УДК 004.9

АНАЛИЗ МЕТОДИК ОЦЕНКИ ИСПОЛЬЗОВАНИЯ ПРОГРАММНЫХ ИНТЕРФЕЙСОВ ПРИЛОЖЕНИЙ НА ЦЕЛЕСООБРАЗНОСТЬ ПРИМЕНЕНИЯ В ГОТОВЫХ ПРОГРАММНЫХ ПРОДУКТАХ С ТОЧКИ ЗРЕНИЯ БЕЗОПАСНОСТИ

Миронов М.А.

Федеральное государственное бюджетное образовательное учреждение высшего образования «МИРЭА – Российский технологический университет» (РТУ МИРЭА), 119454, Россия, г. Москва, проспект Вернадского, 78, e-mail: maxmirrenga@gmail.com

Информационно-коммуникационные технологии делают нашу жизнь проще, однако, также ставят под угрозу личные данные тех, кто ими пользуется. Основной причиной такого противоречия являются ошибки в коде программ, которые заключаются в проблеме возможного небезопасного использования программных интерфейсов приложений (API). Такие ошибки делают программное обеспечение уязвимым для злоумышленников. В данной статье автором ставится задача: приблизить разработчиков к решению этой проблемы, рассмотрев и оценив существующие методики оценки удобства использования и определения проблем безопасности API.

Ключевые слова: API, оценка API, безопасность API, методики оценки API, исследования пользователей.

ANALYSIS OF EVALUATION METHODS OF USE APLICATION PROGRAMMING INTERFACES ON REASONABILITY OF APPLYING IN FINISHED SOFTWARE PRODUCTS FROM THE SECURITY POINT OF VIEW

Mironov M.A.

Federal State Educational Institution of Higher Education "MIREA – Russian Technological University" (RTU MIREA), 119454, Russia, Moscow, Vernadscogo avenue, 78 e-mail: maxmirrenga@gmail.com

Information and communications technologies make our lives easier, however, they also endanger the personal data of those who use them. The main reason for this contradiction is errors in the program code, which are the problem of the possible insecure using of application programming interfaces (APIs). Such errors make software vulnerable to malefactors. In this article, the author sets the task: to bring developers closer to solving this problem by proposing a methodology for evaluating usability and identifying API security issues.

Key words: API, API evaluation, API security, API evaluation methodologies, user studies.

Введение

Поскольку большинство программистов, участвующих в разработке программного обеспечения (ПО), не являются специалистами в области кибербезопасности, то иногда использование ими API сторонних разработчиков, приводит к ошибкам, создающим проблемы с безопасным функционированием разрабатываемых приложений. Так, специалист по безопасности С. Фал вместе с коллегами провел анализ 13 500 Android-приложений и выявил, что 8% из них имеют уязвимости [5]. Также, было установлено, что это напрямую связано с ошибками программистов, включивших API в отсутствии логических доводов для их удобного использования. В аналогичном эксперименте были выявлены уязвимости в приложениях iOS, вызванные ошибками программистов, использовавших API, которые обеспечивали функциональность, связанную с задействованием криптографических протоколов передачи данных (SSL) и (TLS) [5].

В настоящее время нет единственной признанной методики оценки удобства использования API с точки зрения безопасности. Хотя, в практику введены несколько общих методик оценки удобства использования API для разрабатываемого приложения [3-8], но исследования на целесообразность их применения с точки зрения

безопасности в различных ситуациях не проводились. Таким образом, разработка подобной методики является актуальной задачей.

Автором статьи был проведен обзор литературы по существующим методикам оценки удобства использования API. После чего, были выделены пять тех, которые предназначены для применения в различных условиях разработки программных интерфейсов приложений. Далее, проведен анализ методик на их возможное использование в оценке безопасности интегрирования API. Наиболее подходящая методика была описана более детально и скорректирована для тестирования API на безопасность.

1. Методики оценки общего удобства использования API

1.1 Методика исследования пользователей

Исследования в области оценки удобства использования API проводятся совместно с теми, кто будет использовать такие интерфейсы [6]. Такая методика подразумевает набор программистов, которым даются индивидуальные задания, выполнение которых требует использование оцениваемого API. Задания могут отличаться по своей концепции работы с кодом, где используется анализируемый API: написание, чтение и его отладка. Удобство использования отмечается путем отслеживания действий испытывающих пользователей, а также содержательным обзором проделанной работы по окончании выполнения заданий. Некоторые усовершенствованные методики включают коллективное обсуждение решений обнаруженных проблем с API, а также коллективную разработку исправлений [2].

Эксперт по безопасности в области использования API С. Кларк утверждает, что Microsoft анализирует данные обратной связи с пользователями по данной тематике с помощью опросника, основанного на Cognitive Dimensions Framework [4]. Таким образом, как только программист закончил выполнение задания, ему сразу предлагается ответить на список вопросов, каждый из которых подвергается анализу. Разработчик ПО М. Пиччиони совместно со своими коллегами также использовал подобную методику, однако с несколько другой структурой: он предлагал больше вариантов исследований, которые используют невероятные по своему количеству типы задач [2].

Так как исследования такого рода используют реальных программистов в процессе оценки, то возможно выявить реальные проблемы, которые бы могли испытывать все остальные разработчики при использовании анализируемых API в приложениях. Именно поэтому такой подход считается фактически стандартным в процедуре выявления изъянов для применения интерфейсов в тех или иных случаях. Опираясь на полученные отзывы программистов, становится возможным понять: что именно заставляет использовать API не так, как это задумывалось его создателями, и как улучшить удобство использования [7]. Большим и основным недостатком такой методики является то, что привлечение потенциальных испытателей API является дорогостоящим [4]. Более того, для многих из приглашенных программистов задание может оказаться невыполнимым, так как не многие способны применять API, для использования которых необходимо большое количество интерфейсов, классов и методов.

1.2 Методика эвристической оценки

В отличие от предыдущей методики, эвристическая оценка является куда менее затратной. Кроме того, её возможно использовать для оценки, как достаточно объемных API, так и сравнительно небольших. Суть методики заключается в том, что эксперт в области обеспечения удобства использования API и разрабатываемом приложении, проверяет интерфейс в соответствии с набором эвристик и идентифицирует в нем потенциальные проблемы. Так, исследователь в области оценки API Т. Грилл совместно со своими коллегами провели испытание API на удобство использования, в котором использовалась эвристическая оценка для определения возможных уязвимостей в случае некорректного использования интерфейса [7].

Однако, проведенные исследования показывают, что выявленные проблемы при данной методики, существенно отличаются от тех, с которыми разработчики сталкиваются в реальной жизни. Также было установлено, что при эвристической оценке значительное внимание уделяется второстепенным вопросам, нежели чем главным, таким как безопасность использования [7]. Несмотря на то, что данная методика довольно часто находит применение в наши дни в оценке удобства использования API конечными пользователями, в исследованиях недостаточно доказательств, подкрепляющих необходимость использовать эвристическую оценку для выявления коренных проблем API [7].

1.3 Методика прохода API

Данная методика проводится также с использованием приглашенных программистов, как и в методики «Исследования пользователей» [3]. Однако, процесс отличается тем, что координатор испытаний требует от приглашенных последовательно объяснить в соответствии с их понимаем весь исполняемый код, который использует API. Объяснение возможно как устно, параллельно с проходом по коду, так и по завершению

задачи, в письменном виде. Специалист по использованию данной методики П. О'Каллаган отмечает, что она успешно используется в Matlab для оценки удобства использования API, которые они предоставляют [3].

Методика прохода API может быть использован на ранних стадиях разработки интерфейса, даже до начала внедрения и написания документации [3]. Тем самым, ошибки могут быть выявлены на этапе проектирования, что позволяет легко беспрепятственно интегрировать данную методику оценки API в жизненный цикл проекта. Важно отметить, что такой путь занимает меньше времени и требует, куда меньших вложений, чем методика исследования пользователей. Однако большим недостатком является отсутствие сфокусированности на документации к API, что является предпосылкой к появлению проблем, связанных безопасностью в приложениях, где используется интерфейс [3]. Это несомненно связанно с возможной некорректной интерпретацией информации в руководствах по применению.

1.4 Методика концептуальных карт API

Д. Геркен совместно со своими коллегами представил методику оценки удобства использования API, названную методикой концептуальных карт [9]. В данной методике участникам предстоит создать карту, отображающую связь между API и их кодом, который может быть задачей или частью реального приложения. В течение 30-60 минут такой сессии каждый участник должен повторять такую процедуру один раз в неделю общей длительностью в пять недель и, на основе этого, обновлять концептуальную карту. Потенциальные проблемы идентифицируются путем анализа концептуальных карт, созданных участниками. Авторы данной методики отмечают, что они успешно использовали её для оценки удобства использования API под названием Zoomable Object-Oriented Information Landscape (ZOIL). Данная методика не только позволяет выделить проблемы для программиста, который впервые использует API, но и для того, у кого уже есть опыт работы с данным API [8], [9]. Таким образом, методика концептуальных карт определяет проблемы, связанные с такими аспектами, как запоминаемость и обучаемость, которые другие методики не способны выявить. Однако, одним из ограничений такой методики является то, что участникам требуется повторить задачу в течение пяти недель, что означает, что для всей оценки требуется довольно продолжительный интервал времени.

1.5 Методика автоматизированной оценки

Помимо приведенных выше методика, было разработано еще несколько инструментов для оценки удобства использования API, которые в комплексе работают автоматически. К. Р. Де Соуза и Д. Л. Бентолила представили инструмент под названием Matrix, который оценивает удобство использования API, вычисляя его сложность [1]. Данный инструмент предполагает, что удобство использования определяется функцией сложности, которая, в свою очередь, зависит от значений ввода и вывода методов, классов, пакетов и структуры самого API [1]. По сравнению с другими методиками автоматизированная оценка удобства использования API является более дешевой, ограничивается стоимостью соответствующего ПО, менее трудоемкой и не требует опытных оценщиков, приглашенных тестирующих пользователей или уже полностью реализованного API. Кроме того, данную методику можно также легко интегрировать в процесс разработки программного обеспечения, как и в случае с методикой прохода API. Существующие инструменты учитывают только сложность оценки удобства использования, однако, этого недостаточно для проверки API на безопасность применения.

2. Сравнение методологий оценки удобства использования API

Ниже изображена таблица 1, которая показывает сильные и слабые стороны методологий оценки удобства использования API. Угрозы уязвимости систем происходят по причине проблем, связанных с удобством использования API с точки зрения безопасного функционирования, что, в свою очередь, ведет к возникающим ошибкам при включении таких интерфейсов в реальные приложения. Поэтому, можно утверждать, что вовлечение возможных пользователей API в процесс такой оценки имеет крайне важное значение в идентификации проблемных мест [2]. Как видно в таблице 1, три методики используют приглашенных программистов в процессе оценки: исследования пользователей, проход API и методика концептуальных карт API. Данное свойство является наиболее весомым по сравнению с другими, тем самым, для рассмотрения иных методологий, в которых не участвуют программисты, нет никакой необходимости. Оставшиеся три методики были вынесены в отдельную таблицу 2 со свойствами, значения у которых отличались друг от друга.

Таблица 1 - Результаты общей оценки удобства использования API.

Свойство/ Методика	**Методика исследования пользователей**	**Методика эвристической оценки**	**Методика прохода API**	**Методика концептуаль-ных карт API**	**Методика автоматизирован-ной оценки**
Использование	Да	Нет	Да	Да	Нет

пользователей (программистов) при оценке					
Протестировано программистами для общей оценки удобства использования	Да	Нет	Нет	Нет	Нет
Низкие затраты	Нет	Да	Да	Нет	Да
Подходит для оценки API со сложной структурой	Нет	Да	Нет	Нет	Да
Низкозатратно по времени	Нет	Да	Да	Нет	Да
Определение проблем, связанных с чрезмерной сложностью	Да	Нет	Да	Да	Нет
Возможность использовать на ранних стадиях разработки	Нет	Да	Да	Нет	Да
Простота внедрения в жизненный цикл ПО	Нет	Нет	Да	Нет	Да
Возможность обнаружить большое количество разнородных ошибок	Да	Да	Нет	Нет	Нет
Оценки документации API	Да	Да	Нет	Нет	Нет
Опытные оценщики не требуются	Нет	Нет	Нет	Нет	Да
При оценке, API не обязательно должен быть окончательно готов	Нет	Да	Да	Да	Да
Определение проблем запоминаемости, обучаемости	Нет	Нет	Нет	Да	Нет

Таблица 2 - Методики, использующие пользователей программистов при оценке.

Свойство/ Методика	Методика исследования пользователей	Методика прохода API	Методика концептуальных карт API
Протестировано программистами для общей оценки удобства использования	Да	Нет	Нет
Низкие затраты	Нет	Да	Нет
Подходит для оценки API со сложной структурой	Нет	Нет	Нет
Низкозатратно по времени	Нет	Да	Нет
Возможность использовать на ранних стадиях разработки	Нет	Да	Нет
Простота внедрения в жизненный цикл ПО	Нет	Да	Нет
Возможность обнаружить большое количество разнородных ошибок	Да	Нет	Нет
Оценки документации API	Да	Нет	Нет
При оценке, API не обязательно должен быть окончательно готов	Нет	Да	Да
Определение проблем запоминаемости, обучаемости	Нет	Нет	Да

По значениям свойств методики концептуальных карт API, данная методика обладает крайне большим количеством недостатков, следовательно, не подлежит дальнейшему рассмотрению. Что касается двух оставшихся, то стоит применить принцип взвешенной суммы с предварительным переводом качественных параметров в количественные и удалить свойства с одинаковыми значениями для выбора методики, подходящей для оценки удобства использования API с точки зрения безопасного функционирования.

В таблице 3 приведены итоги применения принципа взвешенной суммы. Первым шагом были удалены свойства, значения которых для рассматриваемых методологий совпадают. Вторым шагом были заменены значения «Да» на 1 и «Нет» на 0, так как наличие положительного значения в параметре является плюсом к методики. Третьим шагом были присвоены коэффициенты (веса) каждому из параметров. Тестирование API в работе является самым важным этапом в проверке и оценке его функционирования, именно поэтому первое свойство в таблице 3 получило максимальный вес. Второй по значению вес получило свойство «Оценки документации API», так как, прежде чем приступить к включению API в свой проект, любой начинает с прочтения документации. Обнаружение большого количества разнородных ошибок отошло на второй план, так как обнаруживаемый уровень ошибок может не столь критично влиять на безопасность использования API. Также немаловажно то, что процесс оценки будет стоить разумных денег. Низкозатратность по времени тоже играет немаловажную роль, так как принятие в эксплуатацию [4], в данном случае API, не должно негативно влиять на график работ по проекту. Планирование оценки тоже важный шаг, что означает, что данную оценку возможно запланировать, выделив на нее определенное время и ресурсы. Возможность использования на ранних стадиях разработки безусловно важна [7], однако, уровень отлавливаемых ошибок будет оставлять желать лучшего. Именно по этой же причине проверка на промежуточных стадиях готовности API не дает требуемых результатов оценки.

Таблица 3. Выбор лучшей методики для оценки удобства использования API с точки зрения безопасности с помощью принципа взвешенной суммы.

Свойство/Методика	**Вес (коэффициент)**	**Методика исследования пользователей**	**Методика прохода API**
Протестировано программистами для общей оценки удобства использования	0.19	1	0
Низкие затраты	0.13	0	1
Низкозатратно по времени	0.11	0	1
Возможность использовать на ранних стадиях разработки	0.07	0	1
Простота внедрения в жизненный цикл ПО	0.09	0	1
Возможность обнаружить большое количество разнородных ошибок	0.16	1	0
Оценки документации API	0.2	1	0
При оценке, API не обязательно должен быть окончательно готов	0.05	0	1
Сумма весов	1	0.54	0.45

Основываясь на сумме весов, полученной у методики исследования пользователей (0.54), можно сделать вывод о том, что эта методика - наиболее подходящая для оценки удобства работы с интерфейсами приложения, в том числе и с точки зрения безопасности, так как безопасность функционирования во многих случаях ставится как наиболее приоритетная задача.

3. Проведение оценки безопасности использования API по методике исследования пользователей с помощью CDF

Методика исследования пользователей предполагает собой четыре этапа:

1) Разработка задач для приглашенных программистов;

2) Рекрутинг участников и проведение оценки.
3) Сбор отзывов от участников.
4) Определение проблем удобства использования из отзывов участников.

В следующих подразделах описывается, как выполнять каждый из шагов предлагаемой методики.

Шаг 1. Оценка удобства использования API, основанная на пользовательских исследованиях, требует от участников выполнения некоторых задач с использованием программного интерфейса. В эти задачи может входить написание кода, его отладка, либо анализ [2], [6]. Так, в оценке удобства использования API С.Г. МакЛилан совместно со своими коллегами использовали задачу, суть которой заключалась в прочтении программы, которая была разработана с использованием оцениваемого API [6]. Оценивая удобство использования API, команда разработчиков, возглавляемая М. Пиччиони, использовала задачи, которые требовали от участников произвести доступ к реляционной базе данных с использованием предлагаемых функций API [2]. Выбранная методика будет использовать аналогичные задачи программирования для оценки удобства использования API с точки зрения безопасности.

Шаг 2. Приглашенные программисты должны иметь опыт разработки ПО. Также это должны быть те люди, которые будут использовать API после завершения его разработки. Рекрутинг лучше проводить таким образом, чтобы набирать программистов, которые свободно владеют необходимым языком программирования. Количество участников, требующихся для этого шага – зависит от самой сложности API: чем больше функционала предлагается, тем больше необходимо протестировать, причем в различных ситуациях.

Шаг 3. Как только программист завершает задачу, оценщики сразу же должны получить отзывы об API от участника [4]. Заранее предопределенные вопросники, спланированные интервью и наблюдение за участниками при выполнении задач являются методами, используемыми для сбора сведений об участниках и опыте их работы с API.

Microsoft использует данную методику, в которой они задействовали предопределенный опросник на платформе CDF для оценки удобства использования их API [2], [6]. Использование такого вопросника имеет несколько преимуществ перед созданием конкретного вопросника для каждой оценки. Это снимает необходимость составлять вопросы заново для каждого API [4]. Кроме того, поскольку такой опросник является универсальным, возможно делать сравнения об успешности различных API и их версий. Таким образом, автор статьи рекомендует использовать предопределенный общий вопросник на основе CDF для сбора отзывов об удобстве использования API.

Однако упомянутый выше опросник не может использоваться для оценки удобства использования API с точки зрения безопасности, поскольку соответствующие тесты и вопросы не включены в структуру. Таким образом, рекомендуется расширить структуру, включив следующие измерения: рамки использования API, представленный уровень абстракции кода API, защищенность пользователей, защита от злоупотребления использованием и защита от доменной корреспонденции (подмены домена на домен-двойник с целью кражи информации).

Шаг 4. Поскольку опросник является универсальным, кроме свободной формы ответа в нем обязательно должны присутствовать варианты ответов, предоставленные составителями вопросов. Данная конфигурация позволяет построить статистическую зависимость и выявить главные проблемы, с которыми столкнулось определенное количество испытуемых программистов.

Заключение

В данной статье был проведен обзор пяти методик общей оценки удобства использования API с целью выявления наиболее эффективной для оценки интерфейсов приложений с точки зрения безопасности. С помощью принципа взвешенной суммы была выделена методика исследования пользователей. После чего, был проведен её дальнейший анализ, на основе которого было предложено внести определенные коррективы, связанные с добавлением тематических исследований, связанных с вопросами безопасности, и определить нюансы использования, такие как задействование опросника по типу CDF. Однако, важно подчеркнуть, что использование выбранной методики может оказаться нецелесообразным для оценки удобства использования слишком сложных по структуре API с точки зрения безопасности, так как это ощутимо повышает стоимость проведения оценки вследствие необходимости пригласить больше программистов и провести больше тестов и опросов, требующих составления и анализа.

Список литературы

1. К.Р. Де Соуза и Д. Л. Бентолила «Автоматическая оценка удобства использования API с использованием показателей сложности и визуализации» / - В.: ICSE-Companion, 2015 - 302 с.

2. М. Пиччони, С.А. Фурия, Б. Мейер. «Эмпирическое исследование удобства использования API» / - Е.: ACM / IEEE, 2014 - 248 с.
3. П. О'Каллаган. «Методика прохода API: легкий метод получения ранней обратной связи по API» / - Е.: ACM, 2016 - 360 с.
4. С. Кларк, «Измерение удобства использования api» / - Г.: S1S5, 2014 - 203 с.
5. С. Фал, М. Харбач, М. Смит. «Анализ Android безопасноти SSL» / - Е.: ACM, 2014 - 407 с.
6. С.Г. МакЛилан, А.В. Роеслер. «Создание более удобных API» / Б.: IEEE, 2015 – 680 с.
7. Т. Гриль, О. Полачек и М. Челиги. «Методы применимости API: структурный анализ категорий проблем удобства использования» / Б.: Springer, 2015 – 385 с.
8. Хабр: Проектирование Web API в 7 шагов [Электронный ресурс]. — Режим доступа: https://habr.com/company/geekfamily/blog/256495/ (дата обращения: 17.10.18).
9. Ю. Геркен, Н. Джеттер, Х. Рейтерер “Методика концептуальных карт как инструмент оценки удобства использования API.” / Б.: SIGCHI-ACM, 2017 – 342 с.

References

1. K. R. De Souza, I. L. D has Bentolila "Automatic evaluation of the usability of the API using complexity metrics and visualizations" / - B.: ICSE-companion 2015 - 302 S.
2. M. Piccioni, C. A. Furia, B. Meyer. "An empirical study of the usability of the API"/.: ACM / IEEE, 2014 - 248 p.
3. P. O'callaghan. "Method iterate over API: easy method of obtaining early feedback on the API"/.: ACM, 2016 - 360p.
4. S. Clarke, "Measuring usability API" / - G: S1S5, 2014 - 203 C.
5. S. FAL, M. Harbach, M. Smith. "Analysis of the Android security Protocol SSL"/.: ACM, 2014 - 407 p
6. S. G. MacLean, A.V. Roesler. "Creating a more convenient API": IEEE standard, 2015 – 680 p.
7. T. Grill, O. Polacek M I. Celigi. "API methods for applicability: a structural analysis of categories of usability problems", 2015, Springer.
8. Habr: web API Design in 7 steps [Electronic resource]. — Mode of access: https://habr.com/company/geekfamily/blog/256495/ (accessed: 17.10.18).
9. Yerkin, N. Jeter, H. Reuter " the conceptual map Method as a tool for evaluating the usability of the API.” / B.: SIGCHI-AKM, 2017 – 342 p.