# АВТОМАТИЗИРОВАННЫЙ СТАТИЧЕСКИЙ АНАЛИЗ УЯЗВИМОСТЕЙ МЕССЕНДЖЕРА PEREGRINE С ИСПОЛЬЗОВАНИЕМ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ

С.Н. Петров[1], Е.С. Пакуль[2], А.А. Корчинский[2]

[1] *Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники», г. Минск, Республика Беларусь*

[2] *Учреждение образования «Национальный детский технопарк», г. Минск, Республика Беларусь*

**Аннотация.** Разработан мессенджер имеет клиент-серверную архитектуру и использующий национальные стандарты шифрования. Выполнено статическое тестирование исходного кода мессенджера на уязвимости с использованием сканер L3X и больших языковых моделей для валидации результатов, полученных от сканера. Такой подход позволил отфильтровать до 85% ложноположительных срабатываний.

**Ключевые слова:** системы обмена мгновенными сообщениями, мессенджеры, сквозное шифрование, информационная безопасность, криптографические протоколы, тестирование программного обеспечения.

# AUTOMATED STATIC VULNERABILITY ANALYSIS OF THE PEREGRINE MESSENGER WITH LARGE LANGUAGE MODELS

S.N. Petrov[1], Y.S. Pakul[2], A.A. Korchinsky[2]
[1] *Educational Institution "Belarusian State University of Informatics and Radioelectronics", Minsk, Republic of Belarus*
[2] *Educational Institution "National Children's Technopark", Minsk, Republic of Belarus*

**Abstract.** The developed messenger has a client-server architecture and uses national encryption standards. Static testing of the messenger's source code for vulnerabilities was performed using the L3X scanner and large language models to validate the results obtained from the scanner. This approach allowed filtering out up to 85% of false positives.
**Keywords:** instant messaging systems, instant messengers, end-to-end encryption, information security, cryptographic protocols, software testing.

## Введение

Мессенджеры стали универсальным инструментом для коммуникации и совместной работы. Однако в связи с актуализацией проблематики обеспечения безопасности данных при их передаче и возможными ограничениями (например, запретом на использование) становятся актуальными вопросы разработки альтернативных, безопасных мессенджеров. Целью проекта является создание безопасного мессенджера на языке программирования Rust с защитой пользовательских данных и end-to-end (сквозным) шифрованием на основе национальных криптографических стандартов. При разработке защищенного мессенджера большое внимание было уделено вопросам тестирования продукта. В частности, выполнено статическое тестирование исходного кода мессенджера на наличие уязвимостей.

## Основная часть

Разработанный мессенджер имеет клиент-серверную архитектуру, в которой клиенты взаимодействуют с центральным сервером, пересылающим сообщения от отправителя к получателю, выполняющим аутентификацию пользователей и управление доступом. Основной акцент в проекте сделан на безопасность. Центральным звеном стала разработка библиотеки bee2-rs на языке программирования Rust. Библиотека создана для вызовов функций библиотеки Bee2, написанной на языке программирования C.

У статических анализаторов есть особенность – большое количество ложноположительных результатов. Из-за того, что большинство анализаторов используют регулярные выражения или похожие способы анализа исходного текста, возникает сложность обработки и интерпретации полученных результатов. Отсюда возникла идея – использовать статические анализаторы кода, которые основываются на регулярных выражениях, а после использовать большие языковые модели (LLM) для валидации результатов.

Был проведен статический анализ кода мессенджера с использованием статического анализатора L3X (L3X AI-driven Static Analyzer). С кодом проекта можно ознакомится в репозитории Github по ссылке https://github.com/VulnPlanet/l3x. В данном проекте разработчиками была заложена идея использования валидации результатов сканирования с использованием трех языковых моделей по принципу мажоритарного голосования. Однако на данный момент ими реализована поддержка только облачного API ChatGPT, что создавало риски конфиденциальности и зависимость от внешних сервисов. Для решения этой проблемы авторами была выполнена существенная модификация сканера L3X, а именно осуществлен переход на локальные большие языковые модели Gemma 3 12B, Qwen 3 14B, Phi-4 Reasoning Plus 15B.

Паттерны, используемые для поиска уязвимостей, были разделены на три категории по степени критичности: High, Medium и Low. Категория High относится к критическим уязвимостям. Например, уязвимости в блоке unsafe или использование неинициализированной памяти категорируется как High. В результате сканирования без валидации было обнаружено 1537 уязвимостей, включая 103 High, 1292 Medium и 190 Low уязвимостей. В дальнейшем были использованы большие языковые модели для валидации результатов. Такой подход доказал свою эффективность в значительном снижении уровня ложноположительных срабатываний по сравнению с исходным сканированием только инструментом l3x без использования LLM в качестве валидаторов (до 85% ложноположительных результатов).

Языковая модель Gemma 3 12B подтвердила 6 High, 211 Medium и 7 Low уязвимостей. Однако все 6 High уязвимостей были проверены вручную, и ни одна из них не представляла опасности. Проверки остальных потенциальных уязвимостей также показали отсутствие рисков

Языковая модель Qwen 3 14B подтвердила 51 High уязвимостей, 119 Medium, 85 Low, 1219 False Positive, 63 Unknown. False positive rate составил 83.409%. Из 51 High уязвимости, снова ни одна из них не представляет опасности. В этот раз одна уязвимость была про выделение памяти в коде серверной части, основанное на непроверенном значении,

однако несколькими строками кода выше в той же функции проверка на максимальный размер данных все же осуществляется. Одна уязвимость была про возможное неправильное использование потоков и синхронизации, однако в том же файле для всех данных, которые могут быть получены/изменены в нескольких потоках, использовался тип данных, предотвращающий одновременные чтение или запись с нескольких потоков.

Языковая модель Phi 4 Reasoning Plus (15B) подтвердила 0 High, 0 Medium, 0 Low, 1395 False Positive, 142 Unknown. False positive rate составил 100%. Вероятнее всего это означает, что модель Phi 4 Reasoning Plus не подходит для анализа уязвимостей программного кода.

## Заключение

Общий объем написанного кода составил 10452 строк, из которых 7860 строк составляет код мессенджера и 2592 строки составляет код библиотеки bee2-rs. С исходным кодом мессенджера (серверная и клиентская часть, библиотека bee2-rs) можно ознакомится в репозитории Github по ссылке https://github.com/tpyauheni/peregrine. Был проведен статический анализ кода мессенджера с использованием статического анализатора L3X и больших языковых моделей. Такой подход доказал свою эффективность в значительном снижении уровня ложноположительных срабатываний (до 85%) по сравнению со сканированием только инструментом L3X.

## Сведения об авторах

**Петров С.Н.**, канд. техн. наук, доц., доц. каф. защиты информации, учреждение образования «Белорусский государственный университет информатики и радиоэлектроники», petrov@bsuir.by.

**Пакуль Е. С.**, учащийся, учреждение образования «Национальный детский технопарк», eduartpastado@gmail.com.

**Корчинский А. А.**, учащийся, учреждение образования «Национальный детский технопарк», ak47reborntumpa@gmail.com.

## Information about the authors

**Petrov S.N.**, Cand. of Sci., Associate Professor, Associate Professor of the Information Security Department, Belarusian State University of Informatics and Radioelectronics, petrov@bsuir.by.

**Pakul Y. S.**, student, Educational Institution "National Children's Technopark", eduartpastado@gmail.com.

**Korchinsky A. A.**, student, Educational Institution "National Children's Technopark", ak47reborntumpa@gmail.com.