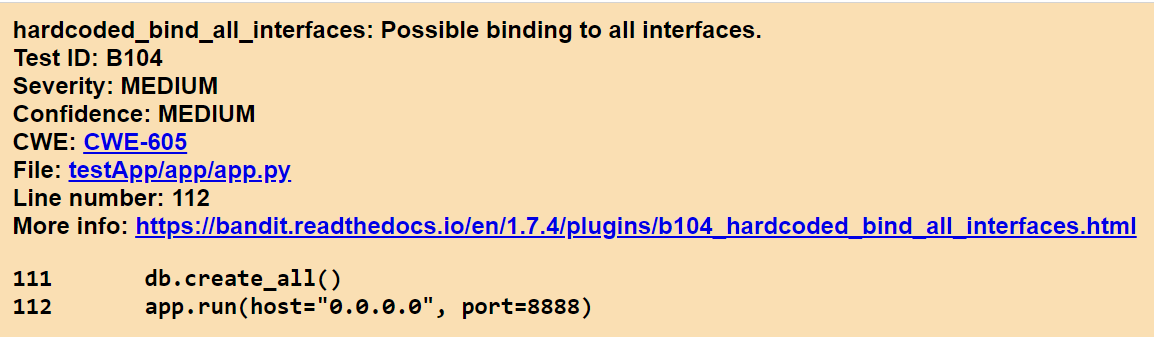
Например, передав в GET запросе параметр вида /api/user?user=\_\_import\_\_('code').InteractiveInterpreter().runsource('import requests;requests.get(\'localhost:3000\')'), значение этого параметра будет интерпретировано как код и выполнится в приложении.

Для предотвращения уязвимости необходимо:

1. Производить валидацию и санитизацию пользовательского ввода. В данном случае, можно использовать regexp для запрета любых символов, кроме буквенно-цифровых;
2. Использовать функцию ast.literal\_eval:

return str(ast.literal\_eval(data))

# CWE-605: Multiple Binds to the Same Port



Уязвимость выявлена в файле testApp/app/app.py.

В приложении выполняется запуск команды:

app.run(host="0.0.0.0", port=8888)

В данной строке производится привязка ко всем сетевым интерфейсам, что потенциально может дать возможность отправлять запросы на непредусмотренные интерфейсы, которые могут быть недостаточно защищены.

Для предотвращения потенциальной уязвимости необходимо явно задать IP адрес сервера:

app.run(host="192.169.0.2", port=8888)

# CWE-259: Use of Hard-coded Password



Уязвимость выявлена в файле testApp/app/app.py.

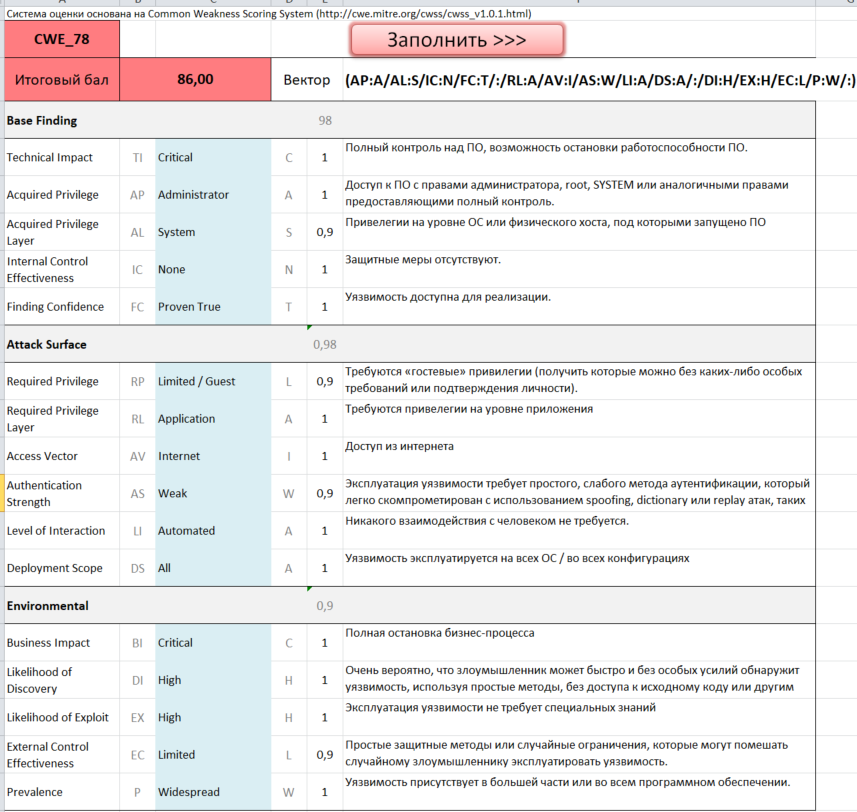
В приложении в открытом виде задается значение параметра secret\_key, используемого для взаимодействия с внешними компонентами:

app.secret\_key = 'Small HackerU vulnerable app secret'

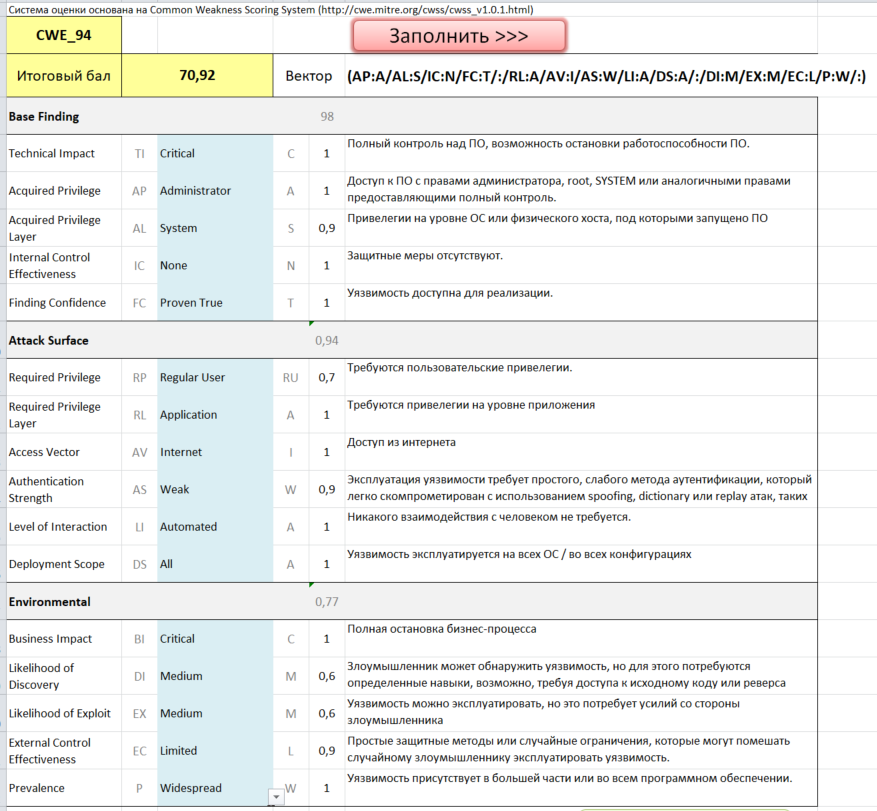
Для предотвращения потенциальной уязвимости необходимо хранить значение данного ключа в зашифрованном (с использованием криптостойких алгоритмов) виде в исходном коде и расшифровывать его с использованием мастер-пароля при развертывании приложения (например, передав его в переменной окружения (app.secret\_key = os.gentenv("SECRET\_API\_KEY")), или (надежнее) использовать внешнее хранилище ключей, например, HashiCorp Vault.

# Оценка уязвимостей в соответствии с Common Weakness Scoring System (CWSS)

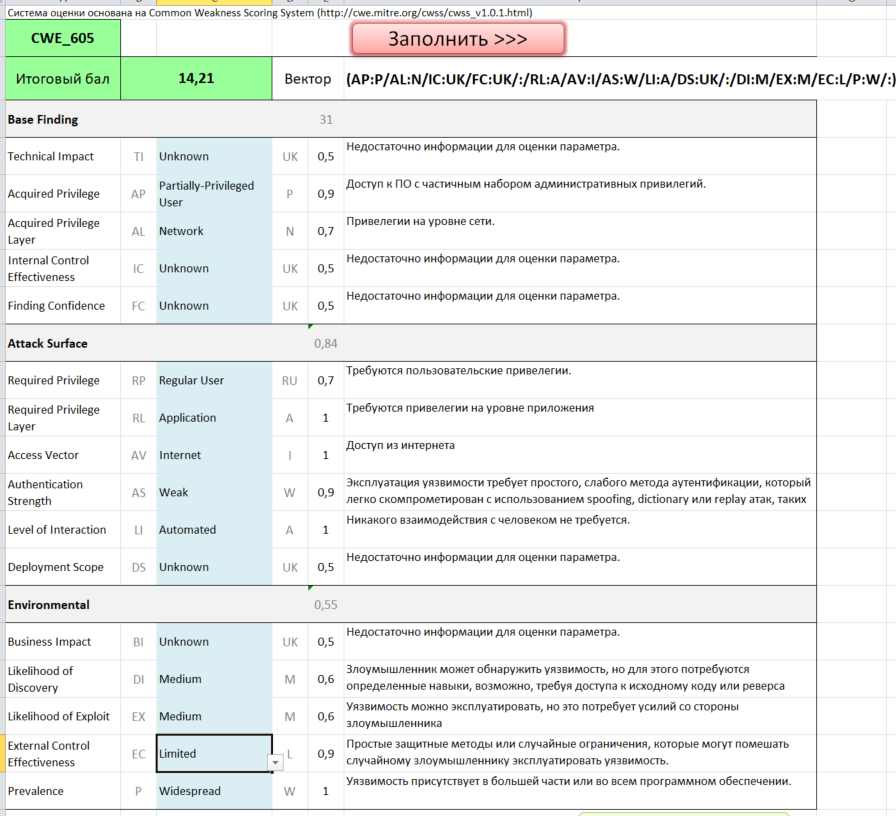
## CWE-78



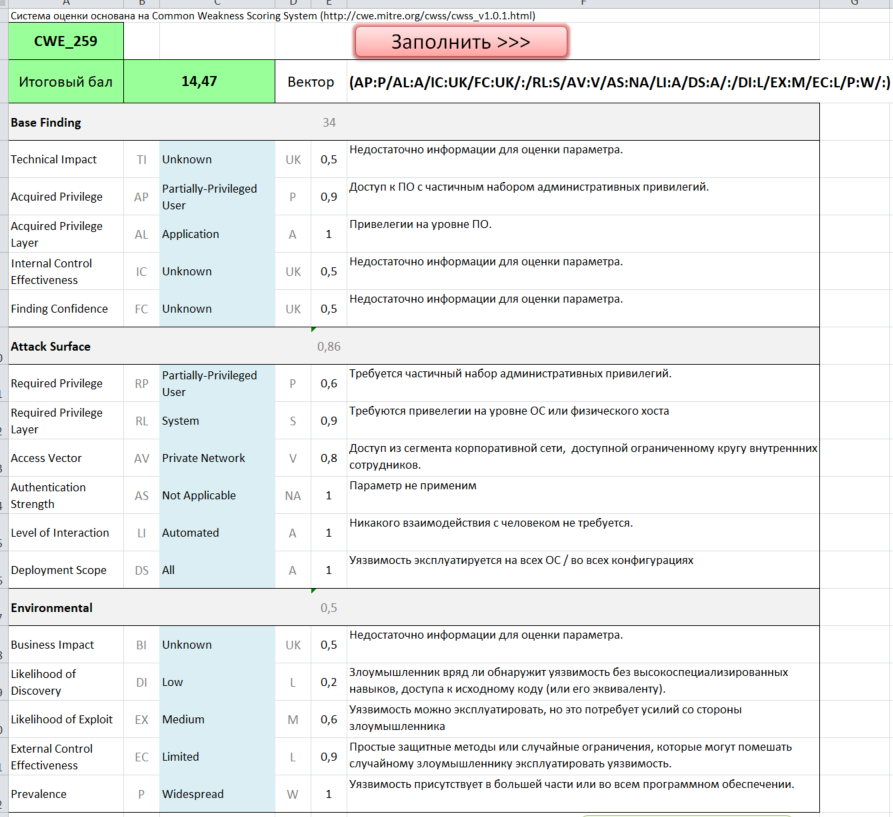
## CWE-94



## CWE-605



## CWE-259



## Summary

Применяемая шкала оценки:

|  |  |  |
| --- | --- | --- |
| Степень критичности | Предпринимаемые меры | Итоговый балл |
| Высокая | Устранение в текущем релизе/ выпуск fix-патча | > 75 |
| Средняя | Планирование исправления в ближайших релизах/ устранение в очередном патче | > 30 и < 75 |
| Низкая | Рекомендуется устранить в будущих релизах | > 10 и < 30 |

|  |  |  |
| --- | --- | --- |
|  |  | Результат (от 1 до 100) |
| 1 | CWE\_78 | 86,00 |
| 2 | CWE\_94 | 70,92 |
| 3 | CWE\_605 | 14,21 |
| 4 | CWE\_259 | 14,47 |