OWASP Open Web Application Security Project | Standard

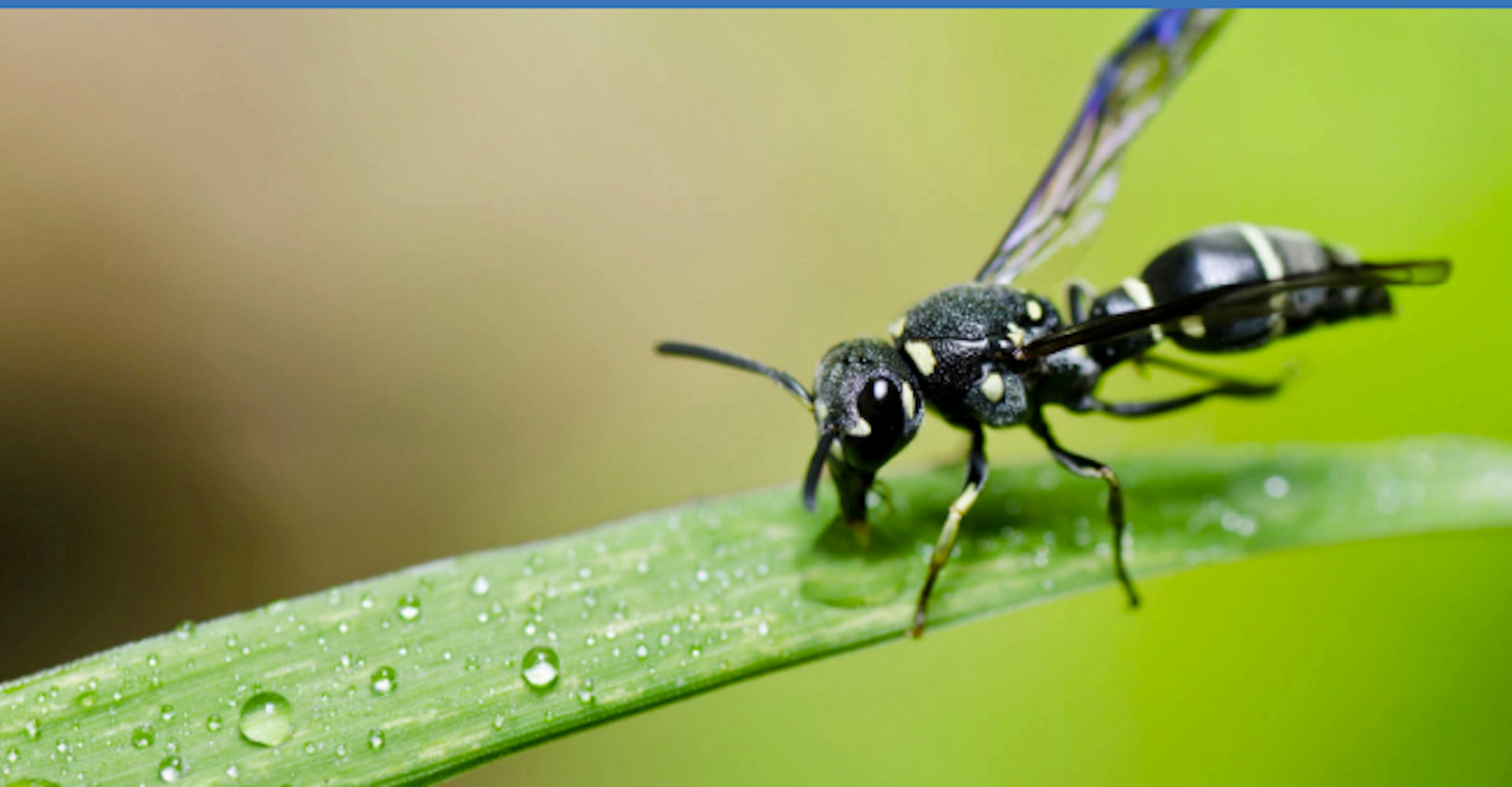
MASVS

Mobile Application Security Verification Standard

(Korean Translation)

Project leaders: Sven Schleier, Jeroen Willemsen and Carlos Holguera

Version 1.2



OWASP Mobile Application Security Verification Standard

Version 1.2 March 23, 2020

목차

서문

Bernhard Mueller, OWASP 모바일 프로젝트

기술 혁명은 빠르게 일어나고 있습니다. 10 년전만 하더라도 스마트폰은 작은 키보드가 달린 투박한 디바이스로 기술에 정통한 비즈니스 사용자를 위한 값비싼 장난감에 불과하였습니다. 오늘날, 스마트폰은 우리 삶의 필수적인 부분입니다. 우리는 정보를 검색하고, 길을 찾고, 사람들과 소통하기 위해 스마트폰에 점점 의존하게 되었으며, 스마트폰은 비즈니스와 일상 생활 어디서나 필요한 존재가 되었습니다.

모든 새로운 기술은 새로운 보안위험을 초래하고 그 변화에 대응하는 것은 보안업계가 직면한 주요 과제 중 하나입니다. 방어적인 측면은 항상 몇 발자국 뒤처집니다. 예를 들어, 많은 사람들의 기본 대응은 오래된 방법들을 적용하는 것이었습니다. 스마트폰은 작은 컴퓨터와 유사하며 모바일 앱은 고전적인 소프트웨어와 마찬가지이기 때문에 보안 요구사항이 동일하다고 생각하지만 그렇지 않습니다. 스마트폰 운영체제는 데스크톱 운영체제와 다르며, 모바일 앱은 웹 앱과 다릅니다. 예를 들어, 시그니처 기반의 바이러스 검색과 같은 전통적인 방법은 최신 모바일 OS 환경에서는 의미가 없습니다: 모바일 앱의 배포 모델과 호환되지 않을뿐만 아니라 샌드박스의 제한으로 인해 기술적으로도 불가능합니다. 또한, 버퍼 오버플로우, XSS 와 같은 몇 가지 취약점 종류는 데스크톱 애플리케이션과 웹 애플리케이션과 달리 (예외는 있지만) 일반 모바일 앱의 맥락에서 별로 의미가 없습니다.

시간이 지남에 따라 우리 업계는 모바일 위협 환경을 더 잘 파악하게 되었습니다. 결국, 모바일 보안은 데이터 보호에 관한 모든 것입니다. 앱은 개인 정보, 사진, 녹음, 노트, 계정 데이터, 비즈니스 정보, 위치 등을 저장합니다. 앱은 매일 사용하는 서비스에 연결하는 클라이언트 역할을 하며, 다른 사람들과 주고 받는 모든 메시지를 처리하는 통신 허브의 역할을 합니다. 다른 사람의 스마트폰을 손상시키면 그 사람의 삶에 아무런 제약 없이 접근할 수 있습니다. 모바일 장치가 너무 쉽게 분실 또는 도난당하고 모바일 맬웨어가 증가하고 있다고 생각하면 데이터 보호의 필요성이 더욱 분명해집니다.

따라서 모바일 앱의 보안 표준은 모바일 앱이 민감한 정보를 처리, 저장 및 보호하는 방법에 초점을 맞추어야 합니다. iOS 나 Android 와 같은 최신 모바일 운영체제는 안전한 데이터 저장과 통신을 위한 뛰어난 API 를 제공하고 있지만, 효과를 발휘하려면 제대로 구현되고 사용되어야 합니다. 데이터 저장, 앱간 통신, 암호화 API 의 올바른 사용 및 안전한 네트워크 통신은 신중하게 고려해야하는 측면 중 일부분에 지나지 않습니다.

산업계의 합의가 필요한 중요한 문제는 '데이터의 기밀성과 무결성을 보호하기 위해 정확히 어디까지 어떻게 할 것인가' 입니다. 예를 들어, 대부분은 모바일 앱이 TLS 교환에서 서버 인증서를 검증해야 한다는 것에 동의할 것입니다. 하지만 SSL 인증서를 피닝하는 것은 어떨까요? 인증서를 검증하지 않으면 취약할까요? 앱이 민감한 데이터를 처리하는 경우 이것이 필수 요건이어야 하나요, 아니면 불필요한 것일까요? OS 가 앱을 샌드박스화해도 SQLite 데이터베이스에 저장된 데이터를 암호화해야합니까? 특정 앱에 적합한 것은 다른 앱에는 쓸모가 없을 수 있습니다. MASVS 는 다양한 위협 시나리오에 맞는 검증 수준을 사용하여 이러한 요구사항을 표준화하려는 시도입니다.

또한 루트 맬웨어 및 원격 관리 도구의 출현으로 인해 모바일 운영체제 자체에 악용 가능한 결함이 있다는 사실을 인식하게 되었고, 따라서 민감한 데이터에 대한 추가 보호를 제공하고 클라이언트 변조를 방지하기 위해 컨테이너 전략이 점점 더 많이 사용되고 있습니다. 상황이 복잡하게 얽혀 있습니다. 하드웨어 지원 보안 기능과 Android for Work 및 Samsung Knox 등의 OS 레벨의 컨테이너 솔루션이 존재하지만, 다양한 디바이스에서 일관성 있게 사용할 수 있는 것은 아닙니다. 미봉책으로 소프트웨어 기반의 보호 대책을 구현할 수 있지만, 안타깝게도 이러한 종류의 보호를 검증하기 위한 표준이나 테스트 프로세스는 없습니다.

그 결과 모바일 앱 보안 테스트 보고서는 체계적이지 못합니다. 예를 들어, 일부 테스터는 Android 앱에서 난독화 또는 루팅 미 탐지를 "보안 결함" 으로 보고합니다. 다른 한편에서는 문자열 암호화, 디버거 감지 또는 통제 흐름 난독화와 같은 조치는 필수로 간주하지 않습니다. 하지만, 이러한 것을 이분법적으로 보는 방법은 복원력이 이항 명제가 아니기 때문에 말이 않됩니다: 방어하려는 특정 클라이언트 측 위협에 따라 다릅니다. 소프트웨어 보호는 무용지물은 아니지만 궁극적으로 우회 할 수 있으므로 보안 통제를 대체할 수는 없습니다.

MASVS 의 전반적인 목표는 모바일 애플리케이션 보안 기준 (MASVS- L1) 을 제시하는 것이며, 심층 방어 조치 (MASVS-L2) 와 클라이언트 측 위협에 대한 보호 (MASVS-R) 도 포함할 수 있도록 허용하는 것입니다. MASVS 는 다음을 달성하기 위한 것입니다:

- 안전한 모바일 애플리케이션을 개발하려는 소프트웨어 아키텍트 및 개발자를 위한 요구사항 제공;
- 모바일 앱 보안 검토에서 테스트할 수 있는 업계 표준 제공;
- 모바일 보안에서 소프트웨어 보호 메커니즘의 역할을 명확히하고 그 효과를 검증하기 위한 요구사항 제공;
- 다양한 사용 사례에 권장되는 보안 수준에 대한 구체적인 권장 사항 제공.

우리는 100 % 업계 합의가 달성 불가능하다는 것을 인식하고 있습니다. 그럼에도 불구하고 MASVS 가 모바일 앱 개발 및 테스트의 모든 단계에서 지침을 제공하는 데 유용하기를 바랍니다. 오픈소스 표준으로서, MASVS 는 시간이 지남에 따라 발전할 것이며, 우리는 어떠한 기여와 제안도 환영합니다.

본 표준에 대하여

모바일 애플리케이션 보안 검증 표준 (MASVS) 1.2 에 오신것을 환영합니다. MASVS 는 iOS 및 Android 에서 안전한 모바일 앱을 설계, 개발, 테스트하는데 필요한 보안 요구사항의 프레임워크를 확립하기 위한 커뮤니티 활동입니다.

MASVS 는 커뮤니티 활동과 업계 피드백의 성과입니다. 우리는 이 표준이 시간이 지남에 따라 발전하는 것을 기대하고 있으며, 커뮤니티의 피드백을 환영합니다.

우리와 연락하는 가장 좋은 방법은 OWASP Mobile Project Slack 채널을 이용하는 것입니다: https://owasp.slack.com/messages/project-mobile_omtg/details/

계정은 다음의 URL 에서 만들 수 있습니다: https://owasp-slack.herokuapp.com/

저작권 및 라이센스

감사의 말

프로젝트 리더	책임 저자	공동저자 및 검토자
Sven Schleier, Jeroen Willemsen and Carlos Holguera	Bernhard Mueller	Alexander Antukh, Mesheryakov Aleksey, Bachevsky Artem, Jeroen Beckers, Vladislav Chelnokov, Ben Cheney, Peter Chi, Lex Chien, Stephen Corbiaux, Manuel Delgado, Ratchenko Denis, Ryan Dewhurst, Tereshin Dmitry, Christian Dong, Oprya Egor, Ben Gardiner, Rocco Gränitz, Henry Hu, Sjoerd Langkemper, Vinícius Henrique Marangoni, Martin Marsicano, Roberto Martelloni, Gall Maxim, Eugen Martynov, Riotaro Okada, Abhinav Sejpal, Stefaan Seys, Yogesh Sharma, Prabhant Singh, Sven Schleier, Nikhil Soni, Anant Shrivastava, Francesco Stillavato, Romuald Szkudlarek, Abderrahmane Aftahi, Abdessamad Temmar, Koki Takeyama, Chelnokov Vladislav, Leo Wang

언어	번역자 및 검토자
중국어 전통	Peter Chi, and Lex Chien, Henry Hu, Leo Wang
중국어 간체	Bob Peng, Harold Zang, Jack S
프랑스어	Romuald Szkudlarek, Abderrahmane Aftahi, Christian Dong (Review)
독일어	Rocco Gränitz, Sven Schleier (Review)
스페인어	Martin Marsicano, Carlos Holguera
일본어	Koki Takeyama, Riotaro Okada (Review)
러시어아	Gall Maxim, Eugen Martynov, Chelnokov Vladislav (Review), Oprya Egor (Review), Tereshin Dmitry (Review)
한국어	Youngjae Jeon, Jeongwon Cho, Jiyou Han, Jiyeon Sung

본 문서는 Jim Manico 가 작성한 OWASP 애플리케이션 보안 검증 표준의 포크로 시작되었습니다.

스폰서

MASVS 와 MSTG 는 모두 커뮤니티에서 자발적으로 생성되고 유지되지만, 때때로 약간의 외부 도움이 필요합니다. 따라서 기술 편집자를 고용할 수 있도록 자금을 기부해 주신 스폰서들에게 감사드립니다. 그들의 후원은 어떤식으로든 MASVS 또는 MSTG 의 내용에 영향을 미치지 않는다는 점을 알려드립니다. 스폰서 패키지는 OWASP Project Wiki에 설명되어 있습니다..

명예 후원자

자선 후원자

다음으로 우리는 OWASP Bay Area Chapter 의 후원에 감사의 말씀을 전합니다. 마지막으로 Leanpub 에서 이 책을 구입하여 후원해 주신 모든 분들께 감사드립니다.

모바일 애플리케이션 보안 검증 표준

MASVS 를 사용하여 모바일 앱의 보안에 대한 신뢰 수준을 설정하는 데 사용할 수 있습니다. 요구사항은 다음과 같은 목표를 염두에 두고 개발되었습니다:

- 메트릭으로 사용 - 개발자 및 애플리케이션 소유자가 기존 모바일 앱을 비교할 수 있는 보안 표준 제공.
- 지침으로 사용 - 모바일 앱 개발 및 테스트의 모든 단계에 대한 지침 제공.
- 구매 시 사용 - 모바일 앱 보안 검증을 위한 기준 제공.

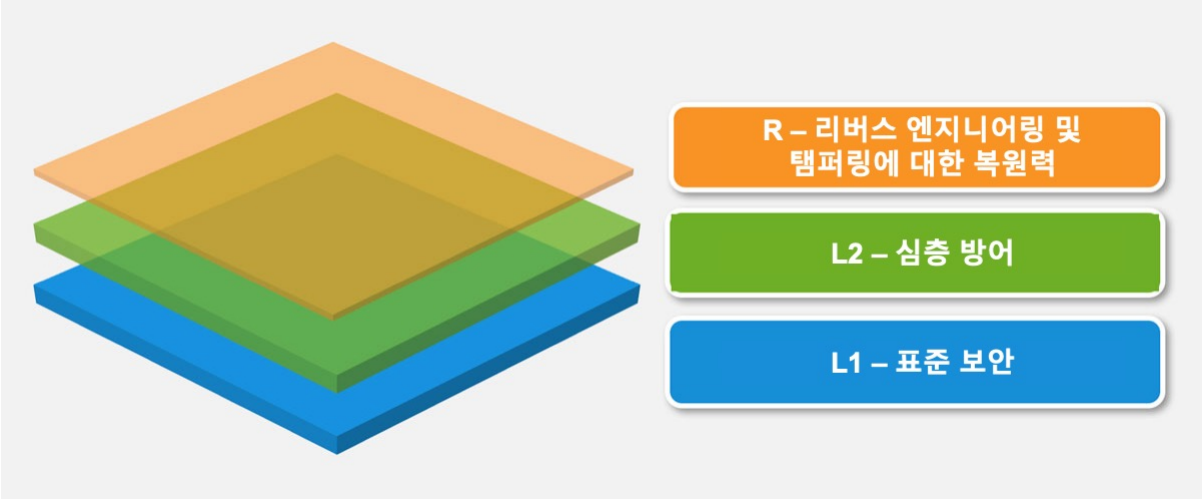
모바일 앱 보안 모델

MASVS 는 두 가지 보안 검증 수준 (MASVS-L1 and MASVS-L2) 과 일련의 리버스 엔지니어링 복원력 요구사항 (MASVS-R) 을 정의하고 있습니다. MASVS-L1 은 모든 모바일 앱에 권장되는 일반적인 보안 요구사항을 포함하고 있으며, MASVS-L2 는 매우 민감한 데이터를 처리하는 앱에 적용해야 합니다. MASVS-R 은 클라이언트측 위협을 방지하는 것이 설계 목표인 경우에 적용할 수 있는 추가 보호 요구사항을 다루고 있습니다.

MASVS-L1 의 요구사항을 충족하면 보안 모범 사례에 따라 일반적으로 취약점이 없는 안전한 앱을 완성할 수 있습니다. MASVS-L2 는 SSL 피닝과 같은 심층 방어 통제를 추가하여 보다 정교한 공격에 내성이 있는 앱을 제공합니다. 모바일 운영체제의 보안 통제에 문제가 없고 최종 사용자가 잠재적인 공격자로 간주되지 않는다고 가정합니다. MASVS-R 의 소프트웨어 보호 요구사항 전부 또는 일부를 충족하면 최종 사용자가 악의적이거나 모바일 OS 가 손상되는 특정 클라이언트측 위협을 방지할 수 있습니다.

I: 모든 앱에서 MASVS-L1 통제를 구현하는 것을 권고하지만, 통제권 구현은 궁극적으로 비즈니스 소유자와 의사 소통을 통해 위험 기반으로 결정되어야 합니다.

II: MASVS-R 에 나열되고 OWASP Mobile Security Testing Guide 에 설명된 소프트웨어 보호 통제는 궁극적으로 우회될 수 있으며, 보안 통제의 대체 수단으로 사용되어서는 안됩니다. 다만, 신규 위협을 명시하고 MASVS-L1 또는 MASVS-L2 에 명시된 MASVS 요구사항을 충족하는 앱의 보호 통제를 추가하기 위한 것입니다.



문서의 구성

MASVS 의 첫 번째 부분에는 보안 모델과 사용 가능한 검증 수준에 대한 설명이 수록되어 있으며, 그 다음이 실제로 표준을 사용하는 방법에 대한 권고사항입니다. 검증 수준에 대한 맵핑과 함께 자세한 보안 요구사항이 두 번째 파트에 나열되어 있습니다. 요구사항은 기술적 목표/범위를 기준으로 8 개 범주 (V1 ~ V8) 로 분류되어 있습니다. MASVS 및 MSTG 전체에서 다음과 같은 명명법이 사용됩니다:

- 요구사항 카테고리*:* MASVS-Vx, 예. MASVS-V2: 데이터 스토리지 및 개인정보보호
- 요구사항*:* MASVS-Vx.y, 예. MASVS-V2.2: “민감한 데이터는 애플리케이션 로그에 기록되지 않습니다.”

세부 검증 수준

MASVS-L1: 표준 보안

MASVS-L1 을 달성한 모바일 앱은 모바일 애플리케이션 보안 모범 사례를 준수합니다. 코드 품질, 민감한 데이터 처리 및 모바일 환경과의 상호작용 측면에서 기본 요구사항을 충족합니다. 보안 통제를 확인하기 위한 테스트 프로세스가 있어야 합니다. 이 수준은 모든 모바일 애플리케이션에 적합합니다.

MASVS-L2: 심층 방어

MASVS-L2 에는 표준 요구사항을 능가하는 고급 보안 통제 기능이 도입되었습니다. MASVS-L2 를 충족하려면 위협 모델이 존재해야하며 보안은 앱 아키텍처와 디자인에 필수적인 부분이어야 합니다. 위협 모델을 기반으로 올바른 MASVS-L2 컨트롤을 선택하고 성공적으로 구현해야 합니다. 이 수준은 모바일 뱅킹 앱처럼 매우 민감한 데이터를 처리하는 앱에 적합합니다.

MASVS-R: 리버스 엔지니어링 및 탬퍼링에 대한 복원력

이 앱은 최첨단 보안을 갖추고 있으며, 민감한 코드나 데이터를 추출하기 위해 변조, 위조, 역공학 등 구체적이고 명확하게 정의된 클라이언트측 공격에 대해서도 탄력적입니다. 이러한 앱은 하드웨어 보안 기능 또는 충분히 강력하고 검증 가능한 소프트웨어 보호 기술을 활용합니다. MASVS-R 은 매우 민감한 데이터를 처리하는 앱에 적용할 수 있으며 지적 재산을 보호하거나 앱을 조작 방지하는 수단으로 사용될 수 있습니다.

권장 사용

애플리케이션은 사전 위험 평가 및 필요한 전반적인 보안 수준에 근거하여 MASVS L1 또는 L2 에 대해 검증할 수 있습니다. L1 은 모든 모바일 앱에 적용할 수 있지만 L2 는 일반적으로보다 민감한 데이터 또는 기능을 처리하는 앱에 권장됩니다. MASVS-R(또는 그 일부) 을 적용하면 적절한 보안 검증 외에도 추가적으로 민감한 데이터의 리패키징 또는 추출과 같은 특정 위협에 대한 복원력을 검증할 수 있습니다.

요약하면 다음과 같은 검증 유형을 사용할 수 있습니다:

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

다른 조합은 다른 등급의 보안 및 복원력을 반영합니다. 목표는 유연성을 허용하는 것입니다: 예를 들어, 모바일 게임은 사용적합성 이유로 2 단계 인증과 같은 MASVS-L2 보안 통제장치를 추가하는 것을 보증하지 않을 수 있지만, 조작 방지에 대한 강력한 비즈니스 요구가 있을 수 있습니다.

선택할 검증 유형

MASVS L2 의 요구사항을 구현하면 보안이 향상되는 동시에 개발 비용이 증가하고 최종 사용자 경험이 악화될 수 있습니다.(전통적인 절충) 일반적으로 L2 는 위험 대비 비용 관점에서 의미가 있을때 앱에 사용해야 합니다.(즉, 기밀성 또는 무결성의 타협으로 인한 잠재적 손실이 추가 보안 통제에서 발생하는 비용보다 더 큰 경우) MASVS 를 적용하기 전에 위험평가가 첫 번째 단계이여야 합니다.

사례

MASVS-L1

- 모든 모바일 앱. MASVS-L1 에는 개발 비용 및 사용자 경험에 합리적인 영향을 줄 수 있는 보안 모범 사례가 나와 있습니다. 상위 레벨 중 하나에 해당하지 않는 앱에 대해서는 MASVS-L1 의 요구사항을 적용하십시오.

MASVS-L2

- 헬스케어 산업: 신분 도용, 사기 지불 또는 다양한 범죄 행위에 사용할 수 있는 개인 식별 정보를 저장하는 모바일 앱. 미국 의료 부문의 경우 규정 준수 고려 사항에는 미국의 의료보험의 양도 및 책임에 관한 법률 (HIPAA) 의 개인정보보호, 보안, 위반 통지 규칙 및 환자 안전 규칙이 포함됩니다.
- 금융 산업: 신용카드번호, 개인정보와 같이 매우 민감한 정보에 액세스하거나 사용자가 송금할 수 있는 앱. 이러한 앱은 사기를 방지하기 위해 추가 보안 통제를 보장해야 합니다. 금융 앱은 신용카드업계 데이터 보안 표준 (PCI DSS), 금융서비스현대화법 (그램 리치 블라일리법, Gramm Leach Bliley Act) 및 사베인-옥슬리법 (Sarbanes-Oxley Act, SOx) 를 준수해야 합니다.

MASVS L1+R

- 지적재산권 (IP) 보호가 비즈니스 목표인 모바일 앱. MASVS-R 에 나열된 복원력 통제는 원래 소스코드를 획득하는 데 필요한 노력을 증가시키고 위변조를 방해하기 위해 사용할 수 있습니다.
- 게임 산업: 경쟁력있는 온라인 게임과 같이 부정행위를 방지하는 데 필수적인 필요가 있는 게임. 부정행위는 많은 양의 사기꾼이 불만을 가진 플레이어 기반으로 이어져 궁극적으로 게임이 실패할 수 있기 때문에 온라인 게임에서 중요한 문제입니다. MASVS-R 은 부정행위자의 부담을 증가 시키기 위해 기본적인 변조 방지 컨트롤을 제공합니다.

MASVS L2+R

- 금융 산업: 사용자가 송금할 수 있는 온라인 뱅킹 앱으로, 코드 삽입이나 손상된 장치의 계측과 같은 기술이 위험에 노출될 수 있습니다. 이 경우 MASVS-R 의 컨트롤을 사용하여 조작을 방해하여 맬웨어 작성자의 부담을 증가 시킬 수 있습니다.
- 설계상 민감한 데이터를 모바일 장치에 저장해야하는 동시에 모든 장치와 운영체제 버전을 지원해야하는 모든 모바일 앱. 이 경우 복원력 통제는 민감한 데이터 추출을 목표로하는 공격자의 노력을 증가시키기 위한 심층 방어 수단으로 사용될 수 있습니다.

- 앱내 구매 기능이 있는 앱은 유료 콘텐츠를 보호하기 위해 서버측 및 MASVS-L2 통제를 이상적으로 사용해야 합니다. 단, 서버측 보호를 이용할 가능성이 없는 경우가 있을 수 있습니다. 이 경우 리버싱 또는 변조 방지력을 높이기 위해 MASVS-R 컨트롤을 추가로 적용해야 합니다.

평가 및 인증

MASVS 인증 및 신뢰 마크에 대한 OWASP 의 견해

OWASP 는 벤더 중립적인 비영리 단체로서 벤더, 검증자 또는 소프트웨어 인증은 실시하지 않습니다.

이러한 모든 보증 주장, 신뢰 마크 또는 인증은 OWASP 에 의해 공식적으로 검증, 등록 또는 인증되지 않으므로, 그러한 견해에 의존하는 조직은 (M)ASVS 인증을 주장하는 제 3 자에 또는 신뢰 마크에 대한 신뢰에 주의를 기울여야 합니다.

이는 조직이 공식적인 OWASP 인증을 요구하지 않는 한, 그러한 보증 서비스를 제공하는 것을 방해해서는 않됩니다.

모바일 앱 인증 지침

모바일 앱이 MASVS 를 준수하는지 확인하기 위해 권장되는 방법은 “오픈북” 검토를 수행하는 것이며, 앱 설계자와 개발자, 프로젝트 문서, 소스 코드 및 각 역할에 대해 적어도 하나 이상의 사용자 계정에 대한 액세스를 포함한 엔드 포인트에 인증된 액세스와 같은 주요 리소스에 대한 액세스 권한을 테스트 기술자에 부여하여야 합니다.

MASVS 는 (클라이언트 측) 모바일 앱의 보안 및 앱과 원격 엔드포인트 간의 네트워크 통신뿐만 아니라, 사용자 인증 및 세션 관리와 관련된 몇 가지 기본적이고 일반적인 요구사항을 다루고 있습니다. 앱과 관련된 원격 서비스 (예: 웹 서비스) 에 대한 특정 요구 사항은 포함하지 않으며, 인증 및 세션 관리와 관련된 제한된 일반적인 요구사항에 대해 안전합니다. 하지만, MASVS V1 은 원격 서비스를 전체 위협 모델에서 다루며, OWASP ASVS 와 같은 적절한 표준에 따라 검증해야 한다고 명시하고 있습니다.

인증 기관은 모든 보고서에 검증의 범위 (특히 중요한 구성요소가 범위 외인 경우), 합격 및 불합격 테스트를 포함한 검증 결과의 요약, 불합격 테스트에 대한 명확한 해결책을 포함해야합니다. 자세한 작업 문서, 스크린 샷 또는 동영상, 문제를 안정적이고 반복적으로 악용하는 스크립트, 프록시 로그 등 테스트의 전자적 기록을 정리한 목록과 같은 관련 메모를 유지하는 것은 업계 표준 관행으로 간주됩니다. 단순히 도구를 실행하고 실패를 보고하는 것만으로는 충분하지 않습니다. 이는 인증 수준의 모든 문제가 철저히 테스트되고 테스트되었다는 충분한 증거를 제공하지 않습니다. 논란이 있을 경우 검증된 모든 요구사항이 실제로 테스트되었음을 입증할 수 있는 충분한 증거가 있어야 합니다.

OWASP 모바일 보안 테스트 가이드 (MSTG) 사용

OWASP MSTG 는 모바일 앱의 보안을 테스트하기 위한 설명서입니다. MASVS 에 나열된 요구사항을 확인하기 위한 기술 프로세스를 설명하고 있습니다. MSTG 에는 MASVS 의 요구사항에 각각 매핑되는 테스트 사례 목록이 포함되어 있습니다. MASVS 요구사항은 수준이 높고 일반적이지만, MSTG 는 모바일 OS 별로 자세한 권장 사항과 테스트 절차를 제공합니다.

자동 보안 테스트 도구의 역할

가능한 경우 효율을 높이기 위해 소스코드 스캐너와 블랙박스 테스트 도구를 사용하는 것이 좋습니다. 하지만 자동화된 도구만으로는 MASVS 검증을 완료할 수 없습니다. 모든 모바일 앱은 서로 다르며, 사용 중인 특정 기술과 프레임워크의 전반적인 아키텍처, 비즈니스 로직 및 기술적 함정을 이해하는 것은 앱의 보안을 검증하기 위한 필수 요건입니다.

기타 용도

고급 보안 아키텍처 지침

모바일 애플리케이션 보안 검증 표준의 가장 일반적인 용도 중 하나는 보안 설계자를 위한 리소스입니다. SABSA 또는 TOGAF 의 두 가지 주요 보안 아키텍처 프레임워크에는 모바일 애플리케이션 보안 아키텍처 검토를 완료하는 데 필요한 많은 정보가 누락되어 있습니다. MASVS 를 사용하면 보안 설계자가 모바일 앱에서 흔히 발생하는 문제에 대해 더 나은 통제를 선택할 수 있으므로 이러한 격차를 해소할 수 있습니다.

상용 시큐어코딩 체크리스트 대체

대부분의 조직에서 MASVS 를 채택하여 두 가지 수준 중 하나를 선택하거나, MASVS 를 분기하여 각 애플리케이션의 위험 수준에 필요한 사항을 도메인별로 변경하여 적용함으로써 이점을 얻을 수 있습니다. 추적성이 유지되는 한 이러한 유형의 분기를 권장하므로 앱이 4.1 요구사항을 통과한 경우 표준이 발전함에 따라 분기 사본과 동일한 의미를 갖습니다.

보안 테스트 방법론의 기초

우수한 모바일 앱 보안 테스트 방법은 MASVS 에 나열된 모든 요구사항을 포함해야 합니다. OWASP 모바일 보안 테스트 가이드 (MSTG) 는 검증 요구사항에 대한 블랙박스와 화이트박스 테스트 사례에 대해 설명하고 있습니다.

자동화된 단위 및 통합 테스트 가이드

MASVS 는 아키텍처 요구사항을 제외하고는 테스트가 가능하도록 설계되었습니다. MASVS 요구사항을 기반으로 한 자동화된 단일, 통합 및 승인 테스트는 지속적인 개발 라이프사이클에 통합할 수 있습니다. 이를 통해 개발자 보안 인식이 향상될 뿐만 아니라, 결과적으로 앱의 전반적인 품질이 향상되고, 릴리즈 전 단계에서 보안 테스트 검증량이 감소합니다.

개발 보안 교육

MASVS 를 사용하여 안전한 모바일 앱의 특성을 정의할 수도 있습니다. 대부분의 “개발 보안” 과정은 간단한 코딩 팁이 포함된 윤리적 해킹 과정입니다. 이는 개발자에게 도움이 되지 않습니다. 대신, 개발 보안 과정에서는 MASVS 를 사용할 수 있으며, MASVS 에 문서화된 사전 예방적 통제에 중점을 두고 상위 10 가지 코드 보안 문제를 중점적으로 다룰 수 있습니다.

V1: 아키텍처, 디자인 및 위협 모델링 요구사항

통제 목표

이상적인 세계에서는 모든 개발 단계에서 보안이 고려될 것입니다. 그러나 실제로 보안은 SDLC 의 마지막 단계에서만 고려되는 경우가 많습니다. 기술적인 통제 외에도, MASVS 는 모바일 앱의 아키텍처를 계획할 때 보안이 명시적으로 해결되었는지 확인하고, 모든 구성 요소의 기능 및 보안 역할이 명확하다는 것을 보장하는 프로세스를 갖추는 것을 요구합니다. 대부분의 모바일 앱은 원격 서비스 클라이언트로 동작하기 때문에 해당 서비스에도 적절한 보안 기준을 적용해야 합니다. - 모바일 앱을 단독으로 테스트하는 것만으로는 충분하지 않습니다.

“V1” 카테고리에는 앱의 아키텍처와 디자인에 관련된 요구사항이 나열되어 있습니다. 따라서, OWASP 모바일 테스트 가이드의 기술 테스트 사례에 매핑되지 않는 카테고리는 이 카테고리가 유일합니다. 위협 모델링, 보안 SDLC, 키 관리와 같은 주제를 다루기 위해 MASVS 사용자는 각 OWASP 프로젝트 또는 아래에 링크된 것과 같은 다른 표준을 참조해야 합니다.

보안 검증 요구사항

MASVS-L1 및 MASVS-L2 의 요구사항은 다음과 같습니다.

#	MSTG-ID	설명	L1	L2
1.1	MSTG-ARCH-1	모든 앱 구성 요소가 필요한 것으로 식별되어야 한다.	✓	✓
1.2	MSTG-ARCH-2	보안 통제는 클라이언트측에서만 적용되는 것이 아니라 각각의 원격 엔드 포인트에서도 적용되어야 한다.	✓	✓
1.3	MSTG-ARCH-3	모바일 앱과 연결되는 모든 원격 서비스에 수준 높은 아키텍처가 정의되어야 하고 해당 아키텍처에서 보안이 지원되어야 한다.	✓	✓
1.4	MSTG-ARCH-4	모바일 앱의 컨텍스트에서 민감한 것으로 간주되는 데이터가 명확하게 식별되어야 한다.	✓	✓
1.5	MSTG-ARCH-5	모든 앱 구성 요소는 비즈니스 기능과 보안 기능이 적용되어야 한다.		✓
1.6	MSTG-ARCH-6	모바일 앱과 연관된 원격 서비스의 위협 모델을 만들어 잠재적인 위협에 대한 대책을 적용하여야 한다.		✓
1.7	MSTG-ARCH-7	모든 보안 통제는 중앙 집중식으로 구현되어야 한다.		✓
1.8	MSTG-ARCH-8	암호화 키 (있는 경우) 를 관리하는 방법에 대한 명시적인 정책이 있으며 암호화 키의 수명주기가 적용되어야 한다. (NIST SP 800-57 등과 같은 키 관리 표준을 준수하는 것이 좋음)		✓
1.9	MSTG-ARCH-9	모바일 앱의 업데이트를 강제화하는 메커니즘이 존재하여야 하다.		✓
1.10	MSTG-ARCH-10	소프트웨어 개발 수명주기의 모든 부분에서 보안을 적용하여야 한다.		✓
1.11	MSTG-ARCH-11	책임 있는 공개 정책이 시행되고 있으며 효과적으로 적용되어야 한다.		✓
1.12	MSTG-ARCH-12	앱은 개인정보 보호법 및 규정을 준수해야 한다.	✓	✓

참고

자세한 내용은 다음을 참조하십시오:

- OWASP Mobile Top 10: M10 (Extraneous Functionality) - https://owasp.org/www-project-mobile-top-10/2016-risks/m10-extraneous-functionality
- OWASP Security Architecture cheat sheet - https://www.owasp.org/index.php/Application_Security_Architecture_Cheat_Sheet
- OWASP Threat 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
- security.txt - https://securitytxt.org/

V2: 데이터 저장 및 개인 정보 요구사항

통제 목표

사용자 자격 증명 및 개인 정보와 같은 민감한 데이터를 보호하는 것이 모바일 보안의 핵심 초점입니다. 먼저, IPC 와 같은 운영체제 메커니즘을 부적절하게 사용하게 되면 민감한 데이터가 의도하지 않게 동일한 장치에서 실행되는 다른 앱에 노출될 수 있습니다. 또한 데이터가 실수로 클라우드 저장소, 백업 또는 키보드 캐시로 유출될 수 있습니다. 또한 모바일 기기는 다른 유형의 기기에 비해 더 쉽게 분실하거나 도난당할 수 있으므로 공격자가 물리적으로 접근하기에 더 용이한 시나리오가 될 수 있습니다. 이 경우 민감한 데이터의 취득을 더 어렵게하기 위해 추가적인 보호대책이 필요합니다.

주의: MASVS 는 앱 중심이므로 MDM 솔루션에 의해 시행되는 것과 같은 장치 수준 정책은 다루지 않습니다. 우리는 기업적 측면에에서 데이터 보안을 보다 강화하기 위해 이러한 정책을 사용하는 것을 권장합니다.

민감한 데이터의 정의

MASVS 의 민감한 데이터는 사용자 자격 증명과 특정 상황에서 민감한 것으로 간주되는 기타 모든 데이터를 포함합니다.

- 신분 도용에 악용될 수 있는 개인 식별 정보 (PII): 사회보장번호 (주민등록번호), 신용카드 번호, 은행 계좌번호, 건강 정보
- 유출될 경우 명예 실추 및 금전적 손실로 이어질 민감한 데이터: 계약 정보, 비공개 계약에 포함된 정보, 관리 정보
- 법률 또는 컴플라이언스 이유로 보호되어야 하는 모든 데이터.

보안 검증 요구사항

대부분의 데이터 공개의 문제는 아래의 간단한 규칙을 따르면 예방할 수 있습니다. 이 장에 나열된 대부분의 통제는 모든 검증 수준에서 필수 사항입니다.

#	MSTG-ID	설명	L1	L2
2.1	MSTG-STORAGE-1	개인 식별 정보 (PII), 사용자 자격 증명 암호화 키 같은 중요한 데이터를 저장할 경우 시스템 자격 증명 저장소를 적절하게 사용하여야 한다.	✓	✓
2.2	MSTG-STORAGE-2	민감한 데이터는 앱 컨테이너 또는 시스템 자격 증명 저장 시설 외부에 저장하지 않아야 한다.	✓	✓
2.3	MSTG-STORAGE-3	민감한 데이터는 응용 프로그램 로그에 기록하지 않아야 한다.	✓	✓
2.4	MSTG-STORAGE-4	민감한 데이터는 아키텍처에서 필요한 부분이 아닌 한 제 3 자와 공유하지 않아야 한다.	✓	✓
2.5	MSTG-STORAGE-5	민감한 데이터를 처리하는 텍스트 입력에서 키보드 캐시가 비활성화 되어야 한다.	✓	✓
2.6	MSTG-STORAGE-6	민감한 데이터는 IPC 메커니즘을 통해 노출되지 않아야 한다.	✓	✓
2.7	MSTG-STORAGE-7	비밀번호 또는 핀과 같은 민감한 데이터는 사용자 인터페이스를 통해 노출되지 않아야 한다.	✓	✓

#	MSTG-ID	설명	L1	L2
2.8	MSTG-STORAGE-8	민감한 데이터는 모바일 운영체제에서 생성된 백업에 포함되지 않아야 한다.		✓
2.9	MSTG-STORAGE-9	민감한 데이터는 앱이 백그라운드로 이동할 때 뷰에서 제거되어야 한다.		✓
2.10	MSTG-STORAGE-10	민감한 데이터는 앱이 필요한 것보다 더 긴 시간 동안 메모리에 유지되지 않아야 하며, 사용 후에는 메모리에서 명시적으로 삭제하여야 한다.		✓
2.11	MSTG-STORAGE-11	앱은 사용자에게 장치 암호를 설정하도록 요구하는 것과 같은 최소한의 장치 액세스 보안 정책을 설정하도록 하여야 한다.		✓
2.12	MSTG-STORAGE-12	앱은 처리되는 개인 식별 정보가 처리되는 방식과 사용자가 앱 사용시 준수해야하는 보안 모범 사례에 대해 통지하여야 한다.		✓
2.13	MSTG-STORAGE-13	민감한 데이터는 모바일 장치 로컬에 저장해서는 안된다. 대신 필요한 경우 원격 엔드포인트에서 데이터를 검색하고 메모리에만 보관하여야 한다.		✓
2.14	MSTG-STORAGE-14	민감한 데이터를 여전히 로컬에 저장해야하는 경우라면, 인증이 필요한 하드웨어 지원 저장소에서 파생된 키를 사용하여 암호화하여야 한다.		✓
2.15	MSTG-STORAGE-15	과도한 인증 시도 실패 후에는 앱의 로컬 저장소를 지워야 한다.		✓

참고

OWASP 모바일 보안 테스트 안내서 (MSTG) 는 이 섹션에 나열된 요구사항을 확인하기 위한 자세한 지침을 제공합니다.

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

자세한 내용은 다음을 참조하십시오:

- OWASP Mobile Top 10: M1 (Improper Platform Usage) - https://owasp.org/www-project-mobile-top-10/2016-risks/m1-improper-platform-usage
- OWASP Mobile Top 10: M2 (Insecure Data Storage) - https://owasp.org/www-project-mobile-top-10/2016-risks/m2-insecure-data-storage
- CWE 117 (Improper Output Neutralization for Logs) - https://cwe.mitre.org/data/definitions/117.html
- CWE 200 (Information Exposure) - https://cwe.mitre.org/data/definitions/200.html
- CWE 276 (Incorrect Default Permissions) - https://cwe.mitre.org/data/definitions/276.html
- CWE 311 (Missing Encryption of Sensitive Data) - https://cwe.mitre.org/data/definitions/311.html
- CWE 312 (Cleartext Storage of Sensitive Information) - https://cwe.mitre.org/data/definitions/312.html

- CWE 316 (Cleartext Storage of Sensitive Information in Memory) - https://cwe.mitre.org/data/definitions/316.html
- CWE 359 (Exposure of Private Information ('Privacy Violation')) - https://cwe.mitre.org/data/definitions/359.html
- CWE 522 (Insufficiently Protected Credentials) - https://cwe.mitre.org/data/definitions/522.html
- CWE 524 (Information Exposure Through Caching) - https://cwe.mitre.org/data/definitions/524.html
- CWE 530 (Exposure of Backup File to an Unauthorized Control Sphere) - https://cwe.mitre.org/data/definitions/530.html
- CWE 532 (Information Exposure Through Log Files) - https://cwe.mitre.org/data/definitions/532.html
- CWE 534 (Information Exposure Through Debug Log Files) - https://cwe.mitre.org/data/definitions/534.html
- CWE 634 (Weaknesses that Affect System Processes) - https://cwe.mitre.org/data/definitions/634.html
- CWE 798 (Use of Hard-coded Credentials) - https://cwe.mitre.org/data/definitions/798.html
- CWE 921 (Storage of Sensitive Data in a Mechanism without Access Control) - https://cwe.mitre.org/data/definitions/921.html
- CWE 922 (Insecure Storage of Sensitive Information) - https://cwe.mitre.org/data/definitions/922.html

V3: 암호화 요구사항

통제 목표

암호화는 모바일 장치에 저장된 데이터를 보호하는데 필수적인 요소입니다. 또한 표준 규칙을 따르지 않을 때 상황이 끔찍하게 잘못될 수 있는 영역이기도 합니다. 이 장의 통제 목적은 검증된 애플리케이션이 다음과 같은 업계 모범 사례에 따라 암호화를 사용하는지 확인하는 것입니다.

- 검증된 암호화 라이브러리 사용
- 암호화 기본 요소의 올바른 선택 및 구성
- 무작위 값이 필요한 경우 적절한 난수 발생기 사용

보안 검증 요구사항

#	MSTG-ID	설명	L1	L2
3.1	MSTG-CRYPTO-1	앱은 암호화의 유일한 방법으로 하드 코드 된 키를 사용하는 암호화에 의존하지 않아야 한다.	✓	✓
3.2	MSTG-CRYPTO-2	앱은 검증된 암호화 알고리즘으로 구현하여야 한다.	✓	✓
3.3	MSTG-CRYPTO-3	앱은 업계 모범 사례를 준수하는 매개 변수로 구성된 특정 유스케이스에 적합한 암호화 알고리즘을 사용하여야 한다.	✓	✓
3.4	MSTG-CRYPTO-4	앱은 보안적인 목적으로 더 이상 사용되지 않고 사라질 암호화 프로토콜과 알고리즘을 사용하지 않아야 한다.	✓	✓
3.5	MSTG-CRYPTO-5	앱은 여러 목적으로 동일한 암호화 키를 재사용하지 않아야 한다.	✓	✓
3.6	MSTG-CRYPTO-6	모든 난수 값은 충분히 안전한 난수 생성기를 사용하여 생성하여야 한다.	✓	✓

참고

OWASP 모바일 보안 테스트 안내서 (MSTG) 는 이 섹션에 나열된 요구사항을 확인하기 위한 자세한 지침을 제공합니다.

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

자세한 내용은 다음을 참조하십시오:

- OWASP Mobile Top 10: M5 (Insufficient Cryptography) - https://owasp.org/www-project-mobile-top-10/2016-risks/m5-insufficient-cryptography
- CWE 310 (Cryptographic Issues) - https://cwe.mitre.org/data/definitions/310.html
- CWE 321 (Use of Hard-coded Cryptographic Key) - https://cwe.mitre.org/data/definitions/321.html

- CWE 326 (Inadequate Encryption Strength) - https://cwe.mitre.org/data/definitions/326.html
- CWE 327 (Use of a Broken or Risky Cryptographic Algorithm) - https://cwe.mitre.org/data/definitions/327.html
- CWE 329 (Not Using a Random IV with CBC Mode) - https://cwe.mitre.org/data/definitions/329.html
- CWE 330 (Use of Insufficiently Random Values) - https://cwe.mitre.org/data/definitions/330.html
- CWE 337 (Predictable Seed in PRNG) - https://cwe.mitre.org/data/definitions/337.html
- CWE 338 (Use of Cryptographically Weak Pseudo Random Number Generator (PRNG)) - https://cwe.mitre.org/data/definitions/338.html

V4: 인증 및 세션 관리 요구사항

통제 목표

대부분의 경우 원격 서비스에 로그인하는 사용자는 전체 모바일 앱 아키텍처에서 필수적인 부분입니다. 대부분의 로직이 엔드포인트에서 발생하더라도 MASVS 는 사용자 계정 및 세션을 관리하는 방법에 대한 몇 가지 기본 요구사항을 정의하고 있습니다.

보안 검증 요구사항

#	MSTG-ID	설명	L1	L2
4.1	MSTG-AUTH-1	앱이 사용자에게 원격 서비스에 대한 액세스를 제공하는 경우 사용자 이름과 암호로의 인증 방식은 원격 엔드 포인트에서 수행되어야 한다.	✓	✓
4.2	MSTG-AUTH-2	상태 저장 세션 관리를 사용하는 경우 원격 엔드 포인트는 무작위로 생성된 세션 식별자를 사용하여 사용자의 자격 증명을 보내지 않고 클라이언트 요청을 인증하여야 한다.	✓	✓
4.3	MSTG-AUTH-3	상태 비 저장 토큰 기반 인증을 사용하는 경우 서버는 보안 알고리즘을 사용하여 서명된 토큰을 제공하여야 한다.	✓	✓
4.4	MSTG-AUTH-4	사용자가 로그아웃하면 원격 엔드 포인트는 기존의 세션을 종료하여야 한다.	✓	✓
4.5	MSTG-AUTH-5	비밀번호 정책이 존재하며 원격 엔드 포인트에서 검증되어야 한다.	✓	✓
4.6	MSTG-AUTH-6	원격 엔드 포인트는 과도한 인증 시도에 대한 보호 메커니즘을 구현하여야 한다.	✓	✓
4.7	MSTG-AUTH-7	사전 정의 된 비 활동 기간 및 액세스 토큰이 만료 된 후 원격 엔드 포인트에서 세션이 무효화되어야 한다.	✓	✓
4.8	MSTG-AUTH-8	생체 인식 인증이 사용되는 경우 이벤트 바인딩 (예: 단순히 “true” 또는 “false” 를 반환하는 API 사용) 되어서는 안된다. 대신 키 체인 및 키 스토어 잠금 해제를 할 때만 사용하여야 한다.		✓
4.9	MSTG-AUTH-9	2 단계 인증 요소는 원격 엔드 포인트에 존재하여야 하며, 2FA 요구 사항이 지속적으로 적용되어야 한다.		✓
4.10	MSTG-AUTH-10	민감한 트랜잭션에는 단계별 인증을 적용하여야 한다.		✓
4.11	MSTG-AUTH-11	앱은 사용자에게 사용자 계정의 모든 민감한 활동을 알려야 한다. 사용자는 장치 목록을 보거나, 상태 정보 (IP 주소, 위치 등) 를 보고 특정 장치를 차단할 수 있어야 한다.		✓
4.12	MSTG-AUTH-12	인증 모델은 원격 엔드포인트에서 정의되고 시행되어야 한다.	✓	✓

참고

OWASP 모바일 보안 테스트 안내서 (MSTG) 는 이 섹션에 나열된 요구사항을 확인하기 위한 자세한 지침을 제공합니다.

- General: Authentication and Session Management - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x04e-Testing-Authentication-and-Session-Management.md
- Android: Testing Local Authentication - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05f-Testing-Local-Authentication.md
- iOS: Testing Local Authentication - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06f-Testing-Local-Authentication.md

자세한 내용은 다음을 참조하십시오:

- OWASP Mobile Top 10: M4 (Insecure Authentication) - https://owasp.org/www-project-mobile-top-10/2016-risks/m4-insecure-authentication
- OWASP Mobile Top 10: M6 (Insecure Authorization) - https://owasp.org/www-project-mobile-top-10/2016-risks/m6-insecure-authorization
- CWE 287 (Improper Authentication) - https://cwe.mitre.org/data/definitions/287.html
- CWE 307 (Improper Restriction of Excessive Authentication Attempts) - https://cwe.mitre.org/data/definitions/307.html
- CWE 308 (Use of Single-factor Authentication) - https://cwe.mitre.org/data/definitions/308.html
- CWE 521 (Weak Password Requirements) - https://cwe.mitre.org/data/definitions/521.html
- CWE 604 (Use of Client-Side Authentication) - https://cwe.mitre.org/data/definitions/604.html
- CWE 613 (Insufficient Session Expiration) - https://cwe.mitre.org/data/definitions/613.html

V5: 네트워크 통신 요구사항

통제 목표

이 장에서 다루는 통제의 목적은 모바일 앱과 원격 서비스 엔드 포인트 간에 교환되는 정보의 기밀성과 무결성을 보장하기 위한 것입니다. 최소한 모바일 앱은 적절한 설정을 적용한 TLS 프로토콜을 사용하여 네트워크 통신을 위한 암호화된 보안 채널을 제공해야 합니다. 레벨 2 에는 SSL 피닝과 같은 심층 방어 조치가 추가되어 있습니다.

보안 검증 요구사항

#	MSTG-ID	설명	L1	L2
5.1	MSTG-NETWORK-1	데이터는 TLS 를 사용하여 네트워크에서 암호화되어야 한다. 보안 채널은 앱 전체에 일관되게 사용하여야 한다.	✓	✓
5.2	MSTG-NETWORK-2	TLS 설정은 현재 모범 사례와 일치하여야 하며, 모바일 운영 체제가 권장 표준을 지원하지 않는 경우 가능한 한 가장 가까운 모범 사례와 일치하여야 한다.	✓	✓
5.3	MSTG-NETWORK-3	보안 채널이 설정되면 앱은 원격 앤드 포인트의 X.509 인증서를 검증하여야 한다. 신뢰할 수 있는 CA 가 서명한 인증서만 허용하여야 한다.	✓	✓
5.4	MSTG-NETWORK-4	앱은 자체 인증서 저장소를 사용하거나 앤드 포인트 인증서 또는 공개 키를 피닝하여야 한다. 그 후 신뢰할 수 있는 CA 가 서명한 경우에도 다른 인증서 또는 키를 제공하는 앤드 포인트와의 연결을 거부할 수 있어야 한다.		✓
5.5	MSTG-NETWORK-5	앱은 등록 및 계정 복구와 같은 중요한 작업을 처리할 때 안전하지 않은 단일 통신 채널 (이메일 또는 SMS) 에 의존하지 않아야 한다.		✓
5.6	MSTG-NETWORK-6	앱은 최신 연결 라이브러리 및 보안 라이브러리에만 의존하여야 한다.		✓

참고

OWASP 모바일 보안 테스트 안내서 (MSTG) 는 이 섹션에 나열된 요구사항을 확인하기 위한 자세한 지침을 제공합니다.

- General: 네트워크 통신 테스트 - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x04f-Testing-Network-Communication.md
- Android: 네트워크 통신 테스트 - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05g-Testing-Network-Communication.md
- iOS: 네트워크 통신 테스트 - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06g-Testing-Network-Communication.md

자세한 내용은 다음을 참조하십시오:

- OWASP Mobile Top 10: M3 (Insecure Communication) - https://owasp.org/www-project-mobile-top-10/2016-risks/m3-insecure-communication
- CWE 295 (Improper Certificate Validation) - https://cwe.mitre.org/data/definitions/295.html
- CWE 296 (Improper Following of a Certificate's Chain of Trust) - https://cwe.mitre.org/data/definitions/296.html
- CWE 297 (Improper Validation of Certificate with Host Mismatch) - https://cwe.mitre.org/data/definitions/297.html
- CWE 298 (Improper Validation of Certificate Expiration) - https://cwe.mitre.org/data/definitions/298.html
- CWE 308 (Use of Single-factor Authentication) - https://cwe.mitre.org/data/definitions/308.html
- CWE 319 (Cleartext Transmission of Sensitive Information) - https://cwe.mitre.org/data/definitions/319.html
- CWE 326 (Inadequate Encryption Strength) - https://cwe.mitre.org/data/definitions/326.html
- CWE 327 (Use of a Broken or Risky Cryptographic Algorithm) - https://cwe.mitre.org/data/definitions/327.html
- CWE 780 (Use of RSA Algorithm without OAEP) - https://cwe.mitre.org/data/definitions/780.html
- CWE 940 (Improper Verification of Source of a Communication Channel) - https://cwe.mitre.org/data/definitions/940.html
- CWE 941 (Incorrectly Specified Destination in a Communication Channel) - https://cwe.mitre.org/data/definitions/941.html

V6: 플랫폼 상호 작용 요구사항

통제 목표

이 그룹의 통제는 앱이 플랫폼 API 와 표준 구성요소를 안전한 방식으로 사용한다는 것을 보장합니다. 또한, 이 통제에는 애플리케이션 간의 통신 (IPC) 을 포함합니다.

보안 검증 요구사항

#	MSTG-ID	설명	L1	L2
6.1	MSTG-PLATFORM-1	앱은 필요한 최소한의 권한만 요구하여야 한다.	✓	✓
6.2	MSTG-PLATFORM-2	외부 소스 및 사용자의 모든 입력에 대해 검증하고 필요한 경우 적절하게 처리하여야 한다. 여기에는 UI 를 통해 수신된 데이터, 인텐트, 사용자 정의 URL 및 네트워크 소스와 같은 IPC 메커니즘이 포함된다.	✓	✓
6.3	MSTG-PLATFORM-3	앱은 메커니즘이 제대로 보호되지 않는 한 사용자 정의 URL 체계를 통해 민감한 기능을 내보내지 않아야 한다.	✓	✓
6.4	MSTG-PLATFORM-4	엡은 메커니즘이 제대로 보호되지 않는 한 IPC 메터니즘을 통해 민감한 기능을 내보내지 않아야 한다.	✓	✓
6.5	MSTG-PLATFORM-5	명시적으로 필요한 경우가 아니면 웹뷰에서 자바스크립트를 사용하지 않아야 한다.	✓	✓
6.6	MSTG-PLATFORM-6	웹뷰는 필요 최소한의 프로토콜 핸들러 세트만 허용하도록 구성되어야 한다. (이상적으로는 https 만 지원) file, tel 및 app-id 와 같은 잠재적으로 위험한 핸들러는 비활성화하여야 한다.	✓	✓
6.7	MSTG-PLATFORM-7	앱의 네이티브 메소드가 웹뷰에 노출되는 경우 웹뷰가 앱 패키지에 포함된 자바스크립트만 렌더링하는지 검증하여야 한다.	✓	✓
6.8	MSTG-PLATFORM-8	객체 역직렬화는 안전한 직렬화 API 를 사용하여 구현하여야 한다.	✓	✓
6.9	MSTG-PLATFORM-9	애플리케이션은 화면 오버레이 공격으로부터 자신을 보호하여야 한다. (Android 만 해당)		✓
6.10	MSTG-PLATFORM-10	WebView 를 종료하기 전에 WebView 의 캐시, 스토리지 및 로드된 리소스 (JavaScript 등) 를 지워야 한다.		✓
6.11	MSTG-PLATFORM-11	민감한 데이터가 입력될 때마다 앱에서 사용자 지정 타사 키보드 사용을 방지하는지 확인하여야 한다.		✓

참고

OWASP 모바일 보안 테스트 안내서 (MSTG) 는 이 섹션에 나열된 요구사항을 확인하기 위한 자세한 지침을 제공합니다.

- Android: 플랫폼 상호 작용 테스트 - https://github.com/OWASP/owasp-mstg/blob/master/

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

자세한 내용은 다음을 참조하십시오:

- OWASP Mobile Top 10: M1 (Improper Platform Usage) - https://owasp.org/www-project-mobile-top-10/2016-risks/m1-improper-platform-usage
- OWASP Mobile Top 10: M7 (Poor Code Quality) - https://owasp.org/www-project-mobile-top-10/2016-risks/m7-client-code-quality
- CWE 20 (Improper Input Validation) - https://cwe.mitre.org/data/definitions/20.html
- CWE 79 (Improper Neutralization of Input During Web Page Generation) - https://cwe.mitre.org/data/definitions/79.html
- CWE 200 (Information Leak / Disclosure) - https://cwe.mitre.org/data/definitions/200.html
- CWE 250 (Execution with Unnecessary Privileges) - https://cwe.mitre.org/data/definitions/250.html
- CWE 672 (Operation on a Resource after Expiration or Release) - https://cwe.mitre.org/data/definitions/672.html
- CWE 749 (Exposed Dangerous Method or Function) - https://cwe.mitre.org/data/definitions/749.html
- CWE 772 (Missing Release of Resource after Effective Lifetime) - https://cwe.mitre.org/data/definitions/772.html
- CWE 920 (Improper Restriction of Power Consumption) - https://cwe.mitre.org/data/definitions/920.html
- CWE 925 (Improper Verification of Intent by Broadcast Receiver) - https://cwe.mitre.org/data/definitions/925.html
- CWE 926 (Improper Export of Android Application Components) - https://cwe.mitre.org/data/definitions/926.html
- CWE 927 (Use of Implicit Intent for Sensitive Communication) - https://cwe.mitre.org/data/definitions/927.html
- CWE 939 (Improper Authorization in Handler for Custom URL Scheme) - https://cwe.mitre.org/data/definitions/939.html

V7: 코드 품질 및 빌드 설정 요구사항

통제 목표

이 통제의 목표는 앱을 개발할 때 기본적인 보안 코딩 방법을 준수하고, 컴파일러에서 제공하는 "무료" 보안 기능이 활성화되도록 하는 것입니다.

보안 검증 요구사항

#	MSTG-ID	설명	L1	L2
7.1	MSTG-CODE-1	앱이 유효한 인증서로 서명 및 프로비저닝되어야 하며, 개인 키가 올바르게 보호되어야 한다.	✓	✓
7.2	MSTG-CODE-2	앱은 릴리 모드로 빌드되어 있어야 한다. (디버그 불가)	✓	✓
7.3	MSTG-CODE-3	네이티브 바이너리에서 디버그 기호가 제거되어야 한다.	✓	✓
7.4	MSTG-CODE-4	디버깅 코드 및 개발자 지원 코드 (예: 테스트 코드, 백도어, 숨겨진 설정) 가 제거되어야 한다. 앱은 자세한 (verbose) 오류나 디버깅 메시지를 기록하지 않아야 한다.	✓	✓
7.5	MSTG-CODE-5	앱에서 사용되는 라이브러리 및 프레임워크 등은 모든 타사 구성 요소를 식별하고 알려진 취약점이 있는지 확인하여야 한다.	✓	✓
7.6	MSTG-CODE-6	앱은 가능한 모든 예외를 포착하고 처리하여야 한다.	✓	✓
7.7	MSTG-CODE-7	보안 통제의 오류 처리 로직은 기본적으로 액세스를 거부하여야 한다.	✓	✓
7.8	MSTG-CODE-8	관리되지 않는 코드에서 메모리는 할당, 해제 및 안전하게 사용되어야 한다.	✓	✓
7.9	MSTG-CODE-9	바이트 코드의 경량화, 스택 보호, PIE 지원 및 자동 참조 카운팅과 같은 툴체인에서 제공하는 무료 보안 기능이 활성화되어야 한다.	✓	✓

참고

OWASP 모바일 보안 테스트 안내서 (MSTG) 는 이 섹션에 나열된 요구사항을 확인하기 위한 자세한 지침을 제공합니다.

- Android: 코드 품질 및 빌드 설정 테스트 - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x05i-Testing-Code-Quality-and-Build-Settings.md
- iOS: 코드 품질 및 빌드 설정 테스트 - https://github.com/OWASP/owasp-mstg/blob/master/Document/0x06i-Testing-Code-Quality-and-Build-Settings.md

자세한 내용은 다음을 참조하십시오:

- OWASP Mobile Top 10: M7 (Poor Code Quality) - https://owasp.org/www-project-mobile-top-10/2016-risks/m7-client-code-quality

- CWE 119 (Improper Restriction of Operations within the Bounds of a Memory Buffer) - https://cwe.mitre.org/data/definitions/119.html
- CWE 89 (Improper Neutralization of Special Elements used in an SQL Command) - https://cwe.mitre.org/data/definitions/89.html
- CWE 388 (7PK - Errors) - https://cwe.mitre.org/data/definitions/388.html
- CWE 489 (Leftover Debug Code) - https://cwe.mitre.org/data/definitions/489.html
- OWASP Mobile Top 10: M7 (Poor Code Quality) - https://owasp.org/www-project-mobile-top-10/2016-risks/m7-client-code-quality
- CWE 20 (Improper Input Validation) - https://cwe.mitre.org/data/definitions/20.html
- CWE 89 (Improper Neutralization of Special Elements used in an SQL Command) - https://cwe.mitre.org/data/definitions/89.html
- CWE 95 (Improper Neutralization of Directives in Dynamically Evaluated Code ('Eval Injection')) - https://cwe.mitre.org/data/definitions/95.html
- CWE 119 (Improper Restriction of Operations within the Bounds of a Memory Buffer) - https://cwe.mitre.org/data/definitions/119.html
- CWE 215 (Information Exposure through Debug Information) - https://cwe.mitre.org/data/definitions/215.html
- CWE 388 (7PK - Errors) - https://cwe.mitre.org/data/definitions/388.html
- CWE 489 (Leftover Debug Code) - https://cwe.mitre.org/data/definitions/489.html
- CWE 502 (Deserialization of Untrusted Data) - https://cwe.mitre.org/data/definitions/502.html
- CWE 511 (Logic/Time Bomb) - https://cwe.mitre.org/data/definitions/511.html
- CWE 656 (Reliance on Security through Obscurity) - https://cwe.mitre.org/data/definitions/656.html
- CWE 676 (Use of Potentially Dangerous Function) - https://cwe.mitre.org/data/definitions/676.html
- CWE 937 (OWASP Top Ten 2013 Category A9 - Using Components with Known Vulnerabilities) - https://cwe.mitre.org/data/definitions/937.html

V8: 복원력 요구사항

통제 목표

이 섹션에서는 민감한 데이터 또는 기능을 처리하거나 액세스하는 앱에 권장되는 심층 방어 조치에 대해 설명합니다. 이러한 통제를 적용하지 않더라도 취약점이 발생하지는 않습니다. 대신 리버스 엔지니어링 및 특정 클라이언트측 공격에 대한 앱의 복원력을 향상 시키기 위한 것입니다.

이 섹션의 통제는 앱의 무단 변조 또는 코드 리버스 엔지니어링으로 인한 위험 평가를 기반으로 필요에 따라 적용하여야 합니다. 비즈니스 위험 및 관련 기술 위협에 대해서는 OWASP 문서 “Technical Risks of Reverse Engineering and Unauthorized Code Modification Reverse Engineering and Code Modification Prevention”(아래 자료 참조) 를 참조하십시오.

아래 목록의 모든 통제가 효과적이기 위해서는 앱이 최소한 MASVS-L1 을 충족해야 합니다.(즉, 견고한 보안 통제가 있어야 함). 뿐만 아니라 V8 의 모든 하위 사항들이 요구됩니다. 예를 들어, “이해 방해” 에 나열된 난독화 통제는 “동적 분석 및 변조 방지” 및 “장치 바인딩” 과 결합되어야 합니다.

메모: 해당 소프트웨어 보호 기능을 보안 통제 대신 사용해서는 안됩니다. MASVR-R 에 나열된 통제는 MASVS 보안 요구사항을 충족하는 앱에 위협별 보호 통제를 추가하기 위한 것입니다.

다음과 같은 고려 사항이 적용됩니다:

1. 위협 모델은 방어하는 클라이언트측 위협을 명확하게 정의해야 합니다. 또한 체계가 제공하는 보호 등급을 지정해야 합니다. 예를 들어, 악성코드 작성자가 앱을 분석하는데 수동 리버스 엔지니어링에 상당한 노력을 투자하도록 만드는 것이 목표일 수 있습니다.
2. 위협 모델은 민감해야 합니다. 예를 들어, 공격자가 화이트박스 전체를 단순히 코드 리프팅 할 수 있다면 화이트박스 구현에서 암호화 키를 숨기는 것이 중요합니다.
3. 보호의 효과는 사용된 특정 유형의 변조 방지 및 난독화를 테스트한 경험이 있는 전문가가 항상 검증해야 합니다.(모바일 보안 테스트 가이드 (MSTG) 의 “리버스 엔지니어링” 및 “소프트웨어 보호 평가” 부분을 참조하십시오.)

동적 분석 및 변조 방지

#	MSTG-ID	설명	R
8.1	MSTG-RESILIENCE-1	앱은 사용자에게 경고하거나 앱을 종료하여 루팅 또는 탈옥 된 기기의 존재를 감지하여야 한다.	✓
8.2	MSTG-RESILIENCE-2	앱은 디버깅을 방지하거나 디버거 연결을 감지하여야 한다. 사용 가능한 모든 디버깅 프로토콜이 포함되어야 한다.	✓
8.3	MSTG-RESILIENCE-3	앱은 자체 샌드박스에서 실행 파일 및 중요한 데이터의 변조를 감지하여야 한다.	✓
8.4	MSTG-RESILIENCE-4	앱은 장치에 널리 사용되는 리버스 엔지니어링 도구 및 프레임워크의 존재를 감지하여야 한다.	✓
8.5	MSTG-RESILIENCE-5	앱은 에뮬레이터에서 실행되고 있는지 여부를 감지하고 대응하여야 한다.	✓

#	MSTG-ID	설명	R
8.6	MSTG-RESILIENCE-6	앱은 자체 메모리 공간에서 코드와 데이터 변조를 감지하여야 한다.	✓
8.7	MSTG-RESILIENCE-7	앱은 각 방어 유형 (8.1~8.6) 에서 여러 메커니즘을 구현하여야 한다. 복원력은 사용된 메커니즘의 독창성의 양 및 다양성과 비례합니다.	✓
8.8	MSTG-RESILIENCE-8	감지 메커니즘은 지연 응답과 스텔스 응답을 포함하여 다양한 종류 응답을 트리거하여야 한다.	✓
8.9	MSTG-RESILIENCE-9	프로그램 난독화가 적용되고, 동적 분석을 통한 역 난독처리를 방해하여야 한다.	✓

장치 바인딩

#	MSTG-ID	설명	R
8.10	MSTG-RESILIENCE-10	앱은 장치 고유의 여러 속성에서 파생되는 장치 지문을 사용하여 '장치 바인딩' 기능을 구현하여야 한다.	✓

이해 방해 (Impede Comprehension)

#	MSTG-ID	설명	R
8.11	MSTG-RESILIENCE-11	앱에 속하는 모든 실행 파일 및 라이브러리는 파일 수준에서 암호화되거나 실행 파일 내의 중요한 코드 및 데이터 세그먼트가 암호화되거나 압축되어야 한다. 간단한 정적 분석은 중요한 코드나 데이터가 노출되지 않아야 한다.	✓
8.12	MSTG-RESILIENCE-12	난독화의 목표가 민감한 계산을 보호하는 것이라면, 현재 공개된 연구를 고려하여 특정 작업에 적합하고 수동 및 자동화된 역 난독화 방법에 대해 강력한 난독화 체계가 사용되어야 한다. 난독화의 효과는 수동 테스트를 통해 검할 필요가 있다. 하드웨어 기반 격리 기능이 난독화 처리보다 우선시 된다.	✓

도청 방해 (Impede Eavesdropping)

#	MSTG-ID	설명	R
8.13	MSTG-RESILIENCE-13	심층 방어로서, 통신 당사자를 확실하게 강화하는 것 외에도, 애플리케이션 레벨 페이로드 암호화를 적용하여 도청을 더욱 방해할 수 있다.	✓

참고

OWASP 모바일 보안 테스트 안내서 (MSTG) 는 이 섹션에 나열된 요구사항을 확인하기 위한 자세한 지침을 제공합니다.

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

자세한 내용은 다음을 참조하십시오:

- OWASP Mobile Top 10: M8 (Code Tampering) - https://owasp.org/www-project-mobile-top-10/2016-risks/m8-code-tampering
- OWASP Mobile Top 10: M9 (Reverse Engineering) - https://owasp.org/www-project-mobile-top-10/2016-risks/m9-reverse-engineering
- OWASP 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

부록 A: 용어집

- **주소 공간 레이아웃 랜덤화 (Address Space Layout Randomization, ASLR)** – 메모리 오염 버그를 더 어렵게 만드는 기술.
- **애플리케이션 보안 (Application Security)** – 애플리케이션 수준 보안은 기본 운영체제 또는 연결된 네트워크와 같은 것에 초점을 맞추는 대신 OSI(Open Systems Interconnect Reference) 모델의 애플리케이션 계층을 구성하는 구성 요소의 분석에 초점을 맞춥니다.
- **애플리케이션 보안 검증 (Application Security Verification)** – OWASP MASVS 에 대한 애플리케이션의 기술적 평가입니다.
- **애플리케이션 보안 검증 보고서 (Application Security Verification Report)** – 특정 애플리케이션에 대해 검증자가 작성한 전체 결과 및 지원 분석을 문서화 한 보고서입니다.
- **인증 (Authentication)** – 애플리케이션 사용자가 요청한 ID 를 확인합니다.
- **자동 검증 (Automated Verification)** – 취약점 시그니처를 사용하여 문제점을 찾는 자동화 된 도구 (동적 분석 도구, 정적 분석 도구 또는 둘 다) 를 사용하는 검증입니다.
- **블랙 박스 테스트 (Black box testing)** – 내부 구조나 작동 방식에 신경 쓰지 않고 애플리케이션의 기능을 검사하는 소프트웨어 테스트 방법입니다.
- **컴포넌트 (Component)** – 다른 구성 요소와 통신하는 관련 디스크 및 네트워크 인터페이스가 있는 독립적인 코드 단위입니다.
- **크로스 사이트 스크립팅 (Cross-Site Scripting, XSS)** – 클라이언트측 스크립트를 콘텐츠에 삽입할 수 있는 웹 애플리케이션에서 일반적으로 볼 수 있는 보안 취약점입니다.
- **암호화 모듈 (Cryptographic module)** – 암호화 알고리즘을 구현하고 암호화 키를 생성하는 하드웨어, 소프트웨어 또는 펌웨어입니다.
- **CWE** – CWE 는 커뮤니티에서 개발한 공통적인 소프트웨어 보안 취약점 목록입니다. 이것은 공통 언어, 소프트웨어 보안 도구의 잣대, 취약점 식별, 완화 및 예방 활동의 기준 역할을 합니다.
- **동적 애플리케이션 보안 테스트 (Dynamic Application Security Testing, DAST)** – 실행 중인 애플리케이션의 보안 취약점을 나타내는 조건을 감지하도록 설계된 테스트 기법입니다.
- **설계 검증 (Design Verification)** – 애플리케이션 보안 아키텍처에 대한 기술 평가입니다.
- **동적 검증 (Dynamic Verification)** – 취약점 시그니처를 사용하여 애플리케이션을 실행하는 동안 문제를 발견하는 자동 도구를 사용하여 확인합니다.
- **전역 고유 식별자 (Globally Unique Identifier, GUID)** – 소프트웨어에서 식별자로 사용되는 고유한 참조 번호입니다.
- **하이퍼 텍스트 전송 프로토콜 (Hyper Text Transfer Protocol, HTTP)** – 분산, 협업 및 하이퍼미디어 정보 시스템을 위한 애플리케이션 프로토콜입니다. 이것은 월드 와이드 웹을 위한 데이터 통신의 기초입니다.
- **하드 코드 된 키 (Hardcoded keys)** – 장치 자체에 저장된 암호화 키입니다.
- **프로세스 간 통신 (Inter Process Communications, IPC)** – 프로세스가 다른 프로세스 또는 커널과 활동을 조율하기 위한 통신입니다.
- **입력 검증 (Input Validation)** – 신뢰할 수 없는 사용자 입력에 대한 표준화 및 유효성 검사입니다.
- **JAVA 바이트 코드 (JAVA Bytecode)** – Java 가상 머신 (JVM) 의 명령 세트입니다. 각각의 바이트 코드는 매개 변수를 전달하기 위한 0 개 이상의 바이트와 함께 명령어 (opcode) 를 나타내는 1 개 또는 2 개 바이트로 구성됩니다.
- **악성 코드 (Malicious Code)** – 개발 과정에서 애플리케이션의 의도된 보안 정책을 우회하기 위해 애플리케이션 소유자에게 잘 알려지지 않은 애플리케이션에 포함된 코드입니다. 바이러스나 웜과 같은 맬웨어와 동일하지 않습니다.
- **맬웨어 (Malware)** – 애플리케이션 사용자와 관리자가 알지 못하세 런타임 동안 애플리케이션에 삽입되어 실행되는

코드입니다.

- **Open Web Application Security Project (OWASP)** – 애플리케이션 소프트웨어의 보안을 향상시키는 데 초점을 맞춘 전 세계 무료 오픈 커뮤니티입니다. 우리의 사명은 사용자와 조직이 애플리케이션 보안 위험에 대한 정보에 근거한 결정을 내릴 수 있도록 애플리케이션 보안을 "가시성" 있게 만드는 것입니다. 참조: https://www.owasp.org/
- **개인 식별 정보 (Personally Identifiable Information, PII)** – 개인 식별 정보는 단독 또는 다른 정보와 함께 사용하여 정보주체를 식별, 접촉 또는 찾아내는 등 개인을 식별하는 데 사용할 수 있는 정보입니다.
- **위치 독립 실행 형식 (Position-independent executable, PIE)** – 기본 메모리의 어딘가에 배치되어 절대 주소에 관계없이 올바르게 실행되는 기계 코드의 본문입니다.
- **공개 키 기반 구조 (Public Key Infrastructure, PKI)** – PKI 는 공개 키를 엔터티의 각 ID 와 바인딩하는 구조입니다. 바인딩은 인증기관 (CA) 의 인증서 등록 및 게시 프로세스에 의해 확립됩니다.
- **정적 애플리케이션 보안 테스트 (Static Application Security Testing, SAST)** – 보안 취약점을 나타내는 코딩 및 설계 조건에 대한 애플리케이션 소스 코드, 바이트 코드 및 이진 파일을 분석하기 위해 설계된 기술 모음입니다. SAST 솔루션은 애플리케이션을 비 작동 상태에서 구석구석까지 분석합니다.
- **SDLC** – 소프트웨어 개발 라이프 사이클.
- **보안 아키텍처 (Security Architecture)** – 보안 통제가 사용되는 위치와 방법을 식별하고 기술하며, 사용자 및 애플리케이션 데이터의 위치와 민감도를 식별하고 기술하는 애플리케이션 설계의 추상화입니다.
- **보안 설정 (Security Configuration)** – 보안 통제 사용 방법에 영향을 미치는 애플리케이션의 런타임 설정입니다.
- **보안 통제 (Security Control)** – 보안 점검 (예: 액세스 통제 점검) 을 수행하거나 호출될 때 보안 효과 (예: 감사 레코드 생성) 를 수행하는 기능 또는 구성 요소입니다.
- **SQL 인젝션 (SQL Injection, SQLi)** – 데이터 기반 애플리케이션을 공격하기 위해 악의적인 SQL 문을 엔트리 포인트에 코드를 삽입하는 공격기법입니다.
- **싱글 사인온 인증 (Single Sign On Authentication, SSO Authentication)** – 사용자가 한 클라이언트에 로그인 한 후 사용자가 사용중인 플랫폼, 기술 또는 도메인에 관계없이 다른 클라이언트에 자동으로 로그인하는 것을 의미합니다. 예를 들어, 구글에 로그인하면 자동으로 유튜브, 드라이브, 메일 서비스에 로그인됩니다.
- **위협 모델링 (Threat Modeling)** – 위협 에이전트, 보안 영역, 보안 통제, 중요한 기술 및 비즈니스 자산을 식별하기 위해 보다 정교한 보안 아키텍처를 개발하기 위한 기술입니다.
- **전송 계층 보안 (Transport Layer Security)** – 인터넷을 통한 통신 보안을 제공하는 암호화 프로토콜입니다.
- **URI and URL** – URI 는 이름 또는 웹 리소스를 식별하는 데 사용되는 문자열입니다. URL 은 종종 자원에 대한 참조로 사용됩니다.
- **사용자 수용 테스트 (User acceptance testing, UAT)** – 운영 환경처럼 작동하도록 테스트 환경에서 프로덕션 전에 행해지는 모든 소프트웨어 테스트를 의미합니다.
- **검토자 (Verifier)** – OWASP MASVS 요구사항에 따라 신청서를 검토하는 사람 또는 팀을 의미합니다.
- **화이트리스트 (Whitelist)** – 허용 된 데이터 또는 작업의 목록입니다. (예: 입력 유효성 검사를 수행할 수 있는 문자 목록)
- **X.509 인증서 (X.509 Certificate)** – X.509 인증서는 널리 사용되는 국제 X.509 공개 키 인프라 (PKI) 표준으로, 공개 키 인증서에 포함 된 사용자, 컴퓨터 또는 서비스 ID 에 속하는지 확인하는 디지털 인증서입니다.

부록 B: 참고 자료

다음 OWASP 프로젝트는 이 표준의 사용자 및 채택자에게 가장 유용할 수 있습니다.

- 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://owasp.org/www-project-mobile-top-10/
- OWASP Reverse Engineering and Code Modification Prevention - https://www.owasp.org/index.php/OWASP_Reverse_Engineering_and_Code_Modification_Prevention_Project

마찬가지로 이 표준의 사용자 및 채택자에게 가장 유용할 수 있는 웹 사이트는 다음과 같습니다.

- 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

변경이력

V1.2 - 2020 년 3 월 7 일 - 국제 릴리스

다음 변경 사항은 릴리스 1.2 의 일부입니다:

- MASVS 의 중국어 간체 번역.
- MASVS 책 표지에서 제목 변경.
- MSTG 에서 Mobile Top 10 및 CWE 를 제거하고 MASVS 의 기존 참조와 병합.

V1.2-RC - 2019 년 10 월 5 일 - 시험판

다음 변경 사항은 시험판 1.2 의 일부입니다:

- 플래그십 상태로 승격됨.
- 요구사항 변경: MSTG-STORAGE-1 “사용 필요”.
- 데이터 보호에 중점을 두기 위해 요구사항 MSTG-STORAGE-13, MSTG-STORAGE-14 및 MSTG-STORAGE-15 요구사항 추가.
- 요구사항 MSTG-AUTH-11 은 상황별 정보를 보존하도록 업데이트.
- 요구 사항 MSTG-CODE-4 는 단순한 디버깅 이상을 포함하도록 업데이트.
- WebView 의 더 안전하게 사용하기 위해 MSTG-PLATFORM-10 요구사항 추가.
- 개발자에게 특히 다중 사용자 앱의 경우 개발자에게 인증 구현을 상기시키기 위해 MSTG-AUTH-12 요구사항 추가.
- 위험 평가 시에 MASVS 를 사용하는 방법에 대한 설명 추가.
- 유료 컨텐츠에 대한 설명 추가.
- L2 애플리케이션에 대한 책임 있는 공개 정책을 포함하기 위해 MSTG-ARCH-11 요구사항 추가.
- 애플리케이션 개발자에게 관련 국제 개인정보보호법을 준수해야 함을 보여주기 위해 MSTG-ARCH-12 요구사항 추가.
- 영어 버전의 모든 참조에 대해 일관된 스타일을 만듦.
- MSTG-PLATFORM-11 요구 사항이 타사 키보드를 통한 스파이 감시에 추가.
- 요구사항 MSTG-MSTG-RESILIENCE-13 이 추가되어 애플리케이션에서 도청을 막을 수 있음.
- MASVS 를 중국어 (ZHTW: 번체), 영어, 독일어, 프랑스어, 일본어, 한국어, 러시아어 및 스페인어로 업데이트했습니다.

V1.1.4 - 2019 년 7 월 4 일 - 서밋 에디션

다음 변경 사항은 릴리스 1.1.4 의 일부입니다:

- 모든 마크다운 문제 해결.
- 프랑스어 및 스페인어 번역 업데이트.
- 변경이력을 중국어 (ZHTW: 번체) 및 일본어로 번역.
- 마크다운 구문 및 URL 의 접근성에 대한 자동 검증.
- 권장 사항과 테스트 사례를 쉽게 찾을 수 있도록 요구사항에 식별 코드를 추가했으며, 향후 버전의 MSTG 에 포함.
- repo 크기를 줄이고.gitignore 에 생성 추가.
- 행동 강령 및 기여 가이드라인 추가.
- 풀 요청 (Pull-Request) 템플릿 추가.

- 깃북 웹 사이트 호스팅에 사용 중인 리포지토리와 동기화 업데이트.
- 모든 번역에 대해 XML/JSON/CSV 를 생성하도록 스크립트 업데이트.
- 서문을 중국어 (ZHTW: 번체) 로 번역.

V1.1.3 - 2019 년 1 월 9 일 - 작은 수정

- 스페인어 버전에서 요구사항 7.1 의 번역 문제 수정
- 감사의 글에 새로운 번역자 추가

V1.1.2 - 2019 년 1 월 3 일 - 후원 및 국제화

다음 변경 사항은 릴리스 1.1.2 의 일부입니다:

- 전자책 구매자에게 감사의 말 추가.
- V4 에서 누락 된 인증 링크 및 업데이트 된 인증 링크 추가.
- 영어 버전 4.7 과 4.8 가 바뀌어 있던 문제 수정.
- 국제 버전 초판 출시!
  - 스페인어 번역 수정. 번역은 현재 영어 (1.1.2) 와 일치.
  - 러시아어 번역 수정. 번역은 현재 영어 (1.1.2) 와 일치.
  - 중국어 (ZHTW: 번체), 프랑스어, 독일어, 일본어를 처음 추가!
- 번역이 용이하도록 문서 간소화.
- 자동 릴리스에 대한 지침 추가.

V1.1.0 - 2018 년 7 월 14 일

다음 변경 사항은 릴리스 1.1.0 의 일부입니다:

- 요구사항 2.6 “민감한 데이터를 포함할 수 있는 텍스트 필드에서 클립보드를 비활성화하여야 한다.” 제거.
- 요구사항 2.2 “민감한 데이터는 앱 컨테이너 또는 시스템 자격 증명 저장 시설 외부에 저장하지 않아야 한다.” 추가.
- 요구사항 2.1 “개인 식별 정보 (PII), 사용자 자격 증명 암호화 키 같은 중요한 데이터를 저장하기 위해 시스템 자격 증명 저장 기능을 적절하게 사용하여야 한다.” 로 수정.

V1.0 - 2018 년 1 월 12 일

다음 변경 사항은 릴리스 1.0 의 일부입니다:

- 8.12 와 동일하게 8.9 삭제
- 4.6 일반적인 표현으로 변경
- 사소한 수정 (오타 등)