DOI 10.37539/2949-1991.2025.29.6.017
УДК 4.056

Дмитриев Дмитрий Валерьевич,
к.т.н., доцент кафедры “Информатика и Системы Управления”,
Нижегородский государственный технический
университет им. Р.Е. Алексеева

Исаев Максим Александрович, магистрант,
Нижегородский государственный технический
университет им. Р.Е. Алексеева

Вайнбаум Денис Алексеевич, магистрант,
Нижегородский государственный технический
университет им. Р.Е. Алексеева

Мельников Роман Васильевич, магистрант,
Нижегородский государственный технический
университет им. Р.Е. Алексеева

ОЦЕНКА СРЕДСТВ МОДЕЛИРОВАНИЯ АУТЕНТИФИКАЦИИ ДЛЯ PYTHON BACKEND-ПРИЛОЖЕНИЙ

Аннотация. В статье представлены исследования методов имитационного моделирования для анализа уязвимостей в системах аутентификации. Проводится сравнение библиотек для моделирования с оценкой их эффективности и выявления наиболее оптимального варианта для проверки процессов аутентификации.

Ключевые слова: Имитационное моделирование, системы моделирования, аутентификация, уязвимости безопасности, атаки на приложения, защита от атак.

Введение

При исследовании механизмов аутентификации веб-приложений критически важно использовать инструменты, позволяющие реалистично имитировать поведение пользователей и злоумышленников. Для комплексного тестирования аутентификационных систем успешно применяются системы имитационного моделирования (СИМ) [1].

К основным преимуществам использования СИМ для анализа аутентификации относятся:

- Реалистичное моделирование процессов входа пользователей с различными сценариями поведения.
- Долгосрочное наблюдение за работой системы при разных нагрузках и типах атак.
- Гибкая настройка параметров аутентификации для поиска оптимальных решений.
- Прогнозирование поведения системы при масштабировании и разработка эффективных стратегий защиты.

Рассмотрим несколько специализированных систем имитационного моделирования, реализованных на языке Python и наиболее подходящих для тестирования серверных компонентов аутентификации.

Существующие системы моделирования на языке Python

SimPy позволяет моделировать системы, в которых события происходят в определенные моменты времени [2]. Это хорошо подходит для моделирования процессов аутентификации, где важны временные характеристики.

С помощью SimPy можно моделировать:

- Последовательность шагов в процессе аутентификации.

- Временные задержки при обработке запросов аутентификации.
- Очереди запросов на сервере аутентификации.
- Временные характеристики различных атак на систему аутентификации.

SimPy позволяет легко изменять параметры модели, такие как время обработки запросов, интервалы между попытками входа, время жизни токенов и т.д. Это помогает оценить производительность и безопасность различных механизмов аутентификации в разных условиях.

Пользователи с определенными интервалами времени пытаются войти в систему. Реализация системы проверки учетных данных происходит с учетом типа пользователя. Проверяется, имеет ли пользователь правильные учетные данные для доступа к системе. Также моделируются три типа механизмов безопасности, а именно: блокировка аккаунтов после нескольких неудачных попыток, защита от DDoS-атак и обнаружение подозрительной активности.

Результаты симуляции представлены в виде диаграмм, отображающих статистику аутентификации, распределение времени обработки запросов, активность по времени суток и эффективность механизмов безопасности.

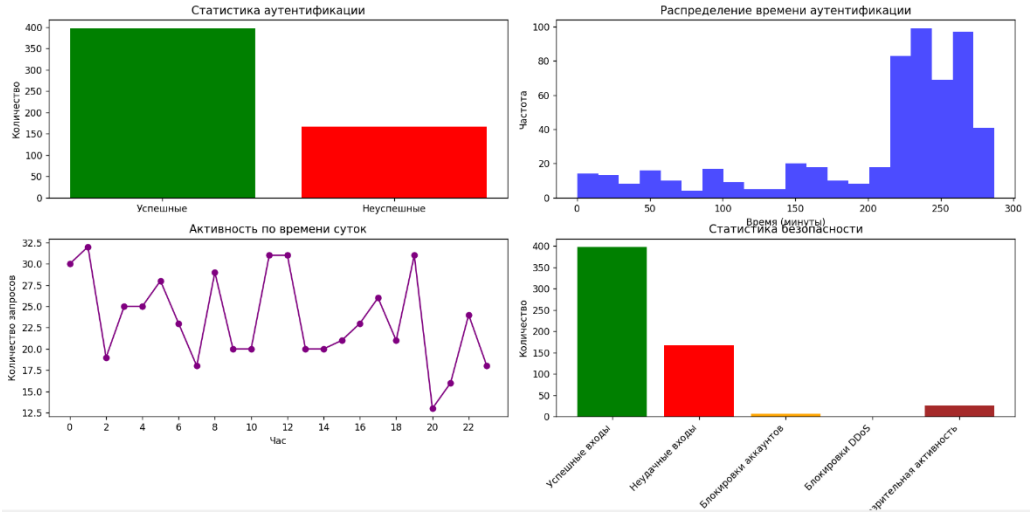

Рисунок 1. – Фрагмент вывода simpy результатов на график

Mesa позволяет создавать модели, где агенты (например, пользователи и серверы) [3] взаимодействуют в определенной среде (например, сетевая инфраструктура). Это дает возможность симулировать различные сценарии аутентификации и анализировать их безопасность.

С помощью Mesa можно моделировать:

- Различные протоколы аутентификации (например, OAuth, JWT).
- Атаки на механизмы аутентификации (брутфорс, перехват сессий).
- Поведение пользователей при аутентификации.
- Нагрузку на серверы аутентификации.

В модели каждый агент имеет свой график активности и вероятность корректного ввода учетных данных. Система аутентификации обрабатывает запросы с учетом ограниченной пропускной способности сервера и применяет различные механизмы защиты. Особое внимание уделяется моделированию пиковых нагрузок, когда частота запросов значительно возрастает.

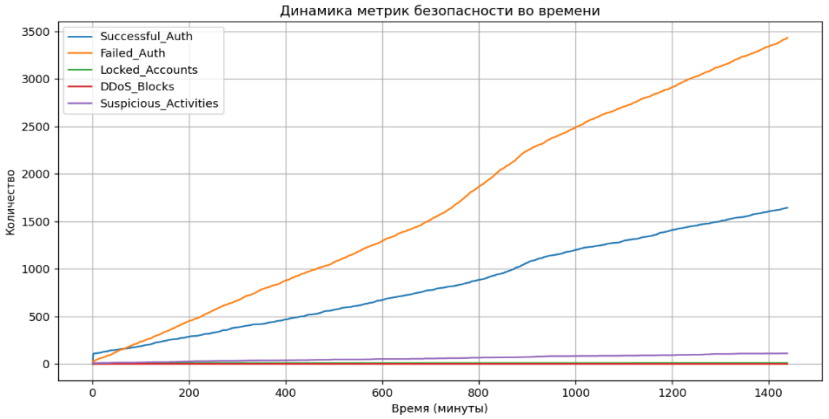

Рисунок 2. – Фрагмент вывода mesa графиков динамики метрик

PyCX предоставляет удобные инструменты для моделирования динамики систем [4], включая дискретные и непрерывные процессы, анализа поведения сложных сетевых и распределенных систем, создания интерактивных симуляций с возможностью изменения параметров в реальном времени и визуализации результатов моделирования.

С помощью PyCX можно моделировать динамику атак на системы аутентификации, такие как brute force, DDoS и session hijacking, исследовать влияние параметров безопасности, таких как время жизни токенов и задержки ответа сервера, на устойчивость системы, анализировать распределение нагрузки на серверы аутентификации при различных сценариях использования и оценивать эффективность различных протоколов, включая OAuth 2.0, JWT и SAML, в условиях повышенной нагрузки или атак.

В отличие от предыдущих моделей, PyCX предоставляет интерактивный интерфейс для наблюдения за процессом моделирования в реальном времени. Пользователь может видеть, как меняются ключевые метрики безопасности с течением времени, и наблюдать за динамикой системы в процессе симуляции.

Модель учитывает различные сценарии использования системы, включая пиковые нагрузки. Это позволяет оценить устойчивость системы к перегрузкам и эффективность механизмов защиты в условиях повышенной активности.

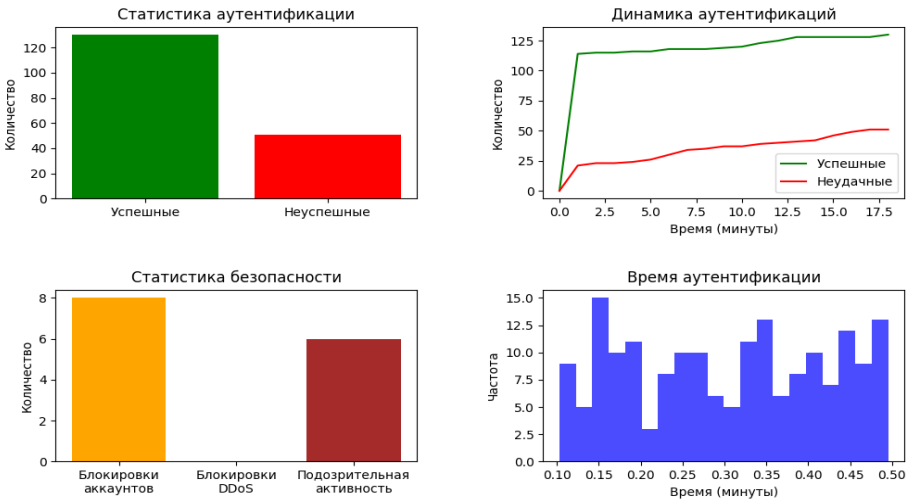

Рисунок 3. – Фрагмент вывода PYCX графиков результатов

Результаты моделирования позволяют проанализировать эффективность системы аутентификации в различных условиях и выявить потенциальные уязвимости при целенаправленных атаках или аномальных паттернах использования.

Заключение

Проведённое исследование позволило проанализировать и сравнить различные системы имитационного моделирования для изучения механизмов аутентификации в веб-приложениях, разработанных на языке Python. В ходе работы были рассмотрены три библиотеки: SimPy, Mesa и PyCX, каждая из которых обладает уникальными характеристиками и подходит для решения определённых задач.

SimPy продемонстрировала высокую эффективность при дискретно-событийном моделировании, особенно при анализе очередей запросов и временных характеристик процессов аутентификации. Однако её возможности ограничены в контексте моделирования сложных взаимодействий между множеством агентов, таких как пользователи, злоумышленники и администраторы системы.

Mesa, как агентно-ориентированный фреймворк, предоставила широкие возможности для моделирования динамических взаимодействий и анализа устойчивости системы к различным видам атак. Её гибкость и мощные инструменты визуализации делают её предпочтительным выбором для комплексного исследования механизмов аутентификации, особенно в условиях пиковых нагрузок и аномальных сценариев использования.

PyCX, несмотря на простоту использования и низкий порог входа, оказалась менее подходящей для масштабируемых и глубоких исследований из-за ограниченной функциональности и производительности.

Система аутентификации тестировалась с применением следующих механизмов защиты: блокировка после неудачных попыток входа – предотвращение атак методом подбора паролей; блокировка DDoS – защита от перегрузки системы массовыми запросами; эвристические алгоритмы – обнаружение подозрительной активности на основе анализа поведения пользователей.

Это позволило протестировать систему для получения ее актуального состояния и выявления уязвимостей механизмов безопасности.

Список литературы:

1. Banks J. Discrete-Event System Simulation / J. Banks, J. S. Carson II, B. L. Nelson, D. M. Nicol. – 5th ed. – Pearson, 2019. – 640 p.

2. Документация SimPy [Электронный ресурс]. – Режим доступа: https://simpy.readthedocs.io/en/latest/ (дата обращения 16.06.2025).

3. Документация Mesa [Электронный ресурс]. – Режим доступа: https://mesa.readthedocs.io/latest/ (дата обращения 16.06.2025).

4. Документация PyCX [Электронный ресурс]. – Режим доступа: https://github.com/hsayama/PyCX (дата обращения 16.06.2025).