

Министерство образования Республики Беларусь

Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Информационные сети. Основы безопасности

ОТЧЁТ  
к лабораторной работе №5  
на тему

**ЗАЩИТА ОТ АТАКИ НА ПЕРЕПОЛНЕНИЕ БУФЕРА**

Выполнил: студент гр. 253503  
Тимошевич К.С.

Проверил: ассистент кафедры  
информатики Герчик А.В.

Минск 2025

## СОДЕРЖАНИЕ

1 Постановка задачи.....	3
2 Ход выполнения работы.....	4
Заключение.....	5
Список использованных источников.....	6

## 1 ПОСТАНОВКА ЗАДАЧИ

В данной лабораторной работе рассматривается проблема безопасности, связанная с переполнением буфера, одной из наиболее распространенных уязвимостей, которая может привести к выполнению произвольного кода, несанкционированному доступу к системе или изменению её поведения [1].

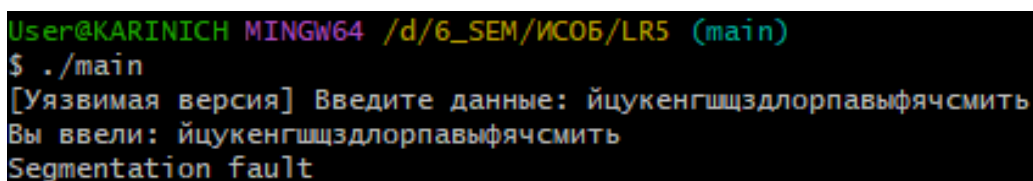
Цель работы – изучить принципы возникновения таких атак, их последствия, а также разработать и протестировать методы защиты на языке C. Для этого сначала анализируются механизмы атак, включая их влияние на выполнение программного кода и возможные способы эксплуатации. Затем разрабатывается уязвимая программа, демонстрирующая проблему, с использованием небезопасных функций работы со строками, таких как *gets*.

После этого создаётся защищённая версия, в которой применяются безопасные методы работы с буферами, включая *fgets* и *strncpy*, а также дополнительные механизмы защиты, такие как контроль длины вводимых данных.

Обе версии программы подвергаются тестированию, чтобы оценить их поведение при различных входных данных и проверить устойчивость к атакам на переполнение буфера. В заключение проводится анализ полученных результатов, выявляются различия между уязвимой и защищённой программами.

## 2 ХОД ВЫПОЛНЕНИЯ РАБОТЫ

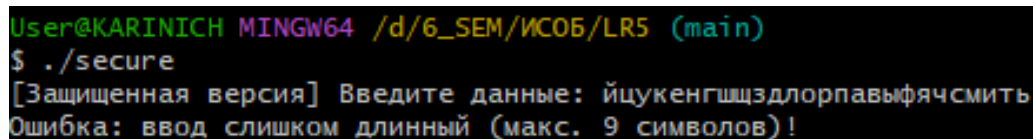
В ходе выполнения лабораторной работы были разработаны и протестированы две версии программы на языке C: уязвимая и защищенная. Уязвимая версия программы использует функцию *gets*, которая не проверяет длину ввода, что делает её подверженной атакам на переполнение буфера. В этой версии программа запрашивает у пользователя ввод данных и выводит их на экран, но при вводе строки, превышающей размер буфера, происходит переполнение, что может привести к повреждению стека и выполнению произвольного кода. Для демонстрации уязвимости программа была протестирована с вводом данных, значительно превышающих размер буфера, что привело к ошибке *Segmentation fault* (рисунок 2.1).



```
User@KARINICH MINGW64 /d/6_SEM/ИСОБ/LR5 (main)
$ ./main
[Уязвимая версия] Введите данные: йцукенгшщздорпавыфячсмить
Вы ввели: йцукенгшщздорпавыфячсмить
Segmentation fault
```

Рисунок 2.1 – Выполнение уязвимой версии программы

Защищённая версия программы была реализована с использованием безопасных методов работы с буферами. Вместо функции *gets* используется *fgets*, которая ограничивает длину ввода и предотвращает переполнение буфера [2]. Также добавлена проверка длины ввода: если введённая строка превышает допустимый размер, программа выводит сообщение об ошибке и завершает выполнение. Это позволяет избежать переполнения буфера и связанных с ним уязвимостей. В ходе тестирования защищённой версии было подтверждено, что программа корректно обрабатывает ввод данных, не превышающих размер буфера, и отклоняет слишком длинные строки, предотвращая потенциальные атаки.



```
User@KARINICH MINGW64 /d/6_SEM/ИСОБ/LR5 (main)
$ ./secure
[Защищенная версия] Введите данные: йцукенгшщздорпавыфячсмить
Ошибка: ввод слишком длинный (макс. 9 символов)!
```

Рисунок 2.2 – Выполнение защищенной версии программы

Для наглядного сравнения обе версии программы были протестированы с одинаковыми входными данными. Уязвимая версия продемонстрировала переполнение буфера и завершилась с ошибкой, в то время как защищённая версия успешно обрабатывала корректные данные и отклоняла некорректные.

## ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы были изучены принципы атак на переполнение буфера и их последствия для безопасности программного обеспечения. Была разработана уязвимая версия программы, использующая небезопасную функцию *gets*, которая продемонстрировала возможность переполнения буфера и повреждения стека при вводе данных, превышающих размер буфера. Это подтвердило, что использование небезопасных функций может привести к уязвимостям, которые могут быть использованы злоумышленниками для выполнения произвольного кода или нарушения работы программы.

Для предотвращения таких атак была разработана защищённая версия программы, в которой использовались безопасные методы работы с буферами, такие как *fgets* и *strncpy*, а также добавлена проверка длины ввода. Тестирование защищённой версии показало, что программа успешно обрабатывает корректные данные и предотвращает переполнение буфера, отклоняя слишком длинные строки. Это подтвердило эффективность применения безопасных методов программирования для защиты от атак на переполнение буфера.

Данная лабораторная работа позволила не только изучить теоретические аспекты атак на переполнение буфера, но и получить практический опыт реализации защитных механизмов, что является важным шагом в формировании навыков безопасного программирования.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Что такое атаки переполнения буфера и способы их предотвращения [Электронный ресурс]. – Режим доступа: <https://uzsoft.uz/chto-takoe-ataki-perepolneniya-bufera-i-kak-oni-predotvrashhayutsya>. – Дата доступа: 05.03.2025

[2] fgets() and gets() in C [Электронный ресурс]. – Режим доступа: <https://www.digitalocean.com/community/tutorials/fgets-and-gets-in-c-programming>. – Дата доступа: 05.03.2025