# Методы совершенствования фаззинг-тестирования с применением машинного обучения

И.А. Мишин[1], О.А. Салтыкова[1]

[1]*Российский университет дружбы народов, Миклухо-Маклая 6, Москва, Россия, 117198*

**Аннотация**

Статья посвящена анализу фаззинг-тестирования – методу динамического бинарного кода программы. Проведенный анализ литературы позволяет утверждать, что, на сегодняшний день, автоматизированное фаззинг-тестирование является достаточно сложным с точки зрения построения алгоритмов, а также крайне востребованным процессом, с точки зрения информационной безопасности. Реализация данного метода анализа динамического бинарного кода программы с применением машинного обучения является наиболее предпочтительной, так как подразумевает достаточно тщательную работу с данными, а именно: анализ входных данных программы, мутация (видоизменение) данных, анализ отчетов об аварийном завершении программы. Проведенное исследование позволяет сделать вывод о том, что робастность является необходимым свойством современного программного обеспечения. Проведена классификация основных типов бинарных уязвимостей.

**Ключевые слова**

Машинное обучение, тестирование на проникновение, информационная безопасность, фаззинг-тестирование, бинарный код, бинарная уязвимость

## 1. Введение

В рамках создания и проектирования сложных информационных систем возникает потребность проверки этой системы на чувствительность ко входным данным, подаваемым некоторым ее компонентам. Случайно или злонамеренно, но иногда неправильно поданные входные данные могут привести к нештатным ситуациям во время работы программного обеспечения. Цель работы – установить проблематику использования автоматизированного фаззинг-тестирования и сделать вывод о необходимости улучшения метода с помощью техник машинного обучения.

## 2. Постановка задачи

Самым убедительным доводом в пользу использования машинного обучения [1, 2, 3] при проведении автоматизированного фаззинг-тестирования может быть его схожесть с алгоритмами «грубого» перебора [4]. Дело в том, что фаззинг подразумевает генерацию огромного количества входных данных, либо же побайтовое изменение данных, находящихся в некотором пуле для изначальной выборки. Подобные операции подразумевают использование огромного количества вычислительных мощностей, а также значительных временных затрат. В общем виде фаззинг-тестирование можно представить в виде схемы на Рисунке 1.

В настоящий момент остро стоит вопрос о робастности программного обеспечения. Робастность – это свойство некоего метода, характеризующее независимость влияния различных помех на результат работы. Схожим свойством обладает и программное обеспечение. Как показывает практика, неправильно сконфигурированные входные данные могут привести не только к некорректным результатам работы программы, но и к её

нештатному поведению в ходе выполнения (возникновение уязвимостей, аварийное завершение и пр.)


**Рисунок 1**: Алгоритм тестирования приложения методом фаззинга

## 3. Вывод

В работе представлена классификация основных типов бинарных уязвимостей, существующих на сегодняшний день. Наглядно показано, как может повлиять эксплуатация этих уязвимостей на процесс работы программного обеспечения, в связи с чем можно определить исторические предпосылки появления такого метода динамического анализа, как автоматизированное фаззинг-тестирование.

## 4. Литература

[1] Мерков, А.Б. Введение в методы статистического обучения / А.Б Мерков. – М: Едиториал УРСС, 2011. – 254 с.
[2] Рубан, А.И. Методы анализа данных: учебное пособие / А.И. Рубан. – Красноярск: ИПЦ КГТУ, 2004. – 319 с.
[3] Саттон, Р.С. Обучение с подкреплением / Р.С. Саттон, Э.Дж. Барто. – М.: ДМК. Пресс, 2020. – 552 с.
[4] Амини, П. Fuzzing: исследование уязвимостей методом грубой силы / П. Амини, М. Саттон, А. Грин. – СПб.: Символ Плюс, 2009. – 506 с.