Государственное учреждение образования

“БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ”

Кафедра: Интеллектуальных информационных технологий

Дисциплина: Проектирование защищенных интеллектуальных информационных систем

Отчет по лабораторной работе №5

**Анализ безопасности кода.**

**Анализ безопасности кода разработанного приложения**

Выполнил:

студент гр.121702

Витковская С. И.

Проверил:

Захаров В. В.

Минск 2024

**Задание:**

1. Разработать приложение, которое поддерживает интерактивный пользовательский интерфейс, с помощью которого можно создать, обновить и удалить данные в/из оперативной памяти. Данные должны быть как конфиденциальные, так и неконфиденциальные. После ввода конфиденциальных данных, выполнить шифрование или хеширование этих данных.

2. Выполнить анализ безопасности разработанного приложения с точки зрения кода.

3. Выполнить анализ оперативной памяти и процессорного времени приложения.

4. Создать дамп оперативной памяти приложения

а) после ввода данных в оперативную память,

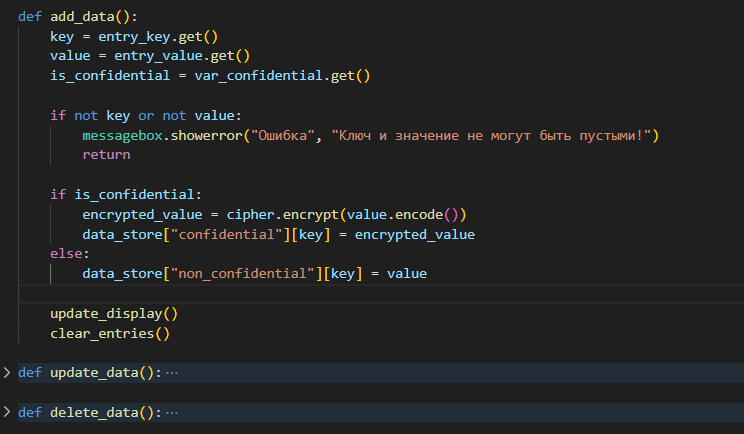
б) после обновления данных в оперативной памяти,

в) после удаления данных из оперативной памяти.

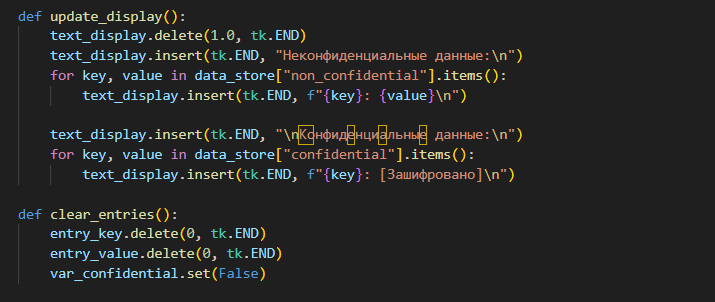
Выполнить анализ дампов. Особое внимание необходимо уделить работе с конфиденциальными данными в оперативной памяти.

**Ход работы:**

1. **Разработка программы**
2. Функции add\_data, update\_data, delete\_data добавляют, изменяют и удаляют данные соответственно, проверяя уровень конфиденциальности и корректность ввода.



1. Функции update\_display и clear\_entries отвечают за обновление пользовательского интерфейса после совершения каких либо действий.



1. Дамп памяти создается с помощью create\_memory\_dump. Первый дамп сохраняет текущее состояние системы, последующие дампы отображают изменения.

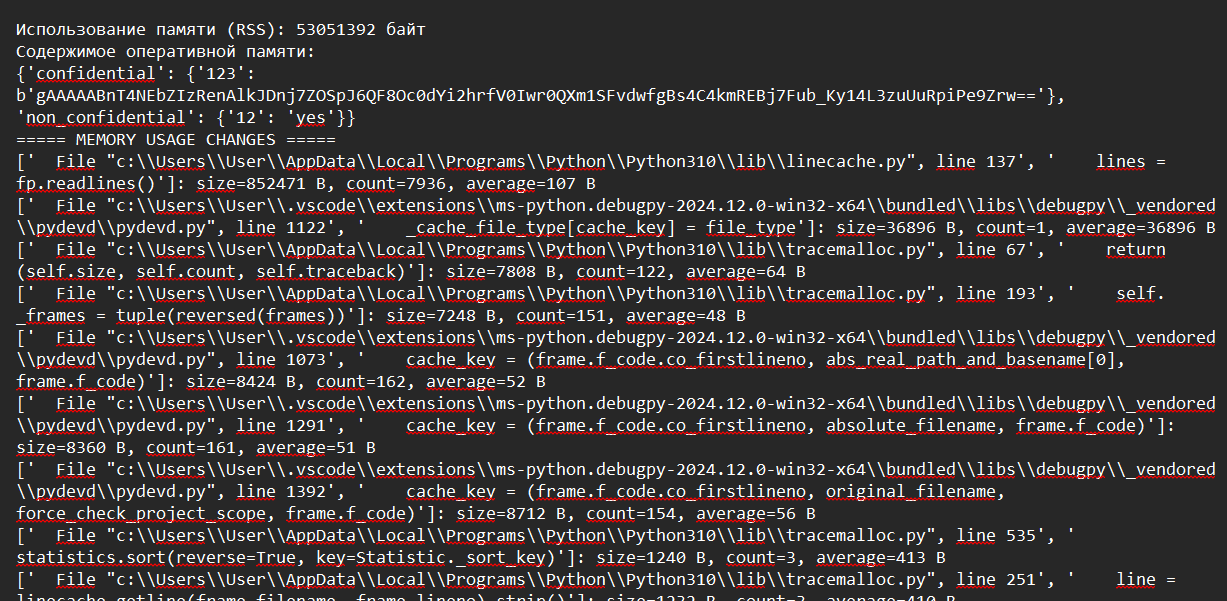


1. **Тестирование программы**

Добавление конфиденциальных данных:

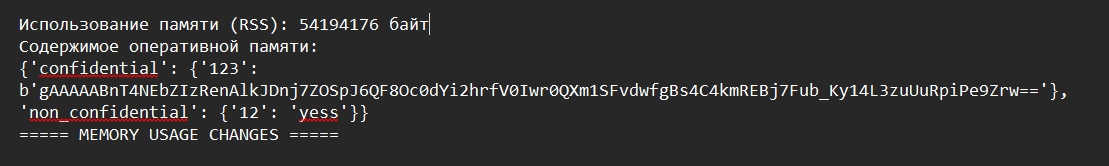
|  |  |
| --- | --- |

Зашифрованные данные не отображаются в интерфейсе, неконфиденциальные отображаются. В том числе в дампе они сохраняются в захэшированном виде.

****

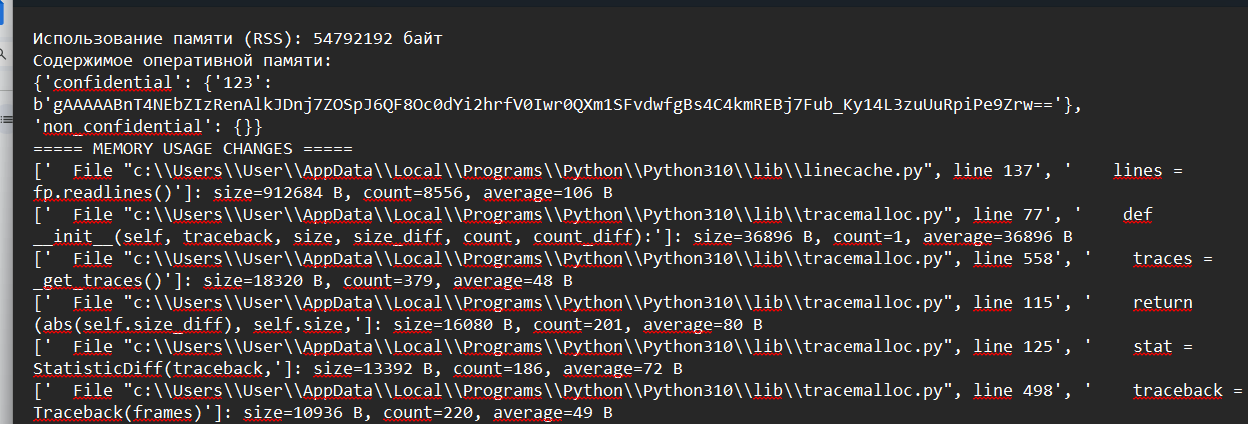
Обновление данных:

|  |  |
| --- | --- |

****

Удаление:

|  |  |
| --- | --- |

****

1. **Анализ безопасности с точки зрения кода**

* Код организован в функции, что упрощает проверку и сопровождение.
* Конфиденциальные данные могут быть защищены через функции хэширования.
* Существуют проверки корректности пользовательского ввода, но нет проверки типов данных, что может привести к ошибкам или некорректному поведению.
* Данные остаются в памяти после использования и могут быть извлечены из дампа памяти, но конфиденциальные данные хранятся в зашифрованном виде.
* Отсутствие журналирования попыток ввода некорректных данных.

1. **Анализ оперативной памяти и процессорного времени приложения**

* Приложение активно использует оперативную память для хранения данных в data\_store и работы с дампами памяти, что позволяет централизовать управление и избежать лишнего выделения памяти.
* Использование tracemalloc помогает отслеживать статистику памяти и находить потенциальные утечки.
* Неоптимальное использование памяти при дампах: дамп записывает содержимое в файл, но сами объекты остаются в оперативной памяти.

**Вывод:**

В ходе лабораторной работы было разработано приложение для создания, обновления и удаления данных в/из оперативной памяти. Данные имеют разный уровень конфиденциальности. Конфид.данные шифруются. Приложение позволяет создавать дампы памяти. Был выполнен анализ безопасности разработанного приложения с точки зрения кода, анализ оперативной памяти и процессорного времени приложения.