OWASP
Open Web Application Security Project

Standard

Mobile AppSec Verification

Version 1.1

Project Leaders: Bernhard Mueller and Sven Schleier

Содержание

Вступительное слово от Bernhard Mueller, OWASP Mobile Project

Технологические революции могут происходить молниеносно: менее десятилетия назад смартфоны были громоздкими устройствами с клавиатурой, и являлись дорогими игрушками для некоторых компаний. Сегодня же смартфоны прочно вошли в нашу жизнь. Мы доверяем им информацию, навигацию, коммуникацию, они являются неотъемлемой составляющей как бизнеса, так и жизни в целом.

Каждая новая технология приносит новые риски безопасности, и попытка поспеть за этими изменениями и есть один из основных вызовов, которые стоят перед индустрией безопасности. Сторона защиты всегда на несколько шагов позади. Например, обычной реакцией многих была бы попытка применить старый подход: смартфоны - это маленькие компьютеры и мобильные приложения - обычное ПО, значит требования безопасности абсолютно такие же? Но так это не работает. Операционные системы смартфонов отличаются от ОС компьютеров и мобильные приложения отличаются от веб приложений. Например, классический метод поиска вирусов на основе сигнатурного анализа не имеет смысл в современных мобильных операционных системах: не только потому, что это не совместимо с моделью распространения мобильных приложений, но еще и потому, что это технически невозможно из-за ограничений среды выполнения приложений (sandbox). Кроме того, некоторые типы уязвимостей, такие как переполнение буфера и XSS, менее релевантны к мобильным приложениям, нежели к десктопным и веб-приложениям (есть исключения).

Со временем наша индустрия получила больше опыта в борьбе с мобильными угрозами. Как оказалось, мобильная безопасность в основном затрагивает хранение данных: приложения хранят нашу личную информацию, изображения, записи голоса и видео, заметки, учетные данные, информацию о бизнесе, местоположения и многое другое. Приложения играют роль клиентов, подключающих нас к сервисам, которыми мы используем ежедневно, и коммуникационных центров, обрабатывающих каждое сообщение, которым мы обмениваемся. Скомпрометировав телефон человека, вы получите неограниченный доступ к его личной жизни. Если учесть, что мобильные устройства легко теряются и похищаются, а мобильные вирусы сейчас активно развиваются, необходимость защиты данных становится ещё более очевидной.

Стандарты безопасности мобильных приложений должны сконцентрироваться на том, как они обрабатывают, хранят и защищают чувствительную информацию. Несмотря на то, что современные мобильные операционные системы, такие как iOS и Android, предоставляют API для безопасного хранения данных и сетевого взаимодействия, чтобы быть эффективным, оно должно быть корректно использовано. Хранение данных, межпроцессное взаимодействие, правильное использование криптографического API и безопасное сетевое взаимодействие - это только некоторые из аспектов, которые требуют внимательного рассмотрения.

Важный вопрос для нашей индустрии, в котором необходимо достичь соглашения: как далеко должны заходить специалисты в вопросах защиты конфиденциальности и целостности данных. Например, многие из нас согласятся, что мобильное приложение должно проверять сертификат сервера во время соединения TLS. Но что насчёт SSL pinning? Невыполнение этого требования ведёт к уязвимости? Должно ли это быть требованием, если приложение обрабатывает чувствительную информацию или это может быть контрпродуктивно? Нужно ли шифровывать данные, хранящиеся в SQlite, несмотря на то, что ОС выполняет приложение в песочнице (sandboxing)? То, что подходит одному приложению, может не подходить для другого. MASVS - это попытка стандартизации требований на основании уровней проверок, подходящих для разных моделей угроз.

Появление вредоносных программ и инструментов удаленного управления заставляют осознать тот факт, что мобильные ОС имеют недостатки, которыми могут воспользоваться злоумышленники. Для дополнительной защиты чувствительных данных и предотвращения фальсификаций на стороне клиента всё шире используются стратегии контейнеризации. Это место, где всё усложняется. Аппаратные и программные

решения, такие как Android for Work и Samsung Knox, существуют, но они доступны не на всех устройствах. В качестве решения можно реализовать дополнительные программные механизмы защиты, но, к сожалению, не существует стандартов или руководств по тестированию для верификации таких методов.

В результате при тестировании информационной безопасности мобильных приложений постоянно возникают разногласия. Например, некоторые тестировщики приложений Android считают, что недостаточная обфускация или отсутствие детектирования root-доступа серьезными уязвимостями. Другие считают, что такие меры, как шифрование строк, обнаружение отладчика или обфускация не являются обязательными. Однако, этот двоякий взгляд не имеет смысла, так как меры защиты зависят от конкретной модели угроз для клиентского приложения. Защита на программном уровне не является бесполезной, но в конечном итоге её всегда можно обойти, так что она не должна быть использована как замена основным требованиям безопасности.

Цель MASVS - предложить общий фундамент для безопасности мобильных приложений (MASVS-L1), усиленные меры защиты (MASVS-L2) и меры защиты против угроз на стороне клиента (MASVS-R). MASVS предназначен для достижения следующих целей:

- Обеспечить требования для архитекторов и разработчиков ПО, стремящихся создавать безопасные мобильные приложения.
- Предложить промышленный стандарт, по которому можно проводить аудит безопасности мобильных приложений.
- Прояснить роль механизмов защиты ПО в мобильной безопасности и предоставить требования для проверки их эффективности.
- Предоставить конкретные рекомендации, для разных уровней безопасности, которые зависят от конкретного варианта использования.

Мы пониманием что 100% согласия в отрасли невозможно достичь. Однако, мы надеемся, что MASVS будет полезен в качестве руководства на всех этапах разработки и тестирования мобильных приложений. MASVS, как открытый стандарт, будет развиваться со временем, и мы рады любым предложениям и вкладу в развитие проекта.

Фронтиспис

О стандарте

Добро пожаловать в стандарт проверки безопасности мобильных приложений (MASVS) 1.1. MASVS - это результат совместных усилий в создании перечня требований информационной безопасности, необходимых для проектирования, разработки и тестирования мобильных приложений на iOS и Android.

MASVS - итог усилий сообщества и обратной связи от представителей индустрии. Мы надеемся на развитие стандарта в будущем и приветствуем обратную связь от сообщества. Лучший способ связаться с нами - через канал OWASP Mobile Project в Slack-е:

https://owasp.slack.com/messages/project-mobile_omtg/details/.

Аккаунты можно создать по этому адресу:

http://owasp.herokuapp.com/.

Авторское право и лицензия

Руководитель проекта	Главные авторы	Авторы и рецензенты	Перевод на русский	Рецензенты перевода на русский
Bernhard Mueller, Sven Schleier	Bernhard Mueller	Stephen Corbiaux, Sven Schleier, Jeroen Willemsen, Anant Shrivastava, Abdessamad Temmar, Alexander Antukh, Roberto Martelloni, Stefaan Seys, Prabhant Singh, Francesco Stillavato, Abhinav Sejpal	Gall Maxim	Oprya Egor, Chelnokov Vladislav, Tereshin Dmitry

Работа над документом была начата как ответвление OWASP Application Security Verification Standard, написанного Jim Manico.

The Mobile Application Security Verification Standard

MASVS может быть использован для подтверждения определенного уровня уверенности в безопасности мобильных приложений. Данные требования были сформированы, ориентируясь на следующие цели:

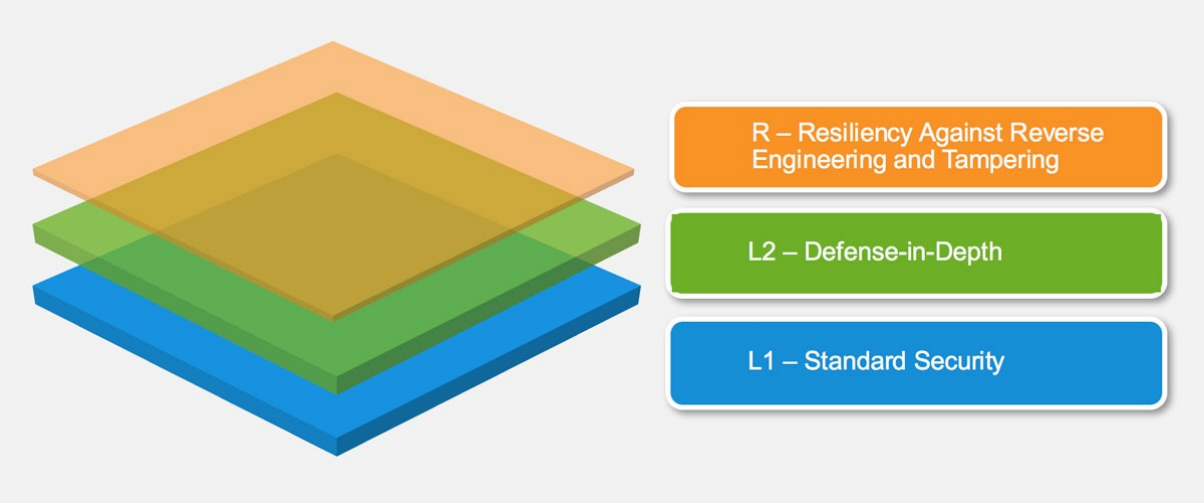
- Использование в качестве метрики: для предоставления стандарта безопасности, по которому разработчики и владельцы мобильных приложений могут проверить свои продукты;
- Использование в качестве руководства: для предоставления рекомендаций во время всех этапов разработки и тестирования;
- Использование во время закупок: как основание для проверки безопасности мобильного приложения.

Модель безопасности мобильных приложений

MASVS определяет 2 строгих уровня проверки безопасности (L1 и L2), а также набор гибких требований для обеспечения защиты от реверс инжиниринга (MASVS-R), адаптируемых для конкретной модели угроз. MASVS-L1 и MASVS-L2 содержат общие требования к безопасности и рекомендуются для всех мобильных приложений (L1) и приложений, работающих с чувствительными данными (L2). MASVS-R охватывает дополнительные меры защиты, которые могут применяться в случае, если предотвращение атак на стороне клиента заложено в дизайн приложения.

Приложение, разработанное в соответствии с требованиями MASVS-L1, защищено лучшими практиками безопасности и не содержит часто встречающиеся уязвимости. MASVS-L2 предусматривает усиленные меры защиты, такие, как SSL pinning, защищающие приложение от более ухищренных атак, при условии, что безопасность операционной системы не скомпрометирована, а конечный пользователь не рассматривается как потенциальный злоумышленник. Реализация всех или хотя бы некоторых защитных техник из MASVS-R затрудняет атаки на стороне клиента, когда конечный пользователь является злоумышленником и/или ОС мобильного устройства скомпрометирована.

Обратите внимание, что программные меры защиты, перечисленные в MASVS-R и OWASP Mobile Testing Guide, можно обойти, и они не должны использоваться в качестве замены основным требованиям безопасности. Вместо этого они должны быть реализованы как специализированное дополнение, направленное на локализацию конкретных угроз, в мобильном приложении, соответствующем требования MASVS L1 или L2.



Структура документа

Первая часть MASVS содержит описание модели безопасности и доступных уровней проверки, рекомендации о том, как использовать стандарт на практике. Подробные требования безопасности и их соответствие с уровнями проверки перечислены во второй части. Требования сгруппированы в восемь категорий (V1 - V8) по принадлежности к технической задаче/области. В MASVS и MSTG используется следующая номенклатура:

- *Категория требований:* MASVS-Vx, например, MASVS-V2: хранение данных и конфиденциальность
- *Требование:* MASVS-Vx.y, например, MASVS-V2.2: «Чувствительные данные не попадают в логи приложения».

Подробное описание уровней верификации

MASVS-L1: Стандартная безопасность

Приложение, удовлетворяющее MASVS-L1, соответствует лучшим практикам обеспечения безопасности мобильных приложений. Данный уровень проверки предоставляет основные требования с точки зрения качества кода, обработки чувствительных данных и взаимодействия с мобильной средой. Для подтверждения соответствия приложения данному уровню безопасности необходимо проведение тестирования. Этот уровень подходит для всех мобильных приложений.

MASVS-L2: Усиленная защита

MASVS-L2 предлагает расширенные средства проверки безопасности, выходящие за рамки стандартных. Для соответствия L2, необходима сформированная модель угроз, и безопасность должна быть неотъемлемой частью архитектуры и дизайна приложения. Этот уровень подходит для приложений, работающих с чувствительными данными, таких, как мобильный банк.

MASVS-R: Устойчивость к реверс инжинирингу и фальсификациям

Приложение имеет встроенные механизмы обеспечения безопасности, устойчиво к конкретным атакам на стороне клиента, таким, как фальсификация, модификация или реверс инжиниринг, направленным на извлечение чувствительных участков кода или данных. Такое приложение либо использует аппаратные средства безопасности, либо достаточно надёжные программные механизмы защиты. MASVS-R применим к приложениям, которые обрабатывают высокочувствительные данные и могут служить средством защиты интеллектуальной собственности.

Рекомендованное использование

Приложения могут быть проверены на соответствие MASVS L1 или L2, основываясь на предварительной оценке рисков и общем понимании требуемого уровня безопасности. L1 применим ко всем мобильным приложениям, в то время как L2 обычно рекомендуется для приложений, которые обрабатывают более чувствительные данные и/или функциональность. MASVS-R (или его части) может быть применен для проверки устойчивости к конкретным угрозам, таким, как переупаковка (repackaging) или извлечение конфиденциальных данных, ***в дополнение*** к основным проверкам безопасности.

Таким образом, доступны следующие типы проверки:

- MASVS-L1
- MASVS-L1+R
- MASVS-L2
- MASVS-L2+R

Различные комбинации отражают различные уровни безопасности и устойчивости к атакам. Цель состоит в том, чтобы обеспечить гибкость: например, игра для смартфона может не гарантировать соответствие таким мерам защиты MASVS-L2, как двухфакторная аутентификация, из-за неудобства использования, но имеет потребность в предотвращении фальсификаций.

Какой тип проверки выбрать

Соответствие требованиям MASVS L2 повышает безопасность, но в то же время увеличивает стоимость разработки и потенциально ухудшает опыт конечного пользователя (классический компромисс между удобством и безопасностью). В общем случае, L2 следует использовать для приложений, рассматриваемых с точки зрения риска и издержек (т.е. когда потенциальная потеря, вызванная нарушением конфиденциальности или целостности, выше, чем затраты, связанные с реализацией дополнительных проверок безопасности). Оценка рисков должна быть первым шагом перед применением MASVS.

Примеры

MASVS-L1

- Все мобильные приложения. В MASVS-L1 перечислены рекомендации по безопасности, которые могут быть выполнены с разумным воздействием на стоимость разработки и пользовательский опыт. Применяйте требования MASVS-L1 для любого приложения, которое не подходит ни под один из более высоких уровней.

MASVS-L2

- Индустрия здравоохранения: мобильные приложения, которые хранят персональные данные, которые могут использоваться для кражи личности, мошеннических платежей или других схем мошенничества. Для сектора здравоохранения США пункты проверок включают в себя Закон об охране и ответственности за информацию, полученную в результате медицинского страховани (HIPAA).
- Финансовая индустрия: приложения, которые обеспечивают доступ к высокочувствительной информации, такой, как номера кредитных карт, персональным данным, или позволяют управлять финансовыми средствами. Эти приложения требуют дополнительных проверок безопасности для предотвращения мошенничества. Финансовым приложениям необходимо обеспечить соблюдение стандартов безопасности данных, таких, как Payment Card Industry Data Security Standard (PCI DSS), Gramm Leech Bliley Act и Sarbanes-Oxley Act (SOX).

MASVS L1+R

- Мобильные приложения, где защита IP является целью бизнеса. Меры защиты, перечисленные в MASVS-R, могут использоваться для увеличения усилий, необходимых для получения исходного кода, и для препятствия попыткам фальсификации или взлома приложения.
- Игровая индустрия: игры, для которых важное значение имеет предотвращение возможности модификации и использования читов, например, многопользовательские онлайн-игры. Читинг является важной проблемой в онлайн-играх, так как большое количество читеров вызывает недовольство у основной массы игроков и в конечном итоге может привести к краху данного игрового продукта. MASVS-R содержит меры защиты, направленные на усложнение задачи читерам.

MASVS L2+R

- Финансовая индустрия: приложения для онлайн банкинга, которые позволяют пользователю управлять финансовыми средствами, для которых инъекции кода и использование специализированного инструментария для взлома на устройствах с jailbreak-ом/root-ом представляют риск. В этом случае,

верификация MASVS-R может быть использована, чтобы препятствовать фальсификации, усложнить задачу авторам вредоносных программ.

- Все мобильные приложения хранят чувствительные данные на мобильном устройстве, и в то же время должны поддерживать широкий спектра устройств и версий операционной системы. В этом случае проверка устойчивости к фальсификациям может использоваться в качестве углублённой защиты, для максимального затруднения извлечения конфиденциальных данных злоумышленником.

Оценка и сертификация

Позиция OWASP в отношении сертификатов MASVS

OWASP является некоммерческой организацией и не сертифицирует вендоров, аудиторов, или программное обеспечение.

OWASP не выдаёт официальные оценки, знаки доверия, или сертификаты, поэтому к организации, утверждающей, что она имеет сертификацию OWASP, следует относиться с осторожностью.

В то же время, отсутствие официальной сертификации OWASP не запрещает организациям предлагать услуги аудита по данному стандарту.

Руководство по сертификации мобильных приложений

Рекомендуемым способом проверки соответствия мобильного приложения стандарту MASVS является аудит методом «открытой книги», означающий, что проверяющие получают полный доступ к ключевым ресурсам, таким как: архитекторы и разработчики приложения, проектная документация, исходный код и авторизованный доступ к бэкенду (как минимум по одной учетной записи для каждой роли).

Важно отметить, что MASVS охватывает только безопасность мобильных приложений (на стороне клиента) и сетевое взаимодействие между приложением и его удалёнными сервисами, а также несколько базовых и общих требований, связанных с аутентификацией пользователя и управлением сессиями. Данный документ не содержит специфичных требований к удаленным сервисам (например, веб-сервисам), связанным с приложением, ограничиваясь набором базовых требований к безопасности аутентификации и управления сессиями. Однако MASVS V1 подчёркивает, что удалённые сервисы должны быть учтены в общей модели угроз и проверены по соответствующим стандартам, таким, как OWASP ASVS.

Проверяющая организация должна включать в отчёт область охвата проверки, резюме о результатах проверки, включая пройденные и непройденные тесты, с ясными указаниями о том, как исправить недостатки. Сохранение подробных документов, скриншотов или видео, сценариев для воспроизведения и эксплуатирования багов, электронных записей, таких как логи перехватывающего запросы прокси-сервера и прочих заметок являются стандартной практикой индустрии. Недостаточно просто запустить инструмент и сообщить об ошибках, это не дает достаточных доказательств того, что проверяющим были проведены все проверки должным образом. В случае возникновения спора должны быть достаточные доказательства для демонстрации того, что соответствие каждому требованию было проверено.

Использование OWASP Mobile Security Testing Guide (MSTG)

OWASP MSTG - это руководство по тестированию безопасности мобильных приложений. В нём описываются технические процессы для проверки требований, перечисленных в MASVS. MSTG включает список тест-кейсов, выстроенных в соответствии с требованиями в MASVS. Хотя требования MASVS являются высокоуровневыми и универсальными, MSTG предоставляет подробные рекомендации и процедуры тестирования для каждой целевой мобильной ОС.

Роль автоматических инструментов тестирования безопасности

Для повышения эффективности рекомендуется использовать статические анализаторы исходного кода и инструменты для blackbox тестирования. Однако невозможно выполнить проверку MASVS, используя только автоматизированные инструменты: каждое мобильное приложение уникально, и понимание общей

архитектуры, бизнес-логики и технических проблем конкретных технологий и фреймворков является обязательным требованием для верификации безопасности приложения.

Другие применения

Как подробное руководство по архитектуре безопасности

Одно из наиболее распространенных применений MASVS - в качестве пособия для архитекторов безопасности. В двух основных методологиях по выстраиванию безопасной архитектуры SABSA и TOGAF отсутствует информация, необходимая для ревью безопасности архитектуры мобильных приложений. MASVS можно использовать для заполнения этих пробелов, позволяя архитекторам безопасности выбирать лучшие требования безопасности, адаптированные для мобильных приложений.

В качестве замены готовых чеклистов написания безопасного кода

Многие организации могут извлечь выгоду из соответствия MASVS, выбрав один из двух уровней проверки, или кастомизировав MASVS для собственных требований, предварительно оценив уровни риска для собственных приложений, беря во внимание область их применения. Мы приветствуем такие ответвления до тех пор, пока сохраняется преемственность, т.е. соответствие приложения требованию 4.1 оригинального стандарта должно означать то же самое в его кастомизированной копии.

В качестве основы для методологий тестирования безопасности

Хорошая методология тестирования безопасности мобильных приложений должна покрывать все требования, перечисленные в MASVS. В гайде OWASP по тестированию мобильных приложений (MSTG) описываются black-box и white-box тест-кейсы для каждого из требований данного стандарта.

Как руководство для автоматического модульного и интеграционного тестирований

MASVS, за исключением некоторых архитектурных требований, может быть использован как руководство для тестирования. Автоматические модульное, интеграционное и приёмочное тестирование на основе требований MASVS могут быть интегрированы в непрерывный жизненный цикл разработки. Это не только повысит уровень осведомлённости разработчиков в области безопасности, но также улучшит общее качество приложений и уменьшит количество находимых уязвимостей на этапе предрелиза.

Для курсов по обучению безопасной разработке

MASVS также может использоваться для определения характеристик безопасных мобильных приложений. Многие курсы «безопасного программирования» - это курсы этичного хакинга с лёгким намеком на подсказки по написанию кода, что, безусловно, не помогает разработчикам. Вместо этого курсы безопасной разработки могут использовать MASVS, уделяя особое внимание проактивным средствам контроля, задокументированным в MASVS, а не, например, списку наиболее часто встречающихся уязвимостей в мобильных приложениях (OWASP Mobile Top 10).

V1: Требования к архитектуре, дизайну и модели угроз

Цель верификации

В идеальном мире безопасность должна приниматься во внимание на всех этапах разработки. Однако на самом деле безопасность часто рассматривается только на поздней стадии SDLC. Помимо технических средств управления, MASVS требует наличия процессов, которые гарантируют, что безопасность была явно учтена при разработке архитектуры мобильного приложения и что функциональные и защитные роли всех компонентов известны. Поскольку большинство мобильных приложений выступают в качестве клиентов у веб-сервисов, необходимо обеспечить применение соответствующих стандартов безопасности: только тестирования изолированного мобильного приложения недостаточно.

В категории «V1» перечислены требования, касающиеся архитектуры и дизайна приложения. Таким образом, это единственная категория, которая не соответствует техническим тестам в OWASP MSTG. Чтобы охватить такие темы, как моделирование угроз, безопасный SDLC, управление ключами, пользователи MASVS должны проконсультироваться с соответствующими проектами OWASP и/или другими стандартами, такими как те, которые приведены ниже.

Требования безопасности

Ниже приведены требования к MASVS-L1 и MASVS-L2.

#	Описание	L1	L2
1.1	Все компоненты приложения идентифицированы и используются.	✓	✓
1.2	Проверки безопасности реализованы не только на клиенте, но и на бэкенде.	✓	✓
1.3	Архитектура мобильного приложения учитывает все удалённые сервисы. Безопасность заложена в архитектуре.	✓	✓
1.4	Определены данные, которые являются чувствительными в контексте мобильного приложения.	✓	✓
1.5	Все компоненты приложения определены с точки зрения бизнес логики и/или безопасности.		✓
1.6	Сформирована модель угроз для мобильного приложения и связанных с ним удаленных сервисов, которая идентифицирует потенциальные угрозы и необходимые контрмеры.		✓
1.7	Все проверки безопасности имеют централизованную реализацию.		✓
1.8	Существует явная политика управления криптографическими ключами (если они есть) и их жизненным циклом. В идеале политика соответствует стандарту управления ключами, например, NIST SP 800-57.		✓
1.9	Существует механизм принудительных обновлений мобильного приложения.		✓
1.10	Безопасность заложена во все этапы жизненного цикла разработки программного обеспечения.		✓

Ссылки

Для дополнительной информации смотрите также:

- OWASP Mobile Top 10: M10 - Extraneous Functionality: https://www.owasp.org/index.php/Mobile_Top_10_2016-M10-Extraneous_Functionality
- OWASP Security Architecture cheat sheet: https://www.owasp.org/index.php/Application_Security_Architecture_Cheat_Sheet
- OWASP Thread modelling: https://www.owasp.org/index.php/Application_Threat_Modeling
- OWASP Secure SDLC Cheat Sheet: https://www.owasp.org/index.php/Secure_SDLC_Cheat_Sheet
- Microsoft SDL: https://www.microsoft.com/en-us/sdl/
- NIST SP 800-57: http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57-Part1-revised2_Mar08-2007.pdf

V2: Требования к конфиденциальности и хранению данных

Цель проверки

Защита чувствительных данных, таких, как данные пользователя для авторизации (логин + пароль) и персональные данные, является ключевым аспектом в безопасности мобильных приложений. Во-первых, чувствительные данные могут быть непреднамеренно раскрыты другим приложениям, работающим на том же устройстве, если механизмы операционной системы, такие как межпроцессное взаимодействие (IPC), используются ненадлежащим образом. Данные также могут непреднамеренно попасть в облачное хранилище, резервную копию или кеш клавиатуры. Кроме того, мобильные устройства могут быть потеряны или украдены легче чем другие типы устройств, поэтому злоумышленник, получивший физический доступ к устройству, является наиболее вероятным сценарием. В этом случае могут быть реализованы дополнительные меры защиты для затруднения извлечения чувствительных данных с устройства.

Следует обратить внимание, что MASVS ориентирован на приложения, и потому не охватывает политики безопасности на уровне устройства (MDM - Mobile Device Managment). Мы поощряем использование таких политик в контексте предприятия для повышения безопасности данных.

Определение чувствительных данных

К чувствительным данным в контексте MASVS относятся как данные пользователя для авторизации, так и любые другие данные, которые считаются чувствительными в конкретном контексте, например:

- Персональные данные, которые могут быть использованы для кражи личности: номера социального страхования, кредитных карт, банковских счетов, информация о здоровье.
- Конфиденциальная информация, которая может привести к репутационному ущербу и/или финансовым потерям, если она скомпрометирована: информация о договорах, информация, охватываемая соглашениями о неразглашении, управленческая информация;
- Любые данные, которые должны быть защищены по закону или внутренним требованиям компании.

Требования безопасности

Подавляющее большинство проблем с раскрытием информации можно предотвратить, следуя простым правилам. Большинство требований, перечисленных в этой главе, являются обязательными для всех уровней проверки.

#	Описание	L1	L2
2.1	Хранилище учетных данных системы используется надлежащим образом для хранения чувствительных данных, таких как персональные данные, данные пользователя для авторизации и криптографические ключи.	✓	✓
2.2	Чувствительные данные хранятся только во внутреннем хранилище приложения, либо в системном хранилище авторизационных данных.	✓	✓
2.3	Чувствительные данные не попадают в логи приложения.	✓	✓
2.4	Никакие чувствительные данные не передаются третьей стороне, если это не является необходимой частью архитектуры.	✓	✓
2.5	Кэш клавиатуры выключен для полей ввода чувствительных данных.	✓	✓

2.6	Буфер обмена выключен для текстовых полей, которые могут содержать чувствительные данные.	✓	✓
2.7	Чувствительные данные недоступны для механизмов межпроцессного взаимодействия (IPC).	✓	✓
2.8	Никакие чувствительные данные, такие как пароли или пин-коды, не видны через пользовательский интерфейс.	✓	✓
2.9	Никакие чувствительные данные не попадают в бэкапы, создаваемые операционной системой.		✓
2.10	Приложение скрывает чувствительные данные с экрана, когда находится в фоновом режиме.		✓
2.11	Приложение не хранит чувствительные данные в памяти дольше, чем необходимо, и полностью удаляет их из памяти после работы с ними.		✓
2.12	Приложение требует от пользователя минимальную настройку доступа к устройству, такую, как установку пин-кода на устройство.		✓
2.13	Приложение информирует пользователя о персональных данных, которые оно обрабатывает, а также о лучших практиках безопасности, которым должен следовать пользователь при использовании приложения.		✓

Ссылки

OWASP MSTG содержит подробные инструкции по верификации требований, перечисленных в этом разделе.

- Для Android - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05d-Testing-Data-Storage.md
- Для iOS - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06d-Testing-Data-Storage.md

Для получения дополнительной информации смотрите также:

- OWASP Mobile Top 10: M2 - Insecure Data Storage: https://www.owasp.org/index.php/Mobile_Top_10_2016-M2-Insecure_Data_Storage
- CWE: https://cwe.mitre.org/data/definitions/922.html

V3: Требования к криптографии

Цель верификации

Криптография является неотъемлимым компонентом защиты данных, хранящихся на мобильном устройстве. Но кроме того, это область, в которой все может пойти не так, особенно когда стандартные правила не соблюдаются. Цель верификационных требований в этой главе состоит в том, чтобы убедиться, что проверяемое приложение использует криптографию в соответствии с лучшими практиками индустрии, такими, как:

- Использование проверенных криптографических библиотек;
- Правильный выбор и настройка криптографических алгоритмов;
- Подходящий генератор случайных чисел там, где это необходимо.

Требования безопасности

#	Описание	L1	L2
3.1	Приложение не использует симметричное шифрование с захардкоженными ключами в качестве единственного метода шифрования.	✓	✓
3.2	Приложение использует проверенные реализации криптографических алгоритмов.	✓	✓
3.3	Приложение использует подходящие криптографические алгоритмы для каждого конкретного случая, с параметрами, которые соответствуют лучшим практикам индустрии.	✓	✓
3.4	Приложение не использует устаревшие и слабые криптографические протоколы и алгоритмы.	✓	✓
3.5	Приложение не использует один и тот же ключ несколько раз.	✓	✓
3.6	Все случайные значения генерируются с использованием безопасного генератора случайных чисел.	✓	✓

Ссылки

OWASP MSTG содержит подробные инструкции по верификации требований, перечисленных в этом разделе.

- Android - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05e-Testing-Cryptography.md
- iOS - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06e-Testing-Cryptography.md

Для получения дополнительной информации смотрите также:

- OWASP Mobile Top 10: M5 - Недостаточное шифрование
- CWE: M5 - Недостаточное шифрование

V4: Требования к аутентификации и управлению сессиями

Цель верификации

В большинстве случаев авторизация в удаленных сервисах являются неотъемлемой частью общей архитектуры мобильного приложения. Несмотря на то, что большая часть логики происходит на бекэнде, MASVS определяет некоторые основные требования, касающиеся управления учетными записями пользователей и сессиями.

Требования безопасности

#	Описание	L1	L2
4.1	Если приложение предоставляет пользователям доступ к удалённым сервисам, на бэкенде должна быть реализована аутентификация, например, по логину и паролю.	✓	✓
4.2	Если используются сессии, бекэнд случайно генерирует идентификаторы сессии для аутентификации клиентских запросов без отправки данных учётной записи.	✓	✓
4.3	Если используется аутентификация на основе токена, сервер предоставляет токен, подписанный с использованием безопасного криптоалгоритма.	✓	✓
4.4	Бэкенд удаляет существующую сессию, когда пользователь выходит из системы.	✓	✓
4.5	На сервере реализована парольная политика.	✓	✓
4.6	На сервере реализован механизм защиты от перебора авторизационных данных.	✓	✓
4.7	Биометрическая аутентификация не является event-bound (т.е. использует только API, которое возвращает «true» или «false»). Вместо этого она основана на разблокировке keychain/keystore.		✓
4.8	Сессии становятся невалидными на бэкенде после определенного периода бездействия, срок действия токена истекает.	✓	✓
4.9	Реализована и поддерживается двухфакторная аутентификация.		✓
4.10	Для выполнения чувствительных транзакций требуется дополнительная или повторная аутентификацию.		✓
4.11	Приложение информирует пользователя обо всех входах в систему с использованием его учётной записи. Пользователи могут просмотреть список устройств, используемых для входа с этого аккаунта, и блокировать определённые устройства.		✓

Ссылки

OWASP MSTG содержит подробные инструкции по верификации требований, перечисленных в этом разделе.

- Для Android - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05f-Testing-Authentication.md
- Для iOS - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06f-Testing-Authentication-and-Session-Management.md

Для получения дополнительной информации смотрите также:

- OWASP Mobile Top 10: M4 - Insecure Authentication, M6 - Insecure Authorization
- CWE: https://cwe.mitre.org/data/definitions/287.html

V5: Требования к сетевому взаимодействию

Цель верификации

Целью требований, перечисленных в этом разделе, является обеспечение конфиденциальности и целостности информации, передаваемой между мобильным приложением и сервером. Как минимум в мобильном приложении должен быть настроен безопасный шифрованный канал передачи данных с использованием протокола TLS с соответствующими настройками. L2 содержит меры усиленной защиты, такие, как SSL pinning.

Требования безопасности

#	Описание	L1	L2
5.1	Данные, передаваемые по сети, шифруются с использованием TLS. Безопасный канал используется для всех сервисов приложения.	✓	✓
5.2	Настройки TLS соответствуют современным лучшим практикам, или максимально приближены к ним, если операционная система не поддерживает рекомендуемые стандарты.	✓	✓
5.3	Приложение верифицирует X.509 сертификаты сервера во время установления защищённого канала. Принимаются только сертификаты, подписанные доверенным удостоверяющим центром (CA).	✓	✓
5.4	В приложении реализован SSL pinning и соединение с серверами, которые предлагают другой сертификат или ключ, даже если они подписаны доверенным центром сертификации (CA) не устанавливается.		✓
5.5	Приложение не полагается на единственный небезопасный канал связи (e-mail или SMS) для таких критических операций, как регистрация и восстановление аккаунта.		✓
5.6	Приложение использует только актуальные версии библиотек для подключения к сети и обеспечения безопасного соединения.		✓

Ссылки

OWASP MSTG содержит подробные инструкции по верификации требований, перечисленных в этом разделе.

- Android - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05g-Testing-Network-Communication.md
- iOS - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06g-Testing-Network-Communication.md

Для получения дополнительной информации смотрите также:

- OWASP Mobile Top 10: M3 - Insecure Communication: https://www.owasp.org/index.php/Mobile_Top_10_2016-M3-Insecure_Communication
- CWE: https://cwe.mitre.org/data/definitions/319.html
- CWE: https://cwe.mitre.org/data/definitions/295.html

V6: Требования к взаимодействию с операционной системой

Цель верификации

Следование требованиям этого раздела обеспечивают безопасное использование API операционной системы. Дополнительно содержатся требования к межпроцессному взаимодействию (IPC).

Требования безопасности

#	Описание	L1	L2
6.1	Приложение запрашивает минимально необходимый набор разрешений.	✓	✓
6.2	Все данные, поступающие из внешних источников и от пользователя, валидируются и санитизируются. Сюда входят данные, полученные через пользовательский интерфейс, механизмы IPC (такие как intent-ы, кастомные URL-схемы) и из сети.	✓	✓
6.3	Приложение не экспортирует чувствительные данные через кастомные URL-схемы, если эти механизмы не защищены должным образом.	✓	✓
6.4	Приложение не экспортирует чувствительные данные через IPC механизмы без должной защиты.	✓	✓
6.5	JavaScript отключен в компонентах WebView, если в нём нет необходимости.	✓	✓
6.6	WebViews сконфигурирован с поддержкой минимального набора протоколов (в идеале только https). Поддержка потенциально опасных URL-схем (таких как: file, tel и app-id) отключена.	✓	✓
6.7	Если нативные методы приложения используются WebView, верифицировать, что исполняются только Javascript объекты данного приложения.	✓	✓
6.8	Десериализация объектов, если она есть, реализована с использованием безопасного API.	✓	✓

Ссылки

OWASP MSTG содержит подробные инструкции по верификации требований, перечисленных в этом разделе.

- Android - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05h-Testing-Platform-Interaction.md
- iOS - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06h-Testing-Platform-Interaction.md

Для получения дополнительной информации смотрите также:

- OWASP Mobile Top 10: M1 - Improper Platform Usage
- CWE: https://cwe.mitre.org/data/definitions/20.html
- CWE: https://cwe.mitre.org/data/definitions/749.html

V7: Требования к качеству кода и настройкам сборки

Цель контроля

Следование требованиям данного раздела обеспечивает использование базовых практик безопасного написания кода при разработке приложения, а также стандартных средств безопасности, встроенных в компилятор.

Требования безопасности

#	Описание	L1	L2
7.1	Приложение подписано валидным сертификатом.	✓	✓
7.2	Приложение было собрано в release режиме с настройками, подходящими для релизной сборки (например, без атрибута debuggable).	✓	✓
7.3	Отладочные символы удалены из нативных бинарных файлов.	✓	✓
7.4	Отладочный код был удален, и приложение не логирует подробные ошибки и отладочные сообщения.	✓	✓
7.5	Все сторонние компоненты, используемые мобильным приложением (библиотеки и фреймворки), идентифицированы и проверены на наличие известных уязвимостей.	✓	✓
7.6	Приложение обрабатывает возможные исключения.	✓	✓
7.7	В логике обработки связанных с безопасностью ошибок по умолчанию запрещается доступ.	✓	✓
7.8	В неконтролируемом коде память выделяется, освобождается и используется безопасно.	✓	✓
7.9	Активированы все стандартные функции безопасности, предусмотренные инструментами разработчика (такие как минификация байт-кода, защита стека, поддержка PIE и ARC).	✓	✓

Ссылки

OWASP MSTG содержит подробные инструкции по проверке требований, перечисленных в этом разделе.

- Android - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05i-Testing-Code-Quality-and-Build-Settings.md
- iOS - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05i-Testing-Code-Quality-and-Build-Settings.md

Для получения дополнительной информации смотрите также:

- OWASP Mobile Top 10: M7 - Client Code Quality
- CWE: https://cwe.mitre.org/data/definitions/119.html
- CWE: https://cwe.mitre.org/data/definitions/89.html
- CWE: https://cwe.mitre.org/data/definitions/388.html
- CWE: https://cwe.mitre.org/data/definitions/489.html

V8: Требования к устойчивости к атакам на стороне клиента

Цель проверки

В этом разделе рассматриваются меры усиленной защиты, рекомендуемые для приложений, которые обрабатывают или предоставляют доступ к чувствительным данным или функциональности. Отсутствие каких-либо из этих элементов защиты не означает наличие уязвимости - напротив, они призваны повысить устойчивость приложения к реверс инжинирингу и конкретным атакам на стороне клиента.

Требования в этом разделе должны применяться по мере необходимости, основываясь на оценке рисков, вызванных несанкционированным вмешательством в приложение и/или восстановлением исходного кода. Мы предлагаем обратиться к документу OWASP «Technical Risks of Reverse Engineering and Unauthorized Code Modification Reverse Engineering and Code Modification Prevention» (см. ссылки ниже) для составления списка бизнес рисков и связанных с ними технических угроз.

Чтобы любое из требований в приведенном ниже списке было эффективным, приложение должно выполнить, по меньшей мере, все MASVS-L1, а также все требования из V8, которые ему предшествуют. Например, если приложение соблюдает требование обфускации из раздела «Противодействие восстановлению логики работы приложения», оно должно также удовлетворять всем требованиям из разделов «Противодействие динамическому анализу и фальсификациям» и «Привязка к устройству».

Обратите внимание, что программные меры защиты из данного раздела не должны использоваться в качестве замены основным требованиям безопасности. Вместо этого они должны быть реализованы как специализированное дополнение, направленное на локализацию конкретных угроз, в мобильном приложении, соответствующем требования MASVS

Следует рассмотреть следующие соображения:

1. Должна быть определена модель угроз, в которой прописаны конкретные атаки на стороне клиента, от которых необходимо защититься. Кроме того, должна быть указана степень защиты, которую следует обеспечить. Например, цель внедрения защитных мер может заключаться в том, чтобы заставить авторов вредоносного ПО, нацеленного на данное приложение, приложить значительные усилия для реверс инжиниринга.
2. Модель угроз должна отвечать здравому смыслу. Например, сокрытие криптографического ключа в whitebox реализации не имеет смысла, если злоумышленник может применить технику «code lifting».
3. Эффективность защиты всегда должна проверяться экспертом, имеющим опыт тестирования методов обфускации и защиты от фальсификации (см. также главы «Реверс инжиниринг» и «Оценка защиты ПО» в OWASP MSTG.

Противодействие динамическому анализу и фальсификациям

#	Описание	R
8.1	Приложение обнаруживает и реагирует на наличие root или jailbreak, либо уведомляя пользователя, либо прекращая работу.	✓
8.2	Приложение не позволяет использовать отладчики и/или обнаруживает и реагирует на использование отладчика. Все доступные протоколы отладки должны быть учтены.	✓
8.3	Приложение обнаруживает и реагирует на внесения изменений в исполняемые файлы и критичные данные в своей песочнице.	✓

8.4	Приложение обнаруживает и реагирует на наличие на устройстве широко используемых инструментов и фреймворков для реверс инжиниринга.	✓
8.5	Приложение обнаруживает и реагирует на запуск на эмуляторе.	✓
8.6	Приложение обнаруживает и реагирует на изменение своего кода и данных в оперативной памяти.	✓
8.7	Приложение реализует несколько механизмов для каждой категории защиты (с 8.1 по 8.6). Обратите внимание, что на устойчивость к атакам влияет количество, разнообразие и оригинальность используемых механизмов.	✓
8.8	Механизмы обнаружения инициируют ответные меры разных типов, включая отложенные и скрытые.	✓
8.9	Обфускация применена в том числе и к тем программным механизмам, которые препятствуют деобфускации методами динамического анализа.	✓

Привязка к устройству

#	Описание	R
8.10	Приложение реализует функциональность привязки экземпляра приложения к устройству, формируя его отпечаток из нескольких свойств, уникальных для устройства.	✓

Противодействие восстановлению логики работы приложения

#	Описание	R
8.11	Все исполняемые файлы и библиотеки, принадлежащие приложению, зашифрованы на файловом уровне, либо важные участки кода и данных зашифрованы внутри исполняемых файлов. Простой статический анализ не позволяет обнаружить важный код или данные.	✓
8.12	Если задачей обфускации является защита конфиденциальных данных, то используется схема обфускации, которая подходит не только для этой задачи, но и защищает от ручной тестирования и автоматизированных деобфускаторов и учитывает последние исследования по данной теме. Эффективность схемы обфускации должна быть проверена с помощью ручного тестирования. Обратите внимание, что использование аппаратных средств защиты (если они поддерживаются устройством) предпочтительнее обфускации.	✓

Ссылки

OWASP MSTG содержит подробные инструкции по верификации соответствия требованиям, перечисленным в этом разделе.

- Android - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05j-Testing-Resiliency-Against-Reverse-Engineering.md
- iOS - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06j-Testing-Resiliency-Against-Reverse-Engineering.md

Для получения дополнительной информации смотрите также:

- OWASP Mobile Top 10: M8 - Code Tampering, M9 - Reverse Engineering
- WASP Reverse Engineering Threats - https://www.owasp.org/index.php/Technical_Risks_of_Reverse_Engineering_and_Unauthorized_Code_Modification
- OWASP Reverse Engineering and Code Modification Prevention -

https://www.owasp.org/index.php/OWASP_Reverse_Engineering_and_Code_Modification_Prevention_Project

Приложение A: Список терминов

- 2FA – Двухфакторная аутентификация (2FA) добавляет второй уровень аутентификации для входа в учетную запись.
- Address Space Layout Randomization (ASLR) – Метод, затрудняющий использование ошибок повреждения памяти.
- Application Security – Безопасность на уровне приложений, сфокусированная на анализе компонентов, которые составляют прикладной уровень модели OSI (Open Systems Interconnection Reference), без рассмотрения безопасности, например, операционной системы или подключенных сетей.
- Application Security Verification – Оценка соответствия требованиям OWASP MASVS.
- Application Security Verification Report – Отчёт, включающий общие результаты проверки и анализа конкретного приложения, проведённые специалистом по безопасности.
- Authentication – Верификация соответствия пользователя приложения заявленной им личности.
- Automated Verification – Использование автоматических инструментов (для динамического и статического анализа), использующих сигнатуры уязвимостей для поиска проблемных участков приложения.
- Black box testing – Метод тестирования ПО, который анализирует функциональность приложения без знания о его внутреннем устройстве и принципах работы.
- Component – автономный блок кода, с соответствующими дисковыми и сетевыми интерфейсами, который взаимодействует с другими компонентами.
- Cross-Site Scripting (XSS) – Уязвимость, обычно обнаруживаемая в веб-приложениях, позволяющая внедрить в контент веб-страницы вредоносный код, исполняемый на стороне клиента.
- Cryptographic module – Аппаратное, программное, или программно-аппаратное обеспечение, реализующее криптографические алгоритмы и/или генерирует криптографические ключи.
- CWE - Общедоступный список часто встречающихся недостатков безопасности программного обеспечения.
- DAST – Динамический анализ безопасности (Dynamic application security testing) - поиск уязвимостей приложения во время его выполнения.
- Design Verification – Техническая оценка безопасности архитектуры приложения.
- Dynamic Verification – Использование автоматических инструментов для поиска уязвимостей во время выполнения приложения.
- Globally Unique Identifier (GUID) – уникальный номер, используемый в качестве идентификатора в программном обеспечении.
- Hyper Text Transfer Protocol (HTTP) – Протокол прикладного уровня для распределенных, совместных, гипермедийных информационных систем. Это основа передачи данных для Всемирной паутины.
- Hardcoded keys – захардкоженные ключи (хранятся непосредственно в коде приложения).
- IPC – межпроцессное взаимодействие (Inter Process Communications), в IPC процессы взаимодействуют друг с другом и ядром, чтобы координировать свою деятельность.
- Input Validation – Канонизация и проверка недоверенных данных, вводимых пользователем.
- JAVA Bytecode – Java байт-код - набор команд виртуальной машины Java (JVM). Каждый байт-код состоит из одного или, в некоторых случаях, двух байтов, которые представляют команду (код операции), а также ноль или более байтов для передачи параметров.
- Malicious Code – вредоносный код, добавленный в приложение во время его разработки без ведома владельца приложения, который обходит заложенную политику безопасности. Не то же самое, что вредоносное ПО, такое как вирус или червь!
- Malware – Исполняемый код, который вводится в приложение во время выполнения без ведома пользователя или администратора приложения.
- Open Web Application Security Project (OWASP) – Проект Open Web Application Security (OWASP) является всемирным бесплатным и открытым сообществом, направленным на повышение безопасности прикладного программного обеспечения. Наша миссия заключается в том, чтобы сделать безопасность

приложений «видимой», чтобы люди и организации могли принимать обоснованные решения о рисках безопасности приложений. http://www.owasp.org/

- Personally Identifiable Information (PII) - Информация, которая может использоваться сама по себе или совместно с другой информацией для идентификации, получения контактов или поиска местонахождения человека.
- PIE – Независимый от положения исполняемый файл (PIE) представляет собой тело машинного кода, которое, будучи помещённым где-то в первичной памяти, выполняется должным образом независимо от его абсолютного адреса.
- PKI – PKI - это соглашение, которое связывает открытые ключи с соответствующими идентификаторами объектов. Связь устанавливается посредством процесса регистрации и выдачи сертификатов в удостоверяющем центре (CA).
- SAST – Статический анализ безопасности (SAST) представляет собой набор техник, предназначенных для анализа исходного кода приложения, байт-кода и бинарных файлов для обнаружения ошибок при написании кода и проектировании, которые приводят к уязвимостям. Решения SAST анализируют приложение «изнутри» в неработающем состоянии.
- SDLC – Жизненный цикл разработки программного обеспечения.
- Security Architecture – Абстракция проектирования приложения, которая идентифицирует и описывает, где и как используются требования безопасности, а также идентифицирует и описывает местоположение и чувствительность данных пользователя и приложения.
- Security Configuration – Конфигурация исполнения приложения, влияющая на то, как используются требования безопасности.
- Security Control – Функция или компонент, который выполняет проверку безопасности (например, проверку контроля доступа) или при вызове, влияет на безопасность (например, генерирует запись аудита).
- SQL Injection (SQLi) – Техника внедрения SQL кода в запрос к БД, используемая для атаки на мобильные и веб-приложения.
- SSO Authentication – Single Sign On (SSO) возникает, когда пользователь входит в аккаунт на одном клиенте, и происходит автоматический вход на других клиентах, независимо от платформы, технологии или домена, которые использует пользователь. Например, когда вы авторизуетесь в Google, вы автоматически авторизуетесь на YouTube, Google Docs и Gmail.
- Threat Modeling – Метод, состоящий в разработке все более совершенных архитектур безопасности для идентификации угроз, зон безопасности, средств контроля безопасности и важных технических и бизнес-активов.
- Transport Layer Security – Криптографические протоколы, обеспечивающие безопасность соединения по Интернету.
- URI/URL/URL fragments – Единый идентификатор ресурса - это строка символов, используемых для идентификации имени или веб-ресурса. URL часто используется в качестве ссылки на ресурс.
- User acceptance testing (UAT) – Тестовая среда, которая ведет себя как производственная среда, где все тестирование программного обеспечения выполняется до перехода в лайв.
- Verifier – Лицо или команда, которая проверяет приложение на соответствие требованиям ASVS OWASP.
- Whitelist – Список разрешённых данных или операций, например список символов, которые могут быть поданы на вход.
- X.509 Certificate – Сертификат X.509 это цифровой сертификат, использующий международную инфраструктуру открытых ключей (PKI) для проверки того, что открытый ключ принадлежит пользователю, компьютеру или сервису, указанному в сертификате.

Приложение B: Ссылки

Следующие проекты OWASP будут полезны для пользователей/последователей этого стандарта:

- OWASP Mobile Security Project - https://www.owasp.org/index.php/OWASP_Mobile_Security_Project
- OWASP Mobile Security Testing Guide - https://www.owasp.org/index.php/OWASP_Mobile_Security_Testing_Guide
- OWASP Mobile Top 10 Risks - https://www.owasp.org/index.php/Projects/OWASP*Mobile_Security_Project*-_Top_Ten_Mobile_Risks
- OWASP Reverse Engineering and Code Modification Prevention - https://www.owasp.org/index.php/OWASP_Reverse_Engineering_and_Code_Modification_Prevention_Project

Следующие веб-сайты также будут полезны для пользователей/последователей этого стандарта:

- MITRE Common Weakness Enumeration - http://cwe.mitre.org/
- PCI Security Standards Council - https://www.pcisecuritystandards.org
- PCI Data Security Standard (DSS) v3.0 Requirements and Security Assessment Procedures https://www.pcisecuritystandards.org/documents/PCI_DSS_v3.pdf