معيار التحقق من أمان التطبيقات 4.0.3

نهائي

أكتوبر 2021

فريق الترجمة إلى اللغة العربية

Aref Shaheed

Mhd Ghassan Al-Habash

الفهرس

البداية

عن المعيار

معيار التحقق من أمان التطبيقات هو قائمة لمتطلبات أو اختبارات أمان التطبيقات التي يمكن استخدامها من قبل مهندسي معمارية التطبيقات Architecture ، والمطورين ، والمختبرين ومحترفي أمن المعلومات وبائعي الأدوات والمستهلكين لتحديد وبناء واختبار والتحقق من التطبيقات الآمنة.

حقوق الطبع والرخصة

النسخة 4.0.3 ، أكتوبر 2021

قادة المشروع

Andrew van der Stock	Daniel Cuthbert	Jim Manico
Josh C Grossman	Elar Lang	

المساهمون الرئيسيون

Abhay Bhargav	Benedikt Bauer	Osama Elnaggar
Ralph Andalis	Ron Perris	Sjoerd Langkemper
Tonimir Kisasondi		

المساهمون والمراجعون الآخرون

Aaron Guzman	Alina Vasiljeva	Andreas Kurtz	Anthony Weems	Barbara Schachner
Christian Heinrich	Christopher Loessl	Clément Notin	Dan Cornell	Daniël Geerts
David Clarke	David Johansson	David Quisenberry	Elie Saad	Erlend Oftedal
Fatih Ersinadim	Filip van Laenen	Geoff Baskwill	Glenn ten Cate	Grant Ongers
hello7s	Isaac Lewis	Jacob Salassi	James Sulinski	Jason Axley
Jason Morrow	Javier Dominguez	Jet Anderson	jeurgen	Jim Newman
Jonathan Schnittger	Joseph Kerby	Kelby Ludwig	Lars Haulin	Lewis Ardern
Liam Smit	lyz-code	Marc Aubry	Marco Schnüriger	Mark Burnett
Philippe De Ryck	Ravi Balla	Rick Mitchell	Riotaro Okada	Robin Wood
Rogan Dawes	Ryan Goltry	Sajjad Pourali	Serg Belkommen	Siim Puustusmaa
Ståle Pettersen	Stuart Gunter	Tal Argoni	Tim Hemel	Tomasz Wrobel
Vincent De Schutter	Mike Jang			

إذا كان هناك نقص في القائمة أعلاه، يرجى تسجيل تذكرة في GitHub ليتم أخذها بعين الاعتبار في التحديثات المستقبلية للإصدار 4.x.

تم بناء معيار التحقق من أمان التطبيقات بتضافر جهود المشاركين في ASVS 1.0 عام 2008 لغاية عام 2016 وصولاً للإصدار 3.0 من هذا المعيار. لقد وضع كل من Mike Boberski و Jeff Willams و Dave Wichers الجزء الأكبر من البنية ومعظم مواد التحقق في ASVS. هناك أيضاً العديد من المساهمين. شكراً لجميع الذين ساهموا سابقاً. للحصول على قائمة شاملة لجميع المساهمين في الإصدارات السابقة، يرجى الرجوع لكل إصدار سابق.

تمهيد

أهلاً بك في الإصدار 4.0 من معيار التحقق من أمان التطبيقات ASVS. إن ASVS هو تضافر جهود مجتمعية لإنشاء إطار عمل لمتطلبات وضوابط الأمان التي تركز على تحديد ضوابط الأمان الوظيفية وغير الوظيفية المطلوبة عند تصميم وتطوير واختبار تطبيقات وخدمات الويب الحديثة.

الإصدار 4.0.3 هو التصحيح الثانوي الثالث للإصدار 4.0 وفيه تم تصحيح الأخطاء الإملائية وجعل المتطلبات أكثر وضوحاً بدون إجراء تغييرات حادة كتغييرات جوهرية في المتطلبات أو إضافة المتطلبات أو حتى تقويتها. ومع ذلك، قد تكون بعض المتطلبات قد ضعفت قليلاً ورأينا أنها مناسبة فتم إبقاؤها، كما تم إزالة بعض المتطلبات الزائدة والمتكررة بشكل كامل (لم يؤثر ذلك على الترقيم فهو لم يتغير).

إن ASVS الإصدار 4.0 هو تتويج لجهود مجتمعية والتعليقات الراجعة من قطاع الصناعة خلال العقد الماضي. لقد حاولنا تقديم استخدام أسهل للمعيار مع مجموعة متنوعة لحالات الاستخدام المختلفة طوال أي دورة تطوير برمجيات آمنة.

نتوقع ألا يكون هناك إجماع بنسبة 100% على محتوى أي معيار لتطبيقات الويب، بما في ذلك ASVS. إن تحليل المخاطر هو دائماً قائم على الحكم الشخصي (ليس موضوعي) مما يجعل هناك تحدي في توحيد أو تعميم هذا المعيار وجعله يناسب الجميع. ومع ذلك، نأمل أن تكون التحديثات الأخيرة التي تم وضعها في هذا الإصدار خطوة في الاتجاه الصحيح، وأن تعزز المفاهيم التي تم تقديمها في هذا المعيار الصناعي المهم.

ما الجديد في الإصدار 4.0

التغيير الأكثر أهمية في هذا الإصدار هو اعتماد إرشادات الهوية الرقمية NIST 800-63-3، وتقديم ضوابط مصادقة حديثة ومتطورة وقائمة على الأدلة. على الرغم من أننا نتوقع أن يتم التراجع عن بعض معايير المصادقة المتقدمة، إلا أننا نرى أنه من الضروري أن يأخذها المعيار بعين الاعتبار، خاصة عندما يكون معيار أمان التطبيقات له اعتباره وقائم على الشواهد.

يجب على معايير أمان التطبيقات أن تحاول تخفيض عدد المتطلبات الفريدة، فلا تضطر المنظمات الممتثلة لاتخاذ قرار بخصوص الضوابط المنافسة أو غير المتوافقة. تتوافق OWASP – TOP 10 2017 و معيار التحقق من أمان التطبيقات الآن مع المعيار NIST 800-63 في المصادقة وإدارة الجلسة. إننا نشجع المعايير الأخرى NIST والآخرين على العمل معنا للوصول لمجموعة مقبولة من ضوابط أمان التطبيقات لتحقيق أقصى درجة أمان وتخفيض تكلفة الامتثال قدر الإمكان.

تمت إعادة ترقيم ASVS 4.0 بالكامل من البداية إلى النهاية. سمح لنا نظام الترقيم الجديد بتقليص الفجوات الناتجة عن الفصول التي اختفت منذ فترة طويلة ، كما سمح لنا بتقسيم الفصول الأطول لتقليل عدد الضوابط التي يتعين على المطور أو الفريق الامتثال لها. على سبيل المثال ، إذا كان التطبيق لا يستخدم JWT ، فلن يكون القسم الخاص بـ JWT في إدارة الجلسة قابلاً للتطبيق.

الجديد في الإصدار 4.0 هو الربط الشامل مع تعداد نقاط الضعف المشهورة Common Weakness Enumeration (CWE)، وهو أحد أكثر الميزات المرغوبة طلباً وشيوعاً والتي تلقيناها بكثرة على مدار العقد الماضي. يسمح تعيين CWE لبائعي الأدوات وأولئك الذين يستخدمون برامج إدارة الثغرات بمطابقة النتائج من الأدوات الأخرى وإصدارات ASVS السابقة إلى 4.0 وما بعده. لإفساح المجال لإدخال CWE ، كان علينا إزالة الحقل "since" ، وقمنا بإعادة ترقيمه بالكامل ، فقد أصبح أقل أهمية مما كان عليه في الإصدارات السابقة من ASVS. لا يحتوي كل عنصر في ASVS على CWE مرتبط به ، ولأن CWE به قدر كبير من التكرار، فقد حاولنا استخدام الأكثر استخدامًا وليس بالضرورة الأقرب تطابقًا. لا يمكن دائمًا تعيين ضوابط التحقق لنقاط الضعف المقابلة لها. نرحب بالمناقشة المستمرة مع مجتمع CWE ومجال أمن المعلومات بشكل عام حول سد هذه الفجوة.

لقد عملنا على تلبية المتطلبات بشكل شامل لمعالجة OWASP – العشر الأوائل 2017 (OWASP – TOP 10 2017) والضوابط الاستباقية لـ OWASP 2018 (OWASP Proactive Controls 2018). نظرًا لأن OWASP – TOP 10 2017 هو الحد الأدنى لتجنب التقصير، فقد تقصدنا وضع العشرة الأوائل كمتطلبات لضوابط المستوى الأول، مما يسهل على الممتثلين لـ OWASP – TOP 10 الارتقاء إلى معيار الأمان الفعلي.

لقد أكدنا على أن المستوى الأول من ASVS 4.0 هو مجموعة شاملة ووافية من المعيار PCI DSS 3.2.1 (الأقسام 6.5) ، لتصميم التطبيقات ، وكتابة الشيفرة المصدرية، والاختبار ، ومراجعات الشيفرة المصدرية الآمنة ، واختبارات الاختراق. استلزم ذلك تغطية لكل من buffer overflow و unsafe memory operations في الفصل 5، و unsafe memory-related compilation flags في الفصل 14، بالإضافة إلى متطلبات التحقق من التطبيقات وخدمة الويب الرائدة في الصناعة.

لقد أكملنا الانتقال بالمعيار ASVS من ضوابط monolithic server-side فقط ، إلى توفير ضوابط أمان لجميع التطبيقات الحديثة وواجهات برمجة التطبيقات APIs. في هذه الأيام ومع وجود البرمجة الوظيفية ، و Server-less API، والهاتف المحمول ، و cloud ، و containers ، و CI / CD و DevSecOps، و federation والكثير، لا يمكننا الاستمرار في تجاهل بنية التطبيقات الحديثة. تم تصميم التطبيقات الحديثة بشكل مختلف تمامًا عن تلك التي تم إنشاؤها عند إصدار ASVS الأصلي في عام 2009. يجب أن ينظر ASVS دائمًا بعيدًا في المستقبل حتى نوفر المشورة السليمة لجمهورنا الأساسي – المطورين – ولقد أوضحنا أو أسقطنا أي شرط يفترض أن التطبيقات يتم تنفيذها على أنظمة تابعة لمؤسسة واحدة.

نظرًا لحجم ASVS 4.0 ، وفضلاً عن رغبتنا في أن نصبح معيار التحقق من أمان التطبيقات الأساسي لجميع معايير التحقق من أمان التطبيقات الأخرى، فقد عزلنا الفصل الخاص بالهاتف المحمول، وقدمنا معيار منفصل للتحقق من أمان تطبيقات الهاتف المحمول Mobile Application Security Verification Standard (MASVS). سيظهر ملحق إنترنت الأشياء ضمن اهتمامات ASVS IoT المستقبلية وذلك في مشروع OWASP لانترنت الأشياء OWASP Internet of Things. لقد قمنا بتضمين معاينة مبكرة لـ IoT ASVS (معيار التحقق من أمان تطبيقات انترنت الأشياء) في الملحق ج. نشكر فريق OWASP Mobile وفريق مشروع OWASP IoT على دعمهم لـ ASVS ، ونتطلع إلى العمل معهم في المستقبل لتوفير معايير مكملة.

أخيرًا ، قمنا بإزالة الضوابط الأقل تأثيرًا. مع الوقت، بدأت ASVS في بالتحول لمجموعة شاملة من الضوابط ، ولكن ليست كل الضوابط متساوية عند صناعة برامج آمنة. هذا الجهد لإزالة الضوابط منخفضة التأثير يمكن أن يذهب أبعد من ذلك. في إصدار مستقبلي من ASVS ، سيساعد نظام تسجيل نقاط الضعف المشترك Common Weakness Scoring System (CWSS) على إعطاء الأولوية لمزيد من الضوابط التي تعتبر مهمة حقًا وتلك التي يجب إيقافها.

اعتبارًا من الإصدار 4.0 ، ستركز ASVS فقط على كونها معايير تطبيقات وخدمات الويب الرائدة، والتي تغطي بنية التطبيقات التقليدية والحديثة ، وممارسات الأمان للمنهجية الرشيقة agile وثقافة DevSecOps.

استخدام ASVS

لـ ASVS هدفين رئيسيين:

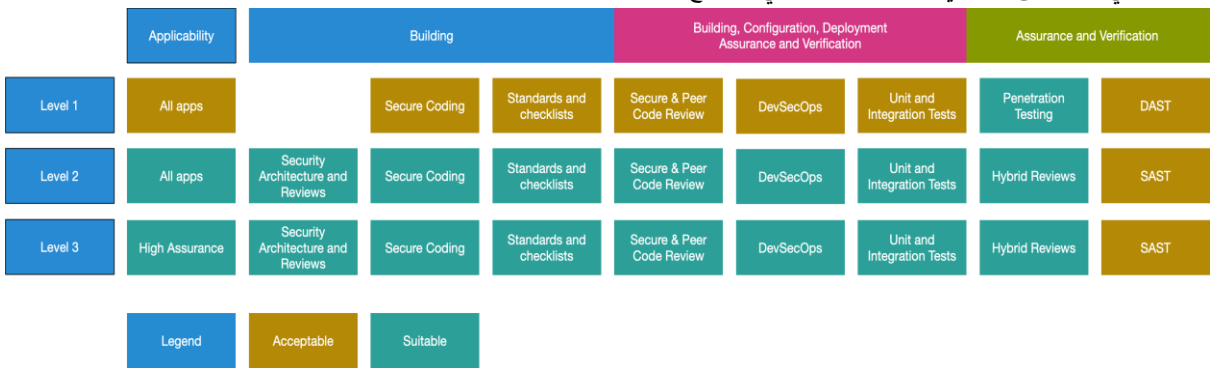
- مساعدة المنظمات على تطوير وصيانة تطبيقات آمنة.
- إتاحة التوافق بين المتطلبات لبائعي خدمات أمن المعلومات، وبائعي أدوات أمن المعلومات، والمستهلكين للتوافق مع متطلباتهم وعروضهم.

مستويات التحقق من أمان التطبيقات

يحدد معيار التحقق من أمان التطبيقات ثلاثة مستويات للتحقق من الأمان، تزداد المتطلبات مع الانتقال بين المستويات للأعلى.

- المستوى 1 من المعيار هو لمستويات أمان منخفضة، وهو قابل لاختبارات الاختراق بشكل كامل.
- المستوى 2 من المعيار هو للتطبيقات التي تتضمن بيانات حساسة والتي تتطلب حماية وهو المستوى المقترح لمعظم التطبيقات.
- المستوى 3 من المعيار وهو للتطبيقات الأكثر أهمية – التطبيقات التي تقوم بتنفيذ أعمال عالية القيمة ، أو تتضمن بيانات طبية حساسة، أو أي تطبيق يتطلب أعلى مستوى من الثقة.

يتضمن كل مستوى من مستويات ASVS قائمة من متطلبات الأمان. يمكن أيضاً تعيين كل متطلب من هذه المتطلبات إلى إمكانيات وميزات أمان خاصة والتي يجب على المطورين أن يقوموا ببنائها في البرنامج.



الشكل 1 – مستويات معيار OWASP للتحقق من أمان التطبيقات 4.0

المستوى 1 هو المستوى الوحيد القابل لاختبارات الاختراق بشكل كامل عن طريق العامل البشري. تتطلب بقية المستويات الوصول إلى الوثائق والشيفرة المصدرية والتكوين configuration والأشخاص المشاركين في عملية التطوير. ومع ذلك ، حتى لو سمحت L1 بإجراء اختبار "الصندوق الأسود" (بدون توثيق ولا شيفرة مصدرية) ، فهو إجراء غير فعال ولا يجب التشجيع على ممارسته. فالمهاجمين يمتلكون متسعاً من الوقت، أما اختبارات الاختراق تنتهي معظمها في غضون أسبوعين. يحتاج المسؤولين عن الحماية إلى بناء ضوابط أمنية، والحماية من نقاط الضعف والعثور عليها وحلها ، واكتشاف الجهات الفاعلة الضارة والرد عليها في وقت مناسب. بشكل أساسي لدى الجهات الفاعلة الضارة وقتًا غير محدود ولا تحتاج إلا لنقطة ضعف واحدة أو غياب ميزة كشف لنجاح أعمالها الضارة. إن اختبار الصندوق الأسود الذي يتم إجراؤه في نهاية التطوير بشكل سريع – أو لا يتم إجراؤه على الإطلاق – غير قادر بشكل تام على مواجهة ذلك.

على مدار الثلاثين عامًا الماضية ، أثبت اختبار الصندوق الأسود مرارًا وتكرارًا أنه لا يغطي مشكلات الأمان الحرجة التي تؤدي مباشرة إلى المزيد من الانتهاكات الهائلة. نحن نشجع بشدة على استخدام مجموعة واسعة من ضمانات الأمان والتحقق ، بما في ذلك استبدال اختبارات الاختراق باختبارات اختراق (هجينة) موجهة بالشيفرة المصدرية penetration tests with source code led في المستوى 1 ، مع إمكانية الوصول الكامل إلى المطورين والوثائق طوال عملية التطوير. لا يتسامح المنظمون الماليون مع عمليات التدقيق المالي الخارجية دون الوصول إلى الدفاتر ، أو المعاملات النموذجية ، أو الأشخاص الذين يقومون بالرقابة. يجب أن تطالب الصناعة والحكومات بنفس معيار الشفافية في مجال هندسة البرمجيات.

نحن نشجع بشدة على استخدام أدوات الأمان في عملية التطوير نفسها. يمكن استخدام أدوات DAST و SAST بشكل مستمر للعثور بسهولة على مشكلات الأمان التي يجب ألا تكون موجودة أبدًا.

إن الأدوات المؤتمتة automated tools وعمليات الفحص عبر الإنترنت online scan غير قادرة على إكمال أكثر من نصف ASVS دون تدخل العامل البشري. إذا كانت هناك حاجة إلى أتمتة اختبار شامل لكل build ، فسيتم استخدام مجموعة من اختبارات الوحدة والتكامل unit and integration tests جنبًا إلى جنب مع عمليات الفحص عبر الإنترنت. أما عيوب منطق الأعمال Business logic flaws واختبار التحكم في الوصول access control testing فهي ممكنة فقط عن طريق العامل البشري. ولذلك يجب تحويلها إلى unit and integration tests.

كيفية استخدام هذا المعيار

واحدة من أفضل الطرق لاستخدام معيار التحقق من أمان التطبيقات هي استخدامه كمخطط عمل لإنشاء قائمة تحقق checklist لكتابة شيفرة مصدرية آمنة، هذه القائمة هي خاصة بتطبيقك أو منصتك أو مؤسستك. سيؤدي استخدام ASVS مع الـ use cases الخاصة بك إلى زيادة التركيز على متطلبات الأمان الأكثر أهمية لمشاريعك وبيئاتك.

المستوى 1: الخطوات الأولى، مؤتمت، أو استعراض المحفظة بشكل كامل

يحقق التطبيق المستوى 1 من ASVS إذا كان يحمي بشكل كاف من الثغرات الأمنية للتطبيقات التي يسهل اكتشافها، وتم تضمينها في OWASP – العشر الأوائل وقوائم التحقق الأخرى المماثلة.

المستوى 1 هو الحد الأدنى الذي يجب أن تسعى جميع التطبيقات لتحقيقه. كما أنه مفيد كخطوة أولى في جهد متعدد المراحل أو عندما لا تخزن التطبيقات البيانات الحساسة أو تتعامل معها ، وبالتالي لا تحتاج إلى ضوابط أكثر صرامة كالضوابط في المستوى 2 أو 3. يمكن التحقق من ضوابط المستوى 1 إما آلياً بواسطة الأدوات المؤتمتة أو ببساطة يدويًا دون الوصول إلى الشيفرة المصدرية. نحن نعتبر المستوى 1 هو الحد الأدنى المطلوب لجميع التطبيقات.

من المرجح أن تكون التهديدات التي يتعرض لها التطبيق من المهاجمين الذين يستخدمون تقنيات بسيطة ومنخفضة الجهد لتحديد نقاط الضعف التي يسهل العثور عليها واستغلالها. هذا على عكس المهاجم الذي سيبذل جهود كبيرة ومركزة لاستهداف التطبيق على وجه التحديد. إذا كانت البيانات التي تتم معالجتها بواسطة التطبيق الخاص بك ذات قيمة عالية ، فنادراً ما ترغب في التوقف عند مراجعة المستوى 1.

المستوى2: معظم التطبيقات

يحقق التطبيق المستوى 2 (أو القياسي) من ASVS إذا كان يحمي بشكل كاف من معظم المخاطر المرتبطة بالبرنامج اليوم.

يضمن المستوى 2 أن ضوابط الأمان مطبقة وفعالة ومستخدمة داخل التطبيق. عادةً ما يكون المستوى 2 مناسبًا للتطبيقات التي تتعامل مع المعاملات transactions الهامة بين الشركات ، بما في ذلك تلك التي تعالج معلومات الرعاية الصحية ، أو تنفذ وظائف مهمة أو حساسة للأعمال ، أو تعالج الأصول الحساسة الأخرى ، أو الصناعات التي تكون فيها السلامة integrity جانبًا مهمًا لحماية أعمالها ، مثل صناعة الألعاب لمكافحة الغش في الألعاب واختراقها.

عادةً ما تكون التهديدات التي تواجه تطبيقات المستوى 2 عبارة عن مهاجمين ماهرين ومتحمسين يركزون على أهداف محددة باستخدام أدوات وتقنيات يتم استخدامها بشكل كبير وفعال في اكتشاف واستغلال نقاط الضعف داخل التطبيقات.

المستوى 3: قيمة عالية ، أو ضمان عالي ، أو سلامة عالية

المستوى 3 من ASVS هو أعلى مستوى للتحقق داخل ASVS. عادةً ما يكون هذا المستوى مخصص للتطبيقات التي تتطلب مستويات كبيرة من التحقق الأمني ، كالتطبيقات في المجالات العسكرية والصحة والسلامة والبنية التحتية الحيوية وما إلى ذلك.

قد تطلب المنظمات ASVS المستوى 3 للتطبيقات التي تؤدي وظائف مهمة ، حيث يمكن لفشل التطبيق أن يؤثر بشكل كبير على عمليات المنظمة ، وحتى على بقائها. ويرد أدناه مثال على إرشادات حول تطبيق ASVS المستوى 3. يحقق التطبيق المستوى 3 من ASVS (أو المستوى المتقدم) إذا كان يحمي بشكل كاف من الثغرات الأمنية المتقدمة للتطبيقات ويوضح أيضًا مبادئ التصميم الآمن الجيد.

يتطلب التطبيق في المستوى 3 من ASVS تحليلًا معمقًا للبنية وكتابة الشيفرة المصدرية والاختبار أكثر من جميع المستويات الأخرى. يتم نمذجة modularized التطبيق الآمن بطريقة هادفة (لتسهيل المرونة وقابلية التوسع والأهم من ذلك كله ، طبقات الأمان) ، وكل وحدة module (مفصولة عن طريق اتصال الشبكة و / أو كيان فيزيائي) تهتم بمسؤولياتها الأمنية الخاصة (الدفاع في العمق defense in depth) ، التي تحتاج إلى توثيقها بشكل صحيح. تشمل المسؤوليات ضوابط لضمان السرية confidentiality (مثل التشفير) ، والسلامة integrity (مثل المعاملات transactions ، والتحقق من صحة المدخلات input validation) ، والتوافر availability (مثل التعامل مع الحمل بأمان handling load gracefully) ، والمصادقة authentication (بما في ذلك بين الأنظمة) ، والتفويض authorization ، والتدقيق auditing (التسجيل logging).

تطبيق المعيار بشكل عملي

التهديدات المختلفة لها دوافع مختلفة. بعض الصناعات لديها معلومات فريدة وأصول تقنية ومتطلبات الامتثال لقوانين خاصة بنطاق عملها.

يتم تشجيع المؤسسات بشدة على النظر بعمق في خصائص المخاطر الفريدة الخاصة بها بناءً على طبيعة أعمالها ، وبناءً على تلك المخاطر ومتطلبات العمل لتحديد مستوى ASVS المناسب.

كيفية الرجوع إلى متطلبات ASVS

يوجد معرف لكل متطلب بالصيغة "<الفصل>.<القسم>.<المتطلب>" حيث كل حقل في هذا المعرف هو رقم ، على سبيل المثال "3.11.1".

– الحقل "<الفصل>" يشير للفصل الذي ينتمي له المتطلب، على سبيل المثال: كل متطلبات "#.#.1" تابعة لفصل المعمارية.

– الحقل "<القسم>" يشير للقسم الذي يتم فيه عرض المتطلب، على سبيل المثال: كل متطلبات "#.11.1" هي قسم معمارية منطق الأعمال في فصل المعمارية.

– الحقل "<المتطلب>" يشير لمتطلب محدد في الفصل والقسم، على سبيل المثال: "3.11.1" هو المتطلب من الإصدار 4.0.3 في هذا المعيار والذي ينص على:

"تحقق من أن جميع تدفقات منطق الأعمال عالية القيمة ، بما في ذلك المصادقة وإدارة الجلسة والتحكم في الوصول آمنة عند استخدام الـ thread ومقاومة لـ time-of-check and time-of-use race conditions".

قد تتغير المعرفات بين الإصدارات المختلفة من المعيار لذلك يفضل في التقارير والمستندات الأخرى أو الأدوات أن يتم استخدام الصيغة التالية: "ن<رقم الإصدار>-<الفصل>.<القسم>.<المتطلب>" ، حيث "<رقم الإصدار>" هو وسم لإصدار ASVS. على سبيل المثال: "ن 4.0.3-1-3-11" هي إشارة واضحة للمتطلب الثالث في معمارية منطق الأعمال في فصل المعمارية من الإصدار 4.0.3. (يمكن اختصار ذلك للصيغة "ن<رقم الإصدار>-<معرف المتطلب>"). ملاحظات: حرف النون الذي يسبق المعرف هو اختصار كلمة نسخة (المقابل في المعيار الإنجليزي هو حرف v وهو اختصار لكلمة version) وتم الترقيم من اليمين إلى اليسار فالمتطلب الخامس الموجود في القسم الرابع من الفصل الأول هو في النسخة العربية 5.4.1 وفي النسخة الإنكليزية 1.4.5 ، لم يتم تغيير الترتيب وإنما اتجاه الترقيم فقط.

ملاحظة: يجب أن تكون الحرف`v` الذي يسبق جزء الإصدار صغير lower case.

إذا لم يتم استخدام حرف النون أو حرف v فهذا يعني أن المعرف يشير للمتطلب الموجود في محتوى آخر إصدار في ASVS. كما يتضح للقارئ أن المعيار ينمو ويتغير وهذا يشكل معضلة، لذلك يجب وضع رقم إصدار ASVS في معرف المتطلب.

إن قائمة متطلبات ASVS متاحة بصيغ مختلفة لدعم الاستخدامات البرمجية كـ CSV و JSON وغيرها من الصيغ المفيدة.

التقييم والشهادة

موقف OWASP من شهادات ASVS وعلامات الثقة

إن OWASP بصفتها منظمة غير ربحية محايدة للبائع، لا تصادق حاليًا على أي بائعين أو محققين أو برامج.

جميع تأكيدات الثقة assurance assertions أو علامات الثقة trust marks أو الشهادات لم يتم فحصها رسميًا أو تسجيلها أو اعتمادها من قبل OWASP، لذلك يجب على المنظمة التي تعتمد على هذه الرؤية أن تكون حذرة بشأن الثقة الموضوعة في أي طرف ثالث أو علامة ثقة تطالب بشهادة ASVS.

ينبغي لهذا الأمر ألا يمنع المنظمات من تقديم خدمات ضمان مشابهة، طالما أنها لا تطالب بشهادة رسمية من OWASP.

دليل إعطاء شهادة أمان للمنظمات

يمكن استخدام معيار التحقق من أمان التطبيقات ككتاب مفتوح للتحقق من التطبيق، بما في ذلك الوصول المفتوح وغير المقيد إلى الموارد الرئيسية متضمنة مهندسي معمارية التطبيقات والمطورين، ووثائق المشروع، والشيفرة المصدرية، والوصول المصادق لاختبار الأنظمة (بما في ذلك الوصول إلى حساب واحد أو أكثر في كل صلاحية role) ، خاصة بالنسبة لعمليات التحقق من المستوى 2 و 3.

تاريخياً، شملت اختبارات الاختراق ومراجعات شيفرة المصدرية الآمنة قضايا "بالاستثناء exception" – أي أن الاختبارات الفاشلة فقط تظهر في التقرير النهائي. يجب على المنظمة المعتمدة أن تدرج في أي تقرير نطاق التحقق (خاصة إذا كان أحد العناصر الأساسية خارج النطاق، مثل مصادقة الدخول الواحد SSO)، وملخص لنتائج التحقق، بما في ذلك الاختبارات الناجحة والفاشلة، مع مؤشرات واضحة حول كيفية حل الاختبارات الفاشلة.

قد لا تنطبق بعض متطلبات التحقق على التطبيق قيد الاختبار. على سبيل المثال، إذا قمت بتنفيذ stateless service layer API دون تنفيذ client implementation لعملائك، فإن العديد من المتطلبات في الفصل الثالث (إدارة الجلسة) لا تنطبق بشكل مباشر. في حالات كهذه قد تستمر المنظمة المعتمدة في المطالبة بالامتثال الكامل لـ ASVS، ولكن يجب أن تشير بوضوح في أي تقرير إلى سبب عدم قابلية تطبيق متطلبات التحقق المستبعدة هذه.

يعتبر حفظ أوراق العمل التفصيلية، ولقطات الشاشة أو الأفلام والبرامج النصية Scripts للاختبار لاستغلال مشكلة ما بشكل موثوق ومتكرر، والسجلات الإلكترونية للاختبار مثل اعتراض السجلات البرمجيات الوسيطة (intercepting Proxy logs) والملاحظات المرتبطة بها مثل cleanup list، كل هذا يعتبر ممارسات معيارية ويمكن أن تكون مفيدة حقا كدليل للنتائج للمطورين الذين يشعرون بالإرتياب. إن تشغيل الأدوات وتقديم تقرير عن حالات الفشل فقط لا يكفي، فهذا لا يوفر دليل كافيا على أن جميع المشاكل على مستوى التصديق certifying قد تم اختبارها بدقة. في حالة حدوث مناقشات، ينبغي أن تكون هناك أدلة كافية لضمان أن كل متطلب تم التحقق منه وأنه بالفعل تم اختباره.

طريقة الاختبار

تتمتع المنظمات المعتمدة بحرية اختيار طرق الاختبار المناسبة، ولكن يجب الإشارة إليها في التقرير.

اعتمادًا على التطبيق قيد الاختبار ومتطلبات التحقق، يمكن استخدام طرق اختبار مختلفة ليكون هناك ثقة مماثلة في النتائج. على سبيل المثال، قد يتم تحليل التحقق من فعالية آليات التحقق من المدخلات input verification mechanisms من خلال اختبار اختراق يدوي أو عن طريق تحليل الشيفرة المصدرية.

دور أدوات اختبار الأمان المؤتمتة

يتم التشجيع على استخدام أدوات اختبار الاختراق المؤتمتة لتوفير أكبر قدر ممكن من التغطية.

لا يمكن إتمام التحقق من ASVS بالكامل باستخدام أدوات اختبار الاختراق المؤتمتة وحدها. ولكن يمكن تنفيذ الغالبية العظمى من المتطلبات في L1 باستخدام الاختبارات المؤتمتة، إن الغالبية الإجمالية من المتطلبات ليست قابلة لاختبار الاختراق المؤتمت.

يرجى ملاحظة أن الخطوط الفاصلة بين الاختبار المؤتمت واليدوي قد أصبحت مموهة مع تقدم ونضوج صناعة أمان التطبيقات. غالبًا ما يتم ضبط الأدوات المؤتمتة يدويًا بواسطة خبراء، وغالبًا ما يستفيد المختبرون اليدويون من مجموعة متنوعة من الأدوات المؤتمتة.

دور اختبارات الاختراق

في الإصدار 4.0، قررنا جعل L1 قابلاً لاختبارات الاختراق بشكل تام دون الوصول إلى الشيفرة المصدرية أو الوثائق أو المطورين. للتوافق مع البند 10 (A10) في OWASP –العشر الأوائل 2017 – يتطلب وجود عنصرين logging، كما يتطلب مقابلات أو لقطات شاشة أو جمع أدلة أخرى ، تمامًا كما يفعلونه في OWASP – العشر الأوائل. ومع ذلك، فإن عملية الاختبار دون الوصول إلى المعلومات الضرورية ليست طريقة مثالية للتحقق الأمني، فهي تفتقد إمكانية الرجوع للشيفرة المصدرية، وتحديد التهديدات والضوابط المفقودة، وإجراء اختبار أكثر شمولاً في إطار زمني أقصر. يلزم الوصول كلما أمكن إلى المطورين والوثائق والشيفرة المصدرية والوصول إلى تطبيق اختباري ببيانات غير التي توجد في البيئة الحقيقية test application with non-production data، عند إجراء تقييم L2 أو L3. تتطلب اختبارات الاختراق هذا المستوى من الوصول عندما يتم إجراؤها على مستويات كهذه، والذي نسميه "المراجعات الهجينة hybrid reviews" أو "اختبارات الاختراق الهجينة hybrid penetration tests".

استخدامات أخرى لـ ASVS

إلى جانب استخدامه لتقييم أمان التطبيقات، فقد حددنا عددًا من الاستخدامات المحتملة الأخرى لـ ASVS.

استخدام المعيار كدليل تفصيلي لمعمارية الآمان

أحد الاستخدامات الأكثر شيوعًا لمعيار التحقق من أمان التطبيقات هو كمورد لمهندسي أمان معمارية التطبيق security architects. يفتقد Sherwood Applied Business Security Architecture (SABSA) قدرًا كبيرًا من المعلومات الضرورية لإكمال مراجعة شاملة لمعمارية أمان التطبيق. يمكن استخدام ASVS لسد هذه الفجوات من خلال السماح لمهندسي الأمن باختيار ضوابط أفضل للمشاكل الشائعة، مثل أنماط حماية البيانات data protection patterns واستراتيجيات التحقق من صحة المدخلات input validation strategies.

استخدام المعيار كبديل لقوائم التحقق الثانوية وغير المستخدمة للشيفرة المصدرية الآمنة On the shelf Secure Coding Checklists

يمكن للعديد من المنظمات الاستفادة من اعتماد ASVS، عن طريق اختيار أحد المستويات الثلاثة، أو عن طريق إنشاء تفريع forking لـ ASVS وتغيير ما هو مطلوب لكل مستوى من مستويات مخاطر التطبيق في مجال معين. نحن نشجع هذا النوع من التفريع طالما يتم الاحتفاظ بتتبع التعديلات، بحيث إذا اجتاز التطبيق المتطلب 4.1، فالنسخ المتفرعة forked copies يجب أن تجتاز نفس المتطلب.

استخدام المعيار كدليل لـ Unit and Integration Tests المؤتمتة

تم تصميم ASVS ليكون قابلاً للاختبار بدرجة عالية، باستثناء متطلبات البنية المعمارية والشيفرة الضارة malicious code. من خلال بناء اختبارات الوحدة والتكامل والتي تختبر fuzz cases وabuse cases محددة، عندها يصبح التطبيق قريب إلى التحقق الذاتي self-verifying مع كل build للتطبيق. على سبيل المثال، يمكن إجراء اختبارات إضافية لمجموعة الاختبار لـ login controller، واختبار المدخلات كبارامتر username لأسماء المستخدمين الافتراضية الشائعة، وaccount enumeration، وbrute forcing ، وحقن LDAP و SQL، و XSS. وبالمثل، يجب أن يتضمن اختبار بارامتر كلمة المرور طول كلمة المرور وكلمات المرور الشائعة وnull byte injection وإزالة البارامتر remove parameter و XSS وغيرها.

استخدام المعيار كتدريب للتطوير البرمجي الآمن

يمكن أيضًا استخدام ASVS لتحديد خصائص البرامج الآمنة. العديد من دورات "التنفيذ البرمجي الآمن" هي ببساطة دورات قرصنة أخلاقية مع مسحة خفيفة من نصائح التنفيذ البرمجي. قد لا يساعد هذا بالضرورة المطورين على كتابة تعليمات برمجية أكثر أمانًا. ولكن يمكن استخدام ASVS في دورات التطوير الآمنة مع التركيز الشديد على الضوابط الاستباقية الموجودة في ASVS، بدلاً من أهم 10 أشياء سلبية لا يجب القيام بها.

استخدام المعيار لقيادة تطوير التطبيقات الرشيقة

يمكن استخدام ASVS في عملية التطوير الرشيقة agile development process كإطار عمل لتحديد المهام المحددة التي يجب أن ينفذها الفريق للحصول على منتج آمن. قد يكون أحد الأساليب: بدءًا من المستوى 1، التحقق من التطبيق أو النظام المحدد وفقًا لمتطلبات ASVS للمستوى المحدد، والعثور الضوابط المفقودة ورفع tickets/tasks محددة في الـ backlog. يساعد هذا في تحديد أولويات مهام محددة (أو grooming)، ويجعل الأمان مرئيًا في منهجية agile. يمكن استخدام هذا أيضًا لتحديد أولويات مهام التدقيق والمراجعة في المؤسسة، حيث يمكن أن يكون متطلب محدد في ASVS دافعًا للمراجعة أو إعادة البناء أو التدقيق لعضو فريق معين ويكون مرئيًا على أنه "مُستحَق debt " في الـ backlog ويجب القيام به في النهاية.

استخدام المعيار كإطار لتوجيه شراء البرامج الآمنة

ASVS هو إطار عمل رائع للمساعدة في شراء البرامج الآمنة أو شراء خدمات التطوير المخصصة. يمكن للمشتري ببساطة أن يضع مطلبًا بأن البرنامج الذي يرغب في شرائه يجب أن يتم تطويره على مستوى ASVS X، ويطلب من البائع إثبات أن البرنامج يفي بالمستوى X من ASVS. يعمل هذا بشكل جيد عند دمجه مع ملحق عقد OWASP للبرنامج الآمن OWASP Secure Software Contract Annex.

ت1: المعمارية، والتصميم ونمذجة التهديدات

الهدف من ضوابط الأمان

أصبح أمان المعمارية فنًا مفقوداً في العديد من المنظمات. لقد ولت أيام مهندس معمارية التطبيق في المؤسسة enterprise architect في عصر DevSecOps. يجب على مجال أمان التطبيقات أن يواكب الجديد ويعتمد مبادئ أمان agile عند إعادة تقديم مبادئ أمان المعمارية الرائدة لممارسي هندسة البرمجيات. المعمارية ليست تنفيذ، ولكنها طريقة للتفكير في مشكلة يحتمل أن يكون لها العديد من الإجابات المختلفة، ولا توجد إجابة واحدة "صحيحة". في كثير من الأحيان، يُنظر إلى الأمان على أنه غير مرن ويتطلب من المطورين تعديل الشيفرة المصدرية بطريقة معينة عندما يجد المطورون طريقة أفضل لحل المشكلة. لا يوجد حل واحد بسيط للمعمارية، والاعتقاد بخلاف ذلك يضر بمجال هندسة البرمجيات.

ربما تتم مراجعة تنفيذ معين لتطبيق ويب بشكل مستمر طوال حياته، تتغير المعمارية نادراً ولكنها تتطور ببطيء. إن معمارية الأمان مماثلة في كل التفاصيل والأوقات على حد سواء – نحتاج إلى المصادقة اليوم ، وسنحتاج المصادقة غدًا ، وسنحتاجها بعد خمس سنوات من الآن. إذا اتخذنا قرارات سليمة اليوم، فيمكننا توفير الكثير من الجهد والوقت والمال إذا اخترنا الحلول المتوافقة مع المعمارية وأعدنا استخدامها. على سبيل المثال، قبل عشرة سنوات، كان استخدام المصادقة متعددة العوامل multi-factor authentication نادراً.

إذا استثمر المطورون في نموذج مزود هوية واحد وآمن single, secure identity provider model، كـ SAML federated identity، فيمكن تحديث مزود الهوية لدمج متطلبات جديدة مثل متطلبات الامتثال لـ NIST 800-63 ، مع عدم تغيير واجهات التطبيق الأصلي. إذا كانت العديد من التطبيقات تشترك في نفس معمارية الأمان وبالتالي نفس المكون، فإنهم جميعًا يستفيدون من هذه الترقية مرة واحدة. ومع ذلك، لن تظل SAML دائمًا أفضل حل مصادقة أو أنسبها – فقد يلزم استبدالها بحلول أخرى مع تغير المتطلبات. مثل هذه التغييرات إما معقدة، ومكلفة للغاية بحيث تتطلب إعادة كتابة كاملة، أو مستحيلة تمامًا بدون معمارية الأمان.

في هذا الفصل، يغطي ASVS الجوانب الأساسية لأي معمارية أمان سليمة: التوافر availability والسرية confidentiality وسلامة المعالجة processing integrity وعدم الإنكار non-repudiation والخصوصية privacy. يجب أن تكون مبادئ الأمان هذه بديهية وأساسية في جميع التطبيقات. إنه من المهم "التحول نحو اليسار shift left"، بدءًا من تمكين المطور من خلال قوائم التحقق من كتابة الشيفرة المصدرية الآمنة secure coding checklists، والتوجيه mentoring والتدريب، وكتابة الشيفرة المصدرية والاختبار، والبناء build ، والنشر deployment ، والتكوين configuration ، والعمليات ، والانهاء بالاختبارات المستقلة يتبع ما سبق للتأكد من أن جميع ضوابط الأمان موجودة وفعالة. الخطوة الأخيرة هي استخدام كل ما سبق في الصناعة، ولكن هذا لم يعد كافيًا عندما يضع المطورون الشيفرة المصدرية في بيئة الإنتاج الحقيقية production عشرات أو مئات المرات يومياً. يجب على محترفي أمان التطبيقات مواكبة تقنيات agile، مما يعني اعتماد أدوات المطور developer tools، وتعلم البرمجة، والعمل مع المطورين بدلاً من انتقاد المشروع بعد شهور من انتقال أي شخص آخر.

ق1.1 دورة حياة تطوير البرمجيات الآمنة

#	التوصيف	L1	L2	L3	CWE
1.1.1	تحقق من استخدام دورة حياة تطوير البرمجيات الآمنة التي تتناول الأمان في جميع مراحل التطوير. (C1)		✓	✓	
2.1.1	تحقق من استخدام نمذجة التهديد عند كل تغيير في التصميم أو في الـ sprint لتحديد التهديدات، والتخطيط للإجراءات المضادة، وتسهيل الاستجابات المناسبة للمخاطر، وتوجيه اختبارات الأمان.		✓	✓	1053
3.1.1	تحقق من أن جميع قصص المستخدم user stories والميزات تحتوي على قيود أمان وظيفية، مثل "بصفتي مستخدمًا، يجب أن أتمكن من عرض ملف التعريف الخاص بي وتعديله. لا ينبغي أن أكون قادرًا على عرض ملف التعريف الخاص بأي شخص آخر أو تعديله"		✓	✓	1110
4.1.1	تحقق من التوثيق والتبرير justification لجميع حدود ثقة التطبيق trust boundaries والمكونات وتدفقات البيانات significant data flows الهامة.		✓	✓	1059
5.1.1	تحقق من التعريف والتحليل الأمني للمعمارية عالية المستوى high-level architecture للتطبيق وجميع الخدمات المتصلة عن بعد connected remote services. (C1)		✓	✓	1059

#	التوصيف	L1	L2	L3	CWE
6.1.1	تحقق من تنفيذ ضوابط أمنية مركزية وبسيطة (الاقتصاد في التصميم economy of design) تم فحصها وتعتبر آمنة وقابلة لإعادة الاستخدام لتجنب الضوابط المكررة أو المفقودة أو غير الفعالة أو غير الآمنة. (C10)		✓	✓	637
7.1.1	تحقق من توفر قائمة تحقق لكتابة شيفرة مصدرية آمنة secure coding checklist ،أو متطلبات الأمان أو إرشادات أو سياسة لجميع المطورين والمختبرين.		✓	✓	637

ق2.1 معمارية المصادقة

عند تصميم المصادقة ، لا يهم أن يكون لديك مصادقة متعددة العوامل ممكّنة للأجهزة القوية إذا كان بإمكان المهاجم إعادة تعيين حساب عن طريق الاتصال بمركز الاتصال call center والإجابة على الأسئلة الشائعة. عند إثبات الهوية، يجب أن تتمتع جميع مسارات المصادقة authentication paths بنفس القوة.

#	التوصيف	L1	L2	L3	CWE
1.2.1	تحقق من استخدام حسابات نظام تشغيل فريدة أو خاصة بامتيازات منخفضة low-privilege لجميع مكونات التطبيق والخدمات والخوادم. (C3)		✓	✓	250
2.2.1	تحقق من أن جميع الاتصالات بين مكونات التطبيق ، بما في ذلك الـ APIs و middleware وطبقات البيانات data layers هي مصادق عليها authenticated. يجب أن تحتوي المكونات على أقل الامتيازات اللازمة.(C3)		✓	✓	306
3.2.1	تحقق من أن التطبيق يستخدم آلية مصادقة واحدة تم فحصها ومن المعروف أنها آمنة ، ويمكن توسيعها لتشمل مصادقة قوية ، ولديه ما يكفي من التسجيل logging والمراقبة لاكتشاف إساءة استخدام الحساب أو الخروقات account abuse or breaches .		✓	✓	306
4.2.1	تحقق من أن جميع مسارات المصادقة وواجهات برمجة تطبيقات إدارة الهوية identity management APIs تستخدم ضوابط أمان قوية وملائمة للمصادقة ، بحيث لا توجد بدائل أضعف من ناحية مخاطر التطبيق.		✓	✓	306

ق3.1 معمارية إدارة الجلسة

هذه الفقرة فارغة للمتطلبات المعمارية المستقبلية المتعلقة بإدارة الجلسة.

ق4.1 معمارية التحكم في الوصول

#	التوصيف	L1	L2	L3	CWE
1.4.1	تحقق من أن نقاط التنفيذ الموثوقة trusted enforcement points كالتي عند بوابات التحكم في الوصول والخوادم والـ serverless functions تفرض ضوابط الوصول. لا تفرض أبدًا ضوابط الوصول على العميل.		✓	✓	602
2.4.1	[تم حذفها ، غير قابلة للتنفيذ]				
3.4.1	[تم حذفها ، مكررة عن 3.1.4]				
4.4.1	تحقق من أن التطبيق يستخدم آلية واحدة للتحكم في الوصول تم فحصها جيدًا للوصول إلى البيانات والموارد المحمية. يجب أن تمر جميع الطلبات من خلال هذه الآلية الفردية لتجنب النسخ واللصق أو المسارات البديلة غير الآمنة insecure alternative paths. (C7)		✓	✓	284
5.4.1	تحقق من استخدام التحكم في الوصول المستند على الخاصية أو الميزة attribute or feature-based access control حيث تتحقق الشيفرة المصدرية من وجود تفويض للمستخدم للوصول للميزة أو لعنصر البيانات بدلاً من التحقق من دوره فقط. يجب أن يتم حصر تخصيص الأذونات Permissions باستخدام الأدوار. (C7)		✓	✓	275

ق5.1 معمارية المدخلات والمخرجات

في الإصدار 4.0 ، ابتعدنا عن مصطلح "من جهة الخادم server-side"" كمصطلح حدود الثقة المحملة loaded trust boundary. لا تزال حدود الثقة مثيرة للقلق – اتخاذ القرارات بشأن المتصفحات غير الموثوق بها أو أجهزة العميل أمر يمكن تجاوزه – ومع ذلك ، في عمليات نشر المعمارية السائدة mainstream architectural deployments اليوم ، تغيرت نقطة إنفاذ الثقة trust enforcement point بشكل كبير . لذلك ، عند استخدام مصطلح "طبقة الخدمة الموثوقة trusted service layer " في ASVS ، فإننا نعني أي نقطة تنفيذ موثوقة trusted enforcement point ، بغض النظر عن الموقع ، مثل خدمة مصغرة microservice والـ serverless API وserver-side والـ trusted API على جهاز عميل لديه إقلاع آمن secure boot أو partner or external APIs، وما إلى ذلك.

يشير مصطلح "العميل غير الموثوق به untrusted client" هنا إلى تقنيات من جهة العميل client-side technologies التي تعرض طبقة العرض التقديمي presentation layer، والتي يشار إليها عادةً باسم تقنيات "الواجهة الأمامية front-end' ". لا يشير مصطلح "التسلسل serialization " هنا فقط إلى إرسال البيانات عبر السلك over the wire مثل مجموعة من القيم أو أخذ بنية JSON وقراءتها ، بل يشير أيضًا إلى تمرير الكائنات المعقدة التي يمكن أن تحتوي على منطق.

#	التوصيف	L1	L2	L3	CWE
1.5.1	تحقق من أن متطلبات المدخلات والمخرجات تحدد بوضوح كيفية التعامل مع البيانات ومعالجتها بناءً على النوع والمحتوى والقوانين واللوائح المعمول بها والامتثال للسياسة الأخرى.		✓	✓	1029
2.5.1	تحقق من عدم استخدام التسلسل عند الاتصال بعملاء غير موثوق بهم. إذا لم يكن ذلك ممكنًا ، فتأكد من فرض ضوابط الحماية الكافية (كالتشفير إذا تم إرسال بيانات حساسة) لمنع هجمات إلغاء التسلسل deserialization attacks بما في ذلك حقن الكائنات object injection.		✓	✓	502
3.5.1	تحقق من أن التحقق من صحة المدخلات input validation يتم فرضه على طبقة خدمة موثوقة trusted service layer. (C5)		✓	✓	602
4.5.1	تحقق من أن ترميز المخرجات output encoding يحدث بالقرب من أو من قبل مفسر الأوامر interpreter الموجه إليه. (C4)		✓	✓	116

ق6.1 معمارية التشفير

يجب تصميم التطبيقات باستخدام معمارية تشفير قوية لحماية أصول البيانات وفقًا لتصنيفها. يعد تشفير كل شيء إهدارًا ، ولا يعد عدم تشفير أي شيء إهمالًا قانونيًا. يجب تحقيق توازن وهذا يكون عادةً أثناء التصميم المعماري أو عالي المستوى high level design ، أو في design sprints أو المسامير المعمارية architectural spikes. إن تصميم أو تعديل التشفير للتنفيذ الآمن أثناء المضي قدماً سيكلف حتماً أكثر بكثير مقارنة ببنائه من البداية.

تعتبر المتطلبات المعمارية جوهرية لأساس الشيفرة المصدرية بأكملها ، وبالتالي تزيد من صعوبة اختبارات الوحدة والتكامل unit or integrate test. تحتاج المتطلبات المعمارية إلى النظر في معايير كتابة الشيفرة المصدرية coding standards ، طوال مرحلة التنفيذ البرمجي، ويجب مراجعتها ضمن معمارية الأمان، أو مراجعات الأقران أو مراجعات الشيفرة المصدرية ، أو عمليات الاسترجاع retrospectives.

#	التوصيف	L1	L2	L3	CWE
1.6.1	تحقق من وجود سياسة واضحة لإدارة مفاتيح التشفير وأن دورة حياة مفتاح التشفير تتبع معيار إدارة المفاتيح مثل NIST SP 800-57.		✓	✓	320
2.6.1	تحقق من أن مستخدمي خدمات التشفير يحمون المواد الأساسية Key material والأسرار الأخرى other secrets باستخدام خزائن المفاتيح key vaults أو البدائل القائمة على APIs.		✓	✓	320
3.6.1	تحقق من أن جميع المفاتيح وكلمات المرور قابلة للاستبدال وأنها جزء من عملية معرفة جيدًا لإعادة تشفير البيانات الحساسة.		✓	✓	320
4.6.1	تحقق من أن المعمارية تتعامل مع الأسرار من جهة العميل client-side secrets – مثل المفاتيح المتناظرة symmetric keys أو كلمات المرور أو الرموز المميزة لواجهة برمجة التطبيقات API tokens – باعتبارها غير آمنة ولا تستخدمها أبدًا لحماية البيانات الحساسة أو الوصول إليها.		✓	✓	320

ق7.1 معمارية الأخطاء، التسجيل والمراقبة

#	التوصيف	L1	L2	L3	CWE
1.7.1	تحقق من استخدام طريقة وتنسيق شائع للتسجيل common logging format and approach داخل النظام. (C9)		✓	✓	1009
2.7.1	تحقق من أن السجلات يتم إرسالها بشكل آمن إلى نظام التحكم عن بعد remote system للتحليل والاكتشاف والتنبيه alerting والتصعيد escalation. (C9)		✓	✓	

ق8.1 معمارية حماية البيانات والخصوصية

#	التوصيف	L1	L2	L3	CWE
1.8.1	تحقق من تحديد وتصنيف جميع البيانات الحساسة ضمن مستويات حماية protection levels.		✓	✓	
2.8.1	تحقق من أن جميع مستويات الحماية لها مجموعة مرتبطة من متطلبات الحماية ، مثل متطلبات التشفير ، ومتطلبات السلامة integrity requirements ، والاحتفاظ retention ، والخصوصية وغير ذلك من متطلبات السرية confidentiality requirements ، وأنه يتم تطبيقها في المعمارية.		✓	✓	

ق9.1 معمارية الاتصالات

#	التوصيف	L1	L2	L3	CWE
1.9.1	تحقق من أن التطبيق يقوم بتشفير الاتصالات بين المكونات، خاصة عندما تتواجد المكونات في حاويات containers أو أنظمة أو مواقع أو مزودي خدمات سحابية cloud providers مختلفة. (C3)		✓	✓	319
2.9.1	تحقق من أن مكونات التطبيق تتحقق من صحة authenticity في كل جانب في حلقة الاتصال communication link لمنع هجمات الشخص في المنتصف person-in-the-middle. على سبيل المثال، يجب أن تتحقق مكونات التطبيق من صحة شهادات TLS وسلاسلها TLS certificates and chains.		✓	✓	295

ق10.1 معمارية البرمجيات الخبيثة

#	التوصيف	L1	L2	L3	CWE
1.10.1	تحقق من أن نظام التحكم في الشيفرة المصدرية source code control system قيد الاستخدام ، مع إجراءات للتأكد من أن عمليات تسجيل الوصول يرافقها قضايا أو تذاكر تغيير issues or change tickets. يجب أن يمتلك نظام التحكم في الشيفرة المصدرية تحكماً في الوصول ومستخدمين يمكن التعرف عليهم للسماح بتتبع أي تغييرات.		✓	✓	284

ق11.1 معمارية منطق الأعمال

#	التوصيف	L1	L2	L3	CWE
1.11.1	تحقق من تعريف وتوثيق جميع مكونات التطبيق من حيث الأعمال أو وظائف الأمان التي توفرها.		✓	✓	1059
2.11.1	تحقق من أن جميع تدفقات منطق الأعمال عالية القيمة high-value business logic flows ، بما في ذلك المصادقة وإدارة الجلسة والتحكم في الوصول ، لا تشترك في الحالة غير المتزامنة unsynchronized state.		✓	✓	362
3.11.1	تحقق من أن جميع تدفقات منطق الأعمال عالية القيمة ، بما في ذلك المصادقة وإدارة الجلسة والتحكم في الوصول ، هي آمنة عند استخدام الـ thread ومقاومة لـ time-of-check and time-of-use race conditions.			✓	367

ق12.1 معمارية رفع الملفات بشكل آمن

#	التوصيف	L1	L2	L3	CWE
1.12.1	[تم حذفها ، مكررة عن 1.4.12]				
2.12.1	تحقق من أن الملفات التي يتم رفعها بواسطة المستخدم – إذا كان هناك حاجة لعرضها أو تحميلها من التطبيق – يتم تقديمها إما من خلال تنزيلات دفق الثماني octet stream downloads ، أو من نطاق مختلف غير مرتبط بنطاق التطبيق unrelated domain ، مثل حاوية تخزين الملفات السحابية cloud file storage bucket. قم بتنفيذ سياسة أمان محتوى مناسبة Content Security Policy (CSP) لتقليل مخاطر هجمات XSS أو الهجمات الأخرى من الملف الذي تم رفعه.		✓	✓	646

ق13.1 معمارية واجهة التطبيقات البرمجية API

هذه الفقرة فارغة للمتطلبات المعمارية المستقبلية المتعلقة بواجهة التطبيقات البرمجية API.

ق14.1 معمارية التكوين

#	التوصيف	L1	L2	L3	CWE
1.14.1	تحقق من فصل مكونات مستويات الثقة المختلفة من خلال ضوابط أمان معرفة جيدًا ، أو قواعد جدار الحماية ، أو API gateways ، أو مخدمات البروكسي العكسية reverse proxies، أو مجموعات الأمان المستندة القائمة على السحابة cloud-based security groups ، أو الآليات المماثلة.		✓	✓	923
2.14.1	تحقق من استخدام التوقيعات الثنائية binary signatures ، والاتصالات الموثوقة trusted connections، ونقاط النهاية التي تم التحقق منها verified endpoints لنشر الـ Binary على الأجهزة البعيدة.		✓	✓	494
3.14.1	تحقق من أن الـbuild pipeline يحذر من المكونات القديمة أو غير الآمنة ويتخذ الإجراءات المناسبة.		✓	✓	1104
4.14.1	تحقق من أن الـ build pipeline يحتوي على خطوة بناء build step للبناء والتحقق من النشر الآمن للتطبيق تلقائياً build and verify the secure deployment of the application ، لا سيما إذا كانت البنية الأساسية للتطبيق معرفة برمجياً software defined ، مثل البرامج النصية لإنشاء بيئة السحابة cloud environment build scripts.		✓	✓	
5.14.1	تحقق من أن عمليات نشر التطبيق تقوم بشكل كاف بـ: استخدام وضع الحماية sandbox و / أو استخدام الحاويات containerize و / أو عزل على مستوى الشبكة لتأخير وردع المهاجمين من مهاجمة التطبيقات الأخرى ، خاصةً عندما يقومون بأفعال حساسة أو خطيرة مثل إلغاء التسلسل deserialization. (C5)		✓	✓	265
6.14.1	تحقق من أن التطبيق لا يستخدم تقنيات غير مدعومة أو غير آمنة أو مهملة deprecated من جهة العميل مثل مكونات NSAPI الإضافية NSAPI plugins أو Flash أو Shockwave أو ActiveX أو Silverlight أو NACL أو تطبيقات Java الصغيرة من جهة العميل client-side Java applets.		✓	✓	477

المراجع

لمزيد من المعلومات، يمكن أيضاً الاطلاع على:

- OWASP Threat Modeling Cheat Sheet
- OWASP Attack Surface Analysis Cheat Sheet
- OWASP Threat modeling
- OWASP Software Assurance Maturity Model Project
- Microsoft SDL
- NIST SP 800-57

ت2: المصادقة

الهدف من ضوابط الأمان

المصادقة هي عملية تأكيد شخص ما (أو شيء ما) على أنه حقيقي وموثوق به وأن الادعاءات التي يقدمها شخص أو جهاز ما صحيحة ومقاومة لانتحال الهوية وتمنع استرداد كلمات المرور أو اعتراضها Interception.

عندما تم إصدار ASVS لأول مرة ، كان اسم المستخدم + كلمة المرور أكثر أشكال المصادقة شيوعًا خارج أنظمة الأمان المشددة. تم قبول المصادقة متعددة العوامل Multi-factor Authentication (MFA) بشكل عام في دوائر الأمان security circles ولكنها نادرًا ما تكون مطلوبة في أي مكان آخر. مع زيادة عدد عمليات اختراق كلمات المرور ، تعارضت فكرة أن أسماء المستخدمين سرية إلى حد ما وكلمات المرور غير معروفة مع العديد من ضوابط الأمان وجعلت منها ضوابط غير مقبولة. على سبيل المثال ، يعتبر NIST 800-63 أسماء المستخدمين والمصادقة المستندة إلى المعرفة Knowledge Based Authentication (KBA) بمثابة معلومات عامة وتعتبر كل من إشعارات الرسائل القصيرة والبريد الإلكتروني على أنها أنواع مصادقة "مقيدة" restricted" authenticator types" وتعتبر كلمات المرور على أنها مخترقة مسبقًا. وهذا في الواقع يجعل المصادقات القائمة على المعرفة ، واستعادة الرسائل القصيرة والبريد الإلكتروني ، وسجل كلمات المرور ، والتعقيد complexity ، وضوابط التدوير rotation controls عديمة الفائدة. لطالما كانت هذه الضوابط أقل فائدة ، وغالبًا ما تجبر المستخدمين على ابتكار كلمات مرور ضعيفة كل بضعة أشهر ، ولكن مع ظهور أكثر من 5 مليارات عملية اختراق لاسم المستخدم وكلمة المرور ، فقد حان الوقت للمضي قدمًا.

من بين جميع الفصول في ASVS ، تغيرت فصول المصادقة وإدارة الجلسة أكثر من غيرها. سيكون تبنّي ممارسة رائدة فعالة وقائمة على الأدلة تحدياً كبيراً بالنسبة للكثيرين ، وهذا أمر جيد تمامًا. علينا أن نبدأ الانتقال إلى مستقبل ما بعد كلمة المرور الآن.

NIST 800-63 - معيار مصادقة حديث قائم على الأدلة

NIST 800-63b هو معيار حديث قائم على الأدلة ، ويمثل أفضل اقتراح متاح ، بغض النظر عن قابلية التطبيق. المعيار مفيد لجميع المنظمات في جميع أنحاء العالم ولكنه وثيق الصلة بشكل خاص بالوكالات الأمريكية وتلك التي تتعامل معها.

يمكن أن تكون مصطلحات NIST 800-63 مربكة بعض الشيء في البداية ، خاصة إذا كنت تستخدم فقط المصادقة التقليدية (اسم المستخدم + كلمة المرور). تعتبر التطورات في المصادقة الحديثة ضرورية ، لذلك يتعين علينا تقديم مصطلحات ستصبح شائعة في المستقبل ، إننا ندرك أن هناك صعوبة الفهم لغاية استقرار الصناعة على هذه المصطلحات الجديدة. لقد قدمنا قائمة مصطلحات في نهاية هذا الفصل للمساعدة. لقد أعدنا صياغة العديد من المتطلبات لتحقيق الغاية الرئيسية منها، بدلاً من نص المتطلب. على سبيل المثال ، يستخدم ASVS مصطلح "كلمة المرور password" عندما يستخدم NIST "السر المحفوظ memorized secret" في جميع فقرات هذا المعيار.

تم ملائمة ت2 المصادقة، ت3 إدارة الجلسات وبدرجة أقل ت4 التحكم في الوصول لتكون مجموعة فرعية متوافقة من ضوابط NIST 800-63b المحددة ، والتي تركز على التهديدات الشائعة ونقاط الضعف الشائعة في المصادقة. عندما يكون الامتثال الكامل لـ NIST 800-63 مطلوبًا ، يرجى الرجوع إلى NIST 800-63.

اختيار مستوى NIST AAL المناسب

حاول معيار التحقق من أمان التطبيقات تعيين ASVS L1 لمتطلبات NIST AAL1 و L2 إلى AAL2 و L3 إلى AAL3. ومع ذلك ، فإن نهج المستوى 1 في ASVS كضوابط "أساسية" قد لا يكون بالضرورة هو مستوى AAL الصحيح للتحقق من تطبيق أو واجهة برمجة تطبيقات API. على سبيل المثال ، إذا كان التطبيق عبارة عن تطبيق من المستوى 3 أو لديه متطلبات تنظيمية ليكون AAL3 ، فيجب اختيار المستوى 3 في الفصلين ت2 و ت3. يجب إجراء اختيار مستوى تأكيد المصادقة Authentication Assertion Level (AAL) المتوافق مع NIST وفقًا لإرشادات NIST 800-63b كما هو موضح في تحديد AAL في NIST 800-63b Section 6.2.

فقرة توضيحية

في كثير من الأحيان تحتاج التطبيقات لمتطلبات تفوق متطلبات المستوى الحالي خاصة إذا كان التطبيق يستخدم المصادقة الحديثة. سابقاً، إذا كان ASVS يتطلب المصادقة متعددة العوامل كمتطلب أساسي ولكن NIST لا يتطلب ذلك. استخدمنا التصميم الاختياري optional designation في هذا الفصل للإشارة إلى الحالات التي توصي ASVS باستخدام الضابط فيها، ولكن لا تفرضه. المفاتيح التالية لتوصيات ASVS ضمن هذا المعيار هي في الجدول التالي.

العلامة	التوصيف
	غير مطلوب
o	موصى به، ولكنه غير مطلوب
✓	مطلوب

ق1.2 أمان كلمة المرور

تتضمن كلمات المرور التي تطلق NIST 800-63 عليها اسم "الأسرار المحفوظة Memorized Secrets"، كلمات المرور وأرقام التعريف الشخصية PIN وأنماط إلغاء القفل unlock patterns واختيار القط الصحيح pick the correct kitten أو عنصر صورة آخر وعبارات المرور passphrases. يتم اعتبارها بشكل عام "شيئًا تعرفه something you know" ، وغالبًا ما تستخدم كمصادقين أحاديي العامل single-factor authenticators. هناك تحديات كبيرة أمام الاستخدام المستمر للمصادقة أحادية العامل ، بما في ذلك المليارات من أسماء المستخدمين وكلمات المرور الصحيحة التي تم تسريبها على الإنترنت ، وكلمات المرور الافتراضية أو الضعيفة ، وجداول قوس قزح rainbow tables والقواميس المنظمة ordered dictionaries لكلمات المرور الأكثر شيوعًا.

يجب أن تشجع التطبيقات المستخدمين بشدة على استخدام المصادقة متعددة العوامل ، ويجب أن تسمح للمستخدمين بإعادة استخدام الرموز المميزة tokens التي يمتلكونها بالفعل ، مثل الرموز المميزة FIDO أو U2F ، أو الارتباط بمزود خدمة الاعتماد credential service provider الذي يوفر مصادقة متعددة العوامل.

يوفر مزودو خدمة الاعتماد Credential Service Providers (CSPs) هوية موحدة للمستخدمين. غالبًا ما يكون لدى المستخدمين أكثر من هوية واحدة federated identity مع العديد من CSPs ، مثل هوية المؤسسة باستخدام Azure AD أو Okta أو Ping Identity أو Google ، أو هوية المستهلك باستخدام Facebook أو Twitter أو Google أو WeChat ، وهذا على سبيل المثال لا الحصر. هذه القائمة ليست إقراراً لهذه الشركات أو الخدمات ، ولكنها مجرد تشجيع للمطورين للنظر في حقيقة أن العديد من المستخدمين لديهم العديد من الهويات الثابتة. يجب أن تفكر المنظمات في التكامل مع هويات المستخدمين الحالية ، وفقًا لملف تعريف المخاطر الخاص بقوة CSP في إثبات الهوية risk profile of the CSP's strength of identity proofing. على سبيل المثال ، من غير المحتمل أن تقبل منظمة حكومية هوية وسائل التواصل الاجتماعي كمعلومات تسجيل دخول للأنظمة الحساسة ، حيث أنه من السهل إنشاء هويات مزيفة أو التخلص منها ، في حين أن شركة ألعاب الهاتف المحمول قد تحتاج إلى الاندماج مع منصات الوسائط الاجتماعية الرئيسية لتنمية قاعدة اللاعبين النشطين.

#	التوصيف	L1	L2	L3	CWE	NIST §
1.1.2	تحقق من أن طول كلمات المرور التي يقوم المستخدم بتعيينها لا يقل عن 12 حرفًا (بعد دمج مسافات متعددة). (C6)	✓	✓	✓	521	5.1.1.2
2.1.2	تحقق من أن كلمات المرور المكونة من 64 حرفًا أو أكثر مسموح بها ويجب ألا تزيد عن 128 حرفًا. (C6)	✓	✓	✓	521	5.1.1.2
3.1.2	تحقق من عدم تنفيذ اقتطاع truncation كلمة المرور. ومع ذلك ، يمكن استبدال المسافات المتعددة المتتالية بمسافة واحدة. (C6)	✓	✓	✓	521	5.1.1.2
4.1.2	تحقق من السماح باستخدام أي حرف Unicode قابل للطباعة ، بما في ذلك الأحرف المحايدة للغة مثل المسافات والرموز التعبيرية في كلمات المرور.	✓	✓	✓	521	5.1.1.2
5.1.2	تحقق من قدرة المستخدمين على تغيير كلمة المرور الخاصة بهم.	✓	✓	✓	620	5.1.1.2
6.1.2	تحقق من أن تابع تغيير كلمة المرور يتطلب كلمة المرور الحالية والجديدة للمستخدم.	✓	✓	✓	620	5.1.1.2

#	التوصيف	L1	L2	L3	CWE	NIST §
7.1.2	تحقق من فحص كلمات المرور المرسلة أثناء عمليات تسجيل الحساب وتسجيل الدخول وتغيير كلمة المرور مقابل مجموعة من كلمات المرور المخترقة إما محليًا (مثل أكثر 1000 كلمة مرور أو 10000 كلمة مرور شائعة تتطابق مع سياسة النظام لكلمة مرور) أو باستخدام واجهة برمجة تطبيقات خارجية. في حالة استخدام واجهة برمجة التطبيقات ، يجب استخدام zero knowledge proof أو أي آلية أخرى لضمان عدم إرسال كلمة مرور بشكل صريح أو استخدامها للتحقق من حالة كلمة المرور. إذا تم اختراقها، يجب أن يطلب التطبيق من المستخدم تعيين كلمة مرور جديدة غير مخترقة. (C6)	✓	✓	✓	521	5.1.1.2
8.1.2	تحقق من توفير مقياس قوة كلمة المرور password strength meter لمساعدة المستخدمين على تعيين كلمة مرور أقوى.	✓	✓	✓	521	5.1.1.2
9.1.2	تحقق من عدم وجود قواعد بناء كلمة مرور password composition rules تحد من نوع الأحرف المسموح بها. يجب ألا تكون هناك متطلبات للأحرف الكبيرة أو الصغيرة أو الأرقام أو الأحرف الخاصة. (C6)	✓	✓	✓	521	5.1.1.2
10.1.2	تحقق من عدم وجود متطلبات دورية لتداول بيانات الاعتماد credential rotation أو متطلبات سجل كلمة المرور password history.	✓	✓	✓	263	5.1.1.2
11.1.2	تحقق من السماح لـ وظيفة "اللصق paste" ، ومساعدات كلمة مرور المتصفح browser password helpers ، ومديري كلمات المرور الخارجية external password managers.	✓	✓	✓	521	5.1.1.2
12.1.2	تحقق من أن المستخدم يمكن أن يختار إما عرض كلمة المرور المقنعة بالكامل masked password مؤقتًا ، أو عرض آخر حرف مكتوب من كلمة المرور مؤقتًا على الأنظمة الأساسية التي لا توجد فيها هذه الوظيفة كوظيفة مضمنة built-in.	✓	✓	✓	521	5.1.1.2

ملاحظة: الهدف من السماح للمستخدم بمشاهدة كلمة المرور الخاصة به أو رؤية الحرف الأخير مؤقتًا هو تحسين إمكانية استخدام إدخال بيانات الاعتماد ، لا سيما فيما يتعلق باستخدام كلمات مرور وعبارات مرور passphrases ومديري كلمات مرور password managers أطول. سبب آخر لتضمين هذا المتطلب هو منع تقارير الاختبار التي تطلب من المنظمات بشكل غير ضروري كتابة تنفيذ برمجي جديد لسلوك حقل كلمة مرور المدمج override built-in platform password field behavior وذلك لإزالة هذه التجربة الأمنية الحديثة سهلة الاستخدام.

ق2.2 الأمان العام للمصادق authenticator

تعد سرعة أداة المصادقة ضرورية للتطبيقات المستقبلية. يجب إعادة بناء refactor محقق التطبيق application verifier للسماح بالمصادقين الإضافيين وفقًا لما يرغبه المستخدم ، فضلاً عن السماح للمصادقين المتقاعدين أو غير الآمنين بطريقة منظمة.

تعتبر NIST البريد الإلكتروني والرسائل القصيرة من أنواع المصادقة "المقيدة" "restricted" authenticator types، ومن المحتمل إزالتها من NIST 800-63 وبالتالي يتم إزالتها أيضاً من ASVS في وقت ما في المستقبل. يجب أن تخطط التطبيقات لخريطة طريق لا تتطلب استخدام البريد الإلكتروني أو الرسائل القصيرة.

#	التوصيف	L1	L2	L3	CWE	NIST §
1.2.2	تحقق من أن ضوابط مضادات الأتمتة anti-automation controls فعالة في تخفيف breached credential testing و brute force وهجمات إغلاق الحساب account lockout attacks. تتضمن هذه الضوابط حظر كلمات المرور المخترقة الأكثر شيوعًا ، أو عمليات الإغلاق الناعمة soft lockouts ، أو تحديد المعدل rate limiting ، أو اختبار CAPTCHA ، أو زيادة التأخيرات باستمرار بين المحاولات، أو قيود على عنوان IP (IP address restrictions)، أو القيود القائمة على المخاطر risk-based restrictions مثل الموقع، أو أول تسجيل دخول على الجهاز، أو المحاولات الأخيرة لإلغاء قفل الحساب ، وغير ذلك. تحقق من أنه لا يمكن إجراء أكثر من 100 محاولة فاشلة في الساعة على حساب واحد.	✓	✓	✓	307	5.2.2 / 5.1.1.2 / 5.1.4.2 / 5.1.5.2
2.2.2	تحقق من أن استخدام المصادق الضعيف weak authenticator (مثل الرسائل القصيرة والبريد الإلكتروني) يقتصر على التحقق الثانوي والموافقة على المعاملات وليس كبديل لطرق المصادقة الأكثر أمانًا. تحقق من تقديم طرق أقوى قبل الطرق الضعيفة ، أو أن المستخدمين على دراية بالمخاطر ، أو أن الإجراءات المناسبة موجودة للحد من مخاطر اختراق الحساب.	✓	✓	✓	304	5.2.10
3.2.2	تحقق من إرسال الإشعارات الآمنة secure notifications إلى المستخدمين بعد التحديثات على تفاصيل المصادقة ، مثل إعادة تعيين بيانات الاعتماد credential resets ، أو تغيير البريد الإلكتروني أو العنوان ، أو تسجيل الدخول من مواقع غير معروفة أو خطرة. يُفضل استخدام إشعارات الدفع push notifications – بدلاً من الرسائل القصيرة أو البريد الإلكتروني – ، ولكن في حالة عدم وجود إشعارات الدفع ، فإن استخدام الرسائل القصيرة أو البريد الإلكتروني مقبول طالما لم يتم الكشف عن أي معلومات حساسة في الإشعار.	✓	✓	✓	620	
4.2.2	تحقق من مقاومة انتحال الهوية impersonation resistance ضد التصيد الاحتيالي phishing ، مثل استخدام المصادقة متعددة العوامل ، أو أجهزة التشفير بقصد cryptographic devices with intent (مثل المفاتيح المتصلة مع الدفع connected keys with a push للمصادقة) ، أو الشهادات من جهة العميل عند مستويات أعلى من AAL.		✓	✓	308	5.2.5
5.2.2	تحقق من أنه في حالة فصل مزود خدمة الاعتماد (CSP) عن تطبيق التحقق من المصادقة ، فإن TLS المصادق عليه بشكل متبادل موجود بين نقطتي النهاية.			✓	319	5.2.6
6.2.2	تحقق من مقاومة إعادة التشغيل replay resistance من خلال الاستخدام الإلزامي لأجهزة كلمات المرور لمرة واحدة One-time Passwords (OTP) أو مصادقوا التشفير cryptographic authenticators أو رموز البحث lookup codes.			✓	308	5.2.8
7.2.2	تحقق من نية المصادقة من خلال طلب إدخال OTP token أو إجراء بدأه المستخدم مثل الضغط على زر على مفتاح جهاز FIDO.			✓	308	5.2.9

ق3.2 دورة حياة المصادق authenticator

المصادقة عبارة عن كلمات مرور ورموز إلكترونية soft tokens ورموز للأجهزة hardware tokens وأجهزة قياس حيوية biometric devices.
تعد دورة حياة المصادقين أمرًا بالغ الأهمية لأمان التطبيق – إذا كان بإمكان أي شخص تسجيل حساب بنفسه بدون دليل على الهوية ، فستكون الثقة أقل في تأكيد الهوية. بالنسبة لمواقع التواصل الاجتماعي مثل Reddit ، هذا جيد تمامًا. بالنسبة للأنظمة المصرفية ، يعد التركيز الأكبر على تسجيل وإصدار بيانات الاعتماد والأجهزة أمرًا بالغ الأهمية لأمن التطبيق.
ملاحظة: لا يشترط أن يكون لكلمات المرور عمر أعظمي أو أن تخضع كلمة المرور للتدوير rotation . يجب فحص كلمات المرور للتأكد إذا تم اختراقها أو لا يتم تغييرها بانتظام.

#	التوصيف	L1	L2	L3	CWE	NIST §
1.3.2	تحقق من أن كلمات المرور الأولية أو رموز التنشيط rotation التي تم إنشاؤها بواسطة النظام ويتم توليدها بشكل عشوائي وآمن، ويجب ألا يقل طولها عن 6 خانات ، وقد تحتوي على أحرف وأرقام ، وتنتهي صلاحيتها بعد فترة قصيرة من الوقت. يجب ألا يُسمح لهذه الأسرار الأولية initial secrets بأن تصبح كلمة مرور طويلة المدى.	✓	✓	✓	330	5.1.1.2 / A.3
2.3.2	تحقق من دعم التسجيل واستخدام أجهزة المصادقة التي يوفرها المستخدم ، مثل رموز U2F أو FIDO.		✓	✓	308	6.1.3
3.3.2	تحقق من إرسال تعليمات التجديد مع إتاحة الوقت الكافي لتجديد المصادقين المقيدين time bound authenticators.		✓	✓	287	6.1.4

ق4.2 تخزين بيانات الاعتماد

يجب على مهندسي المعمارية والمطورين الالتزام بهذا القسم عند بناء أو إعادة بناء الشيفرة المصدرية. لا يمكن التحقق من هذا القسم بالكامل إلا باستخدام مراجعة الشيفرة المصدرية source code review أو من خلال secure unit or integration tests. لا يمكن لاختبار الاختراق تحديد أي من هذه المشكلات.
تم تفصيل قائمة one-way key derivation functions المعتمدة في NIST 800-63 B القسم 5.1.1.2 ، وفي BSI Kryptographische Verfahren: Empfehlungen und Schlussellängen (2018). يمكن اختيار أحدث خوارزمية وطنية أو إقليمية ومعايير طول المفتاح بدلاً من هذه الاختيارات.
لا يمكن تنفيذ اختبار الاختراق للتحقق من متطلبات هذا القسم ، لذلك لم يتم تمييز الضوابط على أنها L1. ومع ذلك ، فإن هذا القسم له أهمية حيوية لأمان بيانات الاعتماد في حالة سرقتها ، لذلك عند تفريع (forking) ASVS للحصول على إرشادات معمارية أو كتابة شيفرة مصدرية أو قائمة مراجعة أو قائمة تحقق لمراجعة الشيفرة المصدرية، يرجى إعادة الضوابط هذه إلى L1 في إصدارك الخاص.

#	التوصيف	L1	L2	L3	CWE	NIST §
1.4.2	تحقق من تخزين كلمات المرور بشكل يقاوم الهجمات التي لا تحتاج لاتصال بالإنترنت. يجب استخدام salt مع كلمات المرور وتجزئتها hashed باستخدام one-way key derivation function أو وظيفة تجزئة كلمة المرور password hashing function. تأخذ وظائف اشتقاق المفاتيح وتجزئة كلمة المرور كل من كلمة المرور وقيمة salt وعامل تكلفة cost factor كمدخلات عند إنشاء hash لكلمة المرور . (C6)		✓	✓	916	5.1.1.2
2.4.2	تحقق من أن قيمة salt يبلغ طوله 32 بتًا على الأقل وأن يتم اختياره بشكل عشوائي لتقليل تصادمات قيمة salt (salt value collisions) بين التجزئات المخزنة .يجب تخزين قيمة salt فريدة والتجزئة الناتجة لكل بيانات اعتماد. (C6)		✓	✓	916	5.1.1.2
3.4.2	تحقق من أنه في حالة استخدام PBKDF2 ، يجب أن يكون عدد التكرار iteration كبيرًا بقدر ما يسمح به أداء خادم التحقق ، وعادةً ما لا يقل عن 100،000 تكرار . (C6)		✓	✓	916	5.1.1.2
4.4.2	تحقق من أنه في حالة استخدام bcrypt، يجب أن يكون عامل العمل work factor كبيرًا بالقدر الذي يسمح به أداء خادم التحقق ، بالحد الأدنى 10. (C6)		✓	✓	916	5.1.1.2

#	التوصيف	L1	L2	L3	CWE	NIST §
5.4.2	تحقق من إجراء تكرار إضافي لـ key derivation function ، باستخدام قيمة salt تكون سرية ومعروفة فقط للمدقق verifier. قم بتوليد قيمة salt باستخدام مولد بت عشوائي معتمد random bit generator [SP 800-90Ar1] وقدّم على الأقل الحد الأدنى من قوة الأمان المحددة minimum security strength في أحدث مراجعة من SP 800-131A. يجب تخزين قيمة salt السرية (secret salt value) بشكل منفصل عن كلمات المرور المجزأة hashed passwords (على سبيل المثال ، في جهاز متخصص مثل وحدة أمان الأجهزة hardware security module).		✓	✓	916	5.1.1.2

عند ذكر المعايير الأمريكية ، يمكن استخدام معيار إقليمي أو محلي بدلاً من أو بالإضافة إلى معيار الولايات المتحدة كما هو مطلوب.

ق5.2 استعادة بيانات الاعتماد

#	التوصيف	L1	L2	L3	CWE	NIST §
1.5.2	تحقق من عدم إرسال التنشيط الأولي initial activation للنظام أو سر الاسترداد initial activation بنص واضح للمستخدم. (C6)	✓	✓	✓	640	5.1.1.2
2.5.2	تحقق من عدم وجود تلميحات كلمة المرور أو المصادقة المستندة إلى المعرفة (ما يسمى "الأسئلة السرية").	✓	✓	✓	640	5.1.1.2
3.5.2	التحقق من أن استعادة بيانات اعتماد كلمة المرور لا يكشف كلمة المرور الحالية بأي شكل من الأشكال. (C6)	✓	✓	✓	640	5.1.1.2
4.5.2	تحقق من عدم وجود الحسابات المشتركة أو الافتراضية (مثل "root" أو "admin" أو "sa").	✓	✓	✓	16	5.1.1.2 / A.3
5.5.2	تحقق من أنه في حالة تغيير عامل المصادقة authentication factor أو استبداله ، يتم إعلام المستخدم بهذا الحدث.	✓	✓	✓	304	6.1.2.3
6.5.2	تحقق من استخدام مسارات استرداد آمنة في حال نسيان كلمة المرور أو استخدام طريقة استعادة أخرى، مثل OTP المستندة إلى الوقت time-based OTP (TOTP) أو soft token آخر، أو دفع الهاتف المحمول mobile push ، أو آلية استرداد أخرى في وضع عدم الاتصال. (C6)	✓	✓	✓	640	5.1.1.2
7.5.2	تحقق من أنه في حالة فقد OTP أو عوامل المصادقة متعددة العوامل ، يتم إجراء إثبات الهوية على نفس المستوى كما هو الحال أثناء التسجيل.		✓	✓	308	6.1.2.3

ق6.2 البحث عن المحقق السري Look-up Secret Verifier

أسرار البحث Look up secrets هي قوائم مُنشأة مسبقًا من الرموز السرية ، على غرار أرقام تفويض المعاملات Transaction Authorization Numbers (TAN) ، أو رموز الاسترداد في تطبيقات التواصل الاجتماعي، أو شبكة تحتوي على مجموعة من القيم العشوائية. يتم توزيعها بشكل آمن على المستخدمين. يتم استخدام رموز البحث هذه مرة واحدة ، وبمجرد استخدام كل عناصر القائمة ، يتم تجاهلها بالكامل. يعتبر هذا النوع من المصادقة "شيئًا لديك something you have".

#	التوصيف	L1	L2	L3	CWE	NIST §
1.6.2	تحقق من إمكانية استخدام أسرار البحث lookup secrets مرة واحدة فقط.		✓	✓	308	5.1.2.2
2.6.2	تحقق من أن أسرار البحث lookup secrets تمتلك عشوائية كافية (112 بت من الانتروبيا entropy) ، أو إذا كان أقل من 112 بت من الانتروبيا ، مع قيمة salt فريدة وعشوائية بطول 32 بت ومجزئة hashed باستخدام تجزئة أحادية الاتجاه معتمدة.		✓	✓	330	5.1.2.2
3.6.2	تحقق من أن أسرار البحث مقاومة للهجمات التي لا تحتاج لانترنت offline attacks ، مثل القيم المتوقعة predictable values.		✓	✓	310	5.1.2.2

ق7.2 المدقق خارج النطاق

في الماضي ، كان المدقق خارج النطاق out of band verifier الأكثر شيوعاً هو عبارة عن بريد إلكتروني أو رسالة نصية قصيرة تحتوي على رابط إعادة تعيين كلمة المرور. يستخدم المهاجمون هذه الآلية الضعيفة لإعادة تعيين الحسابات التي لا يوجد لديهم تحكم بها، مثل الاستيلاء على حساب البريد الإلكتروني لشخص ما وإعادة استخدام أي روابط استرداد reset links تم اكتشافها. هناك طرق أفضل للتعامل مع التحقق خارج النطاق.

المصادقون الآمنون خارج النطاق Secure out of band authenticators هم أجهزة فيزيائية يمكنها التواصل مع المدقق عبر قناة ثانوية آمنة. تشمل الأمثلة دفع الإشعارات إلى الأجهزة المحمولة. يعتبر هذا النوع من المصادقة "شيئًا لديك". عندما يرغب المستخدم في المصادقة ، يرسل تطبيق التحقق رسالة إلى المصادق خارج النطاق عبر اتصال بالمصادق بشكل مباشر أو غير مباشر من خلال خدمة طرف ثالث third party service. تحتوي الرسالة على رمز مصادقة (عادةً ما يكون رقمًا عشوائيًا مكونًا من ستة أرقام أو مربع حوار موافقة مشروطة modal approval dialog). ينتظر تطبيق التحقق استلام رمز المصادقة من خلال القناة الأساسية ويقارن تجزئة القيمة المستلمة بتجزئة رمز المصادقة الأصلي. إذا كانت متطابقة ، يمكن أن يفترض المدقق خارج النطاق أن المستخدم قد قام بالمصادقة.

يفترض ASVS أن عددًا قليلاً فقط من المطورين سوف يطورون مصادقات جديدة خارج النطاق ، مثل دفع الإشعارات ، وبالتالي تنطبق عناصر تحكم ASVS التالية على أدوات التحقق ، مثل واجهة برمجة تطبيقات المصادقة authentication API وتطبيقات تسجيل الدخول الأحادي. في حالة تطوير مصادق جديد خارج النطاق ، يرجى الرجوع إلى NIST 800-63B § 5.1.3.1.

لا يتم السماح للمصادقين غير الآمنين خارج النطاق مثل البريد الإلكتروني و VOIP. مصادقة PSTN و SMS "مقيدة" حاليًا بواسطة NIST ويجب التوقف عن استخدامها واستخدام دفع الإشعارات أو ما شابه ذلك. إذا كنت بحاجة إلى استخدام مصادقة خارج النطاق عبر الهاتف أو الرسائل النصية القصيرة ، فيرجى الاطلاع على الفقرة NIST 800-63B § 5.1.3.3 .

#	التوصيف	L1	L2	L3	CWE	NIST §
1.7.2	تحقق من أن المصادقين خارج النطاق ("المقيدة" في NIST) الذين يستخدمون النص الواضح ، مثل SMS أو PSTN ، لا يتم تقديمهم افتراضيًا ، ويتم تقديم بدائل أقوى مثل دفع الإشعارات أولاً.	✓	✓	✓	287	5.1.3.2
2.7.2	تحقق من انتهاء صلاحية المدقق خارج النطاق لطلبات المصادقة أو الرموز codes أو الرموز المميزة tokens خارج النطاق بعد 10 دقائق.	✓	✓	✓	287	5.1.3.2
3.7.2	تحقق من أن طلبات المصادقة أو الرموز codes أو الرموز المميزة tokens لأداة التحقق خارج النطاق قابلة للاستخدام مرة واحدة فقط ، ولطلب المصادقة الأصلي فقط.	✓	✓	✓	287	5.1.3.2
4.7.2	تحقق من أن المصادق والمحقق خارج النطاق يتواصلان عبر قناة مستقلة آمنة.	✓	✓	✓	523	5.1.3.2
5.7.2	تحقق من أن المدقق خارج النطاق يحتفظ فقط بإصدار مجزأ من رمز المصادقة hashed version of the authentication code.		✓	✓	256	5.1.3.2
6.7.2	تحقق من أن رمز المصادقة الأولي يتم إنشاؤه بواسطة مولد رقم عشوائي آمن ، يحتوي على 20 بت على الأقل من الانتروبيا (عادةً ما يكفي ستة خانات عشوائية رقمية).		✓	✓	310	5.1.3.2

ق8.2 التحقق مرة واحدة أو متعددة العوامل

كلمات المرور أحادية الاستخدام (OTPs) هي رموز physical أو soft تعرض تحديًا عشوائيًا زائفًا لمرة واحدة متغير باستمرار pseudo-random one-time challenge. تجعل هذه الأجهزة التصيد phishing (انتحال الهوية impersonation) أمرًا صعبًا ، ولكنه ليس مستحيلًا. يعتبر هذا النوع من المصادقة "شيئًا لديك". تتشابه الرموز المميزة متعددة العوامل Multi-factor tokens مع رموز OTP أحادية العامل single-factor OTPs ، ولكنها تتطلب رمز PIN صالحًا أو إلغاء قفل المقاييس الحيوية biometric unlocking أو إدخال USB (USB insertion) أو اقتران NFC (NFC pairing) أو بعض القيم الإضافية (مثل حاسبات توقيع المعاملات transaction signing calculators) ليتم إدخالها لإنشاء كلمة المرور لمرة واحدة (OTP) النهائية final OTP.

#	التوصيف	L1	L2	L3	CWE	NIST §
1.8.2	تحقق من أن كلمات المرور لمرة واحدة المستندة إلى الوقت time-based OTPs لها عمر محدد قبل انتهاء صلاحيتها.	✓	✓	✓	613	5.1.4.2 / 5.1.5.2
2.8.2	تحقق من أن المفاتيح المتناظرة symmetric keys المستخدمة للتحقق من OTPs محمية بدرجة كبيرة ، مثل استخدام وحدة أمان للأجهزة hardware security module أو تخزين مفاتيح يعتمد على نظام التشغيل الآمن operating system based key storage.		✓	✓	320	5.1.4.2 / 5.1.5.2
3.8.2	تحقق من استخدام خوارزميات التشفير المعتمدة في إنشاء ، وتوزيع ، والتحقق من OTPs.		✓	✓	326	5.1.4.2 / 5.1.5.2
4.8.2	تحقق من أن كلمات المرور لمرة واحدة المستندة إلى الوقت time-based OTPs يمكن استخدامها مرة واحدة فقط خلال فترة الصلاحية.		✓	✓	287	5.1.4.2 / 5.1.5.2
5.8.2	تحقق من أنه في حالة إعادة استخدام رمز OTP متعدد العوامل المستند إلى الوقت time-based multi-factor OTP token خلال فترة الصلاحية ، يتم تسجيله logged ورفضه مع إرسال إشعارات آمنة إلى حامل الجهاز.		✓	✓	287	5.1.5.2
6.8.2	تحقق من إمكانية إبطال مولد OTP الفيزيائي أحادي العامل physical single-factor OTP generator في حالة السرقة أو أي خسارة أخرى. تأكد من أن الإلغاء ساري المفعول على الفور عبر الجلسات التي تم تسجيل الدخول إليها ، بغض النظر عن الموقع.		✓	✓	613	5.2.1
7.8.2	تحقق من أن المصادقين البيومتريين biometric authenticators مقيدون باستخدامهم فقط كعوامل ثانوية مع أي شيء لديك وشيء تعرفه.		o	✓	308	5.2.3

ق9.2 التحقق من برامج وأجهزة التشفير

مفاتيح أمان التشفير هي بطاقات ذكية smart cards أو مفاتيح FIDO ، حيث يتعين على المستخدم توصيل جهاز التشفير بالكمبيوتر أو إقرانه لإكمال المصادقة. يرسل المحققون تحديًا غير متكرر challenge nonce إلى أجهزة أو برامج التشفير ، ويقوم الجهاز أو البرنامج بإنشاء استجابة اعتماداً على مفتاح تشفير مخزن بشكل آمن.

متطلبات أجهزة وبرامج التشفير أحادي العامل وأجهزة وبرامج التشفير متعددة العوامل هي نفسها ، حيث يثبت التحقق من مصادق التشفير cryptographic authenticator امتلاك عامل المصادقة authentication factor.

#	التوصيف	L1	L2	L3	CWE	NIST §
1.9.2	تحقق من تخزين مفاتيح التشفير المستخدمة في التحقق بشكل آمن ومحمي ضد الكشف ، مثل استخدام وحدة النظام الأساسي الموثوق Trusted Platform Module (TPM) أو وحدة أمان الأجهزة Hardware Security Module (HSM)، أو خدمة نظام التشغيل التي يمكنها استخدام هذا التخزين الآمن.		✓	✓	320	5.1.7.2
2.9.2	تحقق من أن التحدي غير المكرر challenge nonce يبلغ طوله 64 بت على الأقل ، وأنه فريد إحصائيًا أو فريدًا خلال فترة عمر جهاز التشفير cryptographic device .		✓	✓	330	5.1.7.2
3.9.2	تحقق من استخدام خوارزميات التشفير المعتمدة في التوليد والتوزيع seeding والتحقق.		✓	✓	327	5.1.7.2

ق10.2 مصادقة الخدمة

لا يمكن تنفيذ اختبار الاختراق للتحقق من متطلبات هذا القسم، لذلك لا يتضمن أي متطلبات L1، ومع ذلك ، إذا تم استخدامه في معمارية أو تشفير أو مراجعة آمنة للشيفرة المصدرية، فيرجى افتراض أن البرنامج (تمامًا مثل Java Key Store) هو الحد الأدنى من المتطلبات في L1. تخزين الأسرار secrets بشكلها الصريح غير مقبول تحت أي ظرف من الظروف.

#	التوصيف	L1	L2	L3	CWE	NIST §
1.10.2	تحقق من أن الأسرار secrets داخل الخدمة لا تعتمد على بيانات اعتماد ثابتة مثل كلمات المرور أو مفاتيح واجهة برمجة التطبيقات أو الحسابات المشتركة ذات الوصول ذو الامتيازات.		OS assisted	HSM	287	5.1.1.1
2.10.2	تحقق من أنه إذا كانت كلمات المرور مطلوبة لمصادقة الخدمة ، فإن حساب الخدمة الذي تم استخدامه لا يمتلك بيانات اعتماد افتراضية. (على سبيل المثال ، يكون root / root أو admin / admin افتراضيًا في بعض الخدمات أثناء التثبيت).		OS assisted	HSM	255	5.1.1.1
3.10.2	تحقق من تخزين كلمات المرور بحماية كافية لمنع هجمات الاسترداد في وضع عدم الاتصال بالانترنت offline recovery attacks، بما في ذلك الوصول إلى النظام المحلي.		OS assisted	HSM	522	5.1.1.1
4.10.2	تحقق من إدارة كلمات المرور والتكامل مع قواعد البيانات وأنظمة الجهات الخارجية third-party systems والتوزيع seeds والأسرار الداخلية ومفاتيح واجهة برمجة التطبيقات API keys بشكل آمن وتحقق من عدم تضمينها في الشيفرة المصدرية أو تخزينها في مستودعات الشيفرة المصدرية repositories. يجب أن يقاوم هذا التخزين الهجمات في وضع عدم الاتصال بالانترنت. يوصى باستخدام مخزن مفاتيح آمن للبرامج (L1) أو جهاز TPM أو HSM (L3) لتخزين كلمات المرور.		OS assisted	HSM	798	

المتطلبات الإضافية للوكالة الأمريكية

الوكالات الأمريكية لديها متطلبات إلزامية بخصوص NIST 800-63. لطالما كان معيار التحقق من أمان التطبيقات يشكل حوالي 80٪ من ضوابط الأمان التي تنطبق على ما يقارب 100٪ من التطبيقات ، على خلاف الضوابط الأمنية 20٪ المتبقية والتي هي من الضوابط المتقدمة أو تلك التي لديها إمكانية تطبيق محدودة. على هذا النحو ، فإن ASVS هي مجموعة فرعية صارمة من NIST 800-63 ، خاصة لتصنيفات IAL1 / 2 و AAL1 / 2 ، ولكنها ليست شاملة بما فيه الكفاية ، لا سيما فيما يتعلق بتصنيفات IAL3 / AAL3.
نحث الوكالات الحكومية الأمريكية بشدة على مراجعة وتنفيذ NIST 800-63 بالكامل.

قائمة المصطلحات

المصطلح	الاسم الكامل باللغة الانكليزية	المعنى
CSP	Credential Service Provider	يُطلق على مزود خدمة الاعتماد أيضًا اسم مزود الهوية.
المصادق	Authenticator	الكود البرمجي الذي يصادق على كلمة المرور والرمز المميز token و MFA والتأكيد الموحد federated assertion وما إلى ذلك.
المدقق	Verifier	"كيان يتحقق من هوية المدعي عن طريق التحقق من حيازة المدعي والتحكم فيه على واحد أو اثنين من المصادقين باستخدام بروتوكول مصادقة. للقيام بذلك ، قد يحتاج المدقق أيضًا إلى التحقق من صحة بيانات الاعتماد التي تربط المصادق (المصادقين) بمعرف المشترك والتحقق وضعهم ".
OTP	One-time password	كلمة السر لمرة واحدة.

المصطلح	الاسم الكامل باللغة الانكليزية	المعنى
SFA	Single-factor authenticators	المصادقة أحادية العامل ، مثل شيء تعرفه something you know (الأسرار المحفوظة memorized secrets ، وكلمات المرور ، وعبارات المرور passphrases ، وأرقام التعريف الشخصية PINs) ، أو شيء ما أنت عليه something you are (القياسات الحيوية biometrics ، أو بصمات الأصابع fingerprint ، أو مسح الوجه face scans) ، أو شيء لديك something you have (رموز OTP ، جهاز تشفير cryptographic device مثل البطاقة الذكية something you have).
MFA	Multi-factor authentication	مصادقة متعددة العوامل ، والتي تتضمن عاملين منفصلين أو أكثر.

المراجع

لمزيد من المعلومات، يمكن أيضاً الاطلاع على:

- NIST 800-63 - Digital Identity Guidelines
- NIST 800-63 A - Enrollment and Identity Proofing
- NIST 800-63 B - Authentication and Lifecycle Management
- NIST 800-63 C - Federation and Assertions
- NIST 800-63 FAQ
- OWASP Testing Guide 4.0: Testing for Authentication
- OWASP Cheat Sheet - Password storage
- OWASP Cheat Sheet - Forgot password
- OWASP Cheat Sheet - Choosing and using security questions

ت3: إدارة الجلسة

الهدف من ضوابط الأمان

أحد المكونات الأساسية لأي تطبيق مستند إلى الويب web-based application أو واجهة برمجة التطبيقات ذات الحالة stateful API هو الآلية التي يتحكم بها ويحافظ على الحالة للمستخدم أو الجهاز الذي يتفاعل معها. تقوم إدارة الجلسة بتغيير بروتوكول عديم الحالة stateless protocol ليصبح ذو حالة stateful ، وهو أمر بالغ الأهمية للتمييز بين المستخدمين أو الأجهزة المختلفة.

تأكد من أن التطبيق الذي تم التحقق منه يلبي المتطلبات رفيعة المستوى لإدارة الجلسة وهي:

- الجلسات فريدة لكل مستخدم ولا يمكن تخمينها أو مشاركتها.
- يتم إبطال الجلسات invalidated عند عدم الحاجة إليها وتنتهي مهلتها خلال فترات عدم النشاط.

كما لوحظ سابقًا ، تم موائمة هذه المتطلبات لتكون مجموعة فرعية متوافقة من ضوابط NIST 800-63b المحددة ، والتي تركز على التهديدات الشائعة ونقاط ضعف المصادقة التي يتم استغلالها بشكل شائع. تم إلغاء متطلبات التحقق السابقة أو إلغاء خدعها de-duped أو موائمتها في معظم الحالات لتتماشى بشكل قوي مع الغرض من متطلبات NIST 800-63b الإلزامية.

متطلبات التحقق الأمني

ق1.3 أمان إدارة الجلسة الأساسية Fundamental Session Management Security

#	التوصيف	L1	L2	L3	CWE	NIST §
1.1.3	تحقق من أن التطبيق لا يكشف أبدًا عن الرموز المميزة للجلسة session Tokens في بارامترات الرابط URL parameters.	✓	✓	✓	598	

ق2.3 ربط الجلسة

#	التوصيف	L1	L2	L3	CWE	NIST §
1.2.3	تحقق من أن التطبيق ينشئ رمزًا مميزًا جديدًا للجلسة session token عند مصادقة المستخدم. (C6)	✓	✓	✓	384	7.1
3.2.3	تحقق من أن الرموز المميزة للجلسة session token تمتلك ما لا يقل عن 64 بتًا من الانتروبيا entropy. (C6)	✓	✓	✓	331	7.1
3.2.3	تحقق من أن التطبيق يخزن فقط الرموز المميزة للجلسة session token في المتصفح باستخدام طرق آمنة مثل ملفات تعريف الارتباط المؤمنة secured cookies بشكل مناسب (انظر القسم 4.3) أو تخزين جلسة HTML 5 (HTML 5 session storage).	✓	✓	✓	539	7.1
3.2.4	تحقق من توليد رمز الجلسة session token باستخدام خوارزميات التشفير المعتمدة.(C6)		✓	✓	331	7.1

إن TLS أو قناة نقل آمنة أخرى secure transport channel إلزامية لإدارة الجلسة. هذا مغطى في فصل أمان الاتصالات.

ق3.3 إنهاء الجلسة

تمت موائمة مهلات الجلسة Session timeouts مع NIST 800-63 ، مما يسمح بمهلة أطول للجلسة أكثر مما تسمح به معايير الأمان تقليديًا. يجب على المؤسسات مراجعة الجدول أدناه ، وإذا كان من المرغوب فيه الحصول على مهلة أطول استنادًا إلى مخاطر التطبيق ، فيجب أن تكون قيمة NIST هي الحدود العليا لمهلة الخمول للجلسة session idle timeouts.

L1 في هذا السياق هي AAL1 / IAL1، L2 هي AAL2 / IAL2، L3 هي AAL3 / IAL3. بالنسبة إلى AAL2 / IAL2 و AAL3 / IAL3 ، تكون مهلة الخمول idle الأقصر هي الحد الأدنى لأوقات الخمول لتسجيل الخروج أو إعادة المصادقة لاستئناف resume الجلسة.

#	التوصيف	L1	L2	L3	CWE	NIST §
1.3.3	تحقق من أن تسجيل الخروج وانتهاء الصلاحية يبطلان الرمز المميز للجلسة ، بحيث لا يستأنف زر الرجوع أو الطرف المعتمد المتلقين للمعلومات جلسة مصادق عليها ، بما في ذلك عبر الأطراف المعتمدة. (C6)	✓	✓	✓	613	7.1
2.3.3	إذا سمح المصادقون للمستخدمين بالبقاء في وضع تسجيل الدخول ، فتحقق من أن إعادة المصادقة تحدث بشكل دوري عند الاستخدام النشط أو بعد فترة الخمول. (C6)	30 يوم	12 ساعة أو 30 دقيقة من عدم النشاطinactivity ، FA2 اختياري	12 ساعة أو 15 دقيقة من عدم النشاط inactivity ، مع FA2	613	7.2
3.3.3	تحقق من أن التطبيق يمنح خيار إنهاء جميع الجلسات النشطة الأخرى بعد تغيير كلمة المرور بنجاح (بما في ذلك التغيير عبر إعادة تعيين / استرداد كلمة المرور) ، وأن هذا فعال عبر التطبيق ، وتسجيل الدخول الموحد (إن وجد) ، وأي أطراف معتمدة.		✓	✓	613	
4.3.3	تحقق من أن المستخدمين قادرون على عرض و (إعادة إدخال بيانات اعتماد تسجيل الدخول) وتسجيل الخروج من أي أو جميع الجلسات والأجهزة النشطة حاليًا.		✓	✓	613	7.1

ق4.3 إدارة الجلسة المستندة إلى ملفات تعريف الارتباط

#	التوصيف	L1	L2	L3	CWE	NIST §
1.4.3	تحقق من أن الرموز المميزة للجلسة المستندة إلى ملفات تعريف الارتباط cookie-based session tokens مفعل بها السمة "secure". (C6)	✓	✓	✓	614	7.1.1
2.4.3	تحقق من أن الرموز المميزة للجلسة المستندة إلى ملفات تعريف الارتباط cookie-based session tokens مفعل بها السمة "HttpOnly". (C6)	✓	✓	✓	1004	7.1.1
3.4.3	تحقق من أن الرموز المميزة للجلسة المستندة إلى ملفات تعريف الارتباط cookie-based session tokens تستخدم سمة "SameSite" للحد من التعرض لهجمات طلبات التزوير عبر المواقع cookie-based session tokens. (C6)	✓	✓	✓	16	7.1.1
4.4.3	تحقق من أن الرموز المميزة للجلسة المستندة إلى ملفات تعريف الارتباط cookie-based session tokens تستخدم بادئة "__Host-" لذلك يتم إرسال ملفات تعريف الارتباط فقط إلى المضيف الذي قام في البداية بتعيين ملف تعريف الارتباط.	✓	✓	✓	16	7.1.1
5.4.3	تحقق من أنه إذا تم نشر التطبيق تحت اسم نطاق مع تطبيقات أخرى تقوم بتعيين أو استخدام ملفات تعريف ارتباط الجلسة التي قد تكشف عن ملفات تعريف الارتباط للجلسة ، فقم بتعيين سمة path في الرموز المميزة للجلسة المستندة إلى ملفات تعريف الارتباط باستخدام المسار الأكثر دقة ممكنًا. (C6)	✓	✓	✓	16	7.1.1

ق5.3 إدارة الجلسة المستندة إلى الرمز المميز

تتضمن إدارة الجلسة المستندة إلى الرمز المميز مفاتيح JWT و OAuth و SAML و API. من بين هذه المفاتيح ، من المعروف أن مفاتيح API ضعيفة ويجب عدم استخدامها في كود جديد.

#	التوصيف	L1	L2	L3	CWE	NIST §
1.5.3	تحقق من أن التطبيق يسمح للمستخدمين بإبطال revoke رموز OAuth المميزة التي تشكل علاقات ثقة مع التطبيقات المرتبطة.		✓	✓	290	7.1.2
2.5.3	تحقق من أن التطبيق يستخدم الرموز المميزة للجلسة بدلاً من أسرار ومفاتيح واجهة برمجة التطبيقات الثابتة static API secrets and keys ، باستثناء عمليات التنفيذ القديمة legacy implementations.		✓	✓	798	
3.5.3	تحقق من أن الرموز المميزة للجلسة عديمة الحالة stateless session tokens تستخدم التوقيعات الرقمية والتشفير وغيرها من الإجراءات المضادة للحماية من هجمات التلاعب tampering ، والتغليف enveloping ، وإعادة التشغيل reply، والتشفير الفارغ null cipher ، استبدال المفتاح key substitution.		✓	✓	345	

ق6.3 إعادة المصادقة الموحدة Federated Re-authentication

يتعلق هذا القسم بأولئك الذين يكتبون الشيفرة المصدرية للطرف المعتمد Relying Party (RP) أو مزود خدمة الاعتماد Credential Service Provider (CSP). في حالة الاعتماد على التعليمات البرمجية التي تنفذ هذه الميزات ، تأكد من التعامل مع هذه المشكلات بشكل صحيح.

#	التوصيف	L1	L2	L3	CWE	NIST §
1.6.3	تحقق من أن الأطراف المعتمدة Relying Parties (RPs) تحدد الحد الأقصى لوقت المصادقة لمقدمي خدمات الاعتماد (CSPs) وأن CSPs يعيدون مصادقة المستخدم إذا لم يستخدموا جلسة خلال تلك الفترة.			✓	613	7.2.1
2.6.3	تحقق من أن مقدمي خدمات الاعتماد (CSPs) يبلغون الأطراف المعتمدة (RPs) بحدث المصادقة الأخير ، للسماح لـ RPs بتحديد ما إذا كانوا بحاجة إلى إعادة مصادقة المستخدم.			✓	613	7.2.1

ق3.7 الحماية ضد استغلالات إدارة الجلسة

هناك عدد قليل من هجمات إدارة الجلسات ، بعضها يتعلق بتجربة المستخدم (UX) للجلسات. في السابق ، بناءً على متطلبات ISO 27002 ، تطلب ASVS حظر عدة جلسات متزامنة. لم يعد حظر الجلسات المتزامنة مناسبًا ، ليس فقط لأن المستخدمين الحديثين لديهم العديد من الأجهزة أو أن التطبيق عبارة عن واجهة برمجة تطبيقات بدون جلسة متصفح API without a browser session ، ولكن في معظم هذه التطبيقات ، يفوز المصادق الأخير ، والذي غالبًا ما يكون المهاجم. يوفر هذا القسم إرشادات رائدة حول ردع وتأخير واكتشاف هجمات إدارة الجلسة باستخدام التعليمات البرمجية.

وصف الهجوم نصف المفتوح

في أوائل عام 2018 ، تم اختراق العديد من المؤسسات المالية باستخدام ما أطلق عليه المهاجمون "هجمات نصف مفتوحة half-open attacks ". هذا المصطلح عالق في الصناعة. قام المهاجمون بضرب مؤسسات متعددة بقواعد رموز احتكارية مختلفة proprietary code bases ، ويبدو بالفعل أن قواعد كود مختلفة داخل نفس المؤسسات. يستغل الهجوم نصف المفتوح عيبًا في نمط التصميم شائعًا في العديد من أنظمة المصادقة الحالية وإدارة الجلسة والتحكم في الوصول.

يبدأ المهاجمون هجومًا نصف مفتوح بمحاولة قفل بيانات الاعتماد أو إعادة تعيينها أو استردادها. يعيد نمط تصميم إدارة الجلسة الشائع استخدام كائنات / نماذج objects/models لجلسة ملف تعريف المستخدم بين غير مصادق unauthenticated، ونصف مصادق half-authenticated (إعادة تعيين كلمة المرور ، نسيان اسم المستخدم) ، ورمز مصادق بالكامل fully authenticated code. يملأ نمط التصميم هذا كائن جلسة صالحًا أو رمزًا مميزًا يحتوي على ملف تعريف الضحية ، بما في ذلك تجزئة كلمة المرور password hashes والأدوار rules. إذا لم يتحقق التحكم في الوصول في وحدات التحكم أو أجهزة التوجيه بشكل صحيح من تسجيل المستخدم للدخول بشكل كامل ، فسيكون المهاجم قادرًا على التصرف كمستخدم. يمكن أن تشمل الهجمات تغيير كلمة مرور المستخدم إلى قيمة معروفة ، وتحديث عنوان البريد الإلكتروني لإجراء إعادة تعيين كلمة مرور صالحة ، أو تعطيل المصادقة متعددة العوامل أو تسجيل جهاز MFA جديد ، أو الكشف عن مفاتيح واجهة برمجة التطبيقات API keys أو تغييرها ، وما إلى ذلك.

#	التوصيف	L1	L2	L3	CWE	NIST §
1.7.3	تحقق من التطبيق يضمن جلسة تسجيل دخول كاملة وصالحة أو يتطلب إعادة المصادقة أو التحقق الثانوي قبل السماح بأي معاملات حساسة أو تعديلات على الحساب.	✓	✓	✓	306	

المراجع

لمزيد من المعلومات، يمكن أيضاً الاطلاع على:

- OWASP Testing Guide 4.0: Session Management Testing
- OWASP Session Management Cheat Sheet
- Set-Cookie __Host- prefix details

ت4: التحكم في الوصول

الهدف من ضوابط الأمان

التفويض Authorization هو مفهوم السماح بالوصول إلى الموارد فقط لمن يُسمح لهم باستخدامها. تأكد من أن التطبيق الذي يتم التحقق منه يفي بالمتطلبات عالية المستوى التالية:

- يمتلك الأشخاص الذين يصلون إلى الموارد بيانات اعتماد صالحة للقيام بذلك.
- يرتبط المستخدمون بمجموعة معرفة جيداً من الأدوار والامتيازات.
- البيانات الوصفية metadata للدور والإذن محمية من الإعادة replay أو العبث tampering.

متطلبات التحقق الأمني

ق1.4 التصميم العام للتحكم في الوصول

#	التوصيف	L1	L2	L3	CWE
1.1.4	تحقق من أن التطبيق يفرض قواعد التحكم في الوصول على طبقة خدمة موثوقة trusted service layer ، خاصةً إذا كان التحكم في الوصول من جانب العميل موجودًا ويمكن تجاوزه.	✓	✓	✓	602
2.1.4	تحقق من أن جميع خصائص المستخدم attributes والبيانات ومعلومات السياسة المستخدمة بواسطة عناصر التحكم في الوصول لا يمكن التلاعب بها من قبل المستخدمين النهائيين ما لم يتم التفويض بذلك على وجه التحديد.	✓	✓	✓	639
3.1.4	تحقق من وجود مبدأ الامتيازات الأقل least privilege – يجب أن يكون المستخدمون قادرين فقط على الوصول إلى الوظائف وملفات البيانات وعناوين URL ووحدات التحكم والخدمات والموارد الأخرى التي يمتلكون تفويضًا محددًا لها. هذا يعني الحماية ضد الانتحال spoofing ورفع الامتيازات elevation of privilege. (C7)	✓	✓	✓	285
4.1.4	[تم حذفها ، مكررة عن 3.1.4]				
5.1.4	تحقق من أن ضوابط الوصول تفشل بشكل آمن بما في ذلك عند حدوث استثناء exception.(C10)	✓	✓	✓	285

ق2.4 التحكم في الوصول لمستوى التشغيل

#	التوصيف	L1	L2	L3	CWE
1.2.4	تحقق من أن البيانات الحساسة وواجهات برمجة التطبيقات محمية ضد هجمات مرجع الكائن المباشر غير الآمن Insecure Direct Object Reference (IDOR) التي تستهدف إنشاء السجلات وقراءتها وتحديثها وحذفها ، مثل إنشاء أو تحديث سجل شخص آخر أو عرض سجلات الجميع أو حذف جميع السجلات.	✓	✓	✓	639
2.2.4	تحقق من أن التطبيق أو إطار العمل يفرض آلية قوية لمكافحة CSRF (anti-CSRF mechanism) لحماية الوظائف المصادق عليها ، وأن مكافحة الأتمتة anti-automation الفعالة أو مكافحة CSRF تحمي الوظائف غير المصادق عليها.	✓	✓	✓	352

ق3.4 اعتبارات أخرى للتحكم في الوصول

#	التوصيف	L1	L2	L3	CWE
1.3.4	تحقق من أن واجهات الإدارة تستخدم المصادقة متعددة العوامل المناسبة لمنع الاستخدام غير المصرح به.	✓	✓	✓	419
2.3.4	تحقق من تعطيل تصفح الدليل directory browsing ما لم يكن ذلك مطلوبًا بشكل متعمد. بالإضافة إلى ذلك ، يجب ألا تسمح التطبيقات باكتشاف البيانات الوصفية metadata للملف أو المجلد أو الكشف عنها ، مثل مجلدات Thumbs.db أو .DS_Store أو .git أو .svn.	✓	✓	✓	548
3.3.4	تحقق من أن التطبيق لديه تفويض إضافي (مثل المصادقة التكيفية أو التصعيد step up or adaptive authentication) للأنظمة الأقل أهمية lower value systems ، و / أو فصل المهام للتطبيقات عالية القيمة high value applications لفرض ضوابط مكافحة الاحتيال anti-fraud وفقًا لمخاطر التطبيق والاحتيال السابق.		✓	✓	732

المراجع

لمزيد من المعلومات، يمكن أيضاً الاطلاع على:

- OWASP Testing Guide 4.0: Authorization
- OWASP Cheat Sheet: Access Control
- OWASP CSRF Cheat Sheet
- OWASP REST Cheat Sheet

ت5: التحقق من الصحة Validation والتعقيم Sanitization والترميز Encoding

الهدف من ضوابط الأمان

إن أكثر نقاط ضعف أمان تطبيقات الويب شيوعًا هي الفشل في التحقق بشكل صحيح من المدخلات الواردة Input validation من العميل أو البيئة قبل استخدامها مباشرة دون أي ترميز للمخرجات output encoding. تؤدي هذه المشكلة تقريبًا إلى جميع نقاط الضعف المهمة في تطبيقات الويب ، مثل البرمجة النصية عبر الموقع Cross-Site Scripting (XSS) ، وحقن قواعد البيانات SQL injection ، وحقن المترجم interpreter injection ، وهجمات الإعدادات المحلية / Unicode (locale/Unicode attacks)، وهجمات نظام الملفات file system attacks ، وتدفقات المخزن المؤقت buffer overflows.

تأكد من أن التطبيق الذي يتم التحقق منه يفي بالمتطلبات عالية المستوى التالية:

- التحقق من صحة المدخلات ومعمارية ترميز المخرجات output encoding architecture لها pipeline متفق عليه لمنع هجمات الحقن.
- يتم كتابة بيانات المدخلات بشكل صارم strongly typed ، ويتم التحقق من صحتها ومن النطاق range أو الطول ، أو في أسوأ الأحوال ، يتم تعقيمها sanitized أو تصفيتها filtered.
- يتم ترميز بيانات المخرجات encoded أو تفعيل أحرف الهروب escaped وفقًا لسياق البيانات context of the data بحيث تكون أقرب ما يمكن للمترجم interpreter.

في هندسة تطبيقات الويب الحديثة ، أصبح ترميز المخرجات أكثر أهمية من أي وقت مضى. من الصعب توفير التحقق القوي من صحة المدخلات في سيناريوهات معينة ، لذا فإن استخدام واجهة برمجة تطبيقات أكثر أمانًا safer API مثل الاستعلامات ذات البارامترات parameterized queries أو إطارات عمل القوالب التلقائية auto-escaping templating frameworks أو اختيار طريقة ترميز المخرجات بعناية أمر بالغ الأهمية لأمان التطبيق.

ق1.5 التحقق من صحة المدخلات

إن استخدام قوائم السماح الإيجابية positive allow lists وكتابة البيانات بشكل صارم strong data typing في ضوابط التحقق من صحة المدخلات المطبقة بشكل صحيح يمكّن من القضاء على أكثر من 90٪ من جميع هجمات الحقن. يمكن لفحوصات الطول والنطاق Length and range checks أن تقلل هذا بدرجة أكبر. إن البناء باستخدام التحقق الآمن من صحة المدخلات هو مطلوب في معمارية التطبيق، وسرعة التصميم ، وكتابة الشيفرة المصدرية، واختبارات الوحدة والتكامل unit and integration testing. على الرغم من أن العديد من هذه العناصر لا يمكن العثور عليها في اختبارات الاختراق ، إلا أن نتائج عدم تنفيذها توجد عادةً في ق3.5 – متطلبات ترميز المخرجات ومنع الحقن. يوصى المطورين ومراجعي الشيفرة المصدرية الآمنة secure code reviewers بالتعامل مع هذا القسم كما لو كان L1 مطلوبًا لجميع العناصر لمنع عمليات الحقن.

#	التوصيف	L1	L2	L3	CWE
1.1.5	تحقق من أن التطبيق لديه دفاعات ضد هجمات تلوث مدخلات HTTP parameter pollution) HTTP attacks)، خاصةً إذا كان إطار عمل التطبيق application framework لا يميز مصدر مدخلات الطلب (GET أو POST أو ملفات تعريف الارتباط cookies أو الرؤوس headers أو متغيرات البيئة environment variables).	✓	✓	✓	235
2.1.5	تحقق من أن إطارات العمل تحمي من هجمات تخصيص المدخلات الجماعية mass parameter assignment attacks ، أو أن التطبيق لديه إجراءات مضادة للحماية من تعيين قيم المدخلات بشكل غير آمن unsafe parameter assignment ، مثل وضع علامة على الحقول على أنها خاصة أو مشابهة marking fields private. (C5)	✓	✓	✓	915
3.1.5	تأكد من التحقق من صحة جميع المدخلات (حقول نموذج HTML (HTML form fields) وطلبات REST وبارامترات URL ورؤوس HTTP وملفات تعريف الارتباط Cookies و batch files و RSS feeds وما إلى ذلك) باستخدام التحقق الإيجابي (قوائم السماح). (C5)	✓	✓	✓	20
4.1.5	تحقق من أن البيانات المنظمة مكتوبة بشكل صارم وتم التحقق من صحتها مقابل مخطط schema محدد بما في ذلك الأحرف المسموح بها والطول والنمط (مثل أرقام بطاقة الائتمان، البريد الالكتروني وأرقام الهواتف ، أو التحقق من صحة حقلين مرتبطين ، مثل التحقق من الضاحية ومطابقة الرمز البريدي). (C5)	✓	✓	✓	20
5.1.5	تحقق من أن عمليات إعادة التوجيه إلى الخلف وإعادة التوجيه إلى الأمام (URL redirects and forwards) لعناوين URL تسمح فقط بالوجهات التي تظهر في قائمة السماح ، أو تظهر تحذيرًا عند إعادة التوجيه إلى محتوى يحتمل أن يكون غير موثوق به.	✓	✓	✓	601

ق2.5 التعقيم Sanitization ووضع الحماية Sandboxing

	التوصيف	L1	L2	L3	CWE
1.2.5	تحقق من أن جميع مدخلات HTML غير الموثوق بها من محرري WYSIWYG أو ما شابهها قد تم تعقيمها بشكل صحيح باستخدام HTML sanitizer library أو ميزة في إطار العمل. (C5)	✓	✓	✓	116
2.2.5	تحقق من تعقيم البيانات غير المهيكلة unstructured لفرض تدابير السلامة مثل الأحرف والطول المسموح بهما.	✓	✓	✓	138
3.2.5	تحقق من أن التطبيق يقوم بتعقيم مدخلات المستخدم قبل المرور إلى أنظمة البريد للحماية من هجمات حقن SMTP أو IMAP.	✓	✓	✓	147
4.2.5	تحقق من أن التطبيق يتجنب استخدام eval() أو ميزات تنفيذ التعليمات البرمجية الديناميكية الأخرى dynamic code execution features. في حالة عدم وجود بديل ، يجب تعقيم sanitized أي مدخلات للمستخدم يتم تضمينه أو استخدام وضع الحماية sandboxed له قبل تنفيذه.	✓	✓	✓	95
5.2.5	تحقق من أن التطبيق يحمي من هجمات حقن القالب template injection attacks من خلال التأكد من أن أي مدخلات للمستخدم يتم تضمينه معقم أو وضع الحماية.	✓	✓	✓	94
6.2.5	تحقق من أن التطبيق يحمي من هجمات SSRF ، عن طريق التحقق من صحة البيانات غير الموثوق بها أو بيانات تعريف ملف HTTP (HTTP file metadata) أو تعقيمها ، مثل أسماء الملفات وحقول مدخلات عنوان URL ، واستخدام قوائم البروتوكولات والنطاقات والمسارات والمنافذ المسموح بها.	✓	✓	✓	918
7.2.5	تحقق من أن التطبيق يقوم بتعقيم أو تعطيل أو وضع الحماية للمحتوى القابل للبرمجة Scalable Vector Graphics (SVG) الذي يوفره المستخدم ، خاصةً فيما يتعلق بـ XSS الناتج عن البرامج النصية المضمنة inline scripts و ForeignObject.	✓	✓	✓	159
8.2.5	تحقق من أن التطبيق يقوم بتعقيم أو تعطيل أو وضع الحماية لمحتوى لغة قالب التعبير user-supplied scriptable أو لغة قالب التعبير expression template language التي يوفرها المستخدم ، مثل Markdown أو CSS أو XSL Stylesheets أو BBCode أو ما شابه ذلك.	✓	✓	✓	94

ق3.5 ترميز المخرجات ومنع الحقن

يعد ترميز المخرجات بالقرب من المترجم أو بجواره أمرًا بالغ الأهمية لأمن أي تطبيق. عادةً ، ترميز المخرجات هو غير دائم not persisted ، ولكن يتم استخدامه لجعل المخرجات آمنة في سياق المخرجات المناسب للاستخدام الفوري. سيؤدي الفشل في ترميز المخرجات إلى تطبيق غير آمن وقابل للحقن.

#	التوصيف	L1	L2	L3	CWE
1.3.5	تحقق من أن ترميز المخرجات مناسب للمترجم والسياق المطلوب. على سبيل المثال ، استخدم المرمزات encoders خصيصًا لقيم HTML ، وسمات HTML (HTML attributes) ، وJavaScript ، وبارامترات URL ، ورؤوس HTTP ، و SMTP ، وغيرها كما يتطلب السياق ، لا سيما من المدخلات غير الموثوق بها (مثل الأسماء التي تحتوي على Unicode أو الفواصل العليا ، مثل ʻ أو O'Hara). (C4)	✓	✓	✓	116
2.3.5	تحقق من أن ترميز المخرجات يحافظ على مجموعة الأحرف واللغة التي اختارها المستخدم، بحيث تكون أي نقطة محرف Unicode (Unicode character point) صالحة ويتم التعامل معها بأمان. (C4)	✓	✓	✓	176
3.3.5	تحقق من العناية بالسياق context-aware ، ويفضل أن يكون مؤتمتًا – أو في أسوأ الأحوال ، يدويًا – يحمي من XSS المنعكس والمخزن والمستند إلى DOM (reflected, stored, and DOM based XSS). (C4)	✓	✓	✓	79
4.3.5	تحقق من أن اختيار البيانات أو استعلامات قاعدة البيانات (مثل SQL و HQL و ORM و NoSQL) تستخدم الاستعلامات ذات المدخلات parameterized أو ORMs أو أطر عمل الكيانات entity frameworks أو أن تكون محمية بطريقة أخرى من هجمات حقن قواعد البيانات. (C3)	✓	✓	✓	89

#	التوصيف	L1	L2	L3	CWE
5.3.5	تحقق من أنه في حالة عدم وجود parameterized أو تقنيات أكثر أمانًا ، يتم استخدام ترميز المخرجات الخاص بالسياق للحماية من هجمات الحقن ، مثل استخدام SQL escaping للحماية من حقن SQL. (C3, C4)	✓	✓	✓	89
6.3.5	تحقق من أن التطبيق يحمي من هجمات حقن JSON وهجمات JSON eval و JavaScript expression evaluation. (C4)	✓	✓	✓	830
7.3.5	تحقق من أن التطبيق يحمي من الثغرات الأمنية كحقن LDAP ، أو أنه يتم تنفيذ ضوابط أمان محددة لمنع حقن LDAP. (C4)	✓	✓	✓	90
8.3.5	تحقق من أن التطبيق يحمي من حقن أوامر نظام التشغيل وأن استدعاءات نظام التشغيل تستخدم استعلامات نظام تشغيل ذات مدخلات parameterized OS queries أو تستخدم ترميز إخراج سطر الأوامر السياقي contextual command line output encoding. (C4)	✓	✓	✓	78
9.3.5	تحقق من أن التطبيق يحمي من هجمات تضمين الملفات المحلية Local File Inclusion (LFI) أو تضمين الملفات عن بُعد Remote File Inclusion (RFI).	✓	✓	✓	829
10.3.5	تحقق من أن التطبيق يحمي من حقن XPath أو هجمات حقن XML. (C4)	✓	✓	✓	643

ملاحظة: استخدام الاستعلامات ذات المدخلات parameterized queries أو الهروب من SQL (parameterized queries) لا يكفي دائمًا ؛ لا يمكن الهروب من أسماء الجداول والأعمدة ، ORDER BY وما إلى ذلك. يؤدي تضمين البيانات التي قدمها المستخدم التي تم تجاوزها في هذه الحقول إلى فشل الاستعلامات أو حقن SQL.

ملاحظة: يسمح تنسيق SVG صراحةً بـ ECMA script في جميع السياقات تقريبًا ، لذلك قد لا يكون من الممكن حظر جميع SVG XSS vectors تمامًا. إذا كان تحميل SVG مطلوبًا ، فنحن نوصي بشدة إما بتقديم هذه الملفات التي تم تحميلها كـ text/plain أو استخدام مجال محتوى منفصل يوفره المستخدم separate user supplied content domain لمنع XSS الناجح من الاستيلاء على التطبيق.

ق4.5 الذاكرة Memory والسلسلة String والشيفرة المصدرية غير المُدارة unmanaged code

سيتم تطبيق المتطلبات التالية فقط عندما يستخدم التطبيق لغة أنظمة systems language أو شيفرة مصدرية غير مُدارة unmanaged code.

#	التوصيف	L1	L2	L3	CWE
1.4.5	تحقق من أن التطبيق يستخدم سلسلة آمنة للذاكرة memory-safe string ونسخ ذاكرة أكثر أمانًا و المؤشر pointer arithmetic لاكتشاف أو منع stack, buffer, heap overflows.		✓	✓	120
2.4.5	تحقق من أن سلاسل التنسيق format strings لا تأخذ مدخلات يحتمل أن تكون معادية ، وأنها ثابتة.		✓	✓	134
3.4.5	تحقق من استخدام تقنيات التحقق من صحة الإشارة والمجال والمدخلات sign, range, and input validation techniques لمنع integer overflows.		✓	✓	190

ق5.5 متطلبات منع فك التسلسل

#	التوصيف	L1	L2	L3	CWE
1.5.5	تحقق من أن الكائنات المتسلسلة تستخدم فحوصات سلامة integrity checks أو أنها مشفرة لمنع إنشاء كائن معاد hostile object creation أو التلاعب بالبيانات data tampering. (C5)	✓	✓	✓	502
2.5.5	تحقق من أن التطبيق يقيد محللات XML (XML Parsers) بشكل صحيح لاستخدام التكوين الأكثر تقييدًا restrictive configuration وللتأكد من تعطيل الميزات غير الآمنة مثل حل الكيانات الخارجية resolving external entities لمنع هجمات XML eXternal Entity (XXE).	✓	✓	✓	611
3.5.5	تحقق من أن فك تسلسل البيانات غير الموثوق بها يتم تجنبه أو حمايته في كل من التعليمات البرمجية المخصصة ومكتبات الجهات الخارجية (مثل محللات JSON و XML و YAML).	✓	✓	✓	502
4.5.5	تحقق من أنه عند تحليل JSON في المتصفحات أو الخلفيات المستندة إلى JavaScript ، يتم استخدام JSON.parse لتحليل مستند JSON. لا تستخدم eval() لتحليل JSON.	✓	✓	✓	95

المراجع

لمزيد من المعلومات، يمكن أيضاً الاطلاع على:

- OWASP Testing Guide 4.0: Input Validation Testing
- OWASP Cheat Sheet: Input Validation
- OWASP Testing Guide 4.0: Testing for HTTP Parameter Pollution
- OWASP LDAP Injection Cheat Sheet
- OWASP Testing Guide 4.0: Client Side Testing
- OWASP Cross Site Scripting Prevention Cheat Sheet
- OWASP DOM Based Cross Site Scripting Prevention Cheat Sheet
- OWASP Java Encoding Project
- OWASP Mass Assignment Prevention Cheat Sheet
- DOMPurify – Client-side HTML Sanitization Library
- XML External Entity (XXE) Prevention Cheat Sheet

لمزيد من المعلومات حول الهروب التلقائي auto-escaping ، يرجى الاطلاع على:

- Reducing XSS by way of Automatic Context-Aware Escaping in Template Systems
- AngularJS Strict Contextual Escaping
- AngularJS ngBind
- Angular Sanitization
- Angular Security
- ReactJS Escaping
- Improperly Controlled Modification of Dynamically-Determined Object Attributes

لمزيد من المعلومات حول فك التسلسل deserialization ، يرجى الاطلاع على:

- OWASP Deserialization Cheat Sheet
- OWASP Deserialization of Untrusted Data Guide

ت6: التشفير المخزن

الهدف من ضوابط الأمان

تأكد من أن التطبيق الذي يتم التحقق منه يفي بالمتطلبات عالية المستوى التالية:

- تفشل جميع وحدات التشفير بطريقة آمنة ويتم التعامل مع الأخطاء بشكل صحيح.
- يتم استخدام مولد رقم عشوائي مناسب.
- تتم إدارة الوصول إلى المفاتيح بشكل آمن.

ق1.6 تصنيف البيانات

أهم الأصول هي البيانات التي تتم معالجتها أو تخزينها أو نقلها بواسطة تطبيق ما. قم دائمًا بإجراء تقييم لتأثير الخصوصية privacy impact assessment لتصنيف احتياجات حماية البيانات لأي بيانات مخزنة بشكل صحيح.

#	التوصيف	L1	L2	L3	CWE
1.1.6	تحقق من أن البيانات الخاصة المنظمة privacy impact assessment مخزنة بشكل مشفر أثناء الراحة at rest، مثل معلومات التعريف الشخصية Personally Identifiable Information (PII) أو المعلومات الشخصية الحساسة أو البيانات التي يُحتمل أن تكون خاضعة للائحة العامة لحماية البيانات (GDPR) الخاصة بالاتحاد الأوروبي EU's GDPR.		✓	✓	311
2.1.6	تحقق من أن البيانات الصحية المنظمة regulated health data مخزنة بشكل مشفر أثناء الراحة، مثل السجلات الطبية أو تفاصيل الجهاز الطبي أو سجلات البحث مجهولة المصدر.		✓	✓	311
3.1.6	تحقق من أن البيانات المالية المنظمة regulated financial data يتم تخزينها مشفرة أثناء الراحة ، مثل الحسابات المالية أو التخلف عن السداد أو تاريخ الائتمان أو السجلات الضريبية أو سجل الدفع أو المستفيدين أو سجلات البحث أو السوق مجهولة المصدر.		✓	✓	311

ق2.6 الخوارزميات

توضح التطورات الحديثة في التشفير أن الخوارزميات وأطوال المفاتيح الآمنة سابقًا لم تعد آمنة أو كافية لحماية البيانات. لذلك ، ينبغي أن يكون من الممكن تغيير الخوارزميات.

على الرغم من أن هذا القسم لا يتم اختباره بسهولة ، إلا أن المطورين يجب أن يعتبروا هذا القسم بأكمله إلزاميًا على الرغم من أن L1 مفقود من معظم العناصر.

#	التوصيف	L1	L2	L3	CWE
1.2.6	تحقق من أن جميع وحدات التشفير cryptographic modules تفشل بشكل آمن ، وأن الأخطاء يتم معالجتها بطريقة لا تتيح هجمات Padding Oracle.	✓	✓	✓	310
2.2.6	تحقق من استخدام خوارزميات التشفير والأنماط modes والمكتبات التي أثبتت جدواها في الصناعة أو المعتمدة من الحكومة ، بدلاً من التشفير المخصص المبرمَج custom coded cryptography. (C8)		✓	✓	327
3.2.6	تحقق من تكوين متجه تهيئة التشفير encryption initialization vector والنصوص المشفرة cipher و block modes بشكل آمن وباستخدام أحدث التوصيات.		✓	✓	326
4.2.6	تحقق من أن الأرقام العشوائية أو التشفير أو خوارزميات التجزئة أو أطوال المفاتيح أو الدورات rounds أو النصوص المشفرة ciphers أو الأنماط ، يمكن إعادة تكوينها أو ترقيتها أو تبديلها في أي وقت للحماية من فواصل التشفير cryptographic breaks. (C8)		✓	✓	326
5.2.6	تحقق من أنماط الكتلة غير الآمنة insecure block modes المعروفة (مثل ECB ، وما إلى ذلك) ، وأنماط الحشو padding modes (مثل PKCS # 1 v1.5 ، وما إلى ذلك) ، وخوارزميات التشفير ذات أحجام الكتل الصغيرة ciphers with small block sizes (مثل Triple-DES ، و Blowfish ، وما إلى ذلك) ، وخوارزميات التجزئة الضعيفة (مثل MD5 و SHA1 وما إلى ذلك) لا يتم استخدامها ما لم يكن ذلك مطلوبًا للتوافق مع الإصدارات السابقة.		✓	✓	326

#	التوصيف	L1	L2	L3	CWE
6.2.6	تحقق من أن nonces ومتجهات التهيئة initialization vectors وأرقام الاستخدام الفردي الأخرى other single use numbers لا تستخدم الأرقام أكثر من مرة في مفتاح تشفير معين. يجب أن تكون طريقة التوليد مناسبة للخوارزمية المستخدمة.		✓	✓	326
7.2.6	تحقق من مصادقة البيانات المشفرة عبر التوقيعات signatures أو أنماط التشفير المصدق عليها authenticated cipher modes أو HMAC لضمان عدم تغيير النص المشفر من قبل طرف غير مصرح له.			✓	326
8.2.6	تحقق من أن جميع عمليات التشفير تتم في وقت ثابت ، مع عدم وجود عمليات "ماس كهربائي short-circuit " في المقارنات أو الحسابات أو القيم الراجعة returns ، لتجنب تسريب المعلومات.			✓	385

ق3.6 القيم العشوائية

إنه من الصعب توليد الأرقام الزائفة العشوائية الحقيقية True Pseudo-random Number Generation (PRNG) بطريقة صحيحة. بشكل عام ، المصادر الجيدة للإنتروبيا داخل النظام سوف يتم نفاذها سريعًا إذا تم الإفراط في استخدامها ، ولكن المصادر ذات العشوائية الأقل يمكن أن تؤدي إلى مفاتيح وأسرار يمكن التنبؤ بها.

#	التوصيف	L1	L2	L3	CWE
1.3.6	تحقق من أن جميع الأرقام العشوائية وأسماء الملفات العشوائية ومعرفات GUID العشوائية والسلاسل العشوائية يتم توليدها باستخدام مولد الأرقام العشوائي الآمن المشفر المعتمد من وحدة التشفير cryptographic module's approved cryptographically secure random number generator عندما يكون الهدف هو عدم تخمين هذه القيم العشوائية من قبل المهاجم.		✓	✓	338
2.3.6	تحقق من إنشاء المعرفات الفريدة العمومية العشوائية GUIDs باستخدام خوارزمية GUID v4 ، ومولد الأرقام العشوائي الزائف الآمن Cryptographically-secure Pseudo-random Number Generator (CSPRNG). يمكن التنبؤ بمعرفات GUIDs التي تم توليدها باستخدام مولدات الأرقام العشوائية الزائفة الأخرى.		✓	✓	338
3.3.6	تحقق من إنشاء أرقام عشوائية مع الانتروبيا المناسبة حتى عندما يكون التطبيق تحت حمل ثقيل heavy load ، أو أن التطبيق يتدهور بطريقة آمنة في مثل هذه الظروف.			✓	338

ق4.6 إدارة السر

على الرغم من أن هذا القسم لا يتم اختباره بسهولة ، إلا أن المطورين يجب أن يعتبروا هذا القسم بأكمله إلزاميًا على الرغم من أن L1 مفقود من معظم العناصر.

#	التوصيف	L1	L2	L3	CWE
1.4.6	تحقق من استخدام حل إدارة الأسرار secrets management solution مثل خزنة المفاتيح لإنشاء الأسرار وتخزينها والتحكم في الوصول إليها وتدميرها بشكل آمن. (C8)		✓	✓	798
2.4.6	تحقق من عدم تعرض المادة الرئيسية للتطبيق ولكن بدلاً من ذلك تستخدم وحدة أمان معزولة مثل vault لعمليات التشفير. (C8)		✓	✓	320

المراجع

لمزيد من المعلومات، يمكن أيضاً الاطلاع على:

- OWASP Testing Guide 4.0: Testing for weak Cryptography
- OWASP Cheat Sheet: Cryptographic Storage
- FIPS 140-2

ت7: التسجيل ومعالجة الخطأ

الهدف من ضوابط الأمان

الهدف الأساسي من معالجة الأخطاء error handling والتسجيل logging هو توفير معلومات مفيدة للمستخدم والمسؤولين وفرق الاستجابة للحوادث. ليس الهدف هو الحصول على كميات هائلة من السجلات ، ولكن سجلات عالية الجودة تتضمن معلومات مفيدة وبارزة.

غالبًا ما تحتوي السجلات عالية الجودة على بيانات حساسة ، ويجب حمايتها وفقًا لقوانين أو توجيهات خصوصية البيانات المحلية. يجب أن يشمل ذلك:

- عدم جمع أو تسجيل المعلومات الحساسة ما لم يكن ذلك مطلوبًا على وجه التحديد.
- التأكد من التعامل مع جميع المعلومات المسجلة بشكل آمن ومحمي وفقًا لتصنيف البيانات الخاص بها.
- ضمان عدم تخزين السجلات إلى الأبد ، ولكن لها عمر مطلق أقصر ما يمكن.

إذا كانت السجلات تحتوي على بيانات خاصة أو حساسة ، يختلف تعريفها من بلد إلى آخر ، فإن السجلات تصبح من أكثر المعلومات حساسية التي يحتفظ بها التطبيق وبالتالي جذابة للغاية للمهاجمين في حد ذاتها.

من المهم أيضًا التأكد من فشل التطبيق بشكل آمن وأن الأخطاء لا تكشف عن معلومات غير ضرورية.

ق1.7 محتوى السجل

يعد تسجيل المعلومات الحساسة أمرًا خطيرًا – حيث يتم تصنيف السجلات بنفسها ، مما يعني ضرورة تشفيرها وإخضاعها لسياسات الاحتفاظ ويجب الكشف عنها في عمليات تدقيق الأمان. تأكد من الاحتفاظ بالمعلومات الضرورية فقط في السجلات ، وبالتأكيد لا توجد عمليات دفع أو بيانات اعتماد (بما في ذلك الرموز المميزة للجلسة) أو معلومات حساسة أو معلومات شخصية.

إن ق1.7 يغطي OWASP Top 10 2017:A10. وكلا القسمين غير قابلين لاختبار الاختراق ، فهو مهم من أجل:

- ضمان امتثال المطورين الكامل لهذا القسم ، كما لو تم تعيين جميع العناصر على أنها L1.
- تحقق مختبرو الاختراق من الامتثال الكامل لجميع العناصر في ق1.7 عبر المقابلة أو لقطات الشاشة أو التأكيد assertion.

#	التوصيف	L1	L2	L3	CWE
1.1.7	تحقق من أن التطبيق لا يسجل بيانات الاعتماد أو تفاصيل عمليات الدفع. يجب تخزين الرموز المميزة للجلسة فقط في السجلات في شكل مجزأ hashed بطريقة لا يمكن الرجوع فيها irreversible. (C9, C10)	✓	✓	✓	532
2.1.7	تحقق من أن التطبيق لا يسجل بيانات حساسة أخرى على النحو المحدد بموجب قوانين الخصوصية المحلية أو سياسة الأمان ذات الصلة. (C9)	✓	✓	✓	532
3.1.7	تحقق من أن التطبيق يسجل الأحداث المتعلقة بالأمان بما في ذلك أحداث المصادقة الناجحة والفاشلة وفشل التحكم في الوصول وفشل إلغاء التسلسل وفشل التحقق من صحة المدخلات. (C5, C7)		✓	✓	778
4.1.7	تحقق من أن كل سجل حدث log event يتضمن المعلومات الضرورية التي من شأنها أن تسمح بإجراء تحقيق مفصل للجدول الزمني عند وقوع الحدث. (C9)		✓	✓	778

ق2.7 معالجة السجل

يعد التسجيل في الوقت المناسب أمرًا بالغ الأهمية لأحداث التدقيق والفرز triage والتصعيد escalation. تأكد من أن سجلات التطبيق واضحة ويمكن مراقبتها وتحليلها بسهولة إما محليًا أو يتم إرسالها إلى نظام مراقبة عن بُعد remote monitoring system.

إن ق2.7 يغطي OWASP Top 10 2017:A10. As 2017:A10. وكلا القسمين غير قابلين لاختبار الاختراق ، فهو مهم من أجل:

- ضمان امتثال المطورين الكامل لهذا القسم ، كما لو تم تعيين جميع العناصر على أنها L1.
- تحقق مختبرو الاختراق من الامتثال الكامل لجميع العناصر في ق1.7 عبر المقابلة أو لقطات الشاشة أو التأكيد assertion.

#	التوصيف	L1	L2	L3	CWE
1.2.7	تحقق من تسجيل جميع قرارات المصادقة authentication decisions، دون تخزين الرموز المميزة للجلسة أو كلمات المرور الحساسة. يجب أن يتضمن ذلك الطلبات مع الـ metadata ذات الصلة اللازمة للتحقيقات الأمنية security investigations.		✓	✓	778
2.2.7	تحقق من إمكانية تسجيل جميع قرارات التحكم في الوصول وتسجيل جميع القرارات الفاشلة. يجب أن يتضمن ذلك الطلبات مع الـ metadata ذات الصلة اللازمة للتحقيقات الأمنية.		✓	✓	285

ق3.7 متطلبات حماية السجل

السجلات التي يمكن تعديلها أو حذفها بشكل طفيف لا تفيد في التحقيقات والملاحقات القضائية. قد يؤدي الكشف عن السجلات إلى كشف التفاصيل الداخلية حول التطبيق أو البيانات التي يحتوي عليها. يجب توخي الحذر عند حماية السجلات من الكشف أو التعديل أو الحذف غير المصرح به.

#	التوصيف	L1	L2	L3	CWE
1.3.7	تحقق من أن جميع مكونات التسجيل تقوم بترميز البيانات بشكل مناسب لمنع حقن السجل. (C9)		✓	✓	117
2.3.7	[تم حذفها ، مكررة عن 1.3.7]				
3.3.7	تحقق من أن سجلات الأمان محمية من الوصول والتعديل غير المصرح به. (C9)		✓	✓	200
4.3.7	تحقق من مزامنة مصادر الوقت time sources مع الوقت والمنطقة الزمنية الصحيحة. ضع في اعتبارك وبشكل صارم التسجيل في التوقيت العالمي المنسق فقط إذا كانت الأنظمة عالمية للمساعدة في تحليل الأدلة الرقمية بعد الحادث. (C9)		✓	✓	

ملاحظة: من الصعب اختبار ومراجعة ترميز السجل (1.3.7) باستخدام أدوات ديناميكية آلية واختبارات الاختراق ، ولكن يجب على مهندسي المعمارية والمطورين ومراجعي الشيفرة المصدرية اعتباره من متطلبات المستوى 1.

ق4.7 معالجة الأخطاء

الغرض من معالجة الأخطاء هو السماح للتطبيق بتوفير الأحداث الأمنية ذات الصلة للمراقبة والفرز traige والتصعيد. ليس الغرض هو توليد سجلات. عند تسجيل الأحداث المتعلقة بالأمان ، تأكد من تحقق الغاية من وجود السجل ، وأنه يمكن تمييزه بواسطة SIEM أو برامج التحليل.

#	التوصيف	L1	L2	L3	CWE
1.4.7	تحقق من ظهور رسالة عامة عند حدوث خطأ غير متوقع أو حساس للأمان ، من المحتمل أن يكون بمعرف فريد يمكن لموظفي الدعم استخدامه للتحقيق. (C10)	✓	✓	✓	210
2.4.7	تحقق من استخدام معالجة الاستثناءات exception handling (أو مكافئ وظيفي) عبر قاعدة التعليمات البرمجية codebase لحساب حالات الخطأ المتوقعة وغير المتوقعة. (C10)		✓	✓	544
3.4.7	تحقق من تعريف معالج الأخطاء "الملاذ الأخير last resort" والذي سيلتقط كافة الاستثناءات التي لم تتم معالجتها. (C10)		✓	✓	431

ملاحظة: لا تدعم لغات معينة ، مثل Swift و Go – ومن خلال ممارسة التصميم الشائعة – والعديد من اللغات الوظيفية ، الاستثناءات أو معالجات أحداث الملاذ الأخير last resort . في هذه الحالة ، يجب على مهندسي المعمارية والمطورين استخدام نمط أو لغة أو طريقة مريحة لإطار العمل لضمان أن التطبيقات يمكنها التعامل بأمان مع الأحداث الاستثنائية أو غير المتوقعة أو المتعلقة بالأمان.

المراجع

لمزيد من المعلومات، يمكن أيضاً الاطلاع على:

- OWASP Testing Guide 4.0 content: Testing for Error Handling
- OWASP Authentication Cheat Sheet section about error messages

ت8: حماية البيانات

الهدف من ضوابط الأمان

هناك ثلاثة عناصر رئيسية لحماية البيانات: السرية والنزاهة والتوافر Confidentiality, Integrity and Availability (CIA). يفترض هذا المعيار أن حماية البيانات يتم فرضها على نظام موثوق به ، مثل الخادم ، الذي تم تعزيزه ولديه إجراءات حماية كافية.

يجب أن تفترض التطبيقات أن جميع أجهزة المستخدم قد تعرضت للاختراق بطريقة ما. عندما ينقل أحد التطبيقات أو يخزن معلومات حساسة على أجهزة غير آمنة ، مثل أجهزة الكمبيوتر والهواتف والأجهزة اللوحية المشتركة ، يكون التطبيق مسؤولاً عن ضمان تشفير البيانات المخزنة على هذه الأجهزة ولا يمكن الحصول عليها بسهولة أو تغييرها أو الكشف عنها بشكل غير مشروع.

تأكد من أن التطبيق الذي يتم التحقق منه يفي بالمتطلبات عالية المستوى لحماية البيانات التالية:

- السرية Confidentiality: يجب حماية البيانات من المراقبة أو الإفصاح غير المصرح به سواء أثناء النقل أو عند التخزين.
- النزاهة Integrity: يجب حماية البيانات من أن يتم إنشاؤها بشكل ضار أو تغييرها أو حذفها من قبل مهاجمين غير مصرح لهم.
- التوافر Availability : يجب أن تكون البيانات متاحة للمستخدمين المصرح لهم حسب الحاجة.

ق1.8 حماية البيانات العامة

#	التوصيف	L1	L2	L3	CWE
1.1.8	تحقق من أن التطبيق يحمي البيانات الحساسة من التخزين المؤقت cached في مكونات الخادم مثل موازنات التحميل load balancers وذاكرة التخزين المؤقت للتطبيق application caches.		✓	✓	524
2.1.8	تحقق من أن جميع النسخ المخزنة أو المؤقتة cached or temporary للبيانات الحساسة الموجودة على الخادم محمية من الوصول غير المصرح به أو تم إزالتها / إبطالها بعد وصول المستخدم المصرح له إلى البيانات الحساسة.		✓	✓	524
3.1.8	تحقق من أن التطبيق يقلل من عدد البارامترات في الطلب ، مثل الحقول المخفية hidden fields ومتغيرات Ajax (Ajax variables) وقيم ملفات تعريف الارتباط cookies والرؤوس headers.		✓	✓	233
4.1.8	تحقق من أن التطبيق يمكنه الاكتشاف والتنبيه عن عدد غير طبيعي من الطلبات abnormal numbers of requests ، كالكشف من خلال الـ IP أو المستخدم أو إجمالي الطلبات في الساعة أو اليوم أو أي نشاط غير طبيعي في التطبيق.		✓	✓	770
5.1.8	تحقق من إجراء نسخ احتياطية بشكل منتظم للبيانات الهامة وإجراء اختبار استعادة البيانات.			✓	19
6.1.8	تحقق من تخزين النسخ الاحتياطية بشكل آمن لمنع سرقة البيانات أو تلفها.			✓	19

ق2.8 حماية البيانات من جهة المستخدم

#	التوصيف	L1	L2	L3	CWE
1.2.8	تحقق من أن التطبيق يضبط رؤوسًا كافية لمكافحة التخزين المؤقت anti-caching headers بحيث لا يتم تخزين البيانات الحساسة مؤقتًا في المتصفحات الحديثة.	✓	✓	✓	525
2.2.8	تحقق من أن البيانات المخزنة في تخزين المتصفح browser storage (مثل التخزين المحلي local storage) أو تخزين الجلسة session storage أو قاعدة بيانات مفهرسة IndexedDB أو ملفات تعريف الارتباط cookies) لا تحتوي على بيانات حساسة أو معلومات تحديد الهوية الشخصية Personal Identifiable Information (PII).	✓	✓	✓	922
3.2.8	تحقق من مسح البيانات المصادق عليها من تخزين العميل client storage ، مثل متصفح DOM (browser DOM) ، بعد الإنهاء من جهة العميل أو إنهاء الجلسة.	✓	✓	✓	922

ق3.8 البيانات الخاصة الحساسة

يساعد هذا القسم في حماية البيانات الحساسة من الإنشاء أو القراءة أو التحديث أو الحذف دون إذن ، لا سيما في الكميات الكبيرة bulk quantities. الامتثال لهذا القسم يعني الامتثال لـ ت4: التحكم بالوصول ، وعلى وجه الخصوص ق2.4. على سبيل المثال ، للحماية من التحديثات غير المصرح بها أو الكشف عن المعلومات الشخصية الحساسة يتطلب الالتزام بـ م1.2.4. يرجى الالتزام بهذا القسم و ت4 للتغطية الكاملة.

ملاحظة: تؤثر لوائح وقوانين الخصوصية ، مثل مبادئ الخصوصية الأسترالية APP-11 أو GDPR ، بشكل مباشر على كيفية تعامل التطبيقات مع تنفيذ تخزين المعلومات الشخصية الحساسة واستخدامها ونقلها. إن هذا يتراوح من عقوبات شديدة إلى نصيحة بسيطة. يرجى الرجوع إلى القوانين واللوائح المحلية الخاصة بك ، واستشارة متخصص أو محامي خصوصية مؤهل حسب الحاجة.

#	التوصيف	L1	L2	L3	CWE
1.3.8	تحقق من إرسال البيانات الحساسة إلى الخادم في نص أو رؤوس رسالة HTTP (HTTP message body or headers)، وأن بارامترات سلسلة الاستعلام query string parameters من أي طريقة HTTP (HTTP verb) لا تحتوي على بيانات حساسة.	✓	✓	✓	319
2.3.8	تحقق من أن المستخدمين لديهم طريقة لإزالة أو تصدير بياناتهم عند الطلب.	✓	✓	✓	212
3.3.8	تحقق من أن المستخدمين يتم تزويدهم بلغة واضحة فيما يتعلق بجمع واستخدام المعلومات الشخصية المقدمة وأن المستخدمين قد قدموا موافقة الاشتراك opt-in consent لاستخدام تلك البيانات قبل استخدامها بأي شكل من الأشكال.	✓	✓	✓	285
4.3.8	تحقق من تحديد جميع البيانات الحساسة التي تم إنشاؤها ومعالجتها بواسطة التطبيق ، وتأكد من وجود سياسة حول كيفية التعامل مع البيانات الحساسة. (C8)	✓	✓	✓	200
5.3.8	تحقق من تدقيق الوصول إلى البيانات الحساسة (دون تسجيل البيانات الحساسة نفسها) ، إذا تم جمع البيانات بموجب توجيهات حماية البيانات ذات الصلة data protection directives أو عندما يكون تسجيل الوصول مطلوبًا.		✓	✓	532
6.3.8	تحقق من الكتابة فوق (overwritten) المعلومات الحساسة الموجودة في الذاكرة بمجرد عدم الحاجة إليها للتخفيف من هجمات تفريغ الذاكرة memory dumping attacks ، باستخدام الأصفار zeroes أو البيانات العشوائية.		✓	✓	226
7.3.8	تحقق من أن المعلومات الحساسة أو الخاصة المطلوب تشفيرها مشفرة باستخدام خوارزميات معتمدة توفر السرية والنزاهة. (C8)		✓	✓	327
8.3.8	تحقق من أن المعلومات الشخصية الحساسة تخضع لتصنيف الاحتفاظ بالبيانات data retention classification ، مثل حذف البيانات القديمة أو المنتهية صلاحيتها تلقائيًا ، وفقًا لجدول زمني ، أو حسب ما يتطلبه الموقف.		✓	✓	285

عند التفكير في حماية البيانات ، يجب أن يكون الاعتبار الأساسي حول الاستخراج أو التعديل بالجملة bulk extraction or modification أو الاستخدام المفرط excessive usage. على سبيل المثال ، تسمح العديد من أنظمة الوسائط الاجتماعية للمستخدمين فقط بإضافة 100 صديق جديد يوميًا ، ولكن النظام الذي تأتي منه هذه الطلبات ليس مهمًا. قد ترغب منصة مصرفية في حظر أكثر من 5 معاملات في الساعة لتحويل أكثر من 1000 يورو من الأموال إلى مؤسسات خارجية. من المحتمل أن تكون متطلبات كل نظام مختلفة تمامًا ، لذا فإن اتخاذ قرار بشأن "غير طبيعي abnormal " يجب أن يأخذ في الاعتبار نموذج التهديد ومخاطر العمل. المعايير المهمة هي القدرة على اكتشاف ، أو ردع ، أو منع (وهو الأفضل) مثل هذه الإجراءات غير الطبيعية بالجملة abnormal bulk actions.

المراجع

لمزيد من المعلومات، يمكن أيضاً الاطلاع على:

- Consider using Security Headers website to check security and anti-caching headers
- OWASP Secure Headers project
- OWASP Privacy Risks Project
- OWASP User Privacy Protection Cheat Sheet
- European Union General Data Protection Regulation (GDPR) overview
- European Union Data Protection Supervisor – Internet Privacy Engineering Network

ت9: الاتصالات

الهدف من ضوابط الأمان

تأكد من أن التطبيق الذي يتم التحقق منه يحقق المتطلبات عالية المستوى التالية:

• فرض TLS أو تشفير قوي ، بغض النظر عن حساسية المحتوى.

• اتبع أحدث الإرشادات ، بما في ذلك:

– إرشادات التكوين.

– الخوارزميات المفضلة.

• تجنب الخوارزميات الضعيفة أو التي سيتم إهمالها قريبًا ، إلا كملاذ أخير (في حال عدم وجود حل بديل).

• تعطيل الخوارزميات المهملة أو المعروفة غير الآمنة.

وتحقق من تضمين المتطلبات التالية:

• كن على اطلاع دائم بنصائح الصناعة الموصى بها بشأن تكوين TLS الآمن ، حيث يتغير كثيرًا (غالبًا بسبب الانقطاعات الكارثية في الخوارزميات والنصوص المشفرة ciphers الموجودة).

• استخدم أحدث الإصدارات من أدوات مراجعة تكوين TLS لتكوين الترتيب المفضل واختيار الخوارزمية.

• تحقق من التكوين الخاص بك بشكل دوري للتأكد من أن الاتصال الآمن موجود دائمًا وفعال.

ق1.9 أمان اتصالات العميل

تأكد من إرسال جميع رسائل العميل عبر شبكات مشفرة ، باستخدام TLS 1.2 أو أحدث.

استخدم أدوات محدثة لمراجعة تكوين العميل على أساس منتظم.

#	التوصيف	L1	L2	L3	CWE
1.1.9	تحقق من استخدام TLS لجميع اتصالات العميل ، ولا يعود fall back إلى الاتصالات غير الآمنة أو غير المشفرة. (C8)	✓	✓	✓	319
2.1.9	تحقق من استخدام أدوات اختبار TLS المحدثَة والتي تقوم بتمكين مجموعة الخوارزميات القوية cipher suites فقط، مع تعيين مجموعة الخوارزميات القوية على أنها المفضلة.	✓	✓	✓	326
3.1.9	تحقق من تفعيل آخر إصدار موصى به فقط من من بروتوكول TLS والخوارزميات والتكوين ، مثل TLS 1.2 و TLS 1.3. يجب أن يكون أحدث إصدار من بروتوكول TLS هو الخيار المفضل.	✓	✓	✓	326

ق2.9 أمان اتصالات المخدم

اتصالات الخادم Server communications هي أكثر من مجرد HTTP. يجب أن تكون الاتصالات الآمنة من وإلى الأنظمة الأخرى ، مثل أنظمة المراقبة ، وأدوات الإدارة ، والوصول عن بُعد remote access و SSH ، والبرمجيات الوسيطة middleware ، وقاعدة البيانات ، والحواسيب المركزية mainframes ، وأنظمة الشريك أو المصدر الخارجي partner or external source systems – في مكانها الصحيح. كل هذه الأشياء يجب أن تكون مشفرة لمنع "من الصعب من الخارج ، ومن السهل للغاية اعتراضها من الداخل hard on the outside, trivially easy to intercept on the inside."

#	التوصيف	L1	L2	L3	CWE
1.2.9	تحقق من أن الاتصالات من وإلى الخادم تستخدم شهادات TLS الموثوقة. عند استخدام الشهادات التي تم توليدها داخليًا internally generated أو الموقعة ذاتيًا self-signed، يجب تكوين الخادم بحيث يثق فقط في المراجع المصدقة الداخلية internal CAs والشهادات الموقعة ذاتيًا self-signed certificates والتي تم تحديدها ويجب رفض كل الآخرين.		✓	✓	295
2.2.9	تحقق من استخدام الاتصالات المشفرة مثل TLS لجميع الاتصالات الواردة والصادرة ، بما في ذلك اتصالات منافذ الإدارة ، والمراقبة ، والمصادقة ، و API ، أو استدعاء خدمة الويب web service calls ، وقاعدة البيانات ، والسحابة ، serverless ، والحاسوب المركزي mainframe ، والخارجية واتصالات الشركاء partner connections. يجب ألا يعود fall back الخادم إلى البروتوكولات غير الآمنة أو غير المشفرة.		✓	✓	319
3.2.9	تحقق من مصادقة جميع الاتصالات المشفرة بالأنظمة الخارجية التي تتضمن معلومات أو وظائف حساسة.		✓	✓	287
4.2.9	تحقق من تمكين وتكوين إبطال الشهادة المناسبة ، مثل تدبيس بروتوكول حالة الشهادة عبر الإنترنت Online Certificate Status Protocol (OCSP) Stapling.		✓	✓	299
5.2.9	تحقق من تسجيل حالات فشل اتصال TLS من جهة الخادم.			✓	544

المراجع

لمزيد من المعلومات، يمكن أيضاً الاطلاع على:

- OWASP – TLS Cheat Sheet
- OWASP – Pinning Guide
- ملاحظات حول "أنماط TLS المعتمدة":
  - في الماضي ، أشارت ASVS إلى المعيار الأمريكي FIPS 140-2 ، ولكن كمعيار عالمي ، قد يكون تطبيق المعايير الأمريكية صعبًا أو متناقضًا أو محيرًا للتطبيق.
  - تتمثل الطريقة الأفضل لتحقيق التوافق مع القسم 9.1 في مراجعة الأدلة مثل TLS من جانب الخادم من Mozilla (Mozilla's Server Side TLS) أو إنشاء تكوينات جيدة معروفة (generate known good configurations) ، واستخدام أدوات تقييم TLS محدثة ومعروفة للحصول على المستوى المطلوب من الأمان.

ت10: الشيفرة الضارة Malicious Code

الهدف من ضوابط الأمان

تأكد من أن الشيفرة المصدرية تلبي المتطلبات عالية المستوى التالية:

- يتم التعامل مع الأنشطة الضارة بأمان وبشكل صحيح حتى لا تؤثر على بقية أنشطة التطبيق.
- لا تحتوي على قنابل موقوتة زمنية time bombs أو هجمات أخرى على أساس زمني time-based attacks.
- لا تقوم بالاتصال بوجهات خبيثة أو غير مصرح بها "phone home" to malicious or unauthorized destinations.
- لا يحتوي على أبواب خلفية back doors ، أو بيض عيد الفصح Easter eggs ، أو هجمات salami ، أو rootkits ، أو شيفرة مصدرية غير مصرح بها يمكن للمهاجم التحكم فيه.

العثور على الشيفرة البرمجية الضارة هو حكم خاطئ Proof of negative، ويستحيل التحقق من صحته تمامًا. يجب بذل أفضل الجهود لضمان عدم احتواء الشيفرة المصدرية على تعليمات برمجية ضارة أو وظائف غير مرغوب فيها.

ق1.10 سلامة الشيفرة المصدرية

أفضل دفاع ضد التعليمات البرمجية الضارة هو "كن واثقاً، ولكن تحقق trust, but verify ". غالبًا ما يكون إدخال تعليمات برمجية ضارة أو غير مصرح بها في التعليمات البرمجية جريمة جنائية في العديد من الولايات القضائية. يجب أن توضَح السياسات والإجراءات كل العقوبات المتعلقة بالشيفرات المصدرية الخبيثة.

يجب على المطورين الرئيسيين مراجعة عمليات التحقق من التعليمات البرمجية code check-ins بانتظام ، خاصة تلك التي قد تصل إلى وظائف الوقت أو المدخلات / المخرجات أو الشبكة time, I/O, or network functions.

#	التوصيف	L1	L2	L3	CWE
1.1.10	تحقق من استخدام أداة تحليل التعليمات البرمجية التي يمكنها اكتشاف التعليمات البرمجية التي يُحتمل أن تكون ضارة ، مثل وظائف الوقت وعمليات الملفات غير الآمنة واتصالات الشبكة.			✓	749

ق2.10 البحث عن الشيفرة المصدرية

الشيفرة الخبيثة نادرة للغاية ويصعب اكتشافها. يمكن أن تساعد المراجعة اليدوية للشيفرة المصدرية سطرًا بسطر في البحث عن القنابل المنطقية logical bombs، ولكن حتى مراجعي الكود الأكثر خبرة سيبذلوا جهداً كبيراً ويعانوا للعثور على تعليمات برمجية ضارة حتى لو كانوا يعلمون بوجودها.

لا يمكن الالتزام بهذا القسم بدون الوصول الكامل إلى الشيفرة المصدرية ، بما في ذلك المكتبات الخارجية third-party libraries.

#	التوصيف	L1	L2	L3	CWE
1.2.10	تحقق من أن الشيفرة المصدرية للتطبيق والمكتبات الخارجية لا تحتوي على إمكانات جمع البيانات أو الاتصال الهاتفي غير المصرح به unauthorized phone home . في حالة وجود هذه الوظيفة ، يجب الحصول على إذن المستخدم لتشغيلها قبل جمع أي بيانات.		✓	✓	359
2.2.10	تحقق من أن التطبيق لا يطلب أذونات غير ضرورية أو مفرطة للميزات أو المستشعرات المتعلقة بالخصوصية ، مثل جهات الاتصال أو الكاميرات أو الميكروفونات أو الموقع.		✓	✓	272
3.2.10	تحقق من أن الشيفرة المصدرية للتطبيق والمكتبات الخارجية لا تحتوي على أبواب خلفية Backdoors، مثل الحسابات أو المفاتيح المكتوبة داخل الشيفرة المصدرية hard-coded أو الإضافية غير الموثقة ، أو تمويه الشيفرة المصدرية code obfuscation ، أو undocumented binary blobs ، أو rootkits، أو ميزات تصحيح الأخطاء غير الآمنة insecure debugging أو مكافحة التصحيح anti-debugging ، أو وظائف غير آمنة أو مخفية أو منتهية الصلاحية والتي يمكن استخدامها بشكل ضار إذا تم اكتشافها.			✓	507
4.2.10	تحقق من أن الشيفرة المصدرية للتطبيق والمكتبات الخارجية لا تحتوي على قنابل موقوتة time bombs من خلال البحث عن الوظائف ذات الصلة بالتاريخ والوقت.			✓	511

#	التوصيف	L1	L2	L3	CWE
5.2.10	تحقق من أن الشيفرة المصدرية للتطبيق والمكتبات الخارجية لا تحتوي على شيفرة مصدرية ضارة ، مثل هجمات salami أو التجاوزات المنطقية logic bypasses أو القنابل المنطقية logic bombs.			✓	511
6.2.10	تحقق من أن الشيفرة المصدرية للتطبيق والمكتبات الخارجية لا تحتوي على بيض عيد الفصح Easter eggs أو أي وظائف أخرى يحتمل أن تكون غير مرغوب فيها.			✓	507

ق3.10 سلامة التطبيق

بمجرد نشر التطبيق ، لا يزال من الممكن تضمين الشيفرة المصدرية الضارة. تحتاج التطبيقات إلى حماية نفسها من الهجمات الشائعة ، مثل تنفيذ تعليمات برمجية غير موقعة من مصادر غير موثوقة executing unsigned code from untrusted sources وعمليات الاستحواذ على النطاق الفرعي subdomain takeovers.

من المحتمل أن يكون الامتثال لهذا القسم تشغيليًا ومستمرًا.

#	التوصيف	L1	L2	L3	CWE
1.3.10	تحقق من أنه إذا كان التطبيق يحتوي على ميزة التحديث التلقائي للعميل أو الخادم ، فيجب الحصول على التحديثات عبر القنوات الآمنة وتوقيعها رقميًا. يجب أن يتحقق رمز التحديث من صحة التوقيع الرقمي للتحديث قبل تثبيت التحديث أو تنفيذه.	✓	✓	✓	16
2.3.10	تحقق من أن التطبيق يستخدم وسائل حماية السلامة integrity protections ، مثل توقيع الشيفرة المصدرية code signing أو سلامة المصدر الفرعي subresource integrity. يجب ألا يقوم التطبيق بتحميل أو تنفيذ تعليمات برمجية من مصادر غير موثوق بها ، مثل تحميل تضمينات includes، أو وحدات modules، أو مكونات إضافية plugins ، أو شيفرة مصدرية ، أو مكتبات من مصادر غير موثوق بها أو من الإنترنت.	✓	✓	✓	353
3.3.10	تحقق من أن التطبيق يتمتع بالحماية من عمليات الاستحواذ على النطاق الفرعي subdomain takeovers إذا كان التطبيق يعتمد على إدخالات DNS (DNS Entries) أو نطاقات DNS الفرعية (DNS Subdomains)، مثل أسماء المجال منتهية الصلاحية ، أو مؤشرات DNS أو CNAMEs القديمة ، أو المشاريع منتهية الصلاحية في مستودع الشيفرة المصدرية العام public source code repos ، أو واجهات برمجة التطبيقات السحابية العابرة transient cloud APIs ، أو وظائف بدون خادم serverless functions ، أو دلاء التخزين storage buckets (autogen-bucket-id.cloud.example.com) أو ما شابه ذلك. يمكن أن تشمل الحماية التأكد من أن أسماء DNS التي تستخدمها التطبيقات يتم فحصها بانتظام من أجل انتهاء الصلاحية أو التغيير.	✓	✓	✓	350

المراجع

- Hostile Subdomain Takeover, Detectify Labs
- Hijacking of abandoned subdomains part 2, Detectify Labs

ت11: منطق الأعمال

الهدف من ضوابط الأمان

تأكد من أن التطبيق الذي يتم التحقق منه يفي بالمتطلبات عالية المستوى التالية:

- تدفق منطق الأعمال متسلسل ومعالج بالترتيب ولا يمكن تجاوزه.
- يتضمن منطق الأعمال حدودًا لاكتشاف ومنع الهجمات الآلية ، مثل التحويلات المستمرة للأموال الصغيرة ، أو إضافة مليون صديق واحدًا تلو الآخر ، وما إلى ذلك.
- تدفقات منطق الأعمال ذات القيمة العالية تأخذ بعين الاعتبار حالات الإساءة abuse cases والجهات الفاعلة الخبيثة ، ولديها وسائل حماية ضد الانتحال spoofing والعبث tampering والكشف عن المعلومات information disclosure ورفع مستوى هجمات الصلاحيات elevation of privilege attacks.

ق1.11 أمان منطق الأعمال

يعتبر أمن منطق الأعمال فرديًا جدًا لكل تطبيق بحيث لن يتم تطبيق قائمة تحقق واحدة على الإطلاق. يجب تصميم أمان منطق الأعمال للحماية من التهديدات الخارجية المحتملة – لا يمكن إضافته باستخدام جدران حماية تطبيقات الويب أو الاتصالات الآمنة. نوصي باستخدام نمذجة التهديد threat modeling أثناء سباقات التصميم design sprints ، على سبيل المثال باستخدام OWASP Cornucopia أو أدوات مماثلة.

#	التوصيف	L1	L2	L3	CWE
1.1.11	تحقق من أن التطبيق سيعالج فقط تدفقات منطق الأعمال لنفس المستخدم بترتيب خطوات متسلسل وبدون تخطي الخطوات.	✓	✓	✓	841
2.1.11	تحقق من أن التطبيق سيعالج تدفقات منطق الأعمال فقط مع معالجة جميع الخطوات في وقت بشري واقعي ، أي لا يتم إرسال المعاملات transactions بسرعة كبيرة.	✓	✓	✓	799
3.1.11	تحقق من أن التطبيق لديه حدود مناسبة لإجراءات أو معاملات transactions تجارية معينة يتم فرضها بشكل صحيح على أساس كل مستخدم.	✓	✓	✓	770
4.1.11	تحقق من أن التطبيق يحتوي على ضوابط لمكافحة الأتمتة ضد الاستدعاءات المفرطة مثل استخراج البيانات الجماعية mass data exfiltration ، أو طلبات منطق العمل business logic requests ، أو رفع الملفات ، أو هجمات تعطيل الخدمة denial of service attacks.	✓	✓	✓	770
5.1.11	تحقق من أن التطبيق يحتوي على حدود منطق الأعمال أو التحقق من الصحة للحماية من مخاطر أو تهديدات الأعمال المحتملة ، والتي تم تحديدها باستخدام نمذجة التهديد أو منهجيات مماثلة.	✓	✓	✓	841
6.1.11	تحقق من أن التطبيق لا يعاني من مشكلات "وقت التحقق حتى وقت الاستخدام Time Of Check to Time Of Use" (TOCTOU) أو ظروف السباق الأخرى race conditions للعمليات الحساسة.		✓	✓	367
7.1.11	تحقق من مراقبة التطبيق للأحداث أو الأنشطة غير العادية من منظور منطق الأعمال. على سبيل المثال ، محاولات تنفيذ إجراءات خارج النظام أو إجراءات لن يحاول المستخدم العادي تنفيذها أبدًا. (C9)		✓	✓	754
8.1.11	تحقق من أن التطبيق يحتوي على تنبيهات قابلة للتكوين configurable alerting عند اكتشاف هجمات آلية أو نشاط غير عادي.		✓	✓	390

المراجع

لمزيد من المعلومات، يمكن أيضاً الاطلاع على:

- OWASP Web Security Testing Guide 4.1: Business Logic Testing
- يمكن تحقيق مكافحة الأتمتة بعدة طرق ، بما في ذلك استخدام OWASP AppSensor و OWASP Automated Threats to Web Applications
- OWASP AppSensor يمكن أن يساعد أيضًا في اكتشاف الهجمات والاستجابة لها.
- OWASP Cornucopia

ت12: الملفات والموارد

الهدف من ضوابط الأمان

تأكد من أن التطبيق الذي يتم التحقق منه يفي بالمتطلبات عالية المستوى التالية:

- يجب التعامل مع بيانات الملف غير الموثوق Untrusted file data بطريقة آمنة.
- يتم تخزين بيانات الملف غير الموثوق بها التي تم الحصول عليها من مصادر غير موثوق بها خارج مسار جذر الويب web root وبأذونات محدودة limited permissions.

ق1.12 رفع الملف

على الرغم من أن القنابل المضغوطة zip bombs قابلة للاختبار بشكل كبير باستخدام تقنيات اختبار الاختراق ، إلا أنها تعتبر L2 وما فوق لتشجيع التفكير في التصميم والتطوير من خلال اختبار يدوي دقيق ، ولتجنب اختبار الاختراق اليدوي الآلي أو غير الماهر لحالة رفض الخدمة.

#	التوصيف	L1	L2	L3	CWE
1.1.12	تحقق من أن التطبيق لن يقبل الملفات الكبيرة التي قد تملأ مساحة التخزين أو تتسبب في تعطيل الخدمة.	✓	✓	✓	400
2.1.12	تحقق من أن التطبيق يتحقق من الملفات المضغوطة (مثل zip و gz و docx و odt) مقابل الحد الأقصى المسموح به للحجم غير المضغوط ومقابل الحد الأقصى لعدد الملفات قبل فك ضغط الملف.		✓	✓	409
3.1.12	تحقق من فرض الحصة النسبية لحجم الملف file size quota والحد الأقصى لعدد الملفات لكل مستخدم لضمان عدم تمكن مستخدم واحد من ملء التخزين بعدد كبير جدًا من الملفات أو بالملفات الكبيرة جدًا.		✓	✓	770

ق2.12 سلامة الملف

#	التوصيف	L1	L2	L3	CWE
1.2.12	تحقق من أن الملفات التي تم الحصول عليها من مصادر غير موثوقة قد تم التحقق من صحتها لتكون من النوع المتوقع بناءً على محتوى الملف.		✓	✓	434

ق12.3 تنفيذ الملف

#	التوصيف	L1	L2	L3	CWE
1.3.12	تحقق من أن بيانات تعريف لاسم الملف filename metadata التي يرسلها المستخدم لا يتم استخدامها مباشرة بواسطة أنظمة ملفات النظام أو إطار العمل وأن واجهة برمجة تطبيقات URL (URL API) تستخدم للحماية من اجتياز المسار path traversal.	✓	✓	✓	22
2.3.12	تحقق من صحة بيانات تعريف لاسم الملف filename metadata المقدمة من المستخدم أو تجاهلها لمنع الكشف عن الملفات المحلية (Local File Inclusion LFI) أو إنشائها أو تحديثها أو إزالتها.	✓	✓	✓	73
3.3.12	تأكد من التحقق من صحة بيانات التعريف لاسم الملف filename metadata المقدمة من المستخدم أو تجاهلها لمنع الكشف عن الملفات البعيدة أو تنفيذها عبر هجمات تضمين الملفات عن بُعد Remote File Inclusion (RFI) أو هجمات تزوير الطلب من جانب الخادم Server-side Request Forgery (SSRF).	✓	✓	✓	98
4.3.12	تحقق من أن التطبيق يحمي من تنزيل الملف الانعكاسي Reflective File Download (RFD) عن طريق التحقق من صحة أو تجاهل أسماء الملفات التي يرسلها المستخدم في بارامتر JSON أو JSONP أو عنوان URL ، ويجب ضبط قيمة رأس استجابة response Content-Type header بـ text/plain ، ورأس Content-Disposition يجب أن يكون له اسم ملف ثابت.	✓	✓	✓	641
5.3.12	تحقق من أن بيانات التعريف للملف غير الموثوق بها لا تُستخدم مباشرةً مع واجهة برمجة تطبيقات النظام أو المكتبات ، للحماية من حقن أوامر نظام التشغيل OS command injection.	✓	✓	✓	78

#	التوصيف	L1	L2	L3	CWE
6.3.12	تحقق من أن التطبيق لا يتضمن وظائف من مصادر غير موثوق بها ولا ينفذها، مثل شبكات توزيع المحتوى التي لم يتم التحقق منها unverified content distribution networks ، أو مكتبات JavaScript ، أو مكتبات node npm ، أو مكتبات DLL من جانب الخادم.		✓	✓	829

ق4.12 تخزين الملف

#	التوصيف	L1	L2	L3	CWE
1.4.12	تحقق من أن الملفات التي تم الحصول عليها من مصادر غير موثوقة مخزنة خارج جذر الويب web root ، بأذونات محدودة limited permissions.	✓	✓	✓	522
2.4.12	تحقق من أن الملفات التي تم الحصول عليها من مصادر غير موثوقة يتم فحصها بواسطة مضادات الفيروسات antivirus scanners لمنع تحميل وتخديم محتوى ضار معروف.	✓	✓	✓	509

ق5.12 تحميل الملف

#	التوصيف	L1	L2	L3	CWE
1.5.12	تحقق من تكوين طبقة الويب web tier لخدمة الملفات ذات امتدادات محددة لمنع تسرب المعلومات أو الشيفرة المصدرية بشكل غير مقصود. على سبيل المثال ، يجب حظر ملفات النسخ الاحتياطي (مثل .bak) وملفات العمل المؤقتة (مثل .swp) والملفات المضغوطة (.zip. و .tar.gz وما إلى ذلك) وغيرها من الامتدادات التي يشيع استخدامها بواسطة المحررين ما لم يكن ذلك مطلوبًا.	✓	✓	✓	552
2.5.12	تحقق من أن الطلبات المباشرة للملفات التي تم تحميلها لن يتم تنفيذها أبدًا كمحتوى HTML / JavaScript.	✓	✓	✓	434

ق6.12 حماية SSRF

#	التوصيف	L1	L2	L3	CWE
1.6.12	تحقق من تكوين خادم الويب أو التطبيق بقائمة سماحية allow list من الموارد أو الأنظمة التي يمكن للخادم إرسال الطلبات إليها أو تحميل البيانات / الملفات منها.	✓	✓	✓	918

المراجع

لمزيد من المعلومات، يمكن أيضاً الاطلاع على:

- File Extension Handling for Sensitive Information
- Reflective file download by Oren Hafif
- OWASP Third Party JavaScript Management Cheat Sheet

ت13: واجهة برمجة التطبيقات وخدمة الويب

الهدف من ضوابط الأمان

تأكد من أن التطبيق الذي يتم التحقق منه والذي يستخدم واجهات برمجة تطبيقات طبقة الخدمة الموثوقة trusted service layer APIs (التي تستخدم عادةً JSON أو XML أو GraphQL) لديه:

- المصادقة الكافية وإدارة الجلسة والتفويض لجميع خدمات الويب.
- التحقق من صحة المدخلات لجميع البارامترات التي تنتقل من مستوى ثقة أدنى إلى مستوى أعلى.
- ضوابط أمنية فعالة لجميع أنواع واجهات برمجة التطبيقات ، بما في ذلك السحابة وواجهة برمجة التطبيقات بدون خادم cloud and Serverless API.

يرجى قراءة هذا الفصل مع جميع الفصول الأخرى في نفس المستوى ؛ لم نعد نكرر ضوابط المصادقة أو إدارة جلسة API.

ق1.13 أمان خدمة الويب العامة

#	التوصيف	L1	L2	L3	CWE
1.1.13	تحقق من أن جميع مكونات التطبيق تستخدم نفس الترميزات encodings والمحللات parsers لتجنب هجمات التحليل parsing attacks التي تستغل مختلف سلوك URI أو سلوك تحليل الملف file parsing behavior الذي يمكن استخدامه في هجمات SSRF و RFI.	✓	✓	✓	116
2.1.13	[تم حذفها ، مكررة عن 1.3.4]				
3.1.13	تحقق من أن عناوين URL لواجهة برمجة التطبيقات لا تكشف معلومات حساسة ، مثل مفتاح واجهة برمجة التطبيقات API key والرموز المميزة للجلسة session tokens وما إلى ذلك.	✓	✓	✓	598
4.1.13	تحقق من أن قرارات التفويض يتم اتخاذها في كل من URI ، والتي يتم فرضها بواسطة الأمان البرمجي أو التعريفي programmatic or declarative security على وحدة التحكم controller أو وحدة التوجيه router ، وعلى مستوى الموارد ، يتم فرضها بواسطة الأذونات المستندة إلى النموذج model-based permissions.		✓	✓	285
5.1.13	تحقق من رفض الطلبات التي تحتوي على أنواع محتوى غير متوقعة أو مفقودة وذلك برؤوس مناسبة (حالة استجابة HTTP 406 غير مقبول Unacceptable أو 415 نوع وسائط غير مدعوم Unsupported Media Type).		✓	✓	434

ق2.13 خدمة الويب RESTful

التحقق من صحة مخطط JSON (JSON schema) في مرحلة مسودة للتوحيد القياسي draft stage of standardization (انظر المراجع). عند التفكير في استخدام التحقق من صحة مخطط JSON ، وهو أفضل ممارسة لخدمات الويب RESTful ، ضع في اعتبارك استخدام استراتيجيات التحقق من صحة البيانات الإضافية هذه جنبًا إلى جنب مع التحقق من صحة مخطط JSON:

- تحليل التحقق من صحة كائن JSON (JSON Object)، مثل ما إذا كانت هناك عناصر مفقودة أو إضافية.
- التحقق من صحة قيم كائن JSON باستخدام طرق التحقق من صحة المدخلات القياسية ، مثل نوع البيانات ، وتنسيق البيانات ، والطول ، وما إلى ذلك.
- والتحقق من صحة مخطط JSON الرسمي.

بمجرد إضفاء الطابع الرسمي على معيار التحقق من صحة مخطط JSON ، ستقوم ASVS بتحديث نصائحها في هذا المجال. راقب بعناية أي مكتبات التحقق من صحة مخطط JSON قيد الاستخدام ، حيث ستحتاج إلى تحديثها بانتظام حتى يصبح المعيار رسميًا ويتم التخلص من الأخطاء في عمليات التنفيذ المرجعية.

#	التوصيف	L1	L2	L3	CWE
1.2.13	تحقق من أن طرق RESTful HTTP الممكّنة هي خيار صالح للمستخدم أو الإجراء ، مثل منع المستخدمين العاديين من استخدام DELETE أو PUT على واجهة برمجة التطبيقات أو الموارد المحمية.	✓	✓	✓	650
2.2.13	تأكد من أن التحقق صحة مخطط JSON يعمل وموثوق منه قبل قبول المدخلات.	✓	✓	✓	20

#	التوصيف	L1	L2	L3	CWE
3.2.13	تحقق من أن خدمات الويب RESTful التي تستخدم ملفات تعريف الارتباط cookies محمية من التزوير عبر الموقع Cross-Site Request Forgery عبر استخدام واحد أو أكثر مما يلي على الأقل: نموذج إرسال مزدوج لملف تعريف الارتباط double submit cookie pattern أو CSRF nonces أو عمليات التحقق من رأس طلب Origin.	✓	✓	✓	352
4.2.13	[تم حذفها ، مكررة عن 4.1.11]				
5.2.13	تأكد من أن خدمات REST تتحقق صراحةً من نوع المحتوى الوارد ليكون النوع المتوقع ، مثل application / xml أو application / json.		✓	✓	436
6.2.13	تحقق من أن رؤوس الرسائل والحمولة جديرة بالثقة trustworthy ولم يتم تعديلها أثناء النقل. قد يكون طلب تشفير قوي للنقل (TLS فقط) كافياً في كثير من الحالات لأنه يوفر كلاً من السرية والسلامة. يمكن أن توفر التواقيع الرقمية لكل رسالة ضمانًا إضافيًا بالإضافة إلى حماية النقل للتطبيقات عالية الأمان ، ولكنها تجلب معها تعقيدًا ومخاطر إضافية مع الفوائد.		✓	✓	345

ق3.13 خدمة ويب SOAP

#	التوصيف	L1	L2	L3	CWE
1.3.13	تأكد من أن التحقق من صحة مخطط XSD يتم لضمان تكوين مستند XML بشكل صحيح ، متبوعًا بالتحقق من صحة كل حقل مدخلات قبل إجراء أي معالجة لتلك البيانات.	✓	✓	✓	20
2.3.13	تحقق من توقيع حمولة الرسالة باستخدام WS-Security لضمان النقل الموثوق به بين العميل والخدمة.		✓	✓	345

ملاحظة: نظرًا لوجود مشكلات تتعلق بهجمات XXE ضد DTD ، لا ينبغي استخدام التحقق من DTD وتعطيل تقييم DTD لإطار العمل وفقًا للمتطلبات المنصوص عليها في ت14: متطلبات التحقق من التكوين.

ق4.13 GraphQL

#	التوصيف	L1	L2	L3	CWE
1.4.13	تحقق من استخدام قائمة تسمح بالاستعلام أو مجموعة من تحديد العمق وتحديد المقدار combination of depth limiting and amount limiting لمنع تعطيل الخدمة (DoS) لـ GraphQL أو تعبير طبقة البيانات data layer expression كنتيجة للاستعلامات المتداخلة والمكلفة. لمزيد من السيناريوهات المتقدمة ، يجب استخدام تحليل تكلفة الاستعلام query cost analysis.		✓	✓	770
2.4.13	تحقق من أنه يجب تنفيذ GraphQL أو أي منطق تفويض طبقة البيانات data layer authorization logic في طبقة منطق الأعمال بدلاً من طبقة GraphQL.		✓	✓	285

المراجع

لمزيد من المعلومات، يمكن أيضاً الاطلاع على:

- OWASP Serverless Top 10
- OWASP Serverless Project
- OWASP Testing Guide 4.0: Configuration and Deployment Management Testing
- OWASP Cross-Site Request Forgery cheat sheet
- OWASP XML External Entity Prevention Cheat Sheet – General Guidance
- JSON Web Tokens (and Signing)
- REST Security Cheat Sheet
- JSON Schema
- XML DTD Entity Attacks
- Orange Tsai – A new era of SSRF Exploiting URL Parser In Trending Programming Languages

ت14: التكوين

الهدف من ضوابط الأمان

تأكد من أن التطبيق الذي يتم التحقق منه لديه:

- بيئة بناء build environment آمنة وقابلة للتكرار وقابلة للأتمتة automatable.
- أن تكون مكتبات الطرف الثالث Third-party libraries، وإدارة التبعية dependency والتكوين configuration متينة Hardened بحيث لا يتم تضمين المكونات القديمة أو غير الآمنة في التطبيق.

يجب أن يكون تكوين التطبيق الذي يتضمن التشغيل الأساسي فقط آمنًا ليكون على الإنترنت ، مما يعني تكوينًا خارج الصندوق بشكل آمن(safe out of the box).

ق1.14 البناء والنشر Build and Deploy

الأنابيب الخاصة بالبناء Build pipelines هو الأساس للأمان المتكرر – في كل مرة يتم اكتشاف شيء غير آمن ، يمكن حله في الشيفرة المصدرية ، أو السكربتات الخاصة بالبناء أو النشر build or deployment scripts ، واختبارها تلقائيًا. نحن نشجع بشدة على استخدام الأنابيب الخاصة بالبناء Build pipelines مع عمليات فحص الأمان والتبعية المؤتمتة automatic security and dependency checks والتي تحذر أو تكسر البناء warn or break the build لمنع نشر مشكلات الأمان المعروفة في بيئة الإنتاج الحقيقية. تؤدي الخطوات اليدوية التي يتم إجراؤها بشكل غير منتظم مباشرة إلى أخطاء أمنية يمكن تجنبها.

بما أن الصناعة تنتقل إلى نموذج DevSecOps ، من المهم ضمان استمرار توافر وسلامة النشر deployment والتكوين configuration لتحقيق حالة "جيدة معروفة known good ". في الماضي ، إذا تم اختراق نظام ما ، فقد يستغرق الأمر أيامًا إلى شهور لإثبات عدم حدوث مزيد من الاختراقات. اليوم ، مع ظهور بنية تحتية محددة بالبرمجيات ، وعمليات نشر A / B سريعة rapid A/B deployments بدون توقف ، وإنشاء حاويات آلية للبناء automated containerized builds ، فمن الممكن بشكل تلقائي ومستمر بناء ، وتقوية ، ونشر بديل "معروف بشكل جيد known good"" لأي نظام مخترق.

إذا كانت النماذج التقليدية لا تزال موجودة، فيجب اتخاذ خطوات يدوية لتقوية هذا التكوين وعمل نسخة احتياطية منه للسماح باستبدال الأنظمة المخترقة بأنظمة ذات سلامة عالية وغير مخترقة في الوقت المناسب.

يتطلب الامتثال لهذا القسم نظام بناء مؤتمتًا automated build system ، وإمكانية الوصول إلى السكربتات الخاصة بالبناء أو النشر build or deployment scripts.

#	التوصيف	L1	L2	L3	CWE
1.1.14	تحقق من تنفيذ عمليات بناء التطبيق ونشره بطريقة آمنة وقابلة للتكرار ، مثل أتمتة CI / CD ، وإدارة التكوين المؤتمت automated configuration management، وسكربتات النشر المؤتمت automated deployment scripts.		✓	✓	
2.1.14	تحقق من تكوين أعلام المترجم compiler flags لتمكين جميع عمليات الحماية والتحذيرات المتاحة لـ buffer overflow protections، بما في ذلك التوزيع العشوائي للمكدس stack randomization ، ومنع تنفيذ البيانات data execution prevention ، ولكسر البناء o break the build إذا تم العثور على مؤشر أو ذاكرة أو سلسلة تنسيق أو عدد صحيح أو عمليات السلسلة غير آمنة unsafe pointer, memory, format string, integer, or string operations.		✓	✓	120
3.1.14	تحقق من أن تكوين الخادم مقوّى hardened وفقًا لتوصيات مخدم التطبيق والأطر المستخدمة.		✓	✓	16
4.1.14	تحقق من أن التطبيق والتكوين وجميع التبعيات dependencies يمكن إعادة نشرها باستخدام سكربتات النشر المؤتمت automated deployment scripts، والتي تم إنشاؤها من دفتر تشغيل runbook موثق ومختبر في وقت معقول ، أو استعادتها من النسخ الاحتياطية في الوقت المناسب.		✓	✓	
5.1.14	تحقق من أن المسؤولين المصرح لهم يمكنهم التحقق من سلامة جميع التكوينات ذات الصلة بالأمان لاكتشاف التلاعب.			✓	

ق2.14 التبعية Dependency

تعد إدارة التبعية Dependency أمرًا بالغ الأهمية للتشغيل الآمن لأي تطبيق من أي نوع. يعد الفشل في مواكبة التبعيات القديمة أو غير الآمنة هو السبب الجذري لأكبر الهجمات وأكثرها تكلفة حتى الآن.

ملاحظة: في المستوى 1 ، يتعلق الامتثال بالبند 1.2.14 بالملاحظات أو اكتشافات جانب العميل والمكتبات والمكونات الأخرى ، بدلاً من التحليل الثابت للشيفرة المصدرية static code analysis أو تحليل التبعية dependency analysis الأكثر دقة. يمكن اكتشاف هذه التقنيات الأكثر دقة عن طريق المقابلة كما هو مطلوب.

#	التوصيف	L1	L2	L3	CWE
1.2.14	تحقق من أن جميع المكونات محدثة ، ويفضل استخدام مدقق التبعية dependency checker أثناء وقت البناء build أو الترجمة compile. (C2)	✓	✓	✓	1026
2.2.14	تحقق من إزالة جميع الميزات والوثائق ونماذج التطبيقات والإعدادات غير الضرورية.	✓	✓	✓	1002
3.2.14	تحقق من أنه إذا كانت أصول التطبيق ، مثل مكتبات JavaScript أو CSS أو خطوط الويب ، مستضافة خارجيًا على شبكة توصيل المحتوى Content Delivery Network (CDN) أو مزود خارجي ، فيجب أن يتم استخدام تكامل الموارد الفرعية Subresource Integrity (SRI) للتحقق من سلامة الأصل asset.	✓	✓	✓	829
4.2.14	تحقق من أن مكونات المكتبات الخارجية تأتي من مستودعات repositories محددة مسبقًا وموثوقة وتتم صيانتها باستمرار. (C2)		✓	✓	829
5.2.14	تحقق من الاحتفاظ بقائمة مواد البرمجيات Software Bill of Materials (SBOM) لكافة المكتبات الخارجية third party libraries المستخدمة. (C2)		✓	✓	
6.2.14	تحقق من تخفيض سطح الهجوم attack surface عن طريق وضع الحماية sandbox أو تغليف المكتبات الخارجية لكشف السلوك المطلوب فقط في التطبيق. (C2)		✓	✓	265

ق3.14 الإفصاح الأمني غير المقصود

يجب تقوية التكوينات الخاصة بالبيئة الحقيقية للحماية من الهجمات الشائعة ، مثل debug consoles ، ورفع مستوى هجمات البرمجة النصية عبر المواقع Cross-site Scripting (XSS) و Remote File Inclusion (RFI) ، وللتخلص من "نقاط الضعف" في اكتشاف المعلومات غير المهمة والمتواجدة من تقارير اختبار الاختراق. نادرًا ما يتم تصنيف العديد من هذه المشكلات على أنها مخاطر كبيرة ، ولكنها مرتبطة ببعضها البعض مع نقاط ضعف أخرى. إذا لم تكن هذه المشكلات موجودة بشكل افتراضي ، فإنها ترفع المستوى قبل أن تنجح معظم الهجمات.

#	التوصيف	L1	L2	L3	CWE
1.3.14	[تم حذفها ، مكررة عن 1.4.7]				
2.3.14	تحقق من أن أنماط تصحيح الأخطاء debug modes في إطار عمل التطبيق أو خادم الويب والتطبيق معطلة في البيئة الحقيقية لإزالة ميزات تصحيح الأخطاء debug features ووحدات تحكم المطورين developer consoles وإفصاحات الأمان غير المقصودة unintended security disclosures.	✓	✓	✓	497
3.3.14	تحقق من أن رؤوس HTTP أو أي جزء من استجابة HTTP لا تعرض معلومات إصدار تفصيلية لمكونات النظام.	✓	✓	✓	200

ق4.14 رؤوس أمان HTTP

#	التوصيف	L1	L2	L3	CWE
1.4.14	تحقق من أن كل استجابة HTTP تحتوي على رأس نوع المحتوى Content-Type header. أيضاً يجب أن تحدد مجموعة محارف character set آمنة (على سبيل المثال ، UTF-8 و ISO-8859-1). إذا كان المحتوى هو text / * و / + xml و xml / application فيجب أن يتوافق المحتوى مع رأس نوع المحتوى.	✓	✓	✓	173
2.4.14	تحقق من أن جميع استجابات API تحتوي على رأس Content-Disposition: attachment؛ filename="api.json" (أو اسم ملف آخر مناسب لنوع المحتوى).	✓	✓	✓	116
3.4.14	تحقق من وجود رأس استجابة سياسة أمان المحتوى Content Security Policy (CSP) header التي تساعد في التخفيف من تأثير هجمات XSS مثل ثغرات حقن HTML و DOM و JSON و JavaScript.	✓	✓	✓	1021
4.4.14	تحقق من أن جميع الاستجابات تحتوي على X-Content-Type-Options: nosniff header.	✓	✓	✓	116
5.4.14	تحقق من تضمين Strict-Transport-Security header في جميع الاستجابات ولجميع النطاقات الفرعية subdomains ، مثل Strict-Transport-Security: max-age=15724800; includeSubdomains.	✓	✓	✓	523
6.4.14	تحقق من تضمين Referrer-Policy header مناسب لتجنب كشف المعلومات الحساسة في عنوان URL من خلال الرأس "Referer" لأطراف غير موثوق بها.	✓	✓	✓	116
7.4.14	تحقق من أن محتوى تطبيق الويب لا يمكن تضمينه في موقع جهة خارجية بشكل افتراضي وأن تضمين الموارد الدقيقة مسموح به فقط عند الضرورة باستخدام الرؤوس Content-Security-Policy: frame-ancestors and X-Frame-Options response headers.	✓	✓	✓	1021

ق5.14 متطلبات رأس طلب HTTP

#	التوصيف	L1	L2	L3	CWE
1.5.14	تحقق من أن خادم التطبيق لا يقبل سوى طرق HTTP المستخدمة من قبل التطبيق / واجهة برمجة التطبيقات ، بما في ذلك pre-flight OPTIONS ، والسجلات / التنبيهات بشأن أي طلبات غير صالحة لسياق التطبيق.	✓	✓	✓	749
2.5.14	تحقق من أن Origin header المقدم لا يتم استخدامه للمصادقة أو قرارات التحكم في الوصول ، حيث يمكن للمهاجم تغيير عنوان Origin بسهولة.	✓	✓	✓	346
3.5.14	تحقق من أن Cross-Origin Resource Sharing (CORS) Access-Control-Allow-Origin header يستخدم قائمة سماح صارمة للنطاقات الموثوقة والمجالات الفرعية للمطابقة معها ولا يدعم الأصل "الفارغ" origin "null".	✓	✓	✓	346
4.5.14	تحقق من أن رؤوس HTTP المضافة بواسطة proxy موثوق به أو أجهزة SSO ، مثل رمز الحامل bearer token ، قد تمت مصادقتها بواسطة التطبيق.		✓	✓	306

المراجع

لمزيد من المعلومات، يمكن أيضاً الاطلاع على:

- OWASP Web Security Testing Guide 4.1: Testing for HTTP Verb Tampering
- تساعد إضافة Content-Disposition إلى استجابات واجهة برمجة التطبيقات على منع العديد من الهجمات استنادًا إلى سوء فهم نوع MIME بين العميل والخادم ، ويساعد خيار "filename" على وجه التحديد في منع هجمات تنزيل الملفات المنعكسة Reflected File Download attacks .
- Content Security Policy Cheat Sheet
- Exploiting CORS misconfiguration for BitCoins and Bounties
- OWASP Web Security Testing Guide 4.1: Configuration and Deployment Management Testing
- Sandboxing third party components

الملحق أ: قائمة المصطلحات

- **عشوائية تخطيط مساحة العنوان** Address Space Layout Randomization (ASLR) – أسلوب لجعل استغلال أخطاء فساد الذاكرة أكثر صعوبة.
- **قائمة السماح** Allow list– قائمة بالبيانات أو العمليات المسموح بها ، على سبيل المثال قائمة الأحرف المسموح بها لإجراء التحقق من صحة المدخلات.
- **أمان التطبيق** Application Security – يركز الأمان على مستوى التطبيق على تحليل المكونات التي تشكل طبقة التطبيق الخاصة بالنموذج المرجعي لربط الأنظمة المفتوحة (**OSI Model**) ، بدلاً من التركيز على نظام التشغيل الأساسي أو الشبكات المتصلة على سبيل المثال.
- **التحقق من أمان التطبيق** Application Security Verification – التقييم الفني لتطبيق مقابل OWASP ASVS.
- **تقرير التحقق من أمان التطبيق** Application Security Verification Report – تقرير يوثق النتائج الإجمالية والتحليل الداعم الذي تم إنتاجه بواسطة المدقق لتطبيق معين.
- **المصادقة** Authentication – التحقق من الهوية المطالب بها لمستخدم التطبيق.
- **التحقق المؤتمت** Automated Verification– استخدام الأدوات المؤتمتة (إما أدوات التحليل الديناميكي أو أدوات التحليل الثابتة أو كليهما) التي تستخدم تواقيع الثغرات الأمنية vulnerability signatures للعثور على المشكلات.
- **اختبار الصندوق الأسود** Black box testing– هو طريقة لاختبار البرنامج الذي يفحص وظائف التطبيق دون النظر إلى هياكله أو أعماله الداخلية.
- **المكوِّن** Component – وحدة تعليمات برمجية قائمة بذاتها ، مع واجهات القرص والشبكة المرتبطة التي تتواصل مع المكونات الأخرى.
- **البرمجة النصية عبر المواقع** Cross-Site Scripting (XSS) – ثغرة أمنية توجد عادةً في تطبيقات الويب مما يسمح بحقن البرامج النصية من جانب العميل في المحتوى.
- **وحدة التشفير** Cryptographic module – الأجهزة و / أو البرامج و / أو البرامج الثابتة التي تنفذ خوارزميات التشفير و / أو تنشئ مفاتيح التشفير.
- **تعداد نقاط الضعف الشائعة** Common Weakness Enumeration (CWE) – قائمة مطورة من قبل المجتمع لنقاط ضعف أمان البرامج الشائعة. إنها بمثابة لغة مشتركة، وعصا قياس لأدوات أمان البرامج، وكخط أساس لتحديد نقاط الضعف ، والتخفيف ، وجهود الوقاية.
- **التحقق من التصميم** Design Verification – التقييم الفني لمعمارية أمان التطبيق.
- **اختبار أمان التطبيق الديناميكي** Dynamic Application Security Testing (DAST) – تم تصميم التقنيات لاكتشاف الظروف التي تشير إلى وجود ثغرة أمنية في تطبيق في حالته قيد التشغيل.
- **التحقق الديناميكي** Dynamic Verification– استخدام الأدوات المؤتمتة التي تستخدم تواقيع الثغرات الأمنية للعثور على المشكلات أثناء تنفيذ أحد التطبيقات.
- **الهوية السريعة عبر الإنترنت** Fast Identity Online (FIDO) – مجموعة من معايير المصادقة التي تسمح باستخدام مجموعة متنوعة من طرق المصادقة المختلفة بما في ذلك القياسات الحيوية ووحدات النظام الأساسي الموثوقة (Trusted Platform Modules TPM) ورموز أمان USB (USB security tokens) وما إلى ذلك.
- **المعرف الفريد العالمي** Globally Unique Identifier (GUID) – رقم مرجعي فريد يستخدم كمعرف في البرنامج.
- **بروتوكول نقل النص التشعبي** Hyper Text Transfer Protocol (HTTPS) – بروتوكول تطبيق لأنظمة معلومات الوسائط التشعبية الموزعة والتعاونية. إنها أساس اتصال البيانات لشبكة الويب العالمية.
- **المفاتيح المخزنة** Hardcoded keys – مفاتيح التشفير التي يتم تخزينها على نظام الملفات ، سواء أكان ذلك في رمز أو تعليقات أو ملفات.
- **وحدة أمان الأجهزة** Hardware Security Module (HSM) – مكون الأجهزة القادر على تخزين مفاتيح التشفير والأسرار الأخرى بطريقة محمية.
- **إسبات لغة الاستعلام** Hibernate Query Language (HQL) – لغة استعلام تشبه في المظهر لغة SQL المستخدمة بواسطة مكتبة Hibernate ORM.
- **التحقق من صحة المدخلات** Input Validation – توحيد canonicalization والتحقق من صحة مدخلات المستخدم غير الموثوق به.

- **الشيفرة المصدرية الضارة** Malicious Code – شيفرة مصدرية يتم إدخاله في تطبيق ما أثناء تطويره دون علم مالك التطبيق ، والذي يتحايل على سياسة الأمان المقصودة للتطبيق. ليست مثل البرامج الضارة مثل الفيروسات أو الدودة!
- **البرمجيات الخبيثة** Malware – التعليمات البرمجية القابلة للتنفيذ التي يتم إدخالها في التطبيق أثناء وقت التشغيل دون معرفة مستخدم التطبيق أو المسؤول.
- **مشروع أمان تطبيق الويب المفتوح** (OWASP) Open Web Application Security Project – مشروع أمان تطبيق الويب المفتوح (OWASP) هو مجتمع عالمي مجاني ومفتوح يركز على تحسين أمان برامج التطبيقات. مهمتنا هي جعل أمان التطبيق "مرئيًا" ، بحيث يمكن للأفراد والمؤسسات اتخاذ قرارات مستنيرة بشأن مخاطر أمان التطبيق. انظر: https://www.owasp.org/
- **كلمة المرور لمرة واحدة** (OTP) One-time Password – كلمة مرور يتم إنشاؤها بشكل فريد لاستخدامها في مناسبة واحدة.
- **رسم الخرائط العلائقية للكائن** (ORM) Object-relational Mapping – نظام يستخدم للسماح لقاعدة بيانات علائقية / قائمة على الجدول بالإشارة إليها والاستعلام عنها داخل برنامج تطبيق باستخدام نموذج كائن متوافق مع التطبيق.
- **وظيفة اشتقاق المفتاح المعتمد على كلمة المرور 2** (PBKDF2) Password-Based Key Derivation Function 2 – خوارزمية خاصة أحادية الاتجاه تُستخدم لإنشاء مفتاح تشفير قوي من نص مدخلات (مثل كلمة المرور) وقيمة salt عشوائية إضافية ، وبالتالي يمكن استخدامها تجعل من الصعب اختراقها كلمة مرور في وضع عدم الاتصال إذا تم تخزين القيمة الناتجة بدلاً من كلمة المرور الأصلية.
- **معلومات التعريف الشخصية** (PII) Personally Identifiable Information – هي المعلومات التي يمكن استخدامها بمفردها أو مع معلومات أخرى لتحديد شخص واحد أو الاتصال به أو تحديد موقعه ، أو لتحديد شخص في السياق.
- **ملف تنفيذي مستقل عن الموضع** (PIE) Position-independent executable – جسم من كود الآلة body of machine code الذي يتم وضعه في مكان ما في الذاكرة الأساسية primary memory ويتم تنفيذه بشكل صحيح بغض النظر عن عنوانه المطلق absolute address.
- **البنية التحتية للمفتاح العام** (PKI) Public Key Infrastructure – ترتيب يربط المفاتيح العامة public keys بهويات الكيانات الخاصة. يتم إنشاء الارتباط من خلال عملية التسجيل وإصدار الشهادات في ومن قبل سلطة إصدار الشهادات (certificate authority CA).
- **شبكة الهاتف العامة المحولة** (PSTN) Public Switched Telephone Network – شبكة الهاتف التقليدية بما في ذلك هواتف الخطوط الثابتة والهواتف المحمولة.
- **الطرف المعول** (RP) Relying Party – تطبيق يعتمد على مستخدم قام بالمصادقة تجاه مزود مصادقة منفصل. يعتمد التطبيق على نوع من الرموز المميزة أو مجموعة من التأكيدات الموقعة signed assertions التي يوفرها مزود المصادقة للوثوق في ما يدعيه المستخدم.
- **اختبار أمان التطبيق الثابت** (SAST) Static application security testing – مجموعة من التقنيات المصممة لتحليل الشيفرة المصدرية للتطبيق ورمز البايت Byte Code والثنائيات Binaries لظروف كتابة الشيفرة المصدرية والتصميم التي تدل على وجود ثغرات أمنية. تحلل طرق SAST التطبيق من "الداخل إلى الخارج" وهو في حالة عدم التشغيل.
- **دورة حياة تطوير البرمجيات** (SDLC) Software development lifecycle – العملية التدريجية التي يتم من خلالها تطوير البرنامج من المتطلبات الأولية إلى النشر والصيانة.
- **معمارية الأمان** Security Architecture– تعبير مجرد لتصميم التطبيق يحدد ويصف مكان وكيفية استخدام عناصر التحكم في الأمان ، كما يحدد ويصف موقع وحساسية كل من بيانات المستخدم والتطبيق.
- **تكوين الأمان** Security Configuration – تكوين زمن التشغيل للتطبيق الذي يؤثر على كيفية استخدام عناصر التحكم في الأمان.
- **ضابط الأمان** Security Control – وظيفة أو مكون يقوم بإجراء فحص أمني (على سبيل المثال ، فحص التحكم في الوصول) أو عند استدعائه يؤدي إلى تأثير أمني (مثل إنشاء سجل تدقيق).
- **تزوير الطلب من جانب الخادم** (SSRF) Server-side Request Forgery – هجوم يسيء استخدام الوظائف الموجودة على الخادم لقراءة أو تحديث الموارد الداخلية عن طريق توفير أو تعديل عنوان URL الذي ستقوم الشيفرة المصدرية التي يتم تشغيلها على الخادم بقراءة البيانات أو إرسالها إليها.
- **مصادقة الدخول الموحد** (SSO) Single Sign-on Authentication – يحدث هذا عندما يقوم المستخدم بتسجيل الدخول إلى تطبيق واحد ثم يتم تسجيل الدخول تلقائيًا إلى تطبيقات أخرى دون الحاجة إلى إعادة المصادقة. على سبيل المثال ، إذا تم تسجيل الدخول إلى Google ، عند الوصول إلى خدمات Google الأخرى مثل YouTube ومحرّر مستندات Google و Gmail ، سيتم تسجيل دخولك تلقائيًا.

- **حقن SQL** (SQLi) SQL Injection – تقنية حقن التعليمات البرمجية المستخدمة لمهاجمة التطبيقات التي تعتمد على البيانات ، والتي يتم فيها إدراج استعلامات SQL الضارة في نقطة دخول entry point.
- **SVG** – رسومات موجهة قابلة للتحجيم Scalable Vector Graphics.
- **OTP على أساس الوقت** Time-based OTP – طريقة لإنشاء OTP حيث يعمل الوقت الحالي كجزء من الخوارزمية لإنشاء كلمة المرور.
- **نمذجة التهديدات** Threat Modeling– تقنية تتكون من تطوير بنى أمنية مصقولة بشكل متزايد لتحديد عوامل التهديد ، والمناطق الأمنية ، والضوابط الأمنية ، والأصول التقنية والتجارية الهامة.
- **أمان طبقة النقل** (TLS) Transport Layer Security – بروتوكولات التشفير التي توفر أمان الاتصال عبر اتصال الشبكة.
- **الوحدة النمطية للنظام الأساسي الموثوق به** (TPM) Trusted Platform Module – نوع من HSM يتم إرفاقه عادةً بمكون أجهزة أكبر مثل اللوحة الأم ويعمل بمثابة "جذر الثقة root of trust"" لهذا النظام.
- **المصادقة الثنائية** (2FA) Two-factor authentication – تضيف المستوى الثاني من المصادقة لتسجيل الدخول إلى الحساب.
- **العامل الثاني العالمي** (U2F) Universal 2nd Factor – أحد المعايير التي أنشأتها FIDO خصيصًا للسماح باستخدام مفتاح أمان USB أو NFC كعامل مصادقة ثانٍ.
- **أجزاء URI / URL / URL** URI/URL/URL fragments – معرف الموارد المنتظم هو سلسلة من الأحرف المستخدمة لتحديد اسم أو مورد ويب. غالبًا ما يتم استخدام محدد موقع المعلومات Uniform Resource Locator كمرجع إلى أحد الموارد.
- **المدقق** Verifier – الشخص أو الفريق الذي يقوم بمراجعة طلب مقابل متطلبات OWASP ASVS.
- **ما تراه هو ما تحصل عليه** (WYSIWYG) What You See Is What You Get – نوع من محرر المحتوى الغني الذي يوضح كيف سيبدو المحتوى بالفعل عند تقديمه بدلاً من إظهار الترميز المستخدم للتحكم في العرض.
- **شهادة X.509** X.509 Certificate– شهادة X.509 هي شهادة رقمية تستخدم معيار البنية التحتية للمفتاح العام (PKI) الدولي المقبول على نطاق واسع X.509 للتحقق من أن المفتاح العام ينتمي إلى هوية المستخدم أو الكمبيوتر أو الخدمة المضمنة في الشهادة.
- **الكيان الخارجي لـ XML (XXE)** XML external Entity– نوع من كيان XML يمكنه الوصول إلى المحتوى المحلي أو البعيد عبر معرف نظام معلن. قد يؤدي هذا إلى هجمات حقن مختلفة.

الملحق ب: المراجع

من المرجح أن تكون مشاريع OWASP التالية مفيدة لمستخدمي / متبني هذا المعيار:

مشاريع أواسب الأساسية

1. مشروع OWASP – TOP 10: https://owasp.org/www-project-top-ten/
2. دليل أواسب لاختبار تطبيقات الويب OWASP Web Security Testing Guide: https://owasp.org/www-project-web-security-testing-guide/
3. ضوابط أواسب الاستباقية OWASP Proactive Controls: https://owasp.org/www-project-proactive-controls/
4. إطار عمل أواسب للمعرفة الأمنية OWASP Security Knowledge Framework: https://owasp.org/www-project-security-knowledge-framework/
5. نموذج أواسب لنضج ضمان التطبيق (SAMM) OWASP Software Assurance Maturity Model: https://owasp.org/www-project-samm/

مشروع أواسب لمجموعة أوراق المناقشة

يحتوي هذا المشروع على عدد من أوراق المناقشة التي ستكون ذات صلة بمواضيع مختلفة في ASVS.

يوجد تعيين لـ ASVS يمكن العثور عليه هنا: https://cheatsheetseries.owasp.org/cheatsheets/IndexASVS.html

المشاريع المتعلقة بأمن الجوال

1. مشروع أواسب لأمن الجوال OWASP Mobile Security Project: https://owasp.org/www-project-mobile-security/
2. أواسب – العشر المخاطر الأولى في الجوال OWASP Mobile Top 10 Risks: https://owasp.org/www-project-mobile-top-10/
3. دليل أواسب لاختبار أمان الجوال ومعيار التحقق من أمان تطبيقات الأجهزة المحمولة OWASP Mobile Security Testing Guide and Mobile Application Security Verification Standard:

https://owasp.org/www-project-mobile-security-testing-guide/

مشاريع أواسب المتعلقة بانترنت الأشياء

1. مشروع أواسب لانترنت الأشياء OWASP Internet of Things Project: https://owasp.org/www-project-internet-of-things/

مشاريع أواسب بدون خادم Serverless

1. مشروع أواسب بدون خادم OWASP Serverless Project: https://owasp.org/www-project-serverless-top-10/

مشاريع أخرى

وبالمثل ، من المرجح أن تكون مواقع الويب التالية مفيدة لمستخدمي / متبني هذا المعيار

1. SecLists Github: https://github.com/danielmiessler/SecLists
2. تعداد نقاط الضعف الشائعة MITER: https://cwe.mitre.org/
3. مجلس معايير أمن PCI: https://www.pcisecuritystandards.org
4. متطلبات معيار أمان بيانات PCI (DSS) v3.2.1 وإجراءات تقييم الأمان:
https://www.pcisecuritystandards.org/documents/PCI_DSS_v3-2-1.pdf
5. إطار عمل أمان برامج PCI – متطلبات البرامج الآمنة وإجراءات التقييم:
https://www.pcisecuritystandards.org/documents/PCI-Secure-Software-Standard-v1_0.pdf
6. متطلبات دورة حياة البرامج الآمنة لـ PCI (Secure SLC) وإجراءات التقييم:
https://www.pcisecuritystandards.org/documents/PCI-Secure-SLC-Standard-v1_0.pdf

الملحق ج: متطلبات التحقق من إنترنت الأشياء

كان هذا الفصل في الأساس في الفرع الرئيسي ، ولكن مع العمل الذي قام به فريق OWASP IoT ، فليس من المنطقي الاحتفاظ بمكانين مختلفين حول هذا الموضوع. بالنسبة للإصدار 4.0 ، ننقل هذا إلى الملحق ، ونحث كل من يطلب ذلك ، بدلاً من استخدام مشروع أواسب لانترنت الأشياء الرئيسي

الهدف من ضوابط الأمان

يجب على الأجهزة المضمنة / إنترنت الأشياء:

- الحصول على نفس مستوى عناصر التحكم في الأمان داخل الجهاز كما هو موجود في الخادم ، من خلال فرض ضوابط الأمان في بيئة موثوقة.
- يجب أن يتم تخزين البيانات الحساسة على الجهاز بطريقة آمنة باستخدام التخزين المدعوم من الأجهزة مثل العناصر الآمنة.
- يجب أن تستخدم جميع البيانات الحساسة المرسلة من الجهاز أمان طبقة النقل.

متطلبات التحقق الأمني

#	التوصيف	L1	L2	L3	Since
ض1	تحقق من أن واجهات تصحيح أخطاء طبقة التطبيق application layer debugging interfaces مثل USB و UART والمتغيرات التسلسلية serial variants الأخرى معطلة أو محمية بكلمة مرور معقدة.	✓	✓	✓	4.0
ض2	تحقق من أن مفاتيح التشفير والشهادات فريدة لكل جهاز على حدة.	✓	✓	✓	4.0
ض3	تحقق من تمكين عناصر تحكم حماية الذاكرة مثل ASLR و DEP بواسطة نظام التشغيل المدمج / إنترنت الأشياء embedded/IoT operating system ، إن أمكن.	✓	✓	✓	4.0
ض4	تحقق من أن واجهات التصحيح على الرقاقة on-chip debugging interfaces مثل JTAG أو SWD معطلة أو أن آلية الحماية المتاحة تم تمكينها وتكوينها بشكل مناسب.	✓	✓	✓	4.0
ض5	تحقق من تنفيذ وتمكين التنفيذ execution الموثوق به ، إذا كان ذلك متاحًا على جهاز SoC أو وحدة المعالجة المركزية.	✓	✓	✓	4.0
ض6	تحقق من تخزين البيانات الحساسة والمفاتيح الخاصة والشهادات بشكل آمن في Secure Element أو TPM أو TEE (بيئة تنفيذ موثوقة) أو محمية باستخدام تشفير قوي.	✓	✓	✓	4.0
ض7	تحقق من أن تطبيقات البرامج الثابتة firmware apps protect تحمي البيانات أثناء النقل باستخدام أمان طبقة النقل.	✓	✓	✓	4.0
ض8	تحقق من أن تطبيقات البرامج الثابتة تتحقق من صحة التوقيع الرقمي لاتصالات الخادم.	✓	✓	✓	4.0
ض9	تحقق من أن الاتصالات اللاسلكية مصادقة بشكل متبادل.	✓	✓	✓	4.0
ض10	تحقق من إرسال الاتصالات اللاسلكية عبر قناة مشفرة.	✓	✓	✓	4.0
ض11	تحقق من استبدال أي استخدام لوظائف C المحظورة بوظائف مناسبة بحيث تكون بديلة وآمنة.	✓	✓	✓	4.0
ض12	تحقق من أن كل برنامج ثابت firmware يحتفظ بقائمة مواد للبرنامج تقوم بفهرسة مكونات الطرف الثالث ، والإصدارات ، والثغرات الأمنية المنشورة.	✓	✓	✓	4.0
ض13	تحقق من جميع الشيفرة المصدرية code بما في ذلك الثنائيات Binaries والمكتبات والأطر التابعة لجهات خارجية والتي تتم مراجعتها لبيانات الاعتماد المشفرة (الأبواب الخلفية Backdoors).	✓	✓	✓	4.0

#	التوصيف	L1	L2	L3	Since
ض14	تحقق من أن مكونات التطبيق والبرامج الثابتة firmware ليست عرضة لحقن أوامر نظام التشغيل عن طريق استدعاء أغلفة أوامر shell (shell command wrappers) أو البرامج النصية أو أن عناصر التحكم الأمنية تمنع حقن أوامر نظام التشغيل.	✓	✓	✓	4.0
ض15	تحقق من أن تطبيقات البرامج الثابتة تثبت التوقيع الرقمي بخادم (خوادم) موثوق به.		✓	✓	4.0
ض16	تحقق من وجود مقاومة للتلاعب tamper و / أو ميزات كشف التلاعب Tamper Detection.		✓	✓	4.0
ض17	تحقق من تمكين أي من تقنيات حماية الملكية الفكرية التي توفرها الشركة المصنعة للرقاقة.		✓	✓	4.0
ض18	تحقق من وجود ضوابط الأمان لعرقلة الهندسة العكسية للبرامج الثابتة (على سبيل المثال ، إزالة رموز التصحيح المطول verbose debugging symbols).		✓	✓	4.0
ض19	تحقق من قيام الجهاز بالتحقق من صحة توقيع صورة الإقلاع boot image signature قبل التحميل.		✓	✓	4.0
ض20	تحقق من أن عملية تحديث البرنامج الثابت ليست عرضة لهجمات وقت التحقق مقابل وقت الاستخدام time-of-check vs time-of-use attacks.		✓	✓	4.0
ض21	تحقق من أن الجهاز يستخدم توقيع الرمز code signing والتحقق من صحة ملفات ترقية البرامج الثابتة firmware upgrade files قبل التثبيت.		✓	✓	4.0
ض22	تحقق من أنه لا يمكن إرجاع الجهاز إلى الإصدارات القديمة downgraded (مكافحة التراجع anti-rollback) من البرامج الثابتة الصالحة.		✓	✓	4.0
ض23	تحقق من استخدام منشئ الأرقام العشوائية الزائفة الآمنة المشفرة cryptographically secure pseudo-random number generator على جهاز مضمن embedded device (على سبيل المثال ، باستخدام مولدات الأرقام العشوائية المزودة بشريحة using chip-provided random number generators).		✓	✓	4.0
ض24	تحقق من أن البرنامج الثابت يمكنه إجراء تحديثات تلقائية للبرامج الثابتة وفقًا لجدول زمني محدد مسبقًا.		✓	✓	4.0
ض25	تحقق من أن الجهاز يفحص البرامج الثابتة والبيانات الحساسة عند اكتشاف التلاعب أو استلام رسالة غير صالحة.			✓	4.0
ض26	تحقق من استخدام وحدات التحكم الصغيرة micro controllers فقط التي تدعم تعطيل واجهات تصحيح الأخطاء (مثل JTAG و SWD).			✓	4.0
ض27	تحقق من استخدام وحدات التحكم الصغيرة micro controllers فقط التي توفر حماية كبيرة من هجمات إلغاء السد والقنوات الجانبية decapping and side channel attacks.			✓	4.0
ض28	تحقق من عدم تعرض الآثار الحساسة sensitive traces للطبقات الخارجية للوحة الدائرة المطبوعة outer layers of the printed circuit board.			✓	4.0
ض29	تحقق من أن الاتصال بين الشرائح مشفر (على سبيل المثال ، اتصال اللوحة الرئيسية Main board إلى اللوحة الفرعية daughter board).			✓	4.0
ض30	تحقق من أن الجهاز يستخدم توقيع الرمز code signing والتحقق من صحة الرمز قبل التنفيذ.			✓	4.0
ض31	تحقق من أن المعلومات الحساسة المحفوظة في الذاكرة قد تم استبدالها overwritten بالأصفار بمجرد عدم الحاجة إليها.			✓	4.0
ض32	تحقق من أن تطبيقات البرامج الثابتة تستخدم حاويات kernel للعزل بين التطبيقات.			✓	4.0
ض33	تحقق من أن علامات المترجم الآمن مثل -fPIE، -fstack-protector-all، -Wl، -z، noexecstack، -Wl، -z، noexecheap قد تم تكوينها لإنشاءات البرامج الثابتة.			✓	4.0

#	التوصيف	L1	L2	L3	Since
ض34	تحقق من تكوين وحدات التحكم الصغيرة micro controllers مع حماية الشيفرة المصدرية code protection (إن أمكن).			✓	4.0

المراجع

لمزيد من المعلومات، يمكن أيضاً الاطلاع على:

- OWASP Internet of Things Top 10
- OWASP Embedded Application Security Project
- OWASP Internet of Things Project
- Trudy TCP Proxy Tool