استاندارد وارسی امنیت اپلیکیشن ۴.۰

نسخه نهایی

مارچ ۲۰۱۹

دیباچه

درباره استاندارد

استاندارد وارسی[1] امنیت اپلیکیشن، لیستی از الزامات امنیتی اپلیکیشن و یا آزمون‌هایی است که می‌تواند توسط معماران، توسعه‌دهندگان، آزمون‌کنندگان، متخصصین امنیت، توزیع‌کنندگان[2] ابزار و مصرف‌کنندگان، به‌منظور تعریف، ساخت، آزمون و یا وارسی امنیت اپلیکیشن‌ها استفاده شود.

ترجمه استاندارد

این استاندارد به سفارش مرکز ماهر ایران توسط آزمایشگاه تخصصی آپا دانشگاه فردوسی مشهد تهیه و ترجمه شده است که می تواند بعنوان مرجع بررسی، در فرایند آزمون نفوذپذیری استفاده گردد.

مرکز مدیریت امداد و هماهنگی عملیات رخدادهای رایانه ای

دانشگاه فردوسی مشهد
آزمایشگاه تخصصی آپا

در صورت وجود هر گونه مشکل در ترجمه یازخورد خود را به آدرس ایمیل زیر ارسال کنید :

pourali@cert.um.ac.ir

کپی‌رایت و لایسنس

نسخه ۴.۰.۱ ، خردادماه ۱۳۹۸ (مارچ ۲۰۱۹ میلادی)

[1] Verification
[2] Vendors

مدیران پروژه

- Andrew van der Stock
- Daniel Cuthbert
- Jim Manico
- Josh C Grossman
- Mark Burnett

مترجمان

- اردلان فروغی پور
- فاطمه دلدار
- آرمان قریشی
- بهنام شبیری
- سجاد ایرانمنش
- محمد کاهانی

سرپرست مترجمان

- اردلان فروغی پور
- سجاد پورعلی

مشارکت‌کنندگان و بازرسان

- Osama Elnaggar
- Erlend Oftedal
- Serg Belkommen
- David Johansson
- Tonimir Kisasondi
- Ron Perris
- Jason Axley
- Abhay Bhargav
- Benedikt Bauer
- Elar Lang
- ScriptingXSS
- Philippe De Ryck
- Grog's Axle
- Marco Schnüriger
- Jacob Salassi
- Glenn ten Cate
- Anthony Weems
- bschach
- javixeneize
- Dan Cornell
- hello7s
- Lewis Ardern
- Jim Newman
- Stuart Gunter
- Geoff Baskwill
- Talargoni
- Ståle Pettersen
- Kelby Ludwig
- Jason Morrow
- Rogan Dawes

استاندارد وارسی امنیت اپلیکیشن توسط کسانی ساخته شده است که از ASVS ۱.۰ در سال ۲۰۰۸ تا نسخه ۳.۰ در سال ۲۰۱۶ درگیر بوده‌اند. بخش عمده‌ای از ساختار و آیتم‌های وارسی که امروزه همچنان در ASVS هستند، در اصل توسط Mike Boberski، Jeff Williams و Dave Wichers نوشته شده‌اند، اما مشارکت‌کنندگان بیشتری نیز درگیر بوده‌اند. از همه کسانی که درگیر بوده‌اند سپاسگزاریم. برای دسترسی به لیست جامع تمامی کسانی که در نسخه‌های قبلی مشارکت داشته‌اند لطفاً به همان نسخه رجوع کنید.

پیشگفتار

به نسخه ۴.۰ استاندارد وارسی امنیت اپلیکیشن‌ها خوش آمدید. ASVS تلاشی جامعه‌محور[3]، در دایرکردن چارچوبی برای کنترل‌ها و الزامات امنیتی است که بر روی تعریف کنترل‌های امنیتی کاربردی و غیرکاربردی در طراحی، توسعه و آزمایش برنامه‌های کاربردی وب[4] و خدمات تحت وب[5] تمرکز دارد.

ASVS v4.0 نقطه اوج تلاش جامعه و بازخورد صنعت در دهه گذشته است. تلاش کرده‌ایم که استفاده از ASVS را در مراحل مختلف چرخه توسعه نرم‌افزارهای امن، ساده‌تر کنیم.

انتظار نمی‌رود که این استاندارد به‌طور ۱۰۰٪ مورد توافق همگان واقع شود. تحلیل ریسک همیشه تا حدودی یک امر نظری و وابسته به دیدگاه شخصی افراد است. در نتیجه هرگونه تلاش برای عمومیت دادن مفاهیم در یک چارچوب و استاندارد واحد با چالش‌هایی روبه‌رو است. به هر حال، امیدواریم که جدیدترین به‌روزرسانی‌های انجام‌شده در این نسخه، گامی در جهت صحیح باشد و مفاهیم مطرح‌شده در این استاندارد مهم صنعتی را بهبود بخشد.

تازه‌های نسخه ۴.۰

مهمترین تغییر در این نسخه، افزوده شدن NIST 800-63-3 (رهنمودهای هویت دیجیتالی[6]) است که ارائه‌دهنده کنترل‌های مدرن، مبتنی بر شواهد[7] و پیشرفته احرازاصالت است. اگرچه انتظار داریم که در همسو شدن با یک استاندارد اعتبارسنجی پیشرفته، متحمل سختی‌هایی شویم، اما معتقدیم که لازم است استانداردها با یکدیگر همسو باشند، به خصوص زمانی که استاندارد دیگری مبتنی بر شواهد است.

استانداردهای امنیت اطلاعات باید سعی کنند نیازهای منحصربه‌فرد را به حداقل برسانند، به‌طوری که سازمان‌ها مجبور به انتخاب بین کنترل‌های رقیب و یا ناسازگار نشوند. مستندات OWASP Top 10 2017 و در حال حاضر OWASP Application Security Verification Standard با NIST 800-63 برای احرازاصالت و مدیریت نشست‌ها هماهنگ شده‌اند. ما دیگر نهادهای تنظیم استاندارد را برای همکاری با ما، NIST و دیگران تشویق می‌کنیم تا به یک مجموعه کلی از کنترل‌های امنیتی نرم‌افزار برای به حداکثر رساندن امنیت و به حداقل رساندن هزینه‌های انطباق دست یابیم.

ASVS 4.0 به‌طور کامل از ابتدا تا انتها دوباره شماره‌گذاری شده است. طرح شماره‌گذاری جدید به ما اجازه داد تا شکاف‌ها را از فصل‌ها حذف کنیم و فصل‌های طولانی را به بخش‌های کوچک‌تر تقسیم نماییم تا یک توسعه‌دهنده یا یک تیم، تعداد کنترل‌های کمتری را انطباق دهند. به‌عنوان مثال، اگر یک برنامه از JWT استفاده نکند، کل قسمت JWT در بخش مدیریت نشست‌ها به آن قابل اعمال نیست.

---

[3] Community driven
[4] Web applications
[5] Web services
[6] Digital Identity Guidelines
[7] Evidence based

مورد جدید دیگر در نسخه ۴.۰ وجود یک نگاشت جامع برای سرشماری آسیب‌پذیری‌های عمومی (CWE) است که یکی از مهم‌ترین ویژگی‌های خواسته شده از ما در دهه گذشته بود. نگاشت CWE به تولیدکنندگان ابزارها و کسانی که از نرم‌افزارهای مدیریت آسیب‌پذیری استفاده می‌کنند، این امکان را می‌دهد که نتایج نسخه‌های قبلی ASVS و سایر ابزارها را با نسخه ۴.۰ انطباق دهند. برای ایجاد فضای مناسب برای ورود CWE مجبور به حذف ستون "Since" شدیم که البته پس از شماره‌گذاری جدید، نسبت به نسخه‌های قبلی ASVS دارای معنی کمتری بود. هر آیتم در ASVS دارای یک CWE متناظر نیست، همچنین چون در CWE تا حد زیادی موارد تکراری وجود دارد، تلاش کردیم که به‌جای استفاده الزاماً از نزدیک‌ترین تطابق، از یافته‌هایی که بیشترین استفاده را داشته‌اند استفاده شود. کنترل‌های وارسی، همیشه قابل نگاشت به آسیب‌پذیری‌های معادلی نیستند. ما از گفت‌وگو با جامعه CWE و زمینه‌های امنیت اطلاعات برای بستن هرچه بیشتر این شکاف استقبال می‌کنیم.

تلاش کرده‌ایم که الزامات لازم برای پاسخ‌گویی به نیازهای OWASP Top 10 2017 و OWASP Proactive Controls 2018 را برآورده و از آن نیز پیشتر رویم. از آن‌جایی که OWASP Top 10 2018 فقط حداقل‌هایی است که برای جلوگیری از اهمال‌ها است، ما آگاهانه تمام الزامات را به‌جز ۱۰ نیازمندی اول لاگ کردن، کنترل‌های سطح ۱ کرده‌ایم تا برای متقاضیان OWASP Top 10 راه‌اندازی یک استاندارد امنیتی آسان‌تر باشد.

قصد داریم اطمینان حاصل کنیم که سطح ۱ در ASVS 4.0 مجموعه‌ای جامع از بخش ۶.۵ در سند PCI DSS 3.2.1 برای طراحی برنامه، برنامه‌نویسی، آزمون، بررسی ایمن کد و آزمون نفوذپذیری است. این امر ایجاب می‌کند که سرریزی بافر[8] و عملیات‌های ناامن حافظه در بخش V5 و پرچم‌های زمان کامپایل مربوط به حافظه ناامن در بخش V14 را علاوه بر الزامات وارسی نرم‌افزارهای مطرح در زمینه صنعت و برنامه‌های کاربردی وب نیز مورد پوشش قرار دهیم.

ما تغییر جهت ASVS را از کنترل‌های برنامه‌های یک تکه[9] سمت سرویس‌دهنده، به کنترل‌های امنیتی برای تمام برنامه‌های مدرن و APIها کامل کرده‌ایم. در عصر برنامه‌نویسی[10] تابعی، APIهای بدون سرویس‌دهنده[11]، موبایل، ابری، محفظه‌ها[12]، CI/CD و DevSecOps و موارد دیگر، نمی‌توانیم معماری نرم‌افزارهای مدرن را نادیده بگیریم. برنامه‌های کاربردی مدرن بسیار متفاوت با ASVSای که در سال ۲۰۰۹ منتشر شد، طراحی می‌شوند. ASVS همیشه باید به آینده نگاه کند تا مشاوره‌های صحیحی برای مخاطبان اصلی ما – توسعه‌دهندگان - ارائه شود. هرگونه الزاماتی را که فرض می‌کند برنامه‌ها بر روی سیستم‌های متعلق به یک سازمان واحد اجرا می‌شوند شفاف‌سازی یا رها می‌کنیم.

با توجه به اندازه ASVS 4.0 و همچنین تمایل ما برای تبدیل شدن آن به یک پایه برای تمام تلاش‌های دیگر ASVS، بخش موبایل را به نفع Mobile Application Security Verification Standard (MASVS) حذف کردیم. ضمیمه IoT ASVS در مستندات بعدی در مورد اینترنت اشیا در پروژه OWASP Internet of Things ظاهر می‌شود. ما یک پیش‌نمایش اولیه از IOT ASVS را در ضمیمه

[8] Buffer overflow
[9] Monolithic
[10] Functional
[11] Server-less API
[12] Containers

C گنجانده‌ایم. همچنین از تیم OWASP Mobile Team و تیم OWASP IoT برای حمایت از ASVS تشکر می‌کنیم و منتظریم تا با آن‌ها در آینده برای ارائه استانداردهای مکمل همکاری کنیم.

در آخر، کنترل‌هایی که کمتر تأثیرگزار بودند را حذف کردیم. با گذر زمان ASVS تبدیل به یک مجموعه جامع از کنترل‌ها شده است، اما همه کنترل‌ها هنگام تولید نرم‌افزار امن، یکسان نیستند. این تلاش برای از بین بردن موارد کم اثر، می‌تواند بیشتر نیز باشد. در نسخه آینده ASVS سیستم امتیازدهی به آسیب‌پذیری‌های عمومی (CWSS[13]) در اولویت‌بندی بهتر بین کنترل‌هایی که واقعاً مهم هستند و آن‌هایی که باید کنار گذاشته شوند به ما کمک می‌کند.

ASVS 4.0 منحصراً بر پیشرو بودن به‌عنوان استاندارد برنامه‌ها و سرویس‌های تحت وب، با پوشش دادن معماری نرم‌افزار سنتی و مدرن و شیوه‌های امنیتی چابک[۱۴] و DevSecOps تمرکز خواهد کرد.

استفاده از ASVS

ASVS دو هدف اصلی دارد:

- کمک به سازمان‌ها برای توسعه و نگه‌داری اپلیکیشن‌های امن
- امکان‌پذیر کردن تطبیق نیازها با خدمات ارائه شده، برای فروشندگان سرویس‌های امنیتی، فروشندگان ابزارهای امنیتی و مصرف‌کنندگان

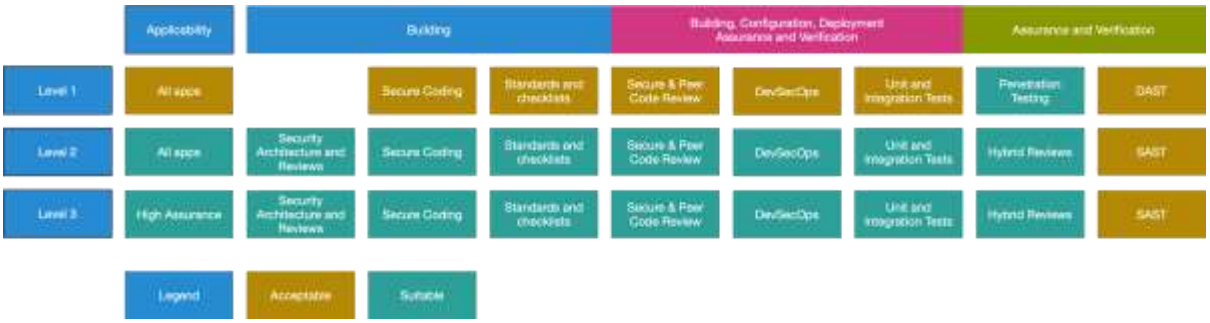
سطوح وارسی امنیت اپلیکیشن

استاندارد وارسی امنیت اپلیکیشن‌ها (ASVS) سه سطح وارسی امنیت (به‌صورت عمقی) به شرح زیر را تعریف می‌کند:

- ASVS سطح ۱ (L1): برای سطوح اطمینان پایین است. می‌توان کاملاً روی آن آزمون نفوذپذیری انجام داد.
- ASVS سطح ۲ (L2): برای اپلیکیشن‌های حاوی داده‌های حساس که نیاز به محافظت دارند. این سطح، سطح پیشنهادی برای بیشتر اپلیکیشن‌ها است.
- ASVS سطح ۳ (L3): برای بحرانی‌ترین اپلیکیشن‌ها مثل اپلیکیشن‌هایی که تراکنش‌های پرارزش انجام می‌دهند، حاوی داده‌های حساس پزشکی هستند، یا هر اپلیکیشنی که بالاترین سطح اعتماد را نیاز داشته باشد.

هر سطح از ASVS دارای لیستی از الزامات امنیتی است. هرکدام از این نیازها را می‌توان به قابلیت‌ها و ویژگی‌های امنیتی خاصی نگاشت کرد که باید توسط توسعه‌دهنده در نرم‌افزار ساخته شده باشد.

[13] Common Weakness Scoring System
[14] Agile



شکل۱- سطوح استاندارد OWASP - وارسی امنیت اپلیکیشن‌ها نسخه ۴

سطح ۱ تنها سطحی است که کاملاً به‌وسیله انسان می‌توان روی آن آزمون نفوذپذیری انجام داد. سایر سطح‌ها نیاز به دسترسی به مستندات، کد منبع، نحوه پیکربندی و افرادی که در فرآیند توسعه دست داشتند دارد. با این حال حتی اگر سطح ۱ امکان آزمون مدل جعبه سیاه (عدم دسترسی به کد منبع و یا مستندات) را بدهد، تضمین آن مؤثر نیست و باید متوقف شود. مهاجمان بدکار[۱۵] زمان زیادی دارند ولی بیشتر آزمون‌های نفوذپذیری نیز در طی چند هفته به پایان می‌رسند. مدافعان باید در زمانی معقول کنترل‌های امنیتی را بسازند و تمام ضعف‌ها را شناسایی و برطرف کنند و محافظت کنند، عاملان بدکار را تشخیص و به آن‌ها پاسخ مناسب دهند. عاملان بدکار بی‌نهایت زمان دارند و فقط به یک دفاع ناقص، یک ضعف یا یک مورد شناسایی نشده نیاز دارند تا موفق شوند. تست جعبه سیاه، که اغلب در انتهای توسعه انجام می‌شود، به سرعت انجام شده و یا به‌طور کلی اصلاً انجام نمی‌شود، به‌طور کامل قادر به مقابله با این عدم تقارن نیست.

در طول بیش از ۳۰ سال گذشته، آزمون جعبه سیاه بارها و بارها ثابت کرده است که مسائل امنیتی حساسی را پیدا نکرده که مستقیماً به شکاف های بزرگ امنیتی منجر شده است. ما به‌شدت استفاده از طیف گسترده‌ای از وارسی‌ها و ضمانت‌های امنیتی، شامل جایگزین کردن آزمون نفوذپذیری با آزمون نفوذ مبتنی بر کد منبع با دسترسی کامل به مستندات و توسعه‌دهندگان در طول فرایند توسعه، در سطح ۱ را تشویق می‌کنیم. حسابرسان مالی بدون دسترسی به مدارک مالی، تراکنش‌های نمونه و افرادی که کنترل این امور را داشته‌اند، قادر به حسابرسی نیستند. صنعت و دولت‌ها نیز باید خواستار این‌گونه استانداردها و شفافیت ها در زمینه مهندسی نرم‌افزار باشند.

ما به شدت از استفاده از ابزارهای امنیتی تشویق می کنیم، اما در طی خود فرایند توسعه، مانند ابزارهای DAST و SAST که به‌طور پیوسته در مراحل ساخت برنامه برای پیدا کردن مسائل امنیتی که هرگز نباید وجود داشته باشد، استفاده می شوند.

ابزارهای خودکار و اسکن‌های آنلاین بدون کمک انسانی قادر به تکمیل بیش از نیمی از ASVS نیستند. اگر یک تست خودکار همه‌جانبه برای هر ایجاد برنامه[۱۶] مورد نیاز باشد، ترکیبی از تست‌های واحد[۱۷] و یکپارچه‌سازی[۱۸] شخصی‌سازی‌شده، به‌همراه

[15] Malicious
[16] Build
[17] Unit test
[18] Integration test

اسکن‌های آنلاین استفاده می‌شود. اشکالات منطق کسب‌وکار[19] و کنترل دسترسی فقط به کمک انسان قابل انجام هستند. این‌ها باید به تست‌های واحد و یکپارچه تبدیل شوند.

نحوه استفاده از این استاندارد

یکی از بهترین راهکارها برای استفاده از استاندارد وارسی امنیت اپلیکیشن‌ها، استفاده از آن به‌عنوان یک طرح کلی برای ایجاد یک چک‌لیست جهت کدنویسی امن خاص برای برنامه، سکو[20] یا سازمان شما است. انتخاب بخش‌های مختلف ASVS بر اساس موارد استفاده خود، تمرکز شما را بر نیازهای امنیتی که برای پروژه‌ها و محیط شما مهم است افزایش می‌دهد.

سطح ۱ – گام‌های اول، خودکار، یا کل نمونه کارها

یک اپلیکیشن زمانی به سطح ۱ از ASVS دست می‌یابد که به اندازه کافی در برابر آسیب‌پذیری‌های امنیتی نرم‌افزار که به‌سادگی کشف می‌شوند، و در OWASP Top 10 و دیگر چک‌لیست‌های مشابه وجود دارد، مقاوم باشد.

سطح ۱ حداقلی است که تمام برنامه‌ها باید برای رسیدن به آن تلاش کنند. همچنین به‌عنوان اولین فاز در پروژه‌های چند فازی و یا زمانی‌که برنامه‌ها اطلاعات حساسی را پردازش نمی‌کنند مفید است و بنابراین به کنترل‌های دقیق و سخت سطح ۲ یا ۳ نیاز ندارند. کنترل‌های سطح ۱ را می‌توان به‌صورت خودکار با ابزارها یا به‌سادگی به روش دستی و بدون دسترسی به کد منبع بررسی نمود. ما سطح ۱ را حداقل مورد نیاز برای همه برنامه‌ها می‌دانیم.

تهدیداتی که به برنامه‌ها وارد می‌شود معمولاً از مهاجمانی است که از تکنیک‌های ساده و با تلاش کم برای شناسایی آسیب‌پذیری‌هایی که به‌سادگی می‌توان کشف و بهره‌برداری کرد، استفاده می‌کنند. این در تضاد با یک مهاجم مصمم است که انرژی خود را روی یک برنامه خاص متمرکز می‌کند. اگر داده‌های پردازش شده توسط برنامه شما پر ارزش و حساس باشند، پیشنهاد نمی‌شود که کار خود را در سطح ۱ متوقف کنید.

سطح۲ – بیشتر برنامه‌ها

یک اپلیکیشن زمانی به سطح ۲ از ASVS دست می‌یابد که به‌اندازه کافی در برابر آسیب‌پذیری‌های امنیتی امروزی مرتبط با نرم‌افزار، مقاوم باشد.

سطح ۲ تضمین می‌کند که کنترل‌های امنیتی در محل صحیح خود، مؤثر، و در داخل نرم‌افزار استفاده می‌شوند. سطح ۲ به‌طور معمول برای اپلیکیشن‌هایی مناسب است که تراکنش‌های مهم کسب‌وکار به کسب‌وکار[21] را به عهده دارند، از جمله آن‌هایی که اطلاعات سلامت و پزشکی را پردازش، کارکردهای حساس تجاری را پیاده‌سازی و یا سایر موارد حساس را پردازش می‌کنند و یا صنایعی که در آن بدون اشکال بودن یک جنبه بحرانی برای محافظت از کسب‌وکار است، مانند صنعت بازی برای جلوگیری از تقلب و هک.

---

[19] Bussines logic
[20] Platform
[21] Business-to-Business

تهدیدات به برنامه‌های سطح ۲ معمولاً مهاجمان ماهر و با انگیزه هستند که از ابزارها و تکنیک‌هایی که زیاد تمرین کرده‌اند و بسیار مؤثرند، در کشف و بهره‌برداری از نقاط ضعف در اپلیکیشن‌ها استفاده می‌کنند.

سطح ۳ – ارزش بالا، اطمینان بالا، امنیت بالا

ASVS سطح ۳ بالاترین سطح وارسی در ASVS است. این سطح معمولاً برای اپلیکیشن‌هایی مورد نیاز است که سطح قابل توجهی از وارسی امنیتی را نیاز دارند، مانند کاربردهایی که در امور نظامی، بهداشت و ایمنی، زیرساخت‌های حیاتی و غیره یافت می‌شود.

سازمان‌ها ممکن است سطح ۳ ASVS را برای اپلیکیشن‌هایی که عملکردهای حیاتی را انجام می‌دهند نیاز داشته باشند، در جایی که خطا به‌طور قابل‌توجهی می‌تواند بر عملیات سازمان تأثیر بگذارد و حتی قابلیت بقای آن را تهدید کند. رهنمودهای نمونه در مورد کاربرد ASVS سطح ۳ در زیر ارائه شده است. یک اپلیکیشن زمانی به سطح ۳ از ASVS دست می‌یابد که به‌اندازه کافی در برابر آسیب‌پذیری های امنیتی پیشرفته مقاوم باشد و همچنین اصول طراحی امنیتی خوب را داشته باشد.

یک برنامه کاربردی در سطح ۳ ASVS نیاز به تحلیل و یا معماری، برنامه‌نویسی و تست عمیق‌تری نسبت به سایر سطوح دارد. یک برنامه امن به‌طور معناداری ماژول‌بندی[۲۲] شده است (برای تسهیل انعطاف‌پذیری، مقیاس‌پذیری و مهم‌تر از همه لایه‌های امنیتی) و هر یک از ماژول‌ها (جداشده از طریق ارتباط شبکه و/یا فیزیکی) مسئولیت‌های امنیتی خود را بر عهده دارد (دفاع در عمق)، که باید به‌درستی مستندسازی شده باشد. این مسئولیت‌ها شامل کنترل‌هایی برای اطمینان از محرمانه بودن (مانند رمزنگاری)، یکپارچگی (مانند تراکنش‌ها، اعتبارسنجی ورودی)، در دسترس بودن (به‌عنوان مثال مدیریت بار به‌صورت مناسب)، احرازاصالت (حتی بین سیستم‌ها)، انکارناپذیری[۲۳]، بررسی مجوزها و حسابرسی (لاگ گرفتن)، می‌باشد.

استفاده از ASVS در عمل

تهدیدهای متفاوت انگیزه‌های متفاوتی دارند. بعضی از صنایع دارایی‌های[۲۴] اطلاعاتی و فن‌آوری منحصربه‌فرد، و یا الزامات انطباق با محدودیت‌های مختص دامنه خود را دارند.

سازمان‌ها به‌شدت تشویق می‌شوند که نقاط حساس کار خود را بررسی کنند و بر اساس آن حساسیت‌ها و الزامات کار خود، سطح مناسب ASVS را تعیین کنند.

ارزیابی و صدور گواهینامه

موضع OWASP در مورد گواهینامه ها و نماد های اعتماد ASVS

OWASP، به‌عنوان یک سازمان غیرفروشنده و غیرانتفاعی، در حال حاضر هیچ فروشنده، تأییدکننده و یا نرم‌افزاری را تأیید نمی‌کند.

---

[22] Modularized
[23] non-repudiation
[24] Asset

همه اظهارنامه‌های اطمینان، نمادهای اعتماد و یا گواهی‌نامه‌ها به‌طور رسمی مورد بررسی، ثبت و یا تأیید OWASP نیستند، بنابراین یک سازمان باید در اعتماد کردن به اشخاص ثالث که ادعای گواهی‌نامه ASVS را دارند محتاط عمل کند.

این نباید سازمان‌ها را از دادن سرویس‌های تضمین و اطمینان باز دارد، بلکه این سازمان‌ها فقط نباید ادعای گواهی رسمی OWASP را داشته باشند.

راهنمایی برای سازمان‌های گواهی‌دهنده

ASVS می‌تواند به‌عنوان یک منبع آزاد برای وارسی اپلیکیشن‌ها استفاده شود، شامل دسترسی آزاد و بدون محدودیت به منابع کلیدی مانند معماران و توسعه‌دهندگان، اسناد پروژه، کد منبع و همچنین دسترسی مجاز به سیستم‌های تست (از جمله دسترسی به یک یا چند حساب در هر نقش)، به‌ویژه برای سطح‌های ۲ و۳.

از لحاظ تاریخی، آزمون نفوذپذیری و بررسی امن بودن کد، فقط شامل موارد نقض بوده که به این معنی است که تنها تست‌های شکست‌خورده در گزارش نهایی نشان داده شده است. سازمان‌های گواهی‌دهنده باید در هر گزارش موارد پیش رو را ذکر کنند: محدوده وارسی (به‌ویژه اگر یک جزء کلیدی خارج از حوزه باشد، مانند احرازاصالت با SSO)، خلاصه‌ای از یافته‌های وارسی، از جمله تست‌های قبول‌شده و شکست‌خورده، با دلایل روشن جهت رفع مشکل تست‌های شکست‌خورده.

برخی الزامات برای وارسی ممکن است در مورد اپلیکیشن‌های تحت آزمایش قابل اجرا نباشد. به‌عنوان مثال، اگر شما یک لایه API بدون‌حالت[۲۵] داشته باشید اما بخش‌های سمت مشتری را پیاده‌سازی نکرده باشید، بسیاری از الزامات در مدیریت نشست‌ها در بخش V3 مستقیماً قابل کابرد نیستند. در چنین مواردی یک سازمان گواهی‌دهنده ممکن است هنوز هم ادعا کند که انطباق کامل با ASVS دارد، اما در هر گزارش باید به وضوح دلیل عدم استفاده از چنین الزامات وارسی را نشان دهد.

نگه‌داشتن دقیق صفحه‌های کاری، تصاویر یا فیلم‌ها، اسکریپت‌ها برای بهره‌برداری[۲۶] مکرر و قابل اعتماد از یک مورد، و گزارش‌های الکترونیکی تست‌ها، مانند رهگیری[۲۷] لاگ‌های پروکسی و یادداشت‌های مرتبط مانند یک لیست پاک‌سازی[۲۸]، به‌عنوان یک تمرین[۲۹] در صنعت استاندارد تلقی می‌شود و به‌عنوان مدرک برای اثبات یافته‌ها به توسعه‌دهنده‌های شکاک استفاده می‌شود. کافی نیست که به‌راحتی فقط یک ابزار را اجرا کنید و در مورد شکست‌ها گزارش دهید؛ این اصلاً شواهد کافی را ندارد که نشان دهد تمامی موضوعات در سطح صدور گواهینامه مورد آزمایش و تست قرار گرفته‌اند. در صورت بروز اختلاف، شواهد کافی برای اثبات این‌که هر بخش به‌طور کامل تست شده است باید وجود داشته باشد.

روش آزمون

سازمان‌های صدور گواهی‌نامه آزاد هستند که روش آزمون مناسب را انتخاب کنند، اما باید آن‌ها را در گزارش ذکر کنند.

---

[25] Stateless API Layer
[26] Exploits
[27] Intercepting
[28] Cleanup
[29] Practice

بسته به اپلیکیشن تحت آزمایش و الزامات وارسی، ممکن است از روش‌های مختلف آزمایش استفاده شود که البته نتایج یکسانی دارند. برای مثال، اعتبارسنجی میزان مؤثر بودن سازوکار تأیید ورودی‌های یک اپلیکیشن، می‌تواند با استفاده از آزمون نفوذپذیری دستی یا با استفاده از تجزیه و تحلیل کد منبع انجام شود.

نقش ابزارهای خودکار آزمون‌های امنیتی

تا حد امکان تشویق می‌شود که از ابزارهای آزمون نفوذپذیری خودکار برای پوشش حداکثری استفاده شود.

امکان اجرای تمامی وارسی‌های ASVS با استفاده از ابزارهای آزمون نفوذپذیری خودکار به‌تنهایی امکان‌پذیر نیست. در حالی که بیشتر موارد در سطح ۱ را می‌توان با استفاده از تست‌های خودکار انجام داد، در نگاه کلی به سه سطح، بیشتر موارد به‌وسیله ابزارهای خودکار قابل تست نیستند.

البته توجه داشته باشید که مرز بین تست خودکار و تست دستی، با رشد صنعت امنیت اپلیکیشن‌ها در حال کمرنگ‌تر شدن است. ابزارهای خودکار اغلب به‌صورت دستی توسط کارشناسان خبره تنظیم می‌شوند و افراد تست‌کننده اغلب از انواع وسیعی از ابزارهای خودکار استفاده می‌کنند.

نقش آزمون نفوذپذیری

در نسخه ۴.۰ تصمیم گرفتیم سطح ۱ به‌طور کامل بدون دسترسی به کد منبع، مستندات و یا توسعه‌دهندگان انجام شود. هرچند، انجام تست بدون دسترسی به اطلاعات لازم یک روش ایده‌آل وارسی امنیت نیست، به‌دلیل عدم امکان بررسی کد منبع، شناسایی تهدیدها و کنترل‌های فراموش‌شده و اجرای یک آزمایش دقیق‌تر در مدت زمان کوتاه‌تر.

در هنگام اجرای یک ارزیابی سطح ۲ یا ۳، در صورت امکان، دسترسی به برنامه‌نویس‌ها، اسناد، کد و دسترسی به یک برنامه آزمایشی با داده‌های غیرساختگی ضروری است. آزمون‌های نفوذپذیری انجام‌شده در این سطوح نیاز به این سطح از دسترسی‌ها دارد، که ما آن را "بررسی هیبریدی" یا "آزمون نفوذپذیری هیبریدی" می‌نامیم.

کاربردهای دیگر ASVS

به غیراز استفاده برای تعیین میزان امنیت یک اپلیکیشن، ASVS کاربرد بالقوه دیگری نیز دارد.

به‌عنوان یک راهنمای تفصیلی معماری امنیتی[30]

یکی از کاربردهای رایج برای ASVS، به‌عنوان یک منبع برای معماران امنیتی است. معماری امنیت کسب‌وکار اعمال شده شروود[31] (SABSA) کمبود اطلاعات زیادی برای تکمیل یک معماری امنیت نرم‌افزاری تمام و کمال دارد. ASVS می‌تواند برای پر کردن این شکاف‌ها با اجازه‌دادن به معماران امنیتی در انتخاب کنترل‌های بهتر برای مشکلات رایج مانند الگوهای محافظت از داده‌ها و راهبردهای سنجش اعتبار ورودی، این کمبود را جبران کند.

---

[30] Security Architecture
[31] Sherwood Applied Business Security Architecture

جایگزینی برای چک‌لیست‌های برنامه‌نویسی امن موجود

بسیاری از سازمان‌ها می‌توانند بااستفاده از ASVS یکی از سه سطح را انتخاب کنند و یا با ایجاد انشعابی[۳۲] از ASVS آن‌چه را که برای هر سطح ریسک برنامه در یک دامنه مخصوص نیاز دارند را تغییر بدهند. ما تا زمانی تشویق به این نوع انشعاب دادن می‌کنیم که قابل ردیابی باشد.

راهنمایی برای تست واحد و تست‌های یکپارچه خودکارسازی شده

ASVS، به استثنای الزامات معماری و کدهای مخرب، به‌گونه‌ای طراحی شده است که قابلیت تست بالایی داشته باشد. با ساختن تست‌های واحد و تست‌های یکپارچه برای فازکردن[33]های مرتبط و موارد سوءاستفاده، برنامه خود تقریباً برای هر ساخت برنامه[۳۴] خودتأیید[۳۵] میشود. به‌عنوان مثال، می‌توان تست‌های اضافی برای مجموعه تست[۳۶] که برای کنترلرهای لاگین استفاده می‌شود، طراحی کرد، مثل تست کردن پارامتر نام کاربری برای نام‌های کاربری پیش‌فرض متداول، شمارش حساب‌های کاربری، جست‌وجوی فراگیر[۳۷]، تزریق LDAP، SQL، XSS و غیره. به‌طور مشابه یک تست روی پارامتر نام کاربری باید شامل بررسی روی گذرواژه‌ها متداول، طول گذرواژه، تزریق بایت Null، حذف پارامتر، XSS و موارد بیشتر باشد.

برای آموزش توسعه امن

ASVS را برای تعریف مشخصات نرم‌افزار امن هم می‌تواند مورد استفاده قرار بگیرد. بسیاری از دوره‌های برنامه‌نویسی امن در واقع دوره‌های هک قانونمند به‌همراه اندکی از نکات برنامه‌نویسی است. الزاماً این دوره‌ها به توسعه‌دهندگان برای نوشتن کدهای امن کمک نمی‌کنند. دوره‌های توسعه امن می‌توانند به‌جای تمرکز روی ده مورد منفی برتری که نباید انجام شود با استفاده از ASVS روی کنترل‌های پیشگیرانه تمرکز کنند.

به‌عنوان یک محرک برای امنیت برنامه‌های چابک[۳۸]

ASVS را می‌توان در فرآیند توسعه چابک به‌عنوان یک چارچوب برای تعریف وظایف[۳۹] خاصی که برای داشتن محصولی امن لازم است پیاده‌سازی شوند، استفاده کرد. یک رویکرد می‌تواند به این‌صورت باشد: با شروع از سطح ۱، سیستم یا برنامه کاربردی خاص خود را با توجه به الزامات ASVS، برای سطح مشخص شده، وارسی کنید و این که چه کنترل‌هایی وجود ندارند را پیدا کرده و آن‌ها را به کارهای ناتمام اضافه کنید. این امر به اولویت‌بندی و مرتب‌کردن وظایف و نمایان ساختن موضوع امنیت در فرآیند توسعه چابک کمک می‌کند. از این چارچوب برای اولویت‌دادن به حسابرسی و مرور وظایف نیز می‌توان استفاده کرد. مثلاً زمانی‌که یک نیازمندی

---

[32] Fork
[33] Fuzz
[34] Build
[35] Self-verified
[36] Test suite
[37] Brute forcing
[38] Agile Applications
[39] Task

ASVS خاص می‌تواند یک محرک برای مرور، اصلاح یا بازرسی برای عضوی از تیم باشد و باید به‌عنوان "بدهی" در کارهای ناتمام قابل رؤیت باشد که بالاخره باید انجام شود.

چارچوبی برای راهنمای خرید نرم‌افزار امن

ASVS چارچوب مناسبی برای کمک به خرید نرم‌افزار یا سرویس‌های توسعه سفارشی است. خریدار می‌تواند به‌راحتی مشخص کند که نرم‌افزاری که می‌خواهد تهیه کند باید الزامات سطح X از ASVS را فراهم کند و از فروشنده درخواست کند تا اثبات نماید که نرم‌افزار سطح X از ASVS را پشتیبانی می‌کند. این امر هنگامی‌که با OWASP Secure Software Contract Annex ترکیب شود به‌خوبی کار می‌کند.

V1: الزامات معماری، طراحی و مدل‌سازی تهدید

هدف کنترل

معماری امنیت تبدیل به یک هنر گمشده در بسیاری از سازمان‌ها شده است. روزهای معماری سازمانی[40] در عصر DevSecOps گذشته‌اند. حوزه امنیت نرم‌افزار باید عقب‌افتادگی خود را جبران کرده و اصول امنیت چابک[41] را، در حالی که اصول معماری امن را به متخصصان نرم‌افزار باز معرفی می‌کند، اتخاذ کند. معماری یک پیاده‌سازی نیست، یک راه فکر کردن درباره یک مشکل است که چندین جواب بالقوه متفاوت دارد و یک جواب "درست" ندارد. بیشتر مواقع امنیت به این شکل دیده شده که غیرمنعطف است و خواستار این است که توسعه‌دهندگان کد خود را به یک شکل خاص تعمیر کنند، در حالی‌که توسعه‌دهندگان خود ممکن است راه بهتری برای حل مشکل بشناسند. برای معماری یک راه‌حل یکتا و ساده وجود ندارد و تظاهر به خلاف این امر یک بدخدمتی به حوزه مهندسی نرم‌افزار است.

پیاده‌سازی خاصی از یک برنامه کاربردی وب محتمل است که به‌طور پیوسته در زمان حیاتش تجدیدنظر شود، ولی معماری کلی آن به‌ندرت تغییر می‌کند، هرچند به‌آرامی تکامل یابد. معماری امنیت نیز دقیقاً همین‌طور است – ما امروز به احرازاصالت نیاز داریم، فردا هم نیاز خواهیم داشت، ۵ سال آینده هم نیاز خواهیم داشت. اگر امروز تصمیمات صحیحی بگیریم می‌توانیم مقدار زیادی در زمان، هزینه و تلاش صرفه‌جویی کنیم، اگر راه‌حل‌های سازگار با استفاده مجدد[42] را انتخاب کنیم. برای مثال در دهه قبل، احرازاصالت چندفاکتوری به‌ندرت پیاده‌سازی می‌شد.

اگر توسعه‌دهندگان در یک مدل تأمین هویت[43] واحد و امن مثل SAML federated identity، سرمایه‌گذاری کرده بودند، مدل تأمین هویت می‌توانست به‌روزرسانی شود تا الزامات های جدید مثل سازگاری با NIST 800-63 را در سیستم وارد کند، در حالی‌که رابط[44]های برنامه کاربردی اصلی را تغییر ندهد. اگر بسیاری از برنامه‌های کاربردی معماری امنیتی یکسانی داشتند، همه از این به‌روزرسانی به شکل یکجا سود می‌بردند. ولی SAML همیشه بهترین و مناسب‌ترین راهکار احرازاصالت باقی نخواهد ماند - ممکن است لازم شود با راهکارهای جدید، با تغییر الزامات عوض شود. تغییرات این چنینی، بدون معماری امنیتی، به‌دلیل لزوم بازنویسی کامل، بسیار پیچیده و پرهزینه و یا به کلی غیرممکن هستند.

در این فصل ASVS ویژگی‌های اصلی هر معماری امنیتی صحیح را پوشش می‌دهد: دردسترس‌بودن، محرمانگی، یکپارچگی فرآیند، انکارناپذیری و حریم خصوصی. هر یک از این اصول امنیتی باید برای هر برنامه کاربردی "از اول ساخته شده" و ذاتی باشد. ضروری است که با شروع کردن از توانمندسازی توسعه‌دهندگان با چک‌لیست‌های کدنویسی امن، مانیتورینگ و آموزش، کدنویسی و تست، ایجاد برنامه[45]، مستقرسازی[46]، پیکربندی[47]، و عملیات[48] و در انتها انجام یک تست مستقل، ایجاد اطمینان از این‌که تمام کنترل‌های امنیتی حاضر و در

---

[40] Enterprise Architecture
[41] Agile
[42] Re-use
[43] Identity provider model
[44] Interface
[45] Build
[46] Deployment
[47] Configuration
[48] Operation

حال کار هستند را به تعجیل انداخت. مرحله آخر قبلاً تمام کاری بود که ما در صنعت انجام می‌دادیم ولی دیگر کافی نیست، از آن‌جایی‌که توسعه‌دهندگان ده‌ها و یا صدها بار در روز کد به محصول نهایی می‌افزایند. متخصصان امنیت نرم‌افزار باید تکنیک‌های چابک خود را به‌روز نگه دارند، که به این معنی است که باید ابزارهای توسعه‌دهندگان را اتخاذ کنند، یاد بگیرند کد بنویسند و با توسعه‌دهندگان کار کنند، به جای این‌که پروژه را زمانی‌که بقیه چند ماه است که از آن گذشته‌اند، نقد کنند.

V1.1: الزامات چرخه حیات توسعه امن نرم‌افزار

#	توضیح	L1	L2	L3	CWE
۱.۱.۱	وارسی کنید که از چرخه حیات توسعه امن و نرم‌افزار، که امنیت را در تمام مراحل توسعه در دستور کار قرار داده است، استفاده شده است.		✓	✓	
۱.۱.۲	وارسی کنید که از مدل‌سازی تهدید برای هر تغییر طراحی و یا برنامه‌ریزی اسپرینت[۴۹] برای مشخص‌کردن تهدیدها، برنامه ریختن برای اقدامات متقابل، تسهیل پاسخ‌گویی به خطرها و راهنمایی تست‌های امنیتی استفاده شده است.		✓	✓	۱۰۵۳
۱.۱.۳	وارسی کنید که ویژگی‌های کاربران دارای محدودیت‌های امنیتی کارا هستند. مثلاً این‌که به‌عنوان یک کاربر من باید بتوانم پروفایل خود را ویرایش کرده و نباید بتوانم پروفایل فرد دیگری را دیده و یا تغییر دهم.		✓	✓	۱۱۱۰
۱.۱.۴	تمام مستندسازی‌ها و دلیل‌آوری‌های محدوده اعتماد اپلیکیشن، اجزاء و جریان داده مهم اپلیکیشن را وارسی کنید.		✓	✓	۱۰۵۹
۱.۱.۵	تعاریف و تحلیل‌های امنیتی معماری سطح بالای اپلیکیشن و تمامی دستگاه‌های متصل به آن را وارسی کنید.		✓	✓	۱۰۵۹
۱.۱.۶	پیاده‌سازی کنترل‌های مرکزی، ساده، بررسی‌شده، امن و قابل استفاده مجدد، برای جلوگیری از کنترل‌های تکراری، گمشده، غیرمؤثر و یا غیرامن، را وارسی کنید.		✓	✓	۶۳۷
۱.۱.۷	دردسترس‌بودن چک‌لیست‌های کدنویسی امن، الزامات امنیتی، خط‌مشی‌ها و سیاست‌ها را برای تمامی توسعه‌دهندگان و یا تست‌کنندگان وارسی کنید.		✓	✓	۶۳۷

[49] Sprint Planning

V1.2: الزامات معماری احرازاصالت

هنگام طراحی احرازاصالت، مهم نیست که شما یک احرازاصالت قوی چندفاکتوری تقویت‌شده با سخت‌افزار دارید، در صورتی‌که یک مهاجم بتواند با تماس گرفتن مرکز تماس و جواب دادن چند سوال کلی حساب را بازنشانی[50] کند. هنگام اثبات هویت تمام مسیرها باید به یک اندازه قدرت داشته باشند.

#	توضیح	L1	L2	L3	CWE
۱.۲.۱	استفاده یک حساب سیستمی یکتا با اختیار کم در یک سیستم‌عامل برای تمام اجزای برنامه کاربردی، سرویس‌ها و سرویس‌دهنده‌ها را وارسی کنید.		✓	✓	۲۵۰
۱.۲.۲	وارسی کنید که ارتباط بین اجزای برنامه کاربردی که شامل APIها نیز می‌شود، میان‌افزارها و لایه‌های داده، احرازاصالت شده‌اند. اجزا باید کمترین اختیارات لازم را داشته باشند.		✓	✓	۳۰۶
۱.۲.۳	وارسی کنید که برنامه کاربردی تنها از یک سازوکار بررسی‌شده که به امن‌بودن شناخته شده، قابل گسترش برای احرازاصالت قوی است و به حد لازم از لاگ کردن و مانیتورینگ کافی بهره‌مند است، استفاده می‌کند.		✓	✓	۳۰۶
۱.۲.۴	وارسی کنید که تمام مسیرهای احرازاصالت و APIهای مدیریت هویت، کنترل‌های امنیتی به شکل منسجم با قدرت‌های برابر پیاده‌سازی شده‌اند و هیچ مسیر ضعیف دیگری در برنامه وجود ندارد.		✓	✓	۳۰۶

V1.3: الزامات معماری و مدیریت نشست

این بخش یک نگهدارنده مکان[51] برای الزامات معماری در آینده است.

V1.4: الزامات معماری کنترل دسترسی

#	توضیح	L1	L2	L3	CWE

[50] Reset
[51] Placeholder

۱.۴.۱	وارسی کنید که نقاط اجرایی معتمد مثل کنترل دسترسی در درواز ه[۵۲] سرویس‌دهنده‌ها و توابع سرویس‌دهنده کنترل دسترسی را انجام می دهند. هیچ وقت کنترل دسترسی را در سمت مشتری انجام ندهید.		✓	✓	۶۰۲
۱.۴.۲	وارسی کنید که راهکار کنترل دسترسی برنامه کاربردی به اندازه کافی انعطاف‌پذیر است تا الزامات برنامه کاربردی را تأمین کند.		✓	✓	۲۸۴
۱.۴.۳	وارسی کنید که قاعده کمترین امتیاز در توابع، فایل‌های داده urlها، کنترل‌ها، سرویس‌ها و دیگر منابع اجرا شده است این به‌منظور جلوگیری از جاسوسی کردن[۵۳] و بالابردن اختیارات است.		✓	✓	۲۷۲
۱.۴.۴	وارسی کنید که برنامه کاربردی از یک سازوکار کنترل دسترسی واحد و به‌خوبی بررسی‌شده، برای دسترسی به منابع و داده‌های محافظت شده استفاده کرده است. تمام درخواست‌ها باید از این یک سازوکار گذر کرده تا از کپی کردن و مسیرهای جایگزین اجتناب شود.		✓	✓	۲۸۴
۱.۴.۵	وارسی کنید که هنگامی‌که کد مجوز کاربر برای یک آیتم ویژگی/داده را بررسی می‌کند، نه فقط نقش آن‌ها را، از کنترل دسترسی مبتنی بر ویژگی یا صفت استفاده شده است. مجوزها هنوز هم باید بر اساس نقش‌ها داده شوند.				۲۷۵

V1.5: الزامات معماری ورودی و خروجی

در نسخه 4.0 از اصطلاح "server-side" به‌عنوان یک اصطلاح محدوده اعتماد لودشده گذر کرده‌ایم. محدوده اعتماد هنوز نگران‌کننده است – زیرا موجب می‌شود که تصمیم‌گیری در دستگاه‌های مشتری و مرورگرهای غیرقابل اعتماد قابل دور زدن باشد. هرچند در مسیر اصلی استقرار معماری‌های امروزه، نقطه اجبار مورد اعتماد[۵۴]، به‌طور چشمگیری تغییر کرده است. بنابراین جایی که اصطلاح "trusted service layer" در ASVS استفاده شده است، جدا از محل آن، مثل میکروسرویس‌ها، APIهای بدون سرویس‌دهنده[۵۵]، سمت سرویس‌دهنده، یک API معتمد بر روی دستگاه‌های کاربر که بوت امن دارد، APIهای همکاران و یا خارجی و غیره، منظور هر نقطه اجبار مورد اعتماد است.

[52] Gateway
[53] Spoofing
[54] Trusted enforcement point
[55] Server-less

#	توضیح	L1	L2	L3	CWE
۱.۵.۱	وارسی کنید که آیا الزامات ورودی و خروجی به وضوح تعریف می‌کند که چگونه داده بر اساس نوع محتوا، قوانین برنامه کاربردی، آیین‌نامه‌ها، و تبعیت از دیگر سیاست‌ها، پردازش شود.		✓	✓	۱۰۲۹
۱.۵.۲	وارسی کنید که serialization هنگام ارتباط با مشتری‌های غیرقابل اعتماد انجام نمی‌شود. اگر این امکان‌پذیر نیست، مطمئن شوید که کنترل‌های یکپارچگی مناسبی (و احتمالاً رمزنگاری، اگر داده حساسی ارسال می‌شود) اجرا شده‌اند تا از حملات deserialization مثل تزریق شی[۵۶] جلوگیری شود.		✓	✓	۵۰۲
۱.۵.۳	وارسی کنید که اعتبارسنجی داده در یک لایه سرویس معتمد انجام می‌شود.		✓	✓	۶۰۲
۱.۵.۴	وارسی کنید که انکدسازی خروجی کنار و یا نزدیک مفسری که برای آن در نظر گرفته شده است، انجام می‌شود.		✓	✓	۱۱۶

V1.6: الزامات معماری رمزنگاری

برنامه‌های کاربردی لازم است که با معماری قوی رمزنگاری طراحی شوند تا از دارایی‌های داده‌ای خود، بر اساس طبقه‌بندی آن‌ها، محافظت شود. رمزنگاری همه‌چیز افراط است، رمز نکردن هیچ چیزی قانوناً سهل‌انگاری است. یک تعادل باید وجود داشته باشد که معمولاً در هنگام طراحی سطح بالا و یا معماری، طراحی اسپرینت[۵۷] و یا اسپایک‌های معماری[۵۸]، رخ می‌دهد. طراحی رمزنگاری و یا تکمیل آن، همان‌طور که جلو می‌روید، به شکل غیرقابل اجتنابی برای پیاده‌سازی امن بسیار پرهزینه‌تر خواهد بود نسبت به این‌که از اول در سیستم قرار می‌گرفت.

الزامات معماری برای کل پایه‌ی کد ذاتی هستند و بنابراین برای انجام تست واحد و تست یکپارچگی، بسیار سخت هستند. الزامات معماری، نیازمند ملاحظه در استاندارد کدنویسی، در تمام فاز کدنویسی است و باید هنگام معماری امن، بازبینی‌های همکاران یا بازبینی کد و یا عقب افتادگی‌ها بازبینی شود.

#	توضیح	L1	L2	L3	CWE

[56] Object Injection
[57] Design Sprint
[58] Architectural Spike

۱.۶.۱	وارسی کنید که یک سیاست آشکار برای تبادل کلید رمزنگاری وجود دارد و این‌که چرخه حیات کلید رمزنگاری از یک استاندارد مدیریت کلید مثل NIST SP 800-57 پیروی می‌کند.		✓	✓	۳۲۰
۱.۶.۲	وارسی کنید که مصرف‌کنندگان سرویس‌های رمزنگاری از جنس کلید و دیگر رازها با استفاده از یک مخزن کلید و یا جایگزین های API محافظت می‌کنند.		✓	✓	۳۲۰
۱.۶.۳	وارسی کنید که تمام کلیدها و رمزها، قابل جایگزینی بوده و قسمتی از یک فرآیند خوب توصیف شده، برای رمزنگاری داده‌های حساس هستند.		✓	✓	۳۲۰
۱.۶.۴	وارسی کنید که کلیدهای متقارن، رمزها، و رازهای API که توسط مشتری تولید و یا مورد استفاده قرار گرفته‌اند، فقط برای محافظت از رازهای کم‌خطر مورد استفاده قرار گرفته‌اند. مثل رمزنگاری حافظه محلی، و یا برای استفاده زودگذر موقت مثل مبهم‌سازی پارامترها. اشتراک‌گذاری رازها با مشتری‌ها از لحاظ معماری یک معادل واضح هست و باید به‌طور مشابه با آن برخورد شود.		✓	✓	۳۲۰

V1.7: الزامات معماری خطاها، انجام لاگ و حسابرسی

#	توضیح	L1	L2	L3	CWE
۱.۷.۱	وارسی کنید که یک فرمت لاگ کردن و رویکرد مشترک در کل سیستم استفاده شده است.		✓	✓	۱۰۰۹
۱.۷.۲	وارسی کنید که لاگ‌ها به شکل امن ترجیحاً به یک سیستم خارجی برای تحلیل، تشخیص، هشدار و تشدید، ارسال می‌شوند.		✓	✓	

V1.8: الزامات معماری حفاظت از داده و حریم خصوصی

#	توضیح	L1	L2	L3	CWE
۱.۸.۱	وارسی کنید که تمامی داده‌های حساس تشخیص داده شده و به سطوح محافظتی طبقه‌بندی شده‌اند.		✓	✓	

۱.۸.۲	وارسی کنید که سطوح محافظت یک مجموعه الزامات محافظتی مرتبط دارند، مثل الزامات رمزنگاری، الزامات یکپارچگی، الزامات نگه‌داری، حریم خصوصی و دیگر محرمانگی‌ها، که به این معماری اعمال شده‌اند.		✓	✓	

V1.9: الزامات معماری ارتباطات

#	توضیح	L1	L2	L3	CWE
۱.۹.۱	وارسی کنید که برنامه کاربردی ارتباطات بین اجزا را رمزنگاری می‌کند. مخصوصاً زمانی‌که این اجزا در محفظه‌ها، سیستم‌ها، سایت‌ها و یا تأمین‌کنندگان رایانش ابری مختلف هستند.		✓	✓	۳۱۹
۱.۹.۲	وارسی کنید که اجزای برنامه کاربردی صحت هویت هر طرف در هر اتصال را بررسی می‌کنند تا از حمله مرد میانی جلوگیری شود. برای مثال، اجزای برنامه کاربردی باید گواهی‌نامه‌های TLS و زنجیره‌ها را بررسی کنند.		✓	✓	۲۹۵

V1.10: الزامات معماری بدافزارها

#	توضیح	L1	L2	L3	CWE
۱.۱۰.۱	وارسی کنید که یک سیستم بررسی کد منبع استفاده شده است، با روال‌هایی برای اطمینان از این‌که check-inها همراه با issueها هستند یا بلیت را تغییر می‌دهند. سیستم بررسی کد منبع باید کنترل دسترسی و کاربران قابل شناسایی داشته باشد که هر تغییر قابل ردیابی باشد.		✓	✓	۲۸۴

V1.11: الزامات معماری منطق کسب‌وکار[۵۹]

#	توضیح	L1	L2	L3	CWE

[59] Business logic

۱.۱۱.۱	تعاریف و مستندات تمامی اجزای برنامه کاربردی از لحاظ قابلیت‌های کارکردی و امنیتی که ارائه می‌کنند را وارسی کنید.		✓	✓	
۱.۱۱.۲	وارسی کنید که تمامی جریان‌های ارزشمند منطق کسب‌وکار مثل احرازاصالت، مدیریت نشست و کنترل دسترسی‌ها، وضعیت‌های غیرهمزمان[۶۰] را به اشتراک نمی‌گذارند.		✓	✓	
۱.۱۱.۳	وارسی کنید که تمامی جریان‌های ارزشمند منطق کسب‌وکار مثل احرازاصالت، مدیریت نشست و کنترل دسترسی‌ها از لحاظ همروندی امن هستند[۶۱] و در مقابل شرایط مسابقه‌ای[۶۲] "زمان بررسی و زمان استفاده" مقاوم هستند.				

V1.12: الزامات معماری بارگذاری فایل امن

#	توضیح	L1	L2	L3	CWE
۱.۱۲.۱	وارسی کنید که فایل‌ها خارج از ریشه‌ی وب ذخیره شده‌اند.		✓	✓	
۱.۱۲.۲	وارسی کنید که فایل‌های بارگذاری‌شده توسط کاربر - در صورتی که نیاز است توسط برنامه کاربردی نمایش داده شده و یا دانلود شود – از طریق octet stream و یا یک دامنه نامربوط مثل یک فضای ذخیره سازی ابری[۶۳]، خدمت‌رسانی[۶۴] می‌شوند. یک سیاست امنیت محتوای مناسب برای کاهش خطر الگوی حمله xss و یا حملات دیگر ناشی از فایل بارگذاری شده را پیاده‌سازی کنید.		✓	✓	

V1.13: الزامات معماری APIها

این یک نگهدارنده مکان[۶۵] برای الزامات معماری در آینده است.

[60] Unsynchronized state
[61] Thread safe
[62] Race condition
[63] Cloud file storage bucket
[64] Serve
[65] Placeholder

V1.14: الزامات معماری تنظیمات

#	توضیح	L1	L2	L3	CWE
۱.۱۴.۱	وارسی کنید که تفکیک اجزا با سطوح مختلف اعتماد از طریق کنترل‌های امنیتی به‌خوبی تعریف‌شده، مثل قوانین دیواره آتش، دروازه‌های API، پروکسی‌های معکوس[66]،کنترل‌های گروه‌های امنیتی برای رایانش ابری و یا راهکارهای مشابه، انجام شده است.		✓	✓	
۱.۱۴.۲	وارسی کنید که اگر فایل‌های باینری را در دستگاه‌های غیرقابل اعتماد مستقر می‌کنید، حتماً از امضاهای دودویی، اتصالات امن و نقاط انتهایی[۶۷] تأیید شده استفاده می‌کنید.		✓	✓	
۱.۱۴.۳	وارسی کنید که خط لوله ایجاد برنامه[۶۸] برای استفاده از اجزای تاریخ گذشته و یا ناامن هشدار داده و اقدامات لازم را انجام می‌دهد.		✓	✓	
۱.۱۴.۴	وارسی کنید که خط لوله ایجاد برنامه، شامل یک مرحله خودکار ایجاد برنامه برای تأیید استقرار امن یک برنامه کاربردی است. مخصوصاً اگر برنامه کاربردی نرم‌افزار محور[۶۹] است، مثل اسکریپت‌های ایجاد برنامه در محیط‌های ابری.		✓	✓	
۱.۱۴.۵	وارسی کنید که استقرار برنامه کاربردی به‌طور مناسب لایه شبکه را سندباکس‌سازی، محفظه‌سازی[۷۰] و یا ایزوله می‌کند تا حمله مهاجمان به دیگر برنامه‌های کاربردی، مخصوصاً زمانی‌که آن‌ها کارهای حساس و یا خطرناکی مثل deserialization انجام می‌دهند، را به تأخیر انداخته و یا تشخیص دهد.		✓	✓	
۱.۱۴.۶	وارسی کنید که برنامه کاربردی از فن‌آوری‌های غیرامن، منسوخ و یا بدون پشتیبانی در سمت مشتری مثل پلاگین‌های NSAPI، Flash، Shockwave، ActiveX، Silverlight، NACL و یا اپلت‌های جاوای سمت مشتری استفاده نمی‌کند.		✓	✓	

منابع

برای اطلاعات بیشتر به منابع زیر رجوع شود:

[66] Reverse Proxies
[67] End point
[68] Build pipeline
[69] Software defined
[70] Containerize

- OWASP Threat Modeling Cheat Sheet
- OWASP Attack Surface Analysis Cheat Sheet
- OWASP Threat modeling
- OWASP Secure SDLC Cheat Sheet
- Microsoft SDL
- NIST SP 800-57

V2: الزامات وارسی احرازاصالت

هدف کنترل

احرازاصالت هنر تأییدکردن یک فرد (یا چیزی) به‌عنوان یک فرد (یا یک چیز) معتبر است و همچنین بررسی این‌که ادعاهای یک فرد یا یک دستگاه صحیح، مقاوم در برابر جعل هویت است و مانع از بازیابی یا ردیابی گذرواژه می‌شود. هنگامی‌که ASVS برای اولین بار منتشر شد، نام کاربری و گذرواژه رایج ترین نوع احرازاصالت (به جز در سیستم هایی با امنیت بسیار بالا) بود. احرازاصالت چندعامله[۷۱] (MFA) معمولاً در محافل امنیتی پذیرفته شده است اما به‌ندرت در جای دیگر موردنیاز است. به‌عنوان مثال، NIST 800-63 نام کاربری و اعتبارسنجی مبتنی بر دانش[۷۲] (KBA) را به‌عنوان اطلاعات عمومی، هشدارهای پیامک و ایمیل به‌عنوان انواع اعتبارسنج‌های "محدودشده" و گذرواژه‌ها را به‌عنوان از پیش لو رفته در نظر می‌گیرد. در واقع، احرازاصالت کنندگان مبتنی بر دانش، بازیابی بر مبنای پیامک و ایمیل، تاریخچه گذرواژه، افزودن پیچیدگی و تغییر دوره‌ای گذرواژه را بی‌فایده در نظر می‌گیرد. این کنترل‌ها همیشه غیرکمک‌کننده بوده‌اند زیرا اغلب کاربران را مجبور می‌کند تا هر چند ماه یکبار گذرواژه‌ها ضعیفی را ارائه دهند، اما با لو رفتن بیش از ۵ میلیارد نام کاربری و گذرواژه، دیگر نمی‌توان از این روش استفاده کرد.

از تمام بخش‌های ASVS، فصل احرازاصالت و مدیریت جلسه بیشترین تغییر را داشته است. پذیرش یک تمرین[۷۳] کارآمد و مبتنی بر شواهد پیشرو برای بسیاری مشکل خواهد بود و این امری طبیعی است. ما باید اکنون از گذرواژه گذشته و به مرحله پسا گذرواژه[۷۴] برسیم.

استاندارد احرازاصالت NIST 800-63 مدرن و مبتنی بر شواهد

استاندارد احرازاصالت NIST 800-63 مدرن و مبتنی بر شواهد است و صرف‌نظر از کاربرد، بهترین توصیه را می‌کند. این استاندارد برای همه سازمان‌ها در سراسر جهان مفید است، به‌ویژه برای سازمان‌های ایالات متحده و کسانی که با سازمان‌های آمریکایی سروکار دارند. اصطلاحات NIST 800-63 در ابتدا ممکن است کمی گیج‌کننده باشد، به‌خصوص اگر برای احرازاصالت فقط از نام کاربری و گذرواژه استفاده می‌کردید. پیشرفت در احرازاصالت مدرن ضروری است، بنابراین ما باید اصطلاحاتی که در آینده تبدیل به اصطلاحاتی طبیعی و پرتکرار خواهد شد را معرفی کنیم. قابل درک است که فهم این اصطلاحات تا قبل از این‌که صنعت از آن‌ها استفاده کند کمی سخت باشد. برای کمک به این موضوع، در انتهای این فصل یک واژه نامه قراردادیم. در موارد بسیار زیادی اصطلاحات را تغییر داده‌ایم تا هدف مورد نیاز ما برآورده شود. در موارد زیادی به جای تغییر کلمه کل اصطلاح را تغییر داده‌ایم تا هدف مورد نیاز ما برآورده شود. به‌عنوان مثال، ASVS از اصطلاح "گذرواژه" استفاده می‌کند در حالی که NIST از " اسرار حفظ شده"[۷۵] در این استاندارد استفاده می‌کند.

---

[71] Multi-factor authentication
[72] Knowledge based authentication
[73] Practice
[74] Post-Password
[75] Memorized secret

در ASVS، بخش V2 - احرازاصالت، V3 - مدیریت نشست و به میزان کمتر، V4 – کنترل دسترسی، طوری تغییر داده شده‌اند که با کنترل‌های انتخاب‌شده استاندارد NIST 800-63b، سازگار شوند و بیشتر بر روی تهدیدات پرتکرار و نقاط ضعف احرازاصالت تمرکز کرده‌اند. در صورت نیاز کامل به NIST 800-63، لطفا با NIST 800-63 مشورت کنید.

انتخاب یک سطح مناسب NIST AAL

استاندارد وارسی امنیتی نرم‌افزار[۷۶] تلاش کرده است تا سطح ۱ ASVS را به الزامات NIST AAL1، سطح ۲ را به AAL2 و سطح ۳ را به AAL3 متصل کند. با این حال، روش ASVS Level 1 به‌عنوان کنترل‌های "ضروری" ممکن است لزوماً برای تأیید یک برنامه یا API در سطح AAL level صحیح نباشد. به‌عنوان مثال، اگر برنامه کاربردی سطح ۳ باشد یا دارای الزامات قانونی برای AAL3 باشد، سطح۳ باید در بخش V2 و V3 – مدیریت نشست انتخاب شود. انتخاب NIST compliant authentication assertion level (AAL) باید طبق دستورالعمل NIST 800-63b انجام شود، همانطور که در انتخاب AAL در بخش NIST 800-63b بخش ۶.۲ قرار دارد

علائم و اختصارات

برنامه‌ها همیشه می‌توانند از سطح الزامات فعلی و گفته شده بالاتر بروند، به‌ویژه اگر بخواهند از احرازاصالت مدرن استفاده کنند. قبلاً، ASVS، MFA[77] را الزامی می دانست اما در NIST، MFA اجباری نیست. بنابراین، از یک علامت اختیاری در این فصل برای نشان دادن جایی که ASVS تشویق می‌کند اما نیازی به کنترل ندارد، استفاده می‌کنیم. علائم زیر در این استاندارد استفاده می‌شوند:

شرح	علامت
لازم نیست	
توصیه شده، اما لازم نیست	o
الزامیست	✓

V2.1: الزامات امنیتی گذرواژه

گذرواژه‌ها، که به نام "اسرار حفظ شده" توسط NIST 800-63 شناخته می‌شوند، شامل گذرواژه‌ها، پین‌ها، باز کردن الگوها، انتخاب kitten صحیح و یا عناصر تصویری دیگر و عبارات عبور[78] هستند. آن‌ها به‌طور کلی بر پایه "چیزی که می‌دانید" در نظر گرفته می شوند و اغلب به‌عنوان یک عامل تأییدکننده مورد استفاده قرار می‌گیرند. چالش‌های قابل توجهی برای استفاده مداوم از احرازاصالت تک‌عامله وجود دارد، از جمله میلیاردها نام کاربری معتبر و گذرواژه موجود در اینترنت، گذرواژه پیش فرض یا ضعیف، جداول رنگین کمانی[79] و واژه‌نامه‌هایی از رایج‌ترین گذرواژه‌ها.

---

[76] Application Security Verification Standard
[77] Multi factor authentication
[78] Passphrases
[79] Rainbow tables

برنامه‌های کاربردی باید کاربران را به استفاده از احرازاصالت چندعامله تشویق کنند و همچنین اجازه دهند کاربران از توکن‌هایی که در حال حاضر دارند مجدداً استفاده نمایند، مانند توکن‌های FIDO یا U2F، و یا به سرویس‌دهنده‌هایی لینک دهند که از احرازاصالت چندعامله پشتیبانی می‌کنند.

ارائه‌دهندگان خدمات اعتبارنامه (CSP[80]) هویت فدرال[۸۱] را برای کاربران فراهم می‌کنند. کاربران اغلب دارای چندین هویت در CSPهای مختلف هستند، مانند هویت سازمانی با استفاده از Azure AD، Okta، Ping Identity و یا گوگل، یا هویت مصرف‌کننده از طریق فیس‌بوک، توییتر، گوگل یا WeChat که تنها چند گزینه مورد استفاده کاربران هستند. این لیست به‌معنی تأیید این شرکت‌ها یا خدمات آن‌ها نیست، بلکه به این منظور است که توسعه‌دهندگان این واقعیت را بدانند که بسیاری از کاربران دارای هویت‌های ثبت‌شده مختلفی هستند. سازمان‌ها باید هویت‌های موجود کاربران را با هم ادغام کنند. به‌عنوان مثال، بعید است یک سازمان دولتی هویت یک کاربر در شبکه اجتماعی را به‌عنوان نحوه ورود به سیستم‌های حساس قبول کند، زیرا ساختن هویت‌های جعلی در شبکه‌های اجتماعی آسان است. در حالی‌که شرکت‌هایی که بازی تلفن همراه تولید می‌کنند از هویت شبکه‌های اجتماعی کاربران استفاده می‌کنند تا کاربران خود را افزایش دهند.

#	توضیح	L1	L2	L3	CWE	§ NIST
۲.۱.۱	وارسی کنید که گذرواژه کاربر حداقل ۱۲ کاراکتر دارد.	✓	✓	✓	۵۲۱	۵.۱.۱.۲
۲.۱.۲	وارسی کنید که گذرواژه‌های ۶۴ کاراکتری یا بیشتر مجاز هستند.	✓	✓	✓	۵۲۱	۵.۱.۱.۲
۲.۱.۳	وارسی کنید که گذرواژه‌ها میتوانند شامل کاراکتر فاصله باشند و مختصرسازی[۸۲] انجام نشده است. چند فاصله پشت سر هم را می‌توان به‌صورت یکپارچه با هم ادغام کرد.	✓	✓	✓	۵۲۱	۵.۱.۱.۲
۲.۱.۴	وارسی کنید که کاراکترهای یونیکد در گذرواژه ها مجاز هستند. یک کد یونیکد به‌عنوان یک کاراکتر در نظر گرفته می‌شود، بنابراین 12 emoji یا 64 kanji باید معتبر و مجاز باشند.	✓	✓	✓	۵۲۱	۵.۱.۱.۲
2.1.5	وارسی کنید که کاربران می‌توانند گذرواژه خود را تغییر دهند.	✓	✓	✓	۶۲۰	۵.۱.۱.۲
2.1.6	وارسی کنید که قابلیت تغییر گذرواژه نیاز به گذرواژه فعلی و جدید کاربر دارد.	✓	✓	✓	۶۲۰	۵.۱.۱.۲

[80] Credential Service Providers
[81] Federated
[82] Truncation

2.1.7	وارسی کنید که گذرواژه‌های ارسال شده در زمان ثبت‌نام حساب، ورود به سیستم و تغییر گذرواژه در مقابل یک مجموعه از گذرواژه‌ها شکسته شده (مانند ۱۰۰۰ یا ۱۰،۰۰۰ رایج‌ترین گذرواژه‌ها، که با سیاست گذرواژه سیستم مطابقت دارند) به شکل محلی یا با استفاده از یک API خارجی بررسی می‌شوند. اگر از یک API استفاده می‌شود باید از سازوکارهایی مانند zero knowledge proof یا دیگر سازوکارها استفاده شود تا اطمینان حاصل شود که گذرواژه به‌صورت شفاف و غیررمزشده[۸۳] ارسال و یا برای شناسایی کاربران استفاده نمی‌شود. اگر گذرواژه به هر دلیلی شکسته شود برنامه باید کاربر را مجبور به تنظیم یک گذرواژه جدید شکسته نشده نماید (C6).	✓	✓	✓	۵۲۱	۵.۱.۱.۲
2.1.8	وارسی کنید که ابزاری برای تشخیص قدرت گذرواژه وجود دارد تا به کاربران کمک کند گذرواژه قوی‌تری را استفاده کنند.	✓	✓	✓	۵۲۱	۵.۱.۱.۲
2.1.9	وارسی کنید که هیچ قانون محدودکننده‌ای برای استفاده از کاراکترها در گذرواژه وجود ندارد. نباید نیازمندی به استفاده از حروف بزرگ یا حروف کوچک واعداد و یا کاراکترهای ویژه وجود داشته باشد.	✓	✓	✓	۵۲۱	۵.۱.۱.۲
2.1.10	وارسی کنید که الزامی به چرخش دوره‌ای اعتبار یا تاریخچه گذرواژه وجود ندارد.	✓	✓	✓	۲۶۳	۵.۱.۱.۲
2.1.11	اطمینان حاصل کنید که قابلیت "paste"، کمک‌های مرورگر برای گذرواژه و استفاده از نرم‌افزارهای خارجی مدیریت گذرواژه مجاز هستند.	✓	✓	✓	۵۲۱	۵.۱.۱.۲
2.1.12	اطمینان حاصل کنید که کاربر می‌تواند انتخاب کند که به‌صورت موقت کل گذرواژه مخفی را مشاهده کند و یا در پلت‌فرم‌هایی که از این قابلیت پشتیبانی نمی‌کنند، به‌طور موقت آخرین کاراکتر تایپ‌شده‌ی گذرواژه را مشاهده کند.	✓	✓	✓	۵۲۱	۵.۱.۱.۲

نکته: هدف از اجازه داشتن کاربر برای مشاهده گذرواژه خود و یا مشاهده آخرین کاراکتر تایپ شده به‌صورت موقت بهبود استفاده به‌هنگام ورود به سیستم مخصوصاً هنگام استفاده از گذرواژه‌های طولانی ، عبارات و یا نرم‌افزارهای مدیریت گذرواژه است. دیگر دلیل

[83] Plain text

آن جلوگیری آزمون‌های است که در گزارش‌های خود سازمان‌ها را مجبور می‌کنند این ویژگی کاربر پسند را با بازنویسی[۸۴] کردن نسبت به حالت پیش‌فرض حذف کنند.

V2.2: الزامات عمومی احرازاصالت

چابکی برای احرازاصالت‌کنندگان برنامه‌های Future-proof ضروری است. تأییدکنندگان برنامه کاربردی را به‌گونه‌ای پیرایش کنید تا به سایر احرازاصالت کنندگان بر اساس هر ترجیحات کاربر اجازه دهند و همچنین احرازاصالت کنندگان منسوخ شده و ناامن را به روش قاعده‌مندی بازنشسته کنند.

NIST ایمیل و پیامک را به‌عنوان احرازاصالت‌کنندگان "محدود" در نظر می‌گیرد و احتمالاً از NIST 800-63 حذف خواهند شد و در نتیجه در آینده از ASVS نیز حذف خواهد شد. برنامه‌های کاربردی باید به‌گونه‌ای طراحی شوند که نیازی به استفاده از ایمیل یا پیامک نداشته باشند.

#	توضیح	L1	L2	L3	CWE	§ NIST
۲.۲.۱	وارسی کنید که کنترل‌های ضد اتوماسیون در کاهش دادن حملات آزمون اطلاعات محرمانه نقض‌شده، جست‌وجوی فراگیر و تحریم حساب کاربری[۸۵] مؤثر هستند. چنین کنترل‌هایی شامل مسدود کردن رایج‌ترین گذرواژه‌ها، تحریم‌های نرم[۸۶]، محدود کردن سرعت، کپچا، افزایش تأخیر در بین تلاش‌ها، محدودیت‌های اعمال‌شده بر آدرس IP و یا محدودیت‌های مبتنی بر ریسک مانند مکان، اولین ورود به سیستم، تلاش‌های اخیر برای باز کردن حساب، یا به شکل‌های مشابه، هستند. اطمینان حاصل کنید که بیش از ۱۰۰ تلاش شکست‌خورده در هر ساعت برای یک حساب کاربری امکان‌پذیر نیست.	✓	✓	✓	۳۰۷	۵.۲.۲ / ۵.۱.۱.۲ / ۵.۱.۴.۲ / ۵.۱.۵.۲
۲.۲.۲	اطمینان حاصل کنید که استفاده از احرازاصالت‌کننده‌های ضعیف (مانند پیامک و ایمیل) محدود به اعتبارسنجی ثانویه و تأیید تراکنش است و نه به‌عنوان جایگزینی برای روش‌های احرازاصالت امن‌تر. تأیید کنید که روش‌های قوی‌تر قبل از روشهای ضعیف ارائه شده، کاربران از خطرات آگاه هستند یا این‌که اقدامات مناسب برای محدود کردن و کاهش خطرات احتمالی صورت گرفته است.	✓	✓	✓	۳۰۷	۵.۲.۱۰

[84] Override
[85] Account lockout
[86] Soft lockout

۲.۲.۳	اطمینان حاصل کنید که پس از به‌روزرسانی جزئیات احرازاصالت، اعلان‌های[۸۷] امن به کاربران ارسال می‌شوند، مانند بازنشانی اعتبارنامه‌ها، تغییرات ایمیل یا آدرس، ورود به سیستم از مکان های ناشناخته یا خطرناک. استفاده از اعلان‌های فشار[۸۸] به‌جای پیامک یا ایمیل ترجیح داده می‌شود، اما در غیاب اعلان‌های فشار پیامک یا ایمیل به‌شرطی قابل قبول است که هیچ‌گونه اطلاعات حساسی در آن افشا نشود.	✓	✓	✓	۶۲۰	
۲.۲.۴	میزان مقاومت جعل هویت در برابر فیشینگ[۸۹] را وارسی کنید، مانند استفاده از احرازاصالت چند فاکتوری، دستگاه‌های رمزنگاری (مانند کلیدهای متصل برای تأیید هویت)، یا در سطح‌های بالاتر AAL، گواهی‌نامه‌های[۹۰] های سمت مشتری.			✓	۳۰۸	۵.۲.۵
۲.۲.۵	اطمینان حاصل کنید که در مواقعی‌که احرازاصالت CSP و برنامه کاربردی از هم جدا شده‌اند، TLS احرازاصالت‌شده متقابل بین دو نقطه انتهایی وجود دارد.			✓	۳۱۹	۵.۲.۶
۲.۲.۶	وارسی کنید که مقاومت در برابر تکرار[91] از طریق استفاده مجاز از دستگاه‌های OTP، احرازاصالت‌کنندگان رمزنگاری‌شده[۹۲] یا کدهای جستجو[۹۳] وجود دارد.			✓	۳۰۸	۵.۲.۸
۲.۲.۷	قصد احرازاصالت را با الزام به واردکردن یک توکن OTP و یا یک عمل آغاز شده توسط کاربر مثل فشردن دکمه و یا کلید سخت‌افزاری FIDO وارسی کنید.			✓	۳۰۸	۵.۲.۹

V2.3: الزامات طول‌عمر احرازاصالت‌کننده

احرازاصالت‌کننده‌ها عبارتند از گذرواژه‌ها، توکن‌های نرم، توکن‌های سخت‌افزاری و دستگاه‌های بیومتریک. طول عمر احرازاصالت‌کننده‌ها برای امنیت یک برنامه حیاتی است - اگر کسی بتواند یک حساب کاربری را بدون هیچ هویتی ثبت کند، اعتماد کمی به ادعای هویت او وجود دارد. برای سایت‌های رسانه‌های اجتماعی مانند Reddit این کاملاً طبیعی است. برای سیستم های بانکی، تمرکز بیشتر بر ثبت و صدور اعتبارات و دستگاه‌ها امری حیاتی برای امنیت برنامه‌ها است.

نکته: گذرواژه‌ها حداکثر عمر ندارند و نیاز نیست به‌طور مداوم تغییر کنند، بلکه گذرواژه‌ها باید به‌طور مداوم در برابر شکسته‌شدن به روش‌های گوناگون

---

[87] Notifications
[88] Push notifications
[89] Phishing
[90] Certificates
[91] Replay resistance
[92] Cryptographic authenticators
[93] Lookup codes

بررسی شوند.

#	توضیح	L1	L2	L3	CWE	§ NIST
۲.۳.۱	اطمینان حاصل کنید که سیستم گذرواژه‌های اولیه یا کدهای فعالسازی را **حتماً** به‌صورت تصادفی و امن تولید کرده باشد، و حتماً باید حداقل ۶ کاراکتر طول داشته باشد و می‌تواند حاوی حروف و اعداد باشد و بعد از یک دوره کوتاه مدت از بین می‌رود. نباید اجازه داده شود تا این رمزهای اولیه به‌جای رمزهای بلندمدت استفاده شوند.	✓	✓	✓	۳۳۰	۵.۱.۱.۲ / A.3
۲.۳.۲	اطمینان حاصل کنید که ثبت‌نام و استفاده از دستگاه‌های تأیید احرازاصالت از قبیل توکن‌های U2F یا FIDO پشتیبانی می‌شود.		✓	✓	۳۰۸	۶.۱.۳
۲.۳.۳	اطمینان حاصل کنید که برای تمدید احرازاصالت‌های محدود به زمان، دستورالعمل‌های تجدید[۹۴] با زمان کافی ارسال می‌شود.		✓	✓	۲۸۷	۶.۱.۴

V2.4: الزامات ذخیره‌سازی اطلاعات محرمانه

طراحان و توسعه‌دهندگان باید به نکات گفته شده در این بخش در هنگام نوشتن و یا اصلاح کد توجه کنند. این بخش به‌طور کامل تنها با استفاده از بررسی کد منبع و یا از طریق تست‌های واحد و یا تست‌های یکپارچه‌سازی امن می‌تواند مورد وارسی قرار گیرد. آزمون نفوذپذیری نمی‌تواند هیچ یک از این مسائل را شناسایی کند.

لیست توابع مورد تأیید تولید کلید یک‌طرفه در بخش 5.1.1.2 از NIST 800-63 B و در BSI Kryptographische Verfahren Empfehlungen und Schlussellängen (2018) شرح داده شده است. جدیدترین الگوریتم‌های ملی یا منطقه‌ای و استانداردهای طول کلید را می‌توان به‌جای این موارد انتخاب کرد.

این بخش نمی‌تواند توسط آزمون نفوذپذیری مورد آزمایش قرار گیرد، بنابراین کنترل‌های آن به‌عنوان L1 علامت‌گذاری نمی‌شوند. با این وجود، این بخش از اهمیت حیاتی برای امنیت اطلاعات، در صورت دزدیده شدن اطلاعات، برخوردار است، بنابراین اگر ASVS را برای یک دستورالعمل نوشتن یا طراحی و یا بازبینی کد استفاده می‌کنید، لطفا این کنترل‌ها را در نسخه خصوصی خود L1 قرار دهید.

#	توضیح	L1	L2	L3	CWE	§ NIST
۲.۴.۱	اطمینان حاصل کنید که گذرواژه‌ها به شکلی ذخیره شده‌اند که در مقابل حملات آفلاین مقاوم هستند. گذرواژه‌ها باید به وسیله salt و با استفاده		✓	✓	۹۱۶	۵.۱.۱.۲

[94] Renewal instructions

	از توابع درهم‌سازی یک‌طرفه که در لیست مورد تأیید (که در ابتدا توضیح داده شد) هستند، درهم‌سازی شوند. این توابع تولید کلید و درهم‌سازی، گذرواژه، salt و cost factor را، هنگام تولید درهم‌سازی گذرواژه، به‌عنوان ورودی می‌گیرند					
۲.۴.۲	اطمینان حاصل کنید که salt حداقل ۳۲ بیت طول داشته باشد و به‌صورت تصادفی انتخاب شود تا موجب کاهش تصادم در درهم‌سازی‌های ذخیره‌شده شود. برای هر یک از موارد محرمانه، مقدار salt و مقدار درهم‌سازی‌شده باید به‌صورت منحصربه‌فرد ذخیره شود (C6).		✓	✓	۹۱۶	۵.۱.۱.۲
۲.۴.۳	اطمینان حاصل کنید که اگر از PBKDF2 استفاده می‌شود تعداد تکرار باید تا حدی که توان سرویس‌دهنده اجازه می‌دهد بزرگ باشد، معمولاً حداقل تعداد آن ۱۰۰،۰۰۰ می‌باشد (C6).		✓	✓	۹۱۶	۵.۱.۱.۲
۲.۴.۴	اطمینان حاصل کنید که اگر bcrypt استفاده می‌شود work factor باید تا حدی که توان سرویس‌دهنده اجازه می‌دهد بزرگ باشد، معمولاً حداقل مقدار آن ۱۳ می‌باشد (C6).		✓	✓	۹۱۶	۵.۱.۱.۲
۲.۴.۵	اطمینان حاصل کنید که قدم‌های اضافی تابع تولید کلید با استفاده از مقدار saltکه مخفی است و تنها احرازاصالت‌کننده مقدار آن را می‌داند، انجام می‌شود. مقدار salt را با استفاده از یک تولیدکننده بیت تصادفی که مورد تأیید است [SP 800-90Ar1] تولید کنید و باید حداقل دارای حداقل قدرت و معیارهای امنیتی مشخص‌شده در آخرین نسخه SP 800-131A باشد. مقدار salt مخفی باید به‌طور جداگانه از درهم‌سازی گذرواژه‌ها ذخیره شده باشد (به‌عنوان مثال، در یک دستگاه خاص منظوره مانند یک ماژول امنیتی سخت‌افزاری).		✓	✓	۹۱۶	۵.۱.۱.۲

در مواردی که استاندارد ایالات متحده ذکر شده است، یک استاندارد منطقه‌ای یا محلی می‌تواند در عوض و یا در کنار استاندارد آمریکا مورد استفاده قرار گیرد.

V2.5: الزامات بازیابی اطلاعات محرمانه

#	توضیح	L1	L2	L3	CWE	§ NIST

۲.۵.۱	وارسی کنید که مقدار فعال‌ساز اولیه[۹۵] تولیدشده توسط سیستم یا مقدار محرمانه بازیابی[۹۶] به‌صورت متن آشکار به کاربر ارسال نمی‌شود.	✓	✓	✓	۶۴۰	۵.۱.۱.۲
۲.۵.۲	اطمینان حاصل کنید که راهنمای[۹۷] گذرواژه یا احرازاصالت مبتنی بر دانش (به اصطلاح "سوالات مخفی") وجود ندارد.	✓	✓	✓	۶۴۰	۵.۱.۱.۲
۲.۵.۳	اطمینان حاصل کنید که بازیابی گذرواژه در هیچ حالتی گذرواژه فعلی را آشکار نمی‌کند.	✓	✓	✓	۶۴۰	۵.۱.۱.۲
۲.۵.۴	وارسی کنید که حساب‌های اشتراکی یا پیش‌فرض وجود ندارد (مثلا "root"، "admin" و یا "sa")	✓	✓	✓	۱۶	۵.۱.۱.۲ / A.3
۲.۵.۵	وارسی کنید که اگر عامل احرازاصالت تغییر کند یا جایگزین شود، کاربر از این رویداد مطلع می‌شود.	✓	✓	✓	۳۰۴	۶.۱.۲.۳
۲.۵.۶	اطمینان حاصل کنید که گذرواژه فراموش‌شده و یا دیگر راه‌های بازیابی از سازوکارهای امن، مثل TOTP و یا دیگر توکن‌های نرم، mobile push و یا دیگر سازوکارهای بازیابی آفلاین، استفاده می‌کنند.	✓	✓	✓	۶۴۰	۵.۱.۱.۲
۲.۵.۷	اطمینان حاصل کنید که اگر OTP یا دیگر عوامل احرازاصالت چندعامله از بین رفته باشند، شواهد اثبات هویت در همان سطح هنگام ثبت‌نام انجام می‌شود.		✓	✓	۳۰۸	۶.۱.۲.۳

V2.6: الزامات وارسی look-up secretها

look-up secretها لیست‌هایی از کدهای مخفی از پیش تولید شده‌اند، مانند شماره مجوز انتقال[۹۸] (TAN)، کدهای بازیابی رسانه‌های اجتماعی یا یک گرید حاوی مجموعه ای از مقادیر تصادفی. این کدها به صورت امن برای کاربران ارسال می‌شوند. این کدها به‌صورت یک‌بار مصرف استفاده می‌شوند، و هنگامی‌که همه کدها مورد استفاده قرار می‌گیرد، لیست look-up secretها از بین می‌رود. این نوع از احرازاصالت بر پایه "چیزی است که شما آن را دارید".

#	توضیح	L1	L2	L3	CWE	§ NIST

[95] Initial activation
[96] Recovery secret
[97] Hint
[98] Transaction Authorization Numbers

۲.۶.۱	اطمینان حاصل کنید که look-up secretها فقط یک بار قابل استفاده هستند.		✓	✓	۳۰۸	۵.۱.۲.۲
۲.۶.۲	اطمینان حاصل کنید که look-up secretها به‌مقدار کافی تصادفی‌اند (۱۱۲ بیت آنتروپی) و یا اگر کمتر از ۱۱۲ بیت آنتروپی است، همراه یک salt ۳۲ بیتی منحصربه فرد و تصادفی با استفاده از یک تابع یک‌طرفه مورد تأیید درهم‌سازی شده‌اند.		✓	✓	۳۳۰	۵.۱.۲.۲
۲.۶.۳	اطمینان حاصل کنید که look-up secretها در مقابل حملات آفلاین مقاوم هستند مانند مقادیر قابل پیش‌بینی.		✓	✓	۳۱۰	۵.۱.۲.۲

V2.7: الزامات وارسی خارج از باند[99]

در گذشته، یکی از رایج‌ترین وارسی‌های خارج از محدوده، یک ایمیل یا پیامک حاوی یک لینک بازنشانی گذرواژه بود. مهاجمان از این سازوکار ضعیف برای بازنشانی حساب‌هایی که هنوز کنترل نمی‌شوند استفاده می‌کردند، مانند به‌دست‌آوردن ایمیل یک شخص واستفاده مجدد از لینک‌های بازیابی کشف‌شده. روش های بهتر برای مدیریت وارسی خارج از باند وجود دارد.

یک روش امن برای احرازاصالت‌کنندگان خارج از محدوده دستگاه‌های فیزیکی هستند که می‌توانند با وارسی‌کننده از طریق یک کانال ثانویه امن ارتباط برقرار کنند. نمونه‌هایی از این مورد شامل جمله اعلان‌های فشار در دستگاه‌های تلفن همراه است. این نوع از احرازاصالت بر پایه "چیزی است که آن را دارید". هنگامی‌که یک کاربر بخواهد احرازاصالت شود، برنامه وارسی‌کننده به‌صورت مستقیم یا غیرمستقیم از طریق یک سرویس شخص ثالث[۱۰۰] ، پیامی را به احرازاصالت کننده خارج از باند ارسال می‌کند. پیام حاوی یک کد تأیید اعتبار (معمولاً یک عدد شش رقمی تصادفی یا یک مدل گفت‌وگو برای تأیید اعتبار) است. برنامه کاربردی منتظر می‌ماند تا کد احرازاصالت را از طریق کانال اصلی دریافت کند و مقدار درهم‌سازی مقدار دریافت‌شده را با مقدار درهم‌سازی از کد تأیید هویت اصلی مقایسه می‌کند. اگر مطابقت داشته باشند، احرازاصالت‌کننده خارج از باند می‌تواند تصدیق کند که کاربر مورد تأیید است.

ASAS فرض می‌کند که تنها تعداد کمی توسعه‌دهنده از احرازاصالت‌کننده‌های خارج از باند جدید مانند اعلان‌های فشار، استفاده می‌کنند، از این رو کنترل‌های ASVS زیر برای احرازاصالت‌کننده‌های مانند API احرازاصالت، برنامه‌های کاربردی و پیاده‌سازی single sign-on استفاده می‌شود. اگر در حال توسعه با استفاده از احرازاصالت خارج ار باند جدید هستید، لطفاً به NIST 800-63B § 5.1.3.1 مراجعه کنید.

[99] Out of Band Verifier
[100] Third party

احرازاصالت‌کننده‌های خارج از باند غیرامن مثل ایمیل و VOIP مجاز نیستند. احرازاصالت PSTN و پیامک در حال حاضر از نظر NIST "محدود" شده‌اند و باید با اعلان‌های فشار یا موارد مشابه جایگزین شوند. اگر شما نیاز به استفاده از تلفن و یاپیامک برای احرازاصالت دارید، لطفا به بخش 5.1.3.3 مراجعه کنید.

#	توضیح	L1	L2	L3	CWE	§ NIST
۲.۷.۱	اطمینان حاصل کنید که احرازاصالت‌کننده‌های خارج از باند که داده‌ها را به‌صورت متن آشکار ارسال می‌کنند، مانندپیامک یا PSTN، (که طبق استاندارد NIST باید به‌صورت محدود استفاده شوند)، به‌طور پیش‌فرض ارائه نمی‌شوند و جایگزین‌های قویتری مانند اعلان فشار ارائه می‌شود.	✓	✓	✓	۲۸۷	۵.۱.۳.۲
۲.۷.۲	اطمینان حاصل کنید که احرازاصالت‌کننده‌های خارج از باند درخواست، کدها یا توکن‌های احرازاصالت خارج از باند را پس از ۱۰ دقیقه منقضی می‌کنند.	✓	✓	✓	۲۸۷	۵.۱.۳.۲
۲.۷.۳	اطمینان حاصل کنید که درخواست، کدها یا توکن‌های احرازاصالت خارج از باند فقط یک‌بار قابل استفاده‌اند و آن هم در صورتی‌که برای درخواست اصلی مورد استفاده قرار گیرند.	✓	✓	✓	۲۸۷	۵.۱.۳.۲
۲.۷.۴	اطمینان حاصل کنید که احرازاصالت‌کننده‌های خارج از باند از طریق کانال امن و مستقل با یکدیگر صحبت می‌کنند.	✓	✓	✓	۵۲۳	۵.۱.۳.۲
۲.۷.۵	اطمینان حاصل کنید که احرازاصالت‌کننده‌های خارج از باند تنها درهم‌سازی کد احرازاصالت را ذخیره می‌کنند.		✓	✓	۲۵۶	۵.۱.۳.۲
۲.۷.۵	اطمینان حاصل کنید که کدهای احرازاصالت اولیه توسط یک تولیدکننده اعداد تصادفی امن تولید شده است که حداقل دارای ۲۰ بیت آنتروپی است (معمولاً یک عدد تصادفی شش رقمی کفایت می‌کند).		✓	✓	۳۱۰	۵.۱.۳.۲

V2.8: الزامات احرازاصالت یک یا چند عاملی یک‌بار مصرف

گذرواژه‌ها یک عاملی یکبار مصرف (OTPs)، توکن‌های فیزیکی و یا نرم‌افزاری هستند که به‌طور مداوم چالش‌های شبه‌تصادفی که تغییر می‌کنند را نشان می‌دهند. این دستگاه‌ها، فیشینگ (جعل هویت) را مشکل می‌کند، اما آن را غیرممکن نمی‌سازند. این نوع از احرازاصالت

بر پایه "چیزی است که آن را دارید". توکن‌های چند عاملی شبیه توکن‌های یک عاملی هستند با این تفاوت که ابتدا باید یک پین کد صحیح، باز کردن قفل بیومتریک، قراردادن USB یا جفت‌کردن NFC و یا برخی از چیزهای اضافی دیگر را وارد کنید تا OTP ساخته و نهایی شود.

#	توضیح	L1	L2	L3	CWE	§ NIST
۲.۸.۱	اطمینان حاصل کنید کهOTP های مبتنی بر زمان یک طول عمر تعریف‌شده، قبل از منقضی شدن، دارند.	✓	✓	✓	۶۱۳	5.1.4.2 / 5.1.5.2
۲.۸.۲	اطمینان حاصل کنید که کلیدهای متقارن مورد استفاده برای تأیید OTPهای ارسال‌شده بسیار امن هستند، مانند استفاده از یک ماژول امنیت سخت‌افزاری یا ذخیره‌سازی کلید مبتنی بر سیستم‌عامل امن.		✓	✓	۳۲۰	5.1.4.2 / 5.1.5.2
۲.۸.۳	اطمینان حاصل کنید که از الگوریتم‌های رمزنگاری مورد تأیید در تولید، seeding و اعتبارسنجی استفاده شده است.		✓	✓	۳۲۶	5.1.4.2 / 5.1.5.2
۲.۸.۴	اطمینان حاصل کنید که OTP مبتنی بر زمان می‌تواند فقط یک بار در زمانی که مجاز است مورد استفاده قرار گیرد.		✓	✓	۲۸۷	5.1.4.2 / 5.1.5.2
۲.۸.۵	اطمینان حاصل کنید که اگر توکن OTP چند عاملی مبتنی بر زمان در طول مدت اعتبار مجددا استفاده شود، لاگ شده و رد می‌شود و با اعلان‌های امن به دارنده دستگاه اطلاع داده می‌شود.		✓	✓	۲۸۷	5.1.5.2
۲.۸.۶	اطمینان حاصل کنید که تولیدکننده OTP یک عاملی فیزیکی، در صورت دزدی و یا گم شدن، قابل باطل‌شدن است. اطمینان حاصل کنید که این باطل‌شدن بلادرنگ و مؤثر در تمام نشست‌های احرازاصالت شده انجام می‌شود و همچنین این عمل باید مستقل از مکان باشد.		✓	✓	۶۱۳	5.2.1
۲.۸.۷	اطمینان حاصل کنید که احرازاصالت‌کننده‌های بیومتریک تنها به‌عنوان عوامل ثانویه احرازاصالت قابل استفاده‌اند، در ارتباط با "چیزی که دارید" یا "چیزی که می‌دانید".		o	✓	۳۰۸	5.2.3

V2.9: الزامات وارسی نرم‌افزارها و وسایل رمزنگاری

کلیدهای امنیتی رمزنگاری عبارتند از کارت‌های هوشمند یا کلید های FIDO، که کاربر باید دستگاه یک رمزنگاری را به کامپیوتر وصل کرده

یا با آن pair کند تا احرازاصالت کامل شود. تأییدکنندگان یک چالش را به دستگاه یا نرم‌افزار رمزنگاری ارسال می‌کنند و دستگاه یا نرم‌افزار پاسخ را بر اساس کلید رمزنگاری ذخیره شده محاسبه می‌کند.

الزامات دستگاه‌ها یا نرم‌افزارهای رمزنگاری تک عاملی و چند عاملی مشابه هستند، زیرا وارسی احرازاصالت‌کننده رمزنگاری مالکیت عامل احرازاصالت را ثابت می‌کند.

#	توضیح	L1	L2	L3	CWE	§ NIST
۲.۹.۱	اطمینان حاصل کنید که کلیدهای رمزنگاری مورد استفاده در تأیید، به‌طور ایمن ذخیره شده و در مقابل فاش شدن محافظت شده‌اند، مانند استفاده از TPM یا HSM، یا یک سرویس سیستم‌عامل که می‌تواند از این ذخیره‌سازی امن استفاده کند.		✓	✓	۳۲۰	5.1.7.2
۲.۹.۲	اطمینان حاصل کنید که چالش حداقل ۶۴ بیت طول دارد و از نظر آماری منحصربه‌فرد است یا در طول زمان زندگی دستگاه رمزنگاری منحصربه‌فرد است.		✓	✓	۳۲۰	5.1.7.2
۲.۹.۳	اطمینان حاصل کنید که الگوریتم‌های رمزنگاری مورد تأیید درتولید، seeding و تأیید به‌کار رفته است.		✓	✓	۳۲۷	5.1.7.2

V2.10: الزامات احرازاصالت سرویس

این بخش قابل آزمون نفوذپذیری نیست، بنابراین هیچ الزام L1 ندارد. با این حال، اگر در معماری و طراحی، برنامه‌نویسی یا بررسی ایمن کد مورد استفاده قرار گیرد، فرض کنید که نرم‌افزار (همانند Java Key Store) حداقل الزامات در سطح L1 است. ذخیره سازی اسرار و گذرواژه‌ها به صورت متن آشکار وغیررمزشده در هیچ شرایط و هیچ سطحی قابل قبول نیست.

#	توضیح	L1	L2	L3	CWE	§ NIST
۲.۱۰.۱	اطمینان حاصل کنید که اسرار یکپارچگی[۱۰۱] بر روی گذرواژه‌ها ثابت، مانند کلیدهای API یا حساب‌های دارای امتیاز مشترک، تکیه نمی‌کنند.		OS assisted	HSM	۲۸۷	5.1.1.1
۲.۱۰.۲	اطمینان حاصل کنید که اگر گذرواژه مورد نیاز باشند، اطلاعات محرمانه یک حساب پیش‌فرض نیستند.		OS assisted	HSM	۲۵۵	5.1.1.1

[101] Integration secrets

۲.۱۰.۳	اطمینان حاصل کنید که گذرواژه‌ها با حفاظت کافی ذخیره می‌شوند تا از حملات بازیابی آفلاین، شامل دسترسی سیستم محلی، جلوگیری به‌عمل آورند.		OS assisted	HSM	۵۲۲	5.1.1.1
۲.۱۰.۴	اطمینان حاصل کنید که گذرواژه‌ها، در ارتباط با پایگاه‌های داده و سیستم‌های شخص ثالث، seedها و اسرار داخلی و کلید های API به‌صورت ایمن مدیریت می‌شوند و در داخل کد یا مخازن[۱۰۲] آن ذخیره نمی‌شوند. چنین ذخیره‌سازی **باید** در مقابل حملات آفلاین مقاوم باشد. استفاده از یک نرم‌افزار key store امن (L1)، ماژول سکوی سخت‌افزاری قابل اطمینان (TPM) یا یک ماژول امنیتی سخت‌افزاری (L3) برای ذخیره گذرواژه توصیه می‌شود.		OS assisted	HSM	۷۹۸	

الزامات اضافی برای نمایندگی‌های ایالات متحده

نمایندگی‌های ایالات متحده دارای الزامات اجباری در مورد NIST 800-63 هستند. استاندارد تأیید امنیتی همیشه در مورد 80٪ کنترل‌هایی است که تقریباً بر روی 100٪ برنامه‌ها اعمال می‌شود، و نه آخرین 20٪ از کنترل‌های پیشرفته یا کنترل‌هایی که کاربرد محدودی دارند به این ترتیب، ASVS یک زیرمجموعه سخت‌گیرانه از NIST 800-63 است، به ویژه برای طبقه‌بندی‌های IAL1/2 و AAL1/2، اما به اندازه کافی جامع نیست، به ویژه در مورد طبقه‌بندی‌های IAL3/AAL3.

ما به شدت سازمان‌های دولتی ایالات متحده را به بررسی و اجرای NIST 800-63 به‌طور کامل توصیه می‌کنیم.

واژه‌نامه

عبارت	معنی
CSP	ارائه‌دهنده خدمات احرازاصالت[۱۰۳]، همچنین یک ارائه‌دهنده هویت نامیده می‌شود.
احرازاصالت کننده	قطعه کدی که گذرواژه، توکن، MFA، federated assertion و غیره را تأیید می‌کند.
وارسی‌کننده[۱۰۴]	"یک نهاد که هویت را با تأیید مالکیت و کنترل یک یا دو احرازاصالت‌کننده با استفاده از یک پروتکل احرازاصالت

[102] Repository
[103] Credential Service Provider
[104] Verifier

	تأیید می‌کند. برای انجام این کار، تأییدکننده ممکن است نیاز داشته باشد که اطلاعات محرمانه‌ای که احرازاصالت‌کننده را به شناسه مشتری وصل می‌کند را اعتبارسنجی کرده و وضعیت آنها را بررسی کند"
OTP	گذرواژه یکبار مصرف
SFA[105]	احرازاصالت‌کنندگان یک عاملی که بر پایه چیزهایی است که شما می‌دانید (اسراری که حفظ کرده‌اید، گذرواژه‌ها، عبارات، PINها)، چیزهایی که شما هستید (اطلاعات بیومتریک، اثر انگشت، اسکن صورت) و یا بر پایه چیزهایی که شما دارید (توکن‌های OTP، یک دستگاه رمزنگاری مانند یک کارت هوشمند).
MFA[106]	احرازاصالت‌کننده‌های چند عاملی که دارای دو یا بیشتر احرازاصالت‌کننده تک عاملی هستند.

مراجع

برای اطلاعات بیشتر به منابع زیر رجوع شود:

- 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

[105] Single factor authenticators
[106] Multi factor authenticators

V3: الزامات وارسی مدیریت نشست

هدف کنترل

یکی از اجزای اصلی هر برنامه مبتنی بر وب یا API حالت‌دار[107] سازوکاری است که حالت یک کاربر یا دستگاهی که با آن در تعامل است را کنترل و نگهداری می‌کند. مدیریت نشست یک پروتکل بدون حالت[108] را به حالت‌دار تغییر می‌دهد که برای تمایز بین کاربران و دستگاه‌های مختلف حیاتی است.

اطمینان حاصل کنید که یک برنامه تأییدشده الزامات مدیریت نشست سطح بالای زیر را برآورده می‌کند:

- نشست‌ها منحصر به فرد هستند و نمی‌توانند حدس زده شوند و یا به اشتراک گذاشته شوند.
- زمانی که دیگر به نشست‌ها نیازی نیست و یا مدت اعتبار آنها تمام شده است، غیرقابل استفاده خواهند شد.

همان‌طور که قبلاً ذکر شد، این الزامات به یک زیرمجموعه سازگار از کنترل‌های NIST 800-63b تبدیل شده‌اند، که بر روی تهدیدات مشترک و نقاط ضعف احرازاصالت که معمولا مورد استفاده قرار می‌گیرد متمرکز شده است. الزامات تأیید قبلی کنار گذاشته شده، انحصاری بوده و یا در اکثرموارد به حالتی که سازگاری با هدف سازگاری با NIST 800-63b تغییر پیدا کرده است.

الزامات وارسی امنیت

V3.1: الزامات پایه مدیریت نشست‌ها

#	توضیح	L1	L2	L3	CWE	§ NIST
۳.۱.۱	وارسی کنید که برنامه هیچ توکن نشستی را در پارامترهای URL یا پیام‌های خطا نشان نمی‌دهد.	✓	✓	✓	598	

V3.2: الزامات انقیاد نشست

#	توضیح	L1	L2	L3	CWE	§ NIST
۳.۲.۱	تأیید کنید که با احرازاصالت کاربر برنامه یک توکن نشست جدید را ایجاد می‌کند. (C6)	✓	✓	✓	384	7.1
۳.۲.۲	تأیید کنید که توکن‌های نشست حداقل 64 بیت آنتروپی دارند. (C6)	✓	✓	✓	331	7.1

[107] Stateful
[108] Stateless

۳.۲.۳	تأیید کنید که برنامه توکن‌های نشست‌ها را در مرورگر فقط با استفاده از روش‌های امن مانند کوکی‌های امن مناسب (بخش 3.4) یا ذخیره‌سازی نشست در HTML 5 ذخیره می‌کند.	✓	✓	✓	539	7.1
۳.۲.۴	تأیید کنید که توکن‌های نشست با استفاده از الگوریتم‌های رمزنگاری تأییدشده تولید می‌شوند. (C6)		✓	✓	331	7.1

TLS یا دیگر کانال انتقال امن برای مدیریت نشست اجباری است. این مورد در فصل امنیت ارتباطات پوشش داده شده است.

V3.3: الزامات خروج و منقضی‌شدن نشست‌ها

زمان انقضای نشست‌ها در استاندارد NIST 800-63 مشخص شده است که زمان‌های انقضای بسیار طولانی‌تری را نسبت به استانداردهای گذشته مجاز می‌داند. شرکت‌ها باید جدول زیر را مطالعه کرده و بر اساس ریسک برنامه کاربردی از جدول به‌عنوان حد بالا برای زمان انقضا استفاده نمایند.

L1 در این زمینه IAL1/AAL1، L2 در این زمینه IAL2/AAL3 و L3 در این زمینه IAL3/AAL3 است. برای IAL2/AAL2 و IAL3/AAL3 زمان انقضای بیکاری کوتاه‌تر، کران پایین زمان‌های بیکاری برای خروج از سیستم یا تأیید اعتبار مجدد برای از سرگیری نشست است.

#	توضیح	L1	L2	L3	CWE	§ NIST
۳.۳.۱	وارسی کنید که خروج و منقضی‌شدن توکن نشست را باطل می‌کند، به‌گونه‌ای که کلید بازگشت مرورگر یا یک بخش متکی به جریان پایین نتوانند یک نشست احرازاصالت‌شده را از سر بگیرند. (C6)	✓	✓	✓	۶۱۳	7.1
۳.۳.۲	اگر احرازاصالت‌کننده به کاربران اجازه می‌دهد که در نشست باقی بمانند، اطمینان حاصل کنید که احرازاصالت مجدد به‌صورت دوره‌ای، چه زمانی که نشست به‌طور فعال استفاده شود و چه پس از یک دوره غیرفعال بودن، اتفاق می افتد.	۳۰ روز	12 ساعت یا 30 دقیقه عدم فعالیت، 2FA اختیاری است	12 ساعت یا 15 دقیقه عدم فعالیت، 2FA اجباری است	۶۱۳	7.2

۳.۳.۳	اطمینان حاصل کنید که برنامه پس تغییر موفقیت‌آمیز گذرواژه تمام نشست های فعال دیگر را متوقف کند و این در سراسر برنامه کاربردی، ورود و هر بخش متکی دیگر مؤثر است.	✓	✓	✓	۶۱۳	
۳.۳.۴	اطمینان حاصل کنید که کاربران قادر به مشاهده و خروج از هر یا همه نشست‌ها و دستگاه‌هایی در حال حاضر فعال هستند، باشند.		✓	✓	۶۱۳	7.1

V3.4: مدیریت نشست مبتنی بر کوکی‌ها

#	توضیح	L1	L2	L3	CWE	§ NIST
۳.۴.۱	اطمینان حاصل کنید که توکن‌های نشست مبتنی بر کوکی دارای ویژگی 'Secure' هستند. (C6)	✓	✓	✓	۶۱۴	7.1.1
۳.۴.۲	اطمینان حاصل کنید که توکن‌های نشست مبتنی بر کوکی دارای ویژگی 'HttpOnly' هستند. (C6)	✓	✓	✓	۱۰۰۴	7.1.1
۳.۴.۳	اطمینان حاصل کنید توکن‌های نشست مبتنی بر کوکی از ویژگی SameSite استفاده می‌کنند تا در حد امکان در معرض حملات CSRF قرار نگیرند. (C6)	✓	✓	✓	۱۶	7.1.1
۳.۴.۴	اطمینان حاصل کنید که توکن‌های نشست مبتنی بر کوکی از پیشوند "-Host_" استفاده می‌کنند (مراجعه کنید به قسمت مراجع) تا محرمانگی کوکی نشست برقرار شود.	✓	✓	✓	۱۶	7.1.1
۳.۴.۵	اطمینان حاصل کنید که اگر برنامه تحت یک نام دامنه با برنامه‌های دیگری منتشر شده است که کوکی‌های نشستی را تنظیم کرده یا استفاده می‌کنند که ممکن است کوکی‌های نشست را بازنویسی یا افشا کنند، صفت مسیر را در توکن‌های نشست مبتنی بر کوکی با دقیق‌ترین مسیر ممکن تنظیم شود. (C6)	✓	✓	✓	۱۶	7.1.1

V3.5: مدیریت نشست مبتنی بر توکن

مدیریت نشست مبتنی بر توکن شامل JWT، OAuth، SAML و کلیدهای API است. از بین این‌ها، کلیدهای API ضعیف هستند و نباید در کدهای جدید استفاده شوند.

#	توضیح	L1	L2	L3	CWE	§ NIST
۳.۵.۱	وارسی کنید که برنامه کاربردی توکن‌های OAuth و توکن‌های تازه‌سازی[۱۰۹] را به خودی خود به‌عنوان حضور یک مشترک[۱۱۰] تلقی نکند و اجازه دهد کاربران رابطه‌های مورد اعتماد بین برنامه‌های کاربردی متصل شده را قطع کنند.		✓	✓	۲۹۰	7.1.2
۳.۵.۲	وارسی کنید که برنامه کاربردی از توکن‌های نشست به جای اسرار و کلیدهای APIهای ایستا استفاده می‌کند، به‌جز در موارد پیاده‌سازی‌های میراثی[۱۱۱].		✓	✓	۷۹۸	
۳.۵.۳	اطمینان حاصل کنید که توکن‌های نشست بدون حالت از امضای دیجیتال، رمزنگاری و دیگر اقدامات متقابل استفاده می‌کنند تا در مقابل حملات دستکاری[۱۱۲]، پوشاندن[۱۱۳]، تکرار، رمز تهی[۱۱۴] و جایگزینی کلید محافظت نمایند.		✓	✓	۳۴۵	

V3.6: احرازاصالت مجدد از یک Federation یا Assertion

این قسمت مربوط به کسانی است که کد یک ارائه دهنده خدمات احرازاصالت (CSP) ویا یک relying party (RP) را می‌نویسند. اگر بر کدی اتکا می کنید که این ویژگی‌ها را پیاده‌سازی کرده است، مطمئن شوید که موارد زیر را به‌درستی مدیریت می‌کند.

#	توضیح	L1	L2	L3	CWE	§ NIST
۳.۶.۱	اطمینان حاصل کنید که RPها حداکثر زمان مجاز برای CSPها را مشخص می‌کند و CSPها مشترک را اگر در زمان مشخص‌شده از نشست خود استفاده نکند، دوباره احرازاصالت می‌کند.			✓	۶۱۳	7.2.1

[109] Refresh tokens
[110] Subscriber
[111] Legacy implementations
[112] Tampering
[113] Enveloping
[114] Null cipher

۳.۶.۲	اطمینان حاصل کنید که CSPها آخرین رویدادهای احرازاصالت را به اطلاع RPها می‌رسانند تا آن‌ها تصمیم بگیرند آیا نیاز به احرازاصالت مجدد هست یا خیر.			✓	۶۱۳	7.2.1

V3.7: دفاع در مقابل سوءاستفاده‌های مدیریت نشست

در گذشته، بر اساس الزامات ASVS،ISO 27002 مسدودکردن چند نشست همزمان را اجباری کرده بود. مسدودکردن چند نشست دیگر مناسب نیست، نه تنها به این دلیل که کاربران مدرن دارای دستگاه‌های مختلف هستند یا برنامه یک API بدون یک نشست مرورگر است، بلکه به این دلیل که در بسیاری از این پیاده‌سازی‌ها، آخرین احرازاصالت‌کننده برنده می‌شود، که اغلب مهاجم است. این بخش راهنمای پیشنهادی را برای بازدارندگی، تأخیر و شناسایی حملات مدیریت نشست با استفاده از کد ارائه می‌دهد.

شرح حمله نیم-باز

در اوایل سال ۲۰۱۸ میلادی، چندین مؤسسه مالی با استفاده از آن‌چه حمله‌کنندگان «حملات نیمه باز» نامیدند، به خطر افتادند. این اصطلاح در صنعت جا افتاده است. مهاجمان موسسات متعددی را با پایه کدهای اختصاصی مختلف مورد حمله قرار دادند و درواقع به نظر می‌رسید که پایه کدهای مختلفی در داخل مؤسسات مشابه وجود دارد. حمله نیم-باز از یک الگوی نادرست طراحی استفاده می‌کند که در بسیاری از احرازاصالت‌کننده‌های موجود، مدیریت نشست‌ها و سیستم‌های کنترل دسترسی وجود دارد.

مهاجمان حمله نیم-باز را با تلاش برای قفل کردن، بازنشانی کردن و یا بازیابی اطلاعات محرمانه شروع می‌کنند. یک الگوی طراحی محبوب مدیریت نشست‌ها کدهای احرازاصالت نشده، نیمه احرازاصالت شده (بازنشانی گذرواژه، فراموشی گذرواژه) و به‌صورت کامل احرازاصالت شده از اشیا/مدل‌های نشست پروفایل کاربر را مجدداً استفاده می‌کند. این الگوی طراحی یک توکن یا نشست معتبر را که شامل پروفایل قربانی از جمله درهم‌سازی گذرواژه و نقش‌هایش است را تولید می‌کند. اگر کنترل دسترسی که کنترل‌کننده‌ها یا مسیریاب‌ها را چک می‌کند مطمئن شود که کاربر به‌طور کامل وارد شده است، مهاجم قادر خواهد بود که به جای کاربر اصلی عمل کند. حمله‌ها ممکن است شامل تغییر گذرواژه به مقدار معلوم، تغییر ایمیل (تا بتواند گذرواژه را بازیابی کند) غیرفعال کردن MFA، ثبت دستگاه MFA جدید، تغییر یا مشاهده کلیدهای APIها و غیره باشد.

#	توضیح	L1	L2	L3	CWE	§ NIST
۳.۷.۱	اطمینان حاصل کنید که برنامه از معتبر بودن نشست ورود اطمینان حاصل می‌کند و یا قبل از هرگونه تغییرات حساس یا تغییر در حساب مجدداً احرازاصالت می‌کند ویا از یک احرازاصالت جانبی استفاده می‌کند.	✓	✓	✓	۷۷۸	

مراجع

برای اطلاعات بیشتر به منابع زیر رجوع شود:

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

V4: الزامات وارسی کنترل دسترسی

هدف کنترل

مجوزداشتن[115] به معنی این است که تنها افرادی که اجازه استفاده از منابع را دارند بتوانند از آن‌ها استفاده کنند. اطمینان حاصل کنید که یک برنامه تأییدشده الزامات سطح بالای زیر را تضمین می‌کند:

- افرادی که به منابع دسترسی دارند مدارک معتبر برای دسترسی به آن‌ها را دارند.
- کاربران دارای یک مجموعه مشخص از نقش‌ها و امتیازات هستند.
- متاداده‌های مربوط به نقش‌ها و اختیارات در برابر حمله تکرار یا دستکاری محافظت می‌شود.

الزامات وارسی امنیت

V4.1: طراحی کنترل دسترسی عمومی

#	توضیح	L1	L2	L3	CWE
۴.۱.۱	اطمینان حاصل کنید که برنامه قوانین کنترل دسترسی را بر روی لایه سرویس مورد اعتماد اجرا می‌کند. مخصوصاً اگر کنترل دسترسی سمت مشتری در دسترس باشد و قابل دور زدن باشد.	✓	✓	✓	۶۰۲
۴.۱.۲	اطمینان حاصل کنید که تمام ویژگی‌های کاربر و داده‌ها و اطلاعات سیاست که توسط کنترل دسترسی استفاده می‌شود، نمی‌تواند توسط کاربران نهایی دستکاری شود مگر این‌که به‌طور خاص مجاز باشند.	✓	✓	✓	۶۳۹
۴.۱.۳	اطمینان حاصل کنید که اصل کمترین امتیاز وجود دارد - کاربران فقط باید قادر به دسترسی به توابع، فایل‌های داده، URLها، کنترل‌کننده‌ها، خدمات و سایر منابعی باشند که برای آن‌ها دارای مجوز خاص هستند. این به معنای حفاظت در برابر جعل و افزایش امتیاز است. (C7)	✓	✓	✓	۲۸۵
۴.۱.۴	اطمینان حاصل کنید که اصل انکار به‌طور پیش فرض وجود دارد که در آن کاربران جدید و یا نقش‌های جدید با حداقل امتیازات یا بدون امتیاز شروع می‌کنند و کاربران/نقش‌ها به ویژگی‌های جدید دسترسی ندارند تا زمانی که دسترسی به صراحت مشخص شود. (C7)	✓	✓	✓	۲۷۶

[115] Authorization

۴.۱.۵	اطمینان حاصل کنید که کنترل دسترسی به‌طور ایمن از کار می‌افتد از جمله زمانی‌که یک استثنا رخ می‌دهد. (C10)	✓	✓	✓	۲۸۵

V4.2: کنترل دسترسی در سطح عملیات

#	توضیح	L1	L2	L3	CWE
۴.۲.۱	اطمینان حاصل کنید که داده‌های حساس و APIها در برابر حملات direct object که هدفشان ایجاد، خواندن، به‌روزرسانی و حذف رکوردها است، مانند ایجاد و یا به‌روزرسانی رکوردهای دیگر، مشاهده رکوردهای همه یا حذف همه رکوردها، محافظت می‌شوند.	✓	✓	✓	۶۳۹
۴.۲.۲	اطمینان حاصل کنید که برنامه یا چارچوب یک سازوکار قوی ضد CSRF را برای حفاظت از قابلیت تأیید هویت تأمین می‌کند ویک سازوکار ضد خودکارسازی یا ضد CSRF مؤثر از قابلیت‌های احرازاصالت‌نشده محافظت می‌کند.	✓	✓	✓	۳۵۲

V4.3: سایر ملاحظات کنترل دسترسی

#	توضیح	L1	L2	L3	CWE
۴.۳.۱	اطمینان حاصل کنید که واسطهای مدیریتی از احرازاصالت چند فاکتوری مناسب برای جلوگیری از استفاده غیرمجاز استفاده می‌کنند.	✓	✓	✓	۴۱۹
۴.۳.۲	تأیید کنید که پویش دایرکتوری[۱۱۶] غیرفعال شده است، مگر این‌که عمداً مورد نظر باشد. علاوه بر این، برنامه‌های کاربردی نباید کشف یا افشای فایل یا دایرکتوری متاداده مانند پوشه‌های Thumbs.db، DS_Store.، .git یا .svn را اجازه دهد.	✓	✓	✓	۵۴۸
۴.۳.۳	اطمینان حاصل کنید که برنامه دارای مجوز اضافی (از قبیل احرازاصالت step up یا adaptive) برای سیستم‌های با ارزش پایین‌تر و/یا جداسازی وظایف برای برنامه‌های با ارزش بالاتر است تا کنترل‌های ضد تقلب را به‌ازای ریسک برنامه کاربردی و تقلب‌های گذشته اعمال کند.		✓	✓	۷۳۲

[116] Directory browsing

مراجع

برای اطلاعات بیشتر به منابع زیر رجوع شود:

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

V5: اعتبار سنجی، پاک‌سازی و الزامات اعتبارسنجی انکدسازی

هدف کنترل

شایع‌ترین ضعف‌های امنیتی در برنامه‌های کاربردی وب، عدم موفقیت در اعتبارسنجی صحیح داده‌های ورودی که از سمت کاربر و یا محیط می‌آیند، قبل از استفاده مستقیم آن‌ها بدون انکدسازی آن‌ها در خروجی است. این ضعف منجر به تقریباً تمام آسیب‌پذیری‌های مهم در برنامه‌های کاربردی وب می‌شود، از قبیل XSS، تزریق SQL، تزریق مفسر، حمله‌های Locale/Unicode، حملات سیستم فایل و سرریزی‌های بافر.

مطمئن شوید که یک برنامه کاربردی تأییدشده الزامات سطح بالای زیر را تضمین می‌کند.

- معماری اعتبارسنجی ورودی و انکدسازی خروجی یک خط لوله مورد توافق دارد تا از حملات تزریق جلوگیری شود.
- داده ورودی قویاً از نظر نوع داده، اعتبارسنجی، محدوده و اندازه بررسی شده و در بدترین حالت پاک‌سازی یا فیلتر شده باشد.
- داده خروجی طبق هر متن داده‌ای، در حد امکان نزدیک به مفسر، انکدسازی یا escape می‌شود.

با معماری مدرن برنامه‌های کاربردی وب، انکدسازی خروجی‌ها مهم‌تر از هر زمان دیگری است. در بعضی از سناریوها ارایه یک اعتبارسنجی قدرتمند مشکل است، پس استفاده از APIهای امن‌تری مثل پرس‌وجوهای پارامتری‌شده، چارچوب‌های قالب‌بندی auto-escaping یا انکدهای به‌دقت انتخاب شده برای خروجی، برای امنیت یک برنامه کاربردی حیاتی است.

V5.1: الزامات اعتبارسنجی ورودی

کنترل‌های اعتبارسنجی ورودی به‌درستی پیاده‌سازی‌شده، با استفاده از لیست‌های سفید مثبت و تعیین نوع قوی داده‌ها، می‌تواند بیش از ۹۰٪ از کل حملات تزریق را از بین ببرد. بررسی طول و دامنه می‌تواند باعث کاهش بیشتر این موارد شود. قراردادن اعتبارسنجی ورودی امن در معماری برنامه، اسپرینت‌های طراحی، کدنویسی و آزمایش واحد و یکپارچه‌سازی مورد نیاز است. اگرچه بسیاری از این موارد در تست‌های نفوذ یافت نمی‌شوند ، نتایج حاصل از عدم اجرای آن‌ها معمولاً در V5.3 وجود دارد – الزامات انکدسازی خروجی و پیشگیری از تزریق. توصیه می‌شود که توسعه‌دهندگان و بازبینی‌کنندگان کد امن برای جلوگیری از تزریق با این بخش طوری رفتار کنند که گویی L1 برای همه موارد لازم است.

#	توضیح	L1	L2	L3	CWE
۵.۱.۱	وارسی کنید که برنامه کاربردی در برابر حملات آلوده‌سازی پارامترهای HTTP مقاوم شده است، مخصوصا اگر چارچوب برنامه کاربردی تفاوتی برای منبع پارامترهای درخواست قائل نمی‌شود (GET، POST، کوکی‌ها یا متغیرهای محیطی).	✓	✓	✓	۲۳۵

۵.۱.۲	وارسی کنید که چارچوب‌ها در برابر حملات انتساب پارامتر انبوه محافظت می‌کنند، یا این‌که برنامه کاربردی اقدامات متقابلی برای حفاظت در برابر انتساب ناامن پارامتر، مثل علامت‌زدن فیلدها به‌صورت خصوصی یا مشابه، دارد.	✓	✓	✓	۹۱۵
۵.۱.۳	وارسی کنید که تمام ورودی‌ها (فیلدهای فرم HTML، درخواست‌های REST، پارامترهای URL، سرآیندهای HTTP، کوکی‌ها، فایل‌های batch، RSS feed ها و غیره) با استفاده از اعتبارسنجی مثبت (لیست سفید) اعتبارسنجی می‌شوند.	✓	✓	✓	۲۰
۵.۱.۴	وارسی کنید داده‌های ساخت‌یافته، دارای انواع داده قوی[۱۱۷] بوده و در برابر یک شمای مشخص که شامل کاراکترهای مجاز، طول مجاز و الگوهای مجاز بوده (به‌عنوان مثال، شماره کارت‌های اعتباری یا تلفن یا وارسی این‌که دو فیلد مربوط به هم ترکیب منطقی دارند، مثل هماهنگ بودن کد پستی و شهر مربوطه) اعتبارسنجی می‌شوند.	✓	✓	✓	۲۰
۵.۱.۵	وارسی کنید که تغییر مسیر[۱۱۸] در URL فقط به مقاصدی قابل انجام است که در لیست سفید هستند، یا هنگامی‌که به یک محتوای به‌طور بالقوه خطرناک تغییر مسیر داده می‌شود یک هشدار به کاربر داده می‌شود.	✓	✓	✓	۶۰۱

V5.2: الزامات پاک‌سازی و سندباکس‌سازی

#	توضیح	L1	L2	L3	CWE
۵.۲.۱	وارسی کنید که تمام ورودی‌های غیرقابل اعتماد HTML از ویرایش‌گرهای WYSIWYG یا مشابه، به‌درستی توسط کتابخانه‌ها پاک‌سازی HTML یا ویژگی‌های چارچوب پاک‌سازی شده‌اند.	✓	✓	✓	۱۱۶
۵.۲.۲	وارسی کنید که تمام داده‌های بدون ساختار، به‌منظور اعمال معیارهای امنیتی مثل کاراکترهای مجاز و طول مجاز، پاک‌سازی شده‌اند.	✓	✓	✓	۱۳۸
۵.۲.۳	وارسی کنید که برنامه کاربردی داده‌های کاربر را قبل از پاس دادن به سیستم‌های ایمیل پاک‌سازی می‌کند تا از تزریق SMTP و IMAP جلوگیری شود.	✓	✓	✓	۱۴۷

[117] Strongly typed
[118] Redirect

۵.۲.۴	وارسی کنید که برنامه کاربردی از استفاده از eval() یا دیگر ویژگی‌های اجرای پویای کد[۱۱۹] اجتناب می‌کند. در جایی که هیچ راه جایگزینی نیست، هر داده گرفته شده از کاربر باید قبل از اجرا پاک‌سازی و سندباکس‌سازی شود.	✓	✓	✓	۹۵
۵.۲.۵	وارسی کنید که برنامه کاربردی، با اطمینان از این‌که تمام داده‌های کاربر پاک‌سازی و سندباکس‌سازی شده‌اند، از حملات تزریق قالب[۱۲۰] محافظت شده است.	✓	✓	✓	۹۴
۵.۲.۶	وارسی کنید که برنامه کاربردی با اعتبارسنجی و پاک‌سازی داده‌های غیرقابل اعتماد، ابرداده‌های مربوط به فایل در HTTP، مثل نام فایل یا فیلدهای ورودی URL، با استفاده از لیست سفیدی از پروتکل‌ها، دامنه‌ها و درگاه‌ها[۱۲۱] در برابر حملات SSRF محافظت شده است.	✓	✓	✓	۹۱۸
۵.۲.۷	وارسی کنید که برنامه کاربردی محتواهای قابل اسکریپت‌نویسی SVG ارائه‌شده توسط کاربر را پاک‌سازی، غیرفعال و سندباکس‌سازی می‌کند، مخصوصاً که آن‌ها با XSSهای منجرشده توسط اسکریپت‌های inline و foreign object مرتبط باشند.	✓	✓	✓	۱۵۹
۵.۲.۸	وارسی کنید که برنامه کاربردی محتواهای قابل اسکریپت‌نویسی یا محتواهای زبان قالب عبارت[۱۲۲] ، مثل Markdown، CSS ، XSL stylesheets، BBCode یا مشابه آن‌ها را پاک‌سازی، غیرفعال یا سندباکس‌سازی می‌کند.	✓	✓	✓	۹۴

V5.3: الزامات انکدسازی و پیشگیری از تزریق

انکدسازی خروجی در کنار یا نزدیک به مفسر مورد استفاده، برای امنیت هر برنامه کاربردی حیاتی است. معمولاً، انکدسازی خروجی پایدار نیست ولی برای امن‌کردن خروجی در زمینه خروجی مربوطه برای استفاده فوری مورد استفاده قرار می‌گیرد. عدم انکدسازی خروجی باعث تولید برنامه کاربردی متزلزل، قابل تزریق و ناامن می‌شود.

#	توضیح	L1	L2	L3	CWE
۵.۳.۱	وارسی کنید که انکدسازی خروجی برای مفسر و حوزه مورد استفاده مناسب است. برای مثال، از انکدرهای مقادیر HTML، ویژگی‌های HTML، جاوااسکریپت، پارامترهای	✓	✓	✓	۱۱۶

[119] Dynamic code execution
[120] Template Injection
[121] Ports
[122] Expression template language

	URL، سرآیندهای HTTP، SMTP و غیره، برای هر زمینه‌ای که نیاز است، مخصوصاً از ورودی‌های غیرقابل اطمینان (مثلا نام‌هایی که با یونیکد هستند و یا نام‌هایی که آپوستروف دارند، مثل ねこ یا O'Hara) استفاده کنید.				
۵.۳.۲	وارسی کنید که انکدسازی خروجی مجموعه کاراکتر و locale انتخاب‌شده توسط کاربر را حفظ کند به‌نحوی که هر کاراکتر یونیکد به‌درستی و به‌صورت امن مدیریت شده است.	✓	✓	✓	۱۷۶
۵.۳.۳	وارسی کنید که escaping خروجی آگاه از متن[۱۲۳]، که ترجیحاً خودکار و یا در بدترین حالت به‌شکل دستی است، در برابر حملات XSS انعکاسی، ذخیره‌شده و مبتنی بر DOM محافظت می‌کند.	✓	✓	✓	۷۹
۵.۳.۴	بررسی کنید که در انتخاب داده‌ها یا در پرس‌وجوهای پایگاه داده (به‌عنوان مثال SQL، HQL، ORM، NoSQL) از پرس‌وجوهای پارامتری‌شده، ORMها و چارچوب‌های موجودیت، و یا سایر مواردی که از حملات تزریق در پایگاه داده جلوگیری می‌کنند، استفاده می‌شود.	✓	✓	✓	۸۹
۵.۳.۵	وارسی کنید که در جاهایی که پرس‌وجوهای پارامتری‌شده و یا راهکارهای امن‌تری موجود نیستند، انکدسازی خروجی خاص متن استفاده شده تا از حملات تزریق جلوگیری کند، مثل استفاده از SQL escaping برای جلوگیری از تزریق SQL.	✓	✓	✓	۸۹
۵.۳.۶	وارسی کنید که برنامه کاربردی در برابر حملات تزریق جاوا اسکریپت یا JSON، شامل حملات eval، remote java script includes، دورزدن CSP، حملات XSS مبتنی بر DOM و ارزیابی عبارات جاوااسکریپت، مقاوم است.	✓	✓	✓	۸۳۰
۵.۳.۷	وارسی کنید که برنامه کاربردی در برابر آسیب‌پذیری‌های تزریق LDAP مقاوم است یا کنترل‌های خاص امنیتی را برای جلوگیری از تزریق LDAP اتخاذ کرده است.	✓	✓	✓	۹۴۶
۵.۳.۸	وارسی کنید که برنامه کاربردی در برابر حملات تزریق دستورات سیستم‌عامل مقاوم است و این‌که فراخوانی‌های سیستمی[۱۲۴] از پرس‌وجوهای پارامتری‌شده سیستم‌عامل یا انکدسازی خروجی خط فرمان متنی[۱۲۵] استفاده می‌کنند.	✓	✓	✓	۷۸

[123] Context
[124] System call
[125] Contextual

۵.۳.۹	وارسی کنید که برنامه کاربردی در برابر حملات شمول فایل‌های محلی[۱۲۶] (LFI) یا شمول فایل‌های راه دور[۱۲۷] (RFI) مقاوم است.	✓	✓	✓	۸۲۹
۵.۳.۱۰	وارسی کنید که برنامه کاربردی در برابر حملات تزریق XPath یا حملات تزریق XML مقاوم است.	✓	✓	✓	۶۴۳

نکته: استفاده از پرس‌وجوهای پارامتری‌شده و یا escapeکردن SQL همیشه کافی نیست. اسامی جداول و ستون‌ها،ORDER BY و غیره را نمی‌شود escape کرد. شمول داده‌های escape شده توسط کاربر در این فیلدها می‌تواند منجر به پرس‌وجوهای شکست‌خورده و یا تزریق SQL شود.

V5.4: الزامات حافظه، رشته و کد مدیریت‌نشده

الزامات زیر تنها در صورتی اعمال می‌شود که برنامه کاربردی از یک زبان سیستمی یا کد مدیریت‌نشده استفاده کند.

#	توضیح	L1	L2	L3	CWE
۵.۴.۱	وارسی کنید که برنامه کاربردی از رشته حافظه امن استفاده می‌کند، و همین‌طورهنگام کپی‌کردن خانه های حافظه وانجام دستورات ریاضی بر روی اشاره‌گرها به‌شکل امن‌تری نیز عمل می‌کند تا از سرریزی پشته، بافر و هیپ جلوگیری شود.		✓	✓	۱۲۰
۵.۴.۲	وارسی کنید که رشته های فرمت[۱۲۸] از ورودی های بالقوه خطرناک استفاده نکنند و ثابت باشند.		✓	✓	۱۳۴
۵.۴.۳	وارسی کنید که تکنیک‌های اعتبارسنجی امضا، محدوده و ورودی مورد استفاده قرار می‌گیرند تا از سرریزی عدد صحیح جلوگیری شود		✓	✓	۱۹۰

V5.5: الزامات جلوگیری از Deserilaztion

[126] Local file Inclusion
[127] Remote file Inclusion
[128] Format Strings

#	توضیح	L1	L2	L3	CWE
۵.۵.۱	وارسی کنید که اشیا serialize شده از بررسی‌های یکپارچگی استفاده می‌کنند یا رمز می‌شوند تا از ایجاد خصومت‌آمیز اشیاء[۱۲۹] و یا تغییر داده جلوگیری شود.	✓	✓	✓	۵۰۲
۵.۵.۲	وارسی کنید که برنامه کاربردی به‌درستی پارسرهای XML را محدود می‌کند تا فقط از محدودکننده‌ترین پیکربندی‌های ممکن استفاده کرده و اطمینان دهد که ویژگی‌های خطرناکی مثل resolveکردن موجودیت‌های خارجی برای جلوگیری از XXE غیرفعال شده است.	✓	✓	✓	۶۱۱

منابع

برای اطلاعات بیشتر به منابع زیر رجوع شود:

- 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(

برای اطلاعات بیشتر در مورد auto-scaping به منابع زیر رجوع شود:

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

---

[129] Hostile object creation

برای اطلاعات بیشتر در مورد deserialization به منابه زیر رجوع شود:

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

V6: الزامات اعتبارسنجی رمزنگاری‌های ذخیره‌شده

هدف کنترل

مطمئن شوید برنامه کاربردی تأییدشده الزامات سطح بالای زیر را تأمین می‌کند.

- تمامی ماژول‌های رمزنگاری به صورت امن با شکست مواجه شوند و خطاها به‌درستی مدیریت شوند.
- یک تولیدکننده اعداد تصادفی مناسب استفاده شده باشد.
- دسترسی به کلیدها به‌طور امن مدیریت شده باشد.

V6.1: طبقه‌بندی داده‌ها

مهم‌ترین دارایی داده‌های پردازش‌شده، ذخیره‌شده یا ارسال‌شده توسط برنامه کاربردی است. همیشه یک ارزیابی حریم خصوصی انجام دهید تا نیازهای حفاظت از داده‌ها را برای هر داده ذخیره‌شده به‌درستی طبقه‌بندی کنید.

#	توضیح	L1	L2	L3	CWE
۶.۱.۱	وارسی کنید که اطلاعات شخصی تنظیم‌شده، هنگام استراحت، به‌شکل رمزنگاری‌شده ذخیره شده‌اند، مانند اطلاعات شناسایی شخصی[۱۳۰] (PII) ، اطلاعات حساس شخصی و یا اطلاعاتی که احتمالاً تحت قوانین GDPR اتحادیه اروپا قرار می‌گیرند.		✓	✓	۳۱۱
۶.۱.۲	وارسی کنید که اطلاعات پزشکی تنظیم‌شده، هنگام استراحت، به‌شکل رمزنگاری‌شده ذخیره شده‌اند، مانند سوابق پزشکی، جزئیات لوازم پزشکی یا رکوردهای تحقیقاتی بی‌نام نشده[۱۳۱].		✓	✓	۳۱۱
۶.۱.۳	وارسی کنید که اطلاعات مالی تنظیم‌شده، هنگام استراحت، به‌شکل رمزنگاری‌شده ذخیره شده‌اند، مانند حساب‌های مالی، پیش‌فرض‌ها یا تاریخچه اعتبار، رکوردهای مالیاتی، سوابق پرداخت، اطلاعات ذینفعان یا رکوردها تحقیقاتی یا بازاری بی‌نام نشده.		✓	✓	۳۱۱

[130] Personally identifiable information
[131] De-anonymized

V6.2: الگوریتم‌ها

پیشرفت‌های اخیر در رمزنگاری به این معنی است که الگوریتم‌های امن و طول‌های کلید گذشته دیگر امن نیستند و برای حفاظت از داده‌ها کافی نیستند. بنابراین، باید الگوریتم‌ها را تغییر داد.

هرچند این قسمت به سادگی قابل آزمون نفوذپذیری نیست، توسعه‌دهندگان باید این قسمت را اجباری تلقی کنند، حتی اگر بیشتر آیتم‌ها L1 نباشند.

#	توضیح	L1	L2	L3	CWE
۶.۲.۱	وارسی کنید که تمام ماژول‌های رمزنگاری به‌شکل امن با شکست مواجه می‌شوند و خطاها به‌گونه‌ای مدیریت می‌شوند که اجازه حملات Padding Oracle را نمی‌دهند.	✓	✓	✓	۳۱۰
۶.۲.۲	وارسی کنید که الگوریتم‌ها، مدها و کتابخانه‌هایی که در صنعت امتحان خود را پس داده‌اند یا توسط دولت‌ها تأیید شده‌اند، به‌جای کدهای شخصی رمزنگاری، استفاده می‌شوند.		✓	✓	۳۲۷
۶.۲.۳	وارسی کنید که بردارهای مقداردهی اولیه رمزنگاری، پیکربندی متون رمزشده و مدهای بلاکی بر اساس آخرین توصیه‌ها و به‌صورت امن پیکربندی شده‌اند.		✓	✓	۳۲۶
۶.۲.۴	وارسی کنید که اعداد تصادفی، الگوریتم‌های درهم‌سازی یا رمزنگاری، طول کلیدها، تعداد دورها، متون رمزشده یا مودها در هر زمانی می‌توانند دوباره پیکربندی، به‌روزرسانی و یا تعویض شوند تا از شکسته‌شدن رمزها جلوگیری شود.		✓	✓	۳۲۶
۶.۲.۵	وارسی کنید که مودهای بلاکی ناامن شناخته‌شده (مثل ECB و غیره)، مدهای پدینگ[۱۳۲] ناامن شناخته‌شده (مثل PKCS#1 v1.5 و غیره)، رمزنگاری با بلوک‌های کوچک (مثل Triple DES، Blowfish و غیره) و الگوریتم‌های درهم‌سازی ضعیف (مثل MD5، SHA1 و غیره) استفاده نشده‌اند، مگر فقط به علت سازگاری رو به عقب.		✓	✓	۳۲۶
۶.۲.۶	وارسی کنید که nonceها، بردارهای مقداردهی اولیه و بقیه اعداد یکبار مصرف، نباید برای هر کلید بیشتر از یکبار استفاده شوند. نحوه تولید آن‌ها نیز باید برای الگوریتم مورد استفاده مناسب باشد.		✓	✓	۳۲۶

[132] Padding

۶.۲.۷	وارسی کنید که داده‌های رمزشده از طریق امضا، مدهای رمزنگاری احرازاصالت شده یا HMAC احرازاصالت شده‌اند تا اطمینان پیدا کنید که متن رمز شده توسط فردی غیرمجاز تغییر پیدا نمی‌کند.			✓	۳۲۶
۶.۲.۸	وارسی کنید که تمامی اعمال رمزنگاری زمان ثابت هستند و عملیات‌های مدار کوتاه[۱۳۳] در مقایسه‌ها، محاسبات یا مقادیر بازگشتی استفاده نشده‌اند تا از نشت اطلاعات جلوگیری شود.			✓	۳۸۵

V6.3: مقادیر تصادفی

بسیار سخت است که یک مولد عدد شبه‌تصادفی[۱۳۴] (PRNG) صحیح، به‌طور صحیح عمل کند. در حالت کلی، منابع خوب آنتروپی در یک سیستم در صورت استفاده بیش از حد تخلیه می‌شوند، ولی منابع با تصادفی بودن کمتر می‌توانند به کلید و رازهای قابل حدس منجر شوند.

#	توضیح	L1	L2	L3	CWE
۶.۳.۱	وارسی کنید که تمام مقادیر تصادفی، نام‌های فایل تصادفی، GUIDهای تصادفی و تمامی رشته‌های تصادفی توسط مولد اعداد تصادفی امن یک ماژول رمزنگاری تأییدشده، تولید شده‌اند، زمانی‌که هدف غیرقابل حدس بودن آن‌ها توسط مهاجم است.		✓	✓	۳۳۸
۶.۳.۲	وارسی کنید که GUIDهای تصادفی تولیدشده توسط الگوریتم GUID v4 و یک مولد اعداد شبه‌تصادفی از لحاظ رمزنگاری امن[۱۳۵] (CSPRNG) تولید شده‌اند. GUIDهای تولیدشده توسط دیگر مولدهای اعداد شبه‌تصادفی ممکن است قابل پیش‌بینی باشند.		✓	✓	۳۳۸
۶.۳.۳	وارسی کنید که اعداد تصادفی با آنتروپی مناسب تولید می‌شوند، حتی اگر برنامه کاربردی زیر بار سنگین قرار گرفته بگیرد، در غیراین‌صورت برنامه کاربردی در چنین مواقعی دچار انحطاط می‌شود.			✓	۳۳۸

[133] Short-circuit
[134] Pseudo-random number generation
[135] Cryptographically-secure pseudo-random number generator

V6.4: مدیریت راز

گرچه این قسمت به‌راحتی قابل آزمون نفوذپذیری نیست ولی توسعه‌دهندگان باید کل این قسمت را اجباری در نظر گرفته حتی اگر بیشتر آیتم‌ها L1 نباشد.

#	توضیح	L1	L2	L3	CWE
۶.۴.۱	وارسی کنید که یک راهکار مدیریت راز مثل یک مخزن کلید برای ایجاد، ذخیره‌سازی، مدیریت دسترسی به رازها و از بین بردن رازها مورد استفاده قرار گرفته است.		✓	✓	۷۹۸
۶.۴.۲	وارسی کنید که جنس کلید در معرض برنامه کاربردی قرار نگرفته است، بلکه به‌جای آن از یک ماژول امنیتی جداگانه مثل یک مخزن برای اعمال رمزنگاری استفاده می‌شود.		✓	✓	۳۲۰

منابع

برای اطلاعات بیشتر به منابع زیر رجوع شود:

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

V7: الزامات اعتبارسنجی مدیریت خطا و گرفتن لاگ

هدف کنترل

هدف اصلی مدیریت خطا و گرفتن لاگ این است که اطلاعات مفیدی را در اختیار کاربر، مدیران و تیم پاسخ‌گویی به حادثه قرار دهد. هدف ایجاد انبوهی از لاگ‌ها نیست، بلکه ایجاد لاگ‌های با کیفیت بالا است که بیشتر سیگنال تولید می‌کنند نه نویز.

لاگ‌های با کیفیت بالا معمولاً شامل اطلاعات حساس است و باید بر اساس قوانین و رهنمودهای محلی حریم خصوصی داده محافظت شود. این شامل موارد زیر است:

- جمع‌آوری و لاگ‌نکردن اطلاعات حساس مگر در مواقعی که به‌طور مشخص مورد نیاز باشد.
- اطمینان از این‌که تمامی اطلاعات لاگ‌شده به‌طور امن مدیریت شده و اطلاعات آن‌ها بر اساس طبقه‌بندی محافظت می‌شوند.
- اطمینان از این‌که لاگ‌ها برای همیشه ذخیره نشده‌اند و یک زمان حیات معین دارند که کوتاه‌ترین زمان ممکن است.

اگر لاگ ها دارای اطلاعات شخصی و یا حساس باشند، که تعاریف آن‌ها کشور به کشور متفاوت است، لاگ‌ها تبدیل به یکی از حساس‌ترین اطلاعات نگهداری شده توسط برنامه کاربردی شده و بنابراین به خودی خود برای مهاجمان بسیار جذاب خواهند بود.

همین‌طور بسیار مهم است که مطمئن شوید برنامه به‌طور امن با شکست مواجه می‌شود و اطلاعات غیرضروری را افشا نمی‌کند.

V7.1: الزامات محتوای لاگ‌ها

لاگ‌کردن اطلاعات حساس خطرناک است. لاگ‌ها خود تبدیل به اطلاعات طبقه‌بندی‌شده می‌شوند، که به این معنی است که آن‌ها نیز باید رمزنگاری شوند، تحت سیاست‌های نگه‌داری قرار گیرند و باید هنگام حسابرسی‌های امنیتی افشا شوند. مطمئن شوید که فقط اطلاعات ضروری در لاگ نگه‌داری می‌شود و به‌خصوص، اطلاعات مالی، اطلاعات هویتی مثل توکن‌های نشست، اطلاعات حساس یا اطلاعات شخصی قابل تشخیص، در لاگ‌ها نگه‌داری نشوند.

V7.1، OWASP Top 10 2017:A10 را پوشش می‌دهد. از آن‌جایی‌که 2017:A10 و این قسمت قابل آزمون نفوذپذیری نیست، مهم است برای:

- توسعه‌دهندگان که از انطباق کامل با این قسمت مطمئن شوند، چون تمام آیتم‌ها به‌عنوان L1 علامت خورده‌اند.
- تست‌نفوذکنندگان که انطباق کامل آیتم‌ها در V7.1 را به‌وسیله مصاحبه، تصاویر لحظه‌ای[۱۳۶] ویا بر اساس اظهارت وارسی کنند.

#	توضیح	L1	L2	L3	CWE

[136] Screenshots

۷.۱.۱	وارسی کنید که برنامه کاربردی اعتبارنامه‌ها[۱۳۷] یا جزئیات پرداخت را لاگ نمی‌کند. توکن‌های نشست باید در لاگ‌ها به‌شکل درهم‌سازی‌شده و غیرقابل بازگشت ذخیره شوند.	✓	✓	✓	۵۳۲
۷.۱.۲	وارسی کنید که برنامه کاربردی اطلاعات حساس دیگری که تحت قوانین محلی حریم خصوصی و سیاست‌های امنیتی مربوطه باشند را لاگ نمی‌کند.	✓	✓	✓	۵۳۲
۷.۱.۳	وارسی کنید که برنامه کاربردی رویدادهای امنیتی مرتبط شامل رویدادهای احرازاصالت موفق و ناموفق، شکست‌های کنترل دسترسی، شکست‌های deserialization و شکست‌های اعتبارسنجی ورودی را لاگ می‌کند.		✓	✓	۷۷۸
۷.۱.۴	وارسی کنید که هر رویداد لاگ شامل اطلاعات لازم برای رسیدگی با جزئیات کامل به یک رویداد در زمانی که آن رویداد اتفاق افتاده است باشد.		✓	✓	۷۷۸

V7.2: الزامات پردازش لاگ

لاگ‌کردن به‌موقع برای بررسی اتفاقات، اولویت‌دهی[۱۳۸] و ترفیع اختیار[۱۳۹] ضروری است. مطمئن شوید که لاگ‌های برنامه کاربردی واضح بوده و به‌سادگی قابل مانیتور و تحلیل شدن به‌شکل محلی یا با اتصال به یک سیستم مانیتورینگ از راه دور هستند.

V7.2، OWASP Top 10 2017:A10 را پوشش می‌دهد. از آجایی‌که 2017:A10 و این قسمت قابل آزمون نفوذپذیری نیستند، مهم است که:

- توسعه‌دهندگان از انطباق کامل با این قسمت مطمئن شوند، زیرا تمام آیتم‌ها به‌عنوان L1 علامت خورده‌اند.
- تست‌نفوذکنندگان انطباق کامل آیتم‌ها در V7.2 را به‌وسیله مصاحبه، تصاویر لحظه‌ای یا بر اساس اظهارت وارسی کنند.

#	توضیح	L1	L2	L3	CWE
۷.۲.۱	وارسی کنید که تمام تصمیمات احرازاصالت بدون ذخیره شناسه‌های نشست حساس و یا گذرواژه‌ها لاگ می‌شوند. این باید شامل همه درخواست ها به همراه ابرداده[۱۴۰]های مرتبط برای تحقیقات امنیتی باشد.		✓	✓	۷۷۸
۷.۲.۲	وارسی کنید که تمام تصمیمات کنترل دسترسی می‌توانند لاگ شوند و تمام تصمیمات شکست‌خورده نیز لاگ می‌شوند. این باید شامل همه درخواست‌ها به‌همراه ابرداده برای تحقیقات امنیتی باشد.		✓	✓	۲۸۵

[137] Credentials
[138] Triage
[139] Escalation
[140] Metadata

V7.3: الزامات محافظت از لاگ‌ها

لاگ‌هایی که به‌سادگی بتوانند تغییر کرده و یا پاک شوند برای تحقیقات و پیگردهای قانونی بی‌فایده‌اند. افشای لاگ‌ها می‌تواند جزئیات درونی برنامه کاربردی و داده‌های درونی آن را آشکار نماید. هنگام محافظت لاگ‌ها از افشا، تغییر یا پاک کردن غیرمجاز، باید احتیاط کامل صورت گیرد.

#	توضیح	L1	L2	L3	CWE
۷.۳.۱	وارسی کنید برنامه کاربردی به‌طور صحیح داده تأمین‌شده توسط کاربر را انکد کرده تا از تزریق لاگ جلوگیری کند.		✓	✓	۱۱۷
۷.۳.۲	وارسی کنید که تمام رویدادها به‌هنگام نمایش در نرم‌افزار نمایش لاگ، از تزریق محافظت می‌شوند.		✓	✓	۱۱۷
۷.۳.۳	وارسی کنید لاگ‌های امنیتی از دسترسی و تغییر غیرمجاز محافظت شده‌اند.		✓	✓	۲۰۰
۷.۳.۴	وارسی کنید منابع زمان به زمان و منطقه زمانی صحیح همزمان شده است. اگر سیستم‌ها جهانی هستند قویا لاگ‌کردن را فقط در فرمت UTC در نظر بگیرید که به تحلیل‌های جرم‌یابی پس از حادثه کمک می‌کند.		✓	✓	

نکته: انکدسازی لاگ‌ها (7.3.1) برای تست و بررسی به‌وسیله‌ی ابزارهای خودکار پویا و ابزارهای آزمون نفوذپذیری سخت است، ولی معماران، توسعه‌دهندگان و بازبینان کد باید این را یک نیازمندی L1 در نظر بگیرند.

V7.4: مدیریت خطا

هدف مدیریت خطا این است که به برنامه کاربردی اجازه دهد تا رویدادهای مرتبط با امنیت را برای مانیتورینگ، اولویت‌دهی و ترفیع اختیار تأمین کند. هدف ایجاد لاگ نیست. هنگام لاگ‌کردن رویدادهای مرتبط با امنیت مطمئن شوید هدفی برای لاگ‌ها وجود دارد و این‌که می‌تواند توسط SIEM یا نرم‌افزار تحلیل متمایز شود.

#	توضیح	L1	L2	L3	CWE
۷.۴.۱	وارسی کنید که یک پیام کلی به کاربر هنگامی‌که یک خطای غیرمنتظره یا امنیتی رخ می‌دهد نمایش داده می‌شود، ترجیحاً با یک شناسه یکتا که تیم پشتیبانی بتواند برای بررسی از آن استفاده کند.	✓	✓	✓	۲۱۰

۷.۴.۲	وارسی کنید که مدیریت استثنا (یا عملکردی مشابه آن) در سرتاسر کد استفاده شده تا شرایط خطای مورد انتظار یا غیرمنتظره را بررسی کند.		✓	✓	۵۴۴
۷.۴.۳	وارسی کنید که یک مدیریت‌کننده خطا که تمامی استثناهای مدیریت‌نشده را می‌گیرد[۱۴۱]، به‌عنوان گزینه آخر تعریف شده است.		✓	✓	۴۶۰s

نکته: برخی زبان‌ها مثل Swift و Go – و بر اساس الگوهای طراحی معمول - بسیاری از زبان‌های تابعی[۱۴۲]، از استثناها یا یک گزینه آخر برای مدیریت رویدادها پشتیبانی نمی‌کنند. در این‌صورت معماران و توسعه‌دهندگان باید از یک الگو، زبان و یا یک راه سازگار با چارچوب استفاده کنند تا مطمئن شوند برنامه کاربردی به شکلی امن استثناها و رویدادهای غیرمنتظره و امنیتی را مدیریت می‌کند.

منابع

برای اطلاعات بیشتر به منابع زیر رجوع شود:

- OWASP Testing Guide 4.0 content: Testing for Error Handling

[141] Catch
[142] Functional Langueges

V8: الزامات وارسی حفاظت از داده

هدف کنترل

برای حفاظت داده صحیح سه عنصر کلیدی وجود دارد. محرمانگی، یکپارچگی و دسترس‌پذیری (CIA). این استاندارد فرض می‌کند که محافظت از داده بر روی یک سیستم مورد اعتماد، مثل یک سرویس‌دهنده، که محکم‌شده[143] و محافظت‌های کافی را دارد، اعمال شده است.

برنامه‌های کاربردی باید فرض کنند که تمامی دستگاه‌های کاربر، به نحوی در معرض خطر قرار گرفته‌اند. جایی که یک برنامه کاربردی اطلاعات حساس را روی یک دستگاه ناامن ذخیره یا به آن ارسال می‌کند، مثل کامپیوترهای اشتراکی، تلفن‌ها و تبلت‌ها، برنامه کاربردی موظف است اطمینان دهد که اطلاعات ذخیره‌شده روی این دستگاه‌ها رمزنگاری شده و نمی‌توان آن‌ها را به‌سادگی به شکل غیرقانونی به دست آورده، تغییر داده و یا افشا کرد.

مطمئن شوید که یک برنامه کاربردی تأییدشده الزامات سطح بالای محافظت از داده زیر را تضمین می‌کند.

- محرمانگی: داده باید هم در زمان ذخیره و هم در زمان ارسال از دیده شدن و فاش شدن غیرمجاز محافظت شده باشد.
- یکپارچگی: داده باید از ایجاد شدن، تغییر کردن و پاک شدن مخرب توسط افراد مهاجم یا غیرمجاز محافظت شود.
- دسترس‌پذیری: داده باید برای کاربران مجاز بر اساس نیاز همیشه در دسترس باشد.

V8.1: محافظت عمومی داده

#	توضیح	L1	L2	L3	CWE
۸.۱.۱	وارسی کنید که برنامه کاربردی از cacheشدن اطلاعات حساس در مؤلفه‌های سرویس‌دهنده مانند توزیع‌کننده های بار[144] و حافظه‌های نهان برنامه کاربردی محافظت می‌کند.		✓	✓	۵۲۴
۸.۱.۲	وارسی کنید که تمام کپی‌های cacheشده و موقت اطلاعات حساس ذخیره‌شده روی سرویس‌دهنده از دسترسی غیرمجاز یا پاک‌کردن/نامعتبرکردن پس از دسترسی کاربر مجاز به اطلاعات حساس محافظت می‌شوند.		✓	✓	۵۲۴
۸.۱.۳	وارسی کنید که برنامه کاربردی تعداد پارامتر های ارسالی در یک درخواست، مثل فیلدهای مخفی، متغیرهای AJAX، کوکی‌ها و مقادیر سرآیند را به حداقل می‌رساند.		✓	✓	۲۳۳

[143] Hardened
[144] Load balancer

۸.۱.۴	وارسی کنید که برنامه کاربردی می‌تواند تعداد غیرطبیعی از درخواست ها را تشخیص داده و هشدار دهد. به‌عنوان مثال به‌وسیله‌ی IP، کاربر، تعداد کل درخواست‌ها در ساعت یا روز یا هر نوع دیگری که برای نرم‌افزار منطقی است.		✓	✓	۷۷۰
۸.۱.۵	وارسی کنید که از داده‌های مهم پشتیبان‌گیری منظمی انجام می‌شود و این‌که تست بازگرداندن[۱۴۵] داده‌ها انجام می‌شود.			✓	۱۹
۸.۱.۶	وارسی کنید که داده‌های پشتیبان به‌طور امن ذخیره می‌شوند تا از سرقت و خرابی آن‌ها جلوگیری شود.			✓	۱۹

V8.2: حفاظت اطلاعات در سمت مشتری

#	توضیح	L1	L2	L3	CWE
۸.۲.۱	وارسی کنید برنامه کاربردی سرآیندهای ضد cacheکردن را به‌طور کافی تنظیم کرده تا از cacheشدن اطلاعات حساس در مرورگرهای مدرن جلوگیری شود.	✓	✓	✓	۵۲۵
۸.۲.۲	وارسی کنید داده‌ای که در حافظه سمت مشتری ذخیره شده‌اند (مانند ذخیره‌سازی محلی HTML5، ذخیره‌سازی نشست، IndexedDB، کوکی‌های معمولی و فلش کوکی‌ها) شامل اطلاعات حساس و یا اطلاعات شناسایی شخصی (PII) نیست.	✓	✓	✓	۹۲۲
۸.۲.۳	وارسی کنید که پس از این‌که نشست کاربر تمام شد، داده‌های احرازاصالت شده ازحافظه ذخیره‌سازی سمت مشتری، مثل DOM مرورگر، پاک‌سازی می‌شوند.	✓	✓	✓	۹۲۲

V8.3: اطلاعات شخصی حساس

این قسمت کمک می‌کند تا اطلاعات شخصی حساس، از ایجادشدن، خوانده‌شدن، به‌روزرسانی‌شدن و یا پاک‌شدن غیرمجاز، به‌خصوص در تعداد بالا محافظت شوند.

تبعیت از این قسمت به تبعیت از کنترل دسترسی قسمت V4 و به خصوص V4.2 اشاره می‌کنند. برای مثال برای حفاظت از به‌روزرسانی های غیرمجاز و فاش شدن اطلاعات شخصی حساس، نیاز به تبعیت از V4.2.1 است. لطفاً برای پوشش کامل، از این قسمت و V4 تبعیت کنید.

[145] Restoration

نکته: قوانین و تنظیمات حریم خصوصی، مثل Australian Privacy Principles APP-11 یا GDPR، به‌طور مستقیم بر رویکرد پیاده‌سازی ذخیره‌سازی، استفاده و ارسال اطلاعات حساس شخصی توسط برنامه کاربردی تاثیر می‌گذارد. این می‌تواند از جریمه‌های بسیار شدید تا نصایح ساده تغییر کند. لطفاً به قوانین و تنظیمات محلی خود رجوع کنید و با یک وکیل و متخصص حریم خصوصی معتبر مشورت کنید.

#	توضیح	L1	L2	L3	CWE
۸.۳.۱	وارسی کنید که اطلاعات حساس از طریق بدنه پیام HTTP یا سرآیندها به سرویس‌دهنده ارسال می‌شوند، و پارامترهای رشته پرس‌وجو از هر لغت HTTP شامل هیچ اطلاعات حساسی نیستند.	✓	✓	✓	۳۱۹
۸.۳.۲	وارسی کنید که کاربران برای پاک‌کردن یا اکسپورت‌کردن داده‌هایشان بر اساس تقاضا راهکاری دارند.	✓	✓	✓	۲۱۲
۸.۳.۳	وارسی کنید که برای کاربران، با زبانی صریح و واضح درباره جمع‌آوری و استفاده از اطلاعات شخصی آن‌ها توضیح داده شده است و قبل از این‌که اطلاعات آن‌ها به هر نحوی استفاده شود، کاربران رضایت انتخابی خود را اعلام کرده‌اند.	✓	✓	✓	۲۸۵
۸.۳.۴	وارسی کنید که تمام اطلاعات حساس تولیدشده و پردازش‌شده در برنامه کاربردی شناسایی شده و سیاستی در مورد نحوه برخورد با اطلاعات حساس وجود دارد.	✓	✓	✓	۲۰۰
۸.۳.۵	وارسی کنید که اگر داده‌ی حساسی، تحت رهنمودهای مرتبط محافظت از داده، جمع‌آوری شده و یا لاگ کردن دسترسی ها مورد نیاز است، دسترسی به اطلاعات حساس حسابرسی[۱۴۶] شود البته بدون درج خود اطلاعات حساس در لاگ‌ها.		✓	✓	۵۳۲
۸.۳.۶	وارسی کنید که اطلاعات حساس نگه‌داری‌شده در حافظه، به‌محض این‌که مورد نیاز نیستند، با صفر یا داده‌های تصادفی بازنویسی[۱۴۷] شوند تا از حملات دامپینگ حافظه[۱۴۸] جلوگیری شود.		✓	✓	۲۲۶
۸.۳.۷	وارسی کنید که اطلاعات حساس و یا شخصی که نیازمند رمزنگاری هستند، با الگوریتم‌های تأییدشده رمزنگاری می‌شوند تا محرمانگی و یکپارچگی حفظ شود.		✓	✓	۳۲۷

[146] Audit
[147] Overwrite
[148] Memory dumping

۸.۳.۸	وارسی کنید که اطلاعات حساس شخصی مشمول طبقه‌بندی نگهداری داده[۱۴۹] می‌شوند، به‌گونه‌ای‌که داده‌های قدیمی یا تاریخ گذشته، به شکل خودکار، به‌صورت منظم یا هنگامی‌که وضعیت اقتضا می‌کند، پاک شوند.		✓	✓	۲۸۵

زمانی‌که حفاظت داده را مد نظر دارید، یک ملاحظه مهم باید در رابطه با استخراج، تغییر و یا استفاده بیش از حد مجاز باشد. برای مثال بسیاری از شبکه‌های اجتماعی فقط اجازه می‌دهند هر کاربر ۱۰۰ دوست جدید را در روز اضافه کند، ولی این‌که درخواست‌ها از چه سیستمی می‌آیند مهم نیست. یک سیستم بانکی ممکن است بخواهد تراکنش‌های بیشتر از ۵ عدد که مقدار بیشتر از ۱۰۰۰ یورو را به مؤسسات خارجی می فرستند بلاک کند. الزامات هر سیستم بسیار متفاوت است و برای تصمیم‌گیری این‌که چه چیزی غیرطبیعی نیست یا هست باید مدل‌سازی تهدید[۱۵۰] و ریسک کسب‌وکار[۱۵۱] در نظر گرفته شود. معیارهای مهم توانایی تشخیص، بازداشتن و ترجیحاً بلاک‌کردن چنین فعالیت‌های غیرنرمالی است.

منابع

برای اطلاعات بیشتر به منابع زیر رجوع شود:

- 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

[149] Data retention classification
[150] Threat Model
[151] Business risk

V9: الزامات وارسی ارتباطات

هدف کنترل

مطمئن شوید که یک برنامه کاربردی تأییدشده الزامات سطح بالای زیر را تضمین می‌کند:

- مستقل از حساسیت دادهای ارسالی، همیشه TLS یا یک رمزنگاری قوی، استفاده شود.
- جدیدترین و مقدم‌ترین توصیه‌های پیکربندی استفاده شده‌اند تا الگوریتم‌ها و رمزهای ارجح فعال شده و سفارش داده شوند.
- الگوریتم‌ها و رمزهای در شرف منسوخ شدن یا ضعیف به‌عنوان آخرین راه چاره انتخاب شده باشند.
- الگوریتم‌ها و رمزهای منسوخ‌شده و یا ناامن غیرفعال شده باشند.

توصیه‌های مقدم صنعت روی پیکربندی TLS، معمولاً به‌خاطر خرابی‌های مصیبت‌بار در الگوریتم‌ها یا رمزهای فعلی مکرراً تغییر می‌کنند. همیشه از آخرین نسخه ابزارهای بررسی تنظیمات TLS (مثل SSLyze یا دیگر اسکنرهای TLS) برای پیکربندی ترتیب و انتخاب الگوریتم ارجح استفاده کنید. پیکربندی‌ها باید به‌شکل دوره‌ای بررسی شوند تا اطمینان حاصل شود که پیکربندی ارتباطات امن، همیشه حاضر و کارا هستند.

V9.1: الزامات امنیت ارتباطات

تمامی ارتباطات مشتری‌ها باید فقط روی مسیرهای ارتباطی رمزشده اتفاق بیفتد. به‌طور خاص، استفاده از TLS 1.2 و یا جدیدتر در مرورگرها و موتورهای جست‌وجوی امروزی موردنیاز است. پیکربندی‌ها باید بااستفاده از ابزارهای آنلاین به‌طور منظم بازبینی شوند تا اطمینان حاصل شود که آخرین روش‌های توصیه‌شده مورد استفاده هستند.

#	توضیح	L1	L2	L3	CWE
۹.۱.۱	وارسی کنید که از TLS امن برای تمامی اتصالات مشتری‌ها استفاده شده است و به پروتکل‌های ناامن یا رمزنشده، بازگشت به عقب نمی‌کند.	✓	✓	✓	۳۱۹
۹.۱.۲	وارسی کنید که از ابزارهای آنلاین یا به‌روز برای تست TLS استفاده شده است که فقط الگوریتم‌ها، رمزها، و پروتکل‌های قوی فعال شده‌اند، و قوی‌ترین مجموعه الگوریتم‌ها و رمزها ترجیح داده شده‌اند.	✓	✓	✓	۳۲۶
۹.۱.۳	وارسی کنید که نسخه‌های قدیمی پروتکل‌های SSL و TLS، الگوریتم‌ها، رمزها و پیکربندی‌ها مثل SSLv2، SSLv3 ، TLS 1.0 و TLS 1.1 غیرفعال شده‌اند. آخرین نسخه TLS باید مجموعه رمز ارجح باشد.	✓	✓	✓	۳۲۶

V9.2: الزامات امنیتی ارتباطات سرویس‌دهنده

ارتباطات سرویس‌دهنده‌ها بیشتر از فقط یک HTTP هستند. اتصالات امن به یا از دیگر سیستم‌ها، مانند سیستم‌های مانیتورینگ، ابزارهای مدیریتی، دسترسی از راه دور و ssh، میان‌افزارها، پایگاه داده، mainframeها، سیستم‌های منابع همکار یا خارجی، باید در جای خود قرار گرفته باشند. تمامی این‌ها باید رمز شده باشند تا از "سخت در خارج، رهگیری بسیار آسان در داخل[۱۵۲]" جلوگیری شود.

#	توضیح	L1	L2	L3	CWE
۹.۲.۱	وارسی کنید که تمامی ارتباطات به و از سرویس‌دهنده، از گواهی‌نامه‌های TLS مورد اعتماد استفاده می‌کنند. وقتی که از گواهی‌نامه‌های تولیدشده داخلی و یا امضا شده توسط خود استفاده می‌شود، سرویس‌دهنده باید به‌گونه‌ای تنظیم شده باشد که فقط به CAهای خاص داخلی و یا CAهای خاص خود امضا شده، اعتماد کرده و بقیه را رد کند.		✓	✓	۲۹۵
۹.۲.۲	وارسی کنید که ارتباطات رمزشده مثل TLS برای تمامی اتصالات ورودی و خروجی، شامل پورت‌های مدیریتی، مانیتورینگ، احرازاصالت، APIها و یا فراخوانی‌های سرویس وب ، پایگاه داده، اتصالات ابری، بدون سرویس‌دهنده[۱۵۳]، mainframeها، اتصالات همکار و یا خارجی، استفاده شده است. سرویس‌دهنده نباید به پروتکل‌های ناامن و رمزنشده بازگشت به عقب کند.		✓	✓	۳۱۹
۹.۲.۳	وارسی کنید که تمام اتصالات رمزشده به سیستم‌های خارجی که شامل داده‌ها یا توابع حساس هستند، احرازاصالت شده‌اند.		✓	✓	۲۸۷
۹.۲.۴	وارسی کنید که لغو گواهی‌نامه‌های مناسب، مثل OCSP[154] Stapling، فعال و تنظیم شده است.		✓	✓	۲۹۹
۹.۲.۵	وارسی کنید که تمامی شکست‌های برقراری اتصال backend لاگ شده‌اند.			✓	۵۴۴

منابع

برای اطلاعات بیشتر به منابع زیر رجوع شود:

- OWASP – TLS Cheat Sheet

[152] Hard on the outside, trivially easy to intercept on the inside
[153] Server-less
[154] Online Certificate Status Protocol

V10: الزامات وارسی کد مخرب

هدف کنترل

اطمینان حاصل کنید که کد الزامات سطح بالای زیر را تضمین می‌کند:

- فعالیت‌های مخرب به‌طور امن و مناسب مدیریت شده تا بقیه برنامه کاربردی را تحت تأثیر قرار ندهد.
- شامل بمب زمانی یا دیگر حملات مبتنی بر زمان نباشد.
- به مقصدهای مخرب یا غیرمجاز "phone home" نکند.
- شامل درهای پشتی، تخم‌مرغ‌های عید پاک[۱۵۵]، حملات سالامی، روت‌کیت‌ها یا کدهای غیرمجازی که توسط مهاجم قابل کنترل هستند، نمی باشد.

یافتن کد مخرب، اثبات این نکته منفی است که اعتبارسنجی کامل غیرممکن است. بهترین تلاش‌ها باید انجام شوند تا اطمینان حاصل شود که کد دارای هیچ قطعه کد مخرب ذاتی یا کارکردهای ناخواسته نیست.

V10.1: کنترل‌های یکپارچگی کد

بهترین دفاع در برابر کد مخرب "اعتماد، اما اعتبارسنجی[۱۵۶]" است. معرفی کدهای غیرمجاز و یا مخرب در کد، معمولاً یک حمله مجرمانه در بسیاری از حوزه‌های قضایی است. پلیس‌ها و روال‌ها باید تحریم‌هایی را با توجه به کد مخرب ایجاد کنند.

توسعه‌دهندگان رهبر باید به‌طور منظم بررسی‌های کد را بازبینی نمایند، مخصوصاً قسمت‌هایی که ممکن است به عملکردهای مربوط به زمان، ورودی/خروجی و یا شبکه دسترسی پیدا کنند.

#	توضیح	L1	L2	L3	CWE
۱۰.۱.۱	وارسی کنید که یک ابزار تحلیل کد استفاده شده است که می‌تواند کدهایی که به‌صورت بالقوه مخرب هستند، مثل توابع زمانی، عملیات فایل ناامن و اتصالات شبکه را تشخیص دهد.			✓	۷۴۹

V10.2: جست‌وجوی کد مخرب

کد مخرب بسیار کم‌یاب بوده و شناسایی آن مشکل است. بازبینی کد به‌صورت دستی و خط به خط می‌تواند به پیدا کردن بمب‌های منطقی کمک نماید ولی حتی با تجربه‌ترین بازبینی‌کنندگان کد هنگام پیدا کردن کد مخرب، حتی اگر از وجود کد مخرب اطمینان دارند، نیز دچار سختی می‌شوند.

انطباق دادن با این قسمت بدون دسترسی کامل به کد منبع، که شامل کتابخانه‌های شخص ثالث می‌شود، امکان‌پذیر نیست.

---

[155] Easter eggs
[156] Trust, but verify

#	توضیح	L1	L2	L3	CWE
۱۰.۲.۱	وارسی کنید که کد منبع برنامه کاربردی و کتابخانه‌های شخص ثالث شامل قابلیت‌های phone home یا جمع‌آوری داده نیست. جایی‌که چنین قابلیت‌هایی موجود است، اجازه کاربر را قبل از جمع‌آوری هر نوع داده‌ای کسب کنید.		✓	✓	۳۵۹
۱۰.۲.۲	وارسی کنید که برنامه کاربردی اجازه‌های غیرضروری یا زیادی به ویژگی‌های مرتبط با حریم خصوصی مثل لیست مخاطبان، دوربین، میکروفون و یا موقعیت را درخواست نمی‌کند.		✓	✓	۲۷۲
۱۰.۲.۳	وارسی کنید که کد منبع برنامه کاربردی و کتابخانه‌های شخص ثالث، درهای پشتی، مانند حساب‌های کاربری یا کلیدهای hard-code شده یا مستندنشده‌ی اضافی، مبهم‌سازی کد، blobهای دودویی مستندنشده، روت‌کیت‌ها یا ضددیباگ‌ها، ویژگی‌های دیباگ ناامن یا دیگر عملیات‌های منقضی‌شده، ناامن یا مخفی که در صورت کشف می‌توانند به‌صورت مخرب مورد استفاده قرار بگیرند، را شامل نمی‌شود.			✓	۵۰۷
۱۰.۲.۴	با جست‌وجوی توابع مرتبط با تاریخ و زمان، وارسی کنید که کد منبع برنامه کاربردی و کتابخانه‌های شخص ثالث شامل بمب‌های زمانی نیست.			✓	۵۱۱
۱۰.۲.۵	وارسی کنید که کد منبع برنامه کاربردی یا کتابخانه‌های شخص ثالث شامل کد مخرب مانند حملات سالامی، دورزدن منطق یا بمب‌های منطقی نیست.			✓	۵۴۴
۱۰.۲.۶	وارسی کنید که کد منبع برنامه کاربردی و کتابخانه‌های شخص ثالث شامل easter egg ها یا دیگر کارکردهای ناخواسته‌ی به‌صورت بالقوه مخرب نیست.				۵۰۷

V10.3: کنترل‌های یکپارچگی برنامه کاربردی مستقرشده

هنگامی‌که یک برنامه کاربردی استقرار پیدا می‌کند، کد مخرب همچنان می‌تواند تزریق شود. برنامه‌های کاربردی باید خودشان را از حملات معمول، مثل اجرای کدهای امضا نشده از منابع غیرقابل اعتماد و تصاحب زیردامنه‌ها[۱۵۷]، محافظت کنند.

رعایت موارد این بخش عملیاتی و پیوسته است.

#	توضیح	L1	L2	L3	CWE

[157] Sub-domain takeovers

۱۰.۳.۱	وارسی کنید که برنامه کاربردی دارای ویژگی به‌روزرسانی خودکار سمت مشتری و یا سرویس‌دهنده است، به‌روزرسانی‌ها باید از طریق کانال‌های امن حاصل شوند و باید امضای دیجیتال داشته باشند. کد بروزرسانی‌کننده باید امضای دیجیتال بروزرسانی را قبل از نصب و یا اجرای آن اعتبارسنجی کند.	✓	✓	✓	۱۶
۱۰.۳.۲	وارسی کنید که برنامه کاربردی حفاظت‌های یکپارچگی مثل امضای کد یا جامعیت زیرمنابع را انجام می‌دهد. برنامه کاربردی نباید کد را از منابع غیرقابل اعتماد، مانند includeهای در حال بارگذاری، ماژول‌ها، پلاگین‌ها، کد یا کتابخانه‌هایی از منابع غیرقابل اعتماد یا اینترنت، بارگذاری یا اجرا کند.	✓	✓	✓	۳۵۳
۱۰.۳.۳	وارسی کنید که برنامه کاربردی در برابر تصاحب زیردامنه محافظت می‌شود، در صورتی‌که برنامه کاربردی به ورودی‌های DNS یا زیردامنه‌های DNS اعتماد کند، مانند نام دامنه منقضی‌شده، اشره‌گرهای DNS یا CNAMEهای قدیمی، پروژه‌های منقضی‌شده در مخازن کد منبع عمومی یا APIهای ابر گذرا، توابع بدون سرویس‌دهنده[۱۵۸]، یا سطل‌های ذخیره (autogen-bucket-id.cloud.example.com) یا موارد مشابه. حمایت‌ها می‌تواند شامل اطمینان از این باشد که نام‌های DNS استفاده‌شده توسط برنامه‌های کاربردی به‌طور مرتب برای انقضا یا تغییر بررسی شوند.	✓	✓	✓	۳۵۰

منابع

برای اطلاعات بیشتر به منابع زیر رجوع شود:

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

[158] Serverless functions

V11: الزامات اعتبارسنجی منطق کسب‌وکار

هدف کنترل

اطمینان حاصل کنید که برنامه کاربردی تأییدشده الزامات سطح بالای زیر را تضمین می‌کند:

- جریان منطق کسب‌وکار ترتیبی است، به‌ترتیب پردازش شده است، و قابل دورزدن نیست.
- منطق کسب‌وکار شامل محدودیت‌هایی برای تشخیص و جلوگیری از حملات خودکارسازی‌شده، مثل انتقال پیوسته مقادیر ریز مالی و یا افزودن یک میلیون دوست جدید به شکل یکجا و مانند آن است.
- جریان‌های ارزشمند منطق کسب‌وکار، حالت‌های سوءاستفاده و افراد مخرب را در نظر گرفته‌اند و در برابر حملات کلاهبرداری[۱۵۹]، دستکاری، انکار، افشای اطلاعات و افزایش امتیاز محافظت‌هایی دارند.

V11.1: الزامات امنیتی منطق کسب‌وکار

منطق کسب‌وکار برای هر برنامه کاربردی به‌قدری اختصاصی است که هیچ‌وقت چک‌لیستی برای آن به‌کار گرفته نخواهد شد. امنیت منطق کسب‌وکار باید به‌صورتی طراحی شده باشد تا در برابر تهدیدات احتمالی خارجی محافظت به‌عمل آورد - نمی‌تواند توسط یک دیواره آتش برنامه کاربردی وب یا ارتباطات امن اضافه شود. توصیه می‌کنیم که از مدل‌سازی تهدید هنگام طراحی‌های مختلف استفاده شود، مثلاً با استفاده از OWASP Cornucopia و یا ابزارهای مشابه.

#	توضیح	L1	L2	L3	CWE
۱۱.۱.۱	وارسی کنید که برنامه کاربردی فقط جریان‌های منطق کسب‌وکاری را پردازش می‌کند که مربوط به یک کاربر و به‌صورت ترتیبی بوده و هیچ مرحله‌ای از قلم نیفتاده باشد.	✓	✓	✓	۸۴۱
۱۱.۱.۲	وارسی کنید که برنامه کاربردی فقط جریان‌های منطق کسب‌وکاری را پردازش می‌کند که تمامی مراحل آن در زمان منطقی برای یک انسان انجام شده است. مثلا تراکنش‌ها با سرعت بیش از حد ارسال نشده باشند.	✓	✓	✓	۷۷۹
۱۱.۱.۳	وارسی کنید که برنامه کاربردی دارای محدودیت‌های مناسبی برای فعالیت‌های کسب‌وکار خاص یا تراکنش‌ها است، که به‌طور صحیح برای هر کاربر، اعمال شده است.	✓	✓	✓	۷۷۰
۱۱.۱.۴	وارسی کنید که برنامه کاربردی دارای کنترل‌های به‌اندازه کافی ضدخودکارسازی برای تشخیص و حفاظت در برابر نشت داده، درخواست‌های بیش از اندازه در منطق کسب‌وکار، بارگذاری بیش از اندازه فایل یا حملات انکار سرویس است.	✓	✓	✓	۷۷۰

[159] Spoofing

۱۱.۱.۵	وارسی کنید که برنامه کاربردی دارای محدودیت‌ها یا اعتبارسنجی منطق کسب‌وکار است تا در برابر خطرات یا تهدیدات احتمالی که توسط مدل‌سازی تهدید و یا روش‌های مشابه شناسایی شده‌اند محافظت نماید.	✓	✓	✓	۸۴۱
۱۱.۱.۶	وارسی کنید که برنامه کاربردی از مشکلات "زمان بررسی به زمان استفاده" (TOCTOU[160]) یا دیگر شرایط رقابتی برای عملیات حساس رنج نمی‌برد.		✓	✓	۳۶۷
۱۱.۱.۷	وارسی کنید که برنامه کاربردی بر اتفاقات و یا فعالیت‌های غیرمعمول از لحاظ منطق کسب‌وکار نظارت می‌کند. به‌عنوان مثال، تلاش‌هایی برای انجام فعالیت‌هایی که خارج از ترتیب هستند یا فعالیت‌هایی که هیچ‌وقت یک کاربر عادی انجام نمی‌دهد.		✓	✓	۷۵۴
۱۱.۱.۸	وارسی کنید که برنامه کاربردی هنگامی‌که حملات خودکارسازی‌شده یا فعالیت‌های غیرعادی شناسایی می‌شوند، دارای هشدارهای قابل پیکربندی است.		✓	✓	۳۹۰

منابع

برای اطلاعات بیشتر به منابع زیر رجوع شود:

- OWASP Testing Guide 4.0: Business Logic Testing
- OWASP Cheat Sheet
- Anti-automation can be achieved in many ways, including the use of OWASP AppSensor and OWASP Automated Threats to Web Applications
- OWASP AppSensor can also help with Attack Detection and Response.
- OWASP Cornucopia

[160] Time Of Check to Time Of Use

V12: الزامات اعتبارسنجی فایل و منابع

هدف کنترل

اطمینان حاصل کنید که برنامه کاربردی تأییدشده الزامات سطح بالای زیر را تأمین می‌کند:

- داده‌های فایل غیرقابل اعتماد باید به‌طور مناسب و به شیوه‌ای امن مدیریت شوند.
- داده‌های فایل غیرقابل اعتماد که از منابع غیرقابل اعتماد به‌دست آمده‌اند در بیرون از مسیر ریشه وب و با مجوزهای محدود ذخیره می‌شوند.

V12.1: الزامات بارگذاری فایل

اگرچه بمب‌های فشرده[۱۶۱] با استفاده از تکنیک‌های آزمون نفوذپذیری تا حد خوبی قابل تست هستند، اما آن‌ها L2 و یا بالاتر در نظر گرفته می‌شوند تا ملاحظات طراحی و توسعه را به‌همراه تست‌های دستی دقیق تشویق کند، و از آزمون نفوذپذیری دستی خودکار و بدون مهارت یک شرط انکار سرویس اجتناب کنند.

#	توضیح	L1	L2	L3	CWE
۱۲.۱.۱	وارسی کنید که برنامه کاربردی فایل‌های بزرگ که می‌توانند حافظه ذخیره‌سازی را پر کرده و یا باعث حمله انکار سرویس شوند را قبول نمی‌کند.	✓	✓	✓	۴۰۰
۱۲.۱.۲	وارسی کنید که فایل‌های فشرده شده برای بمب‌های فشرده بررسی شده‌اند. منظور از بمب‌های فشرده فایل‌های فشرده کوچکی هستند که وقتی از حالت فشرده خارج می‌شوند به فایل‌هایی با حجم بسیار بالا تبدیل شده و محدودیت‌های ذخیره‌سازی فایل را از بین می‌برند.		✓	✓	۴۰۹
۱۲.۱.۳	وارسی کنید که یک کران بالا برای اندازه فایل و حداکثر تعداد فایل‌ها به‌ازای هر کاربر در نظر گرفته شده است تا اطمینان حاصل کنید که یک کاربر، نمی‌تواند حافظه ذخیره‌سازی را با تعداد زیادی فایل و یا فایل‌های بیش از اندازه بزرگ پر کند.		✓	✓	۷۷۰

V12.2: الزامات یکپارچگی فایل

#	توضیح	L1	L2	L3	CWE
۱۲.۲.۱	وارسی کنید که فایل‌های به‌دست آمده از منابع غیرقابل اعتماد، اعتبارسنجی می‌شوند که بر اساس محتوای فایل از نوع مورد انتظار باشند.		✓	✓	۴۳۴

[161] Zip bombs

V12.3: الزامات اجرای فایل

#	توضیح	L1	L2	L3	CWE
۱۲.۳.۱	وارسی کنید که ابرداده‌های نام فایل‌های ارسال‌شده توسط کاربر به‌طور مستقیم با فایل‌های سیستمی یا چارچوبی و URL API استفاده نشده‌اند تا به این وسیله از حمله پیمایش مسیر جلوگیری شود.	✓	✓	✓	۲۲
۱۲.۳.۲	وارسی کنید که ابرداده‌های نام فایل‌های ارسال‌شده توسط کاربر اعتبارسنجی می‌شوند و یا به‌منظور جلوگیری از افشا، ایجاد، به‌روزرسانی، یا حذف فایل‌های محلی (LFI) نادیده گرفته می‌شوند.	✓	✓	✓	۷۳
۱۲.۳.۳	وارسی کنید که ابرداده‌های نام فایل‌های ارسال‌شده توسط کاربر اعتبارسنجی می‌شوند و یا به‌منظور جلوگیری از افشا یا اجرای فایل‌های راه دور (RFI)، که می‌تواند منجر به SSRF نیز بشود، نادیده گرفته می‌شوند.	✓	✓	✓	۹۸
۱۲.۳.۴	وارسی کنید که برنامه کاربردی، با اعتبارسنجی یا نادیده‌گرفتن نام فایل‌های ارسال‌شده توسط کاربر در یک پارامتر JSON، JSONP، یا URL، در برابر دانلود فایل بازتابی RFD[162] محافظت شده است. سرآیند Content-Type در بسته پاسخ باید به text/plain تنظیم شده باشد و سرآیند Content-Disposition باید یک نام فایل ثابت داشته باشد.	✓	✓	✓	۶۴۱
۱۲.۳.۵	وارسی کنید که ابرداده‌های فایل‌های غیرقابل اعتماد به‌صورت مستقیم با API یا کتابخانه‌های سیستمی استفاده نمی‌شوند، تا در برابر تزریق دستورات سیستم‌عامل محافظت شوند.	✓	✓	✓	۷۸
۱۲.۳.۶	وارسی کنید که برنامه کاربردی شامل عملکردی از منابع غیرقابل اعتماد، مثل شبکه‌های توزیع محتوای[۱۶۳] تأییدنشده، کتابخانه‌های جاوااسکریپت، کتابخانه‌های node npm یا DLLهای سمت سرویس‌دهنده نیستند و آن‌ها را اجرا نمی‌کند.		✓	✓	۸۲۹

V12.4: الزامات ذخیره‌سازی فایل

#	توضیح	L1	L2	L3	CWE

[162] Reflective File Download
[163] Content distribution networks

۱۲.۴.۱	وارسی کنید که فایل‌های به‌دست آمده از منابع غیرقابل اعتماد خارج از مسیر ریشه وب، با اختیارات محدود و ترجیحا با یک اعتبارسنجی قوی ذخیره می‌شوند.	✓	✓	✓	۹۲۲
۱۲.۴.۲	وارسی کنید که فایل‌های به‌دست آمده از منابع غیرقابل اعتماد به‌وسیله‌ی اسکنرهای آنتی‌ویروس اسکن شده‌اند تا از بارگذاری محتوای خرابکارانه جلوگیری شود.	✓	✓	✓	۵۰۹

V12.5: الزامات دانلود فایل

#	توضیح	L1	L2	L3	CWE
۱۲.۵.۱	وارسی کنید که لایه وب طوری تنظیم شده است که فقط فایل‌هایی با فرمت خاص را پشتیبانی کند تا از نشت اطلاعات یا کد منبع به‌صورت ناخواسته جلوگیری شود. برای مثال، فایل‌های پشتیبان‌گیری (مانند bak.)، فایل‌های کاری موقتی (مانند swp)، فایل‌های فشرده (zip.، tar.gz. و غیره) و سایر پسوندها که به‌طور معمول توسط ویرایشگران استفاده می‌شوند، باید مگر در صورت نیاز، بلاک شوند.	✓	✓	✓	۵۵۲
۱۲.۵.۲	وارسی کنید که درخواست‌های مستقیم برای بارگذاری فایل‌ها، هیچگاه به‌عنوان محتوای html/JavaScript اجرا نمی‌شوند.	✓	✓	✓	۴۳۴

V12.6: الزامات حفاظت از SSRF

#	توضیح	L1	L2	L3	CWE
۱۲.۶.۱	وارسی کنید که سرویس‌دهنده وب یا برنامه کاربردی با یک لیست سفید از منابع یا سیستم‌هایی که سرویس‌دهنده می‌تواند درخواست‌ها را به آن‌ها بفرستد یا داده/ فایل‌ها را از آن‌ها بارگذاری کند، پیکربندی شده است.	✓	✓	✓	۹۱۸

منابع

برای اطلاعات بیشتر به منابع زیر رجوع شود:

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

V13: الزامات وارسی API و سرویس وب

هدف کنترل

اطمینان حاصل کنید که یک برنامه کاربردی تأییدشده که از APIهای لایه سرویس مورد اعتماد (معمولاً با استفاده از JSON یا XML یا GraphQL) استفاده می‌کند موارد زیر را دارا است:

- احرازاصالت، مدیریت نشست و مجوزدهی متناسب برای تمامی سرویس‌های وب.
- اعتبارسنجی ورودی در تمامی پارامترهایی که از یک سطح اعتماد پایین‌تر به یک سطح اعتماد بالاتر انتقال پیدا می‌کنند.
- کنترل‌های امنیتی مؤثر برای تمامی انواع API، شامل APIهای ابری و بدون سرویس‌دهنده[۱۶۴]

لطفاً این فصل را در کنار تمامی فصل‌های دیگر که در این سطح مشابه هستند بخوانید. ما دیگر نگرانی‌های احرازاصالت و یا مدیریت نشست API را تکرار نمی‌کنیم.

V13.1: الزامات وارسی امنیت کلی سرویس‌های وب

#	توضیح	L1	L2	L3	CWE
۱۳.۱.۱	وارسی کنید که تمامی اجزای برنامه کاربردی از یک encoding و پارسر مشابه استفاده می‌کنند تا از حملات پارسر که از URI یا رفتارهای پارس‌کردن متفاوت بهره‌برداری می‌کنند و می‌تواند در حملات SSRF و RFI استفاده شوند، اجتناب شود.	✓	✓	✓	۱۱۶
۱۳.۱.۲	وارسی کنید که دسترسی به توابع مدیریتی و اجرایی فقط به مدیران دارای مجوز محدود شده است.	✓	✓	✓	۴۱۹
۱۳.۱.۳	وارسی کنید که URLهای API هیچ اطلاعات حساسی مثل کلید API، توکن‌های نشست و غیره را افشا نمی‌کنند.	✓	✓	✓	۵۹۸
۱۳.۱.۴	وارسی کنید که تصمیمات مجوزدهی اتخاذشده، هم در URI، با امنیت برنامه‌ای یا توصیفی در کنترلر یا مسیریاب، و هم در سطح منابع، با مجوزهای مبتنی بر مدل، اجرا می‌شوند.		✓	✓	۲۸۵
۱۳.۱.۵	وارسی کنید که درخواست‌های شامل content type های غیرمنتظره یا فاقد content type با سرآیندهای مناسب رد می‌شوند (وضعیت پاسخ 406 Unacceptable یا 415 Unsupported Media Type)		✓	✓	۴۳۴

[164] Serverless

V13.2: الزامات وارسی سرویس‌های وب RESTful

اعتبارسنجی شمای JSON، در مرحله پیش‌نویس از استانداردسازی است (منابع را ببینید). هنگام مدنظر قرار دادن اعتبارسنجی شمای JSON، که بهترین تمرین برای سرویس‌های وب SOAP است، استفاده از این راهبردهای اضافی اعتبارسنجی داده را در ترکیب با اعتبارسنجی شمای JSON در نظر بگیرید:

- اعتبارسنجی شیء JSON، مانند این‌که آیا عناصر مفقودشده یا اضافی وجود دارند.
- اعتبارسنجی مقادیر شیء JSON، با استفاده از روش‌های اعتبارسنجی ورودی استاندارد مثل نوع داده، فرمت داده، طول و غیره.
- و اعتبارسنجی رسمی شمای JSON.

هنگامی‌که استاندارد اعتبارسنجی شمای JSON رسمی شود، ASVS نصایح خود را برای این قسمت به‌روزرسانی می‌کند. با دقت هر کتابخانه اعتبارسنجی شمای JSON در حال استفاده را مانیتور کنید، زیرا آن‌ها نیاز دارند مرتباً به‌روزرسانی شوند، تا زمانی که این استاندارد رسمی‌شده و تمامی باگ‌ها از پیاده‌سازی‌های مرجع حذف شوند.

#	توضیح	L1	L2	L3	CWE
۱۳.۲.۱	وارسی کنید که متدهای فعال RESTful HTTP انتخاب معتبری برای کاربر یا عمل هستند، مانند جلوگیری از استفاده DELETE یا PUT توسط کاربران عادی بر روی منابع و یا API محافظت‌شده.	✓	✓	✓	۶۵۰
۱۳.۲.۲	وارسی کنید که اعتبارسنجی شمای JSON به‌جا استفاده شده است و قبل از پذیرش ورودی تأیید شده است.	✓	✓	✓	۲۰
۱۳.۲.۳	وارسی کنید که سرویس‌های وب RESTful که از کوکی‌ها استفاده می‌کنند، از Cross Site Request Forgery (CSRF) از طریق استفاده از حداقل یکی یا بیشتر از روش‌های زیر محافظت شده است: الگوهای ارسال کوکی دوگانه و یا سه گانه، nonceهای CSRF، و یا بررسی سرآیند ORIGIN در درخواست‌ها.	✓	✓	✓	۳۵۲
۱۳.۲.۴	وارسی کنید که سرویس‌های REST دارای کنترل‌های ضدخودکارسازی هستند که در برابر فراخوانی‌های بیش از حد، مخصوصاً اگر API احرازاصالت انجام نمی‌دهد، محافظت شوند.		✓	✓	۷۷۹
۱۳.۲.۵	وارسی کنید که سرویس‌های REST به‌طور صریح Content-Type ورودی‌ها را بررسی می‌کنند که از نوع مورد انتظار مثل application/xml یا application/JSON باشند.		✓	✓	۴۳۶

۱۳.۲.۶	وارسی کنید که سرآیندهای پیام و payload قابل اعتماد هستند و در حین انتقال دست‌کاری نشده‌اند. الزام به استفاده از رمزنگاری قوی برای انتقال (فقط TLS) ممکن است در بیشتر موارد کافی باشد از آن‌جایی که هم یکپارچگی و هم محرمانگی را تأمین می‌کند. امضای دیجیتال برای هر پیام در برنامه‌های با امنیت بالا می‌تواند در کنار محافظت‌های انتقال، ضمانت بیشتری را تأمین کند ولی با خود پیچیدگی و خطرهایی به‌همراه دارد که در مقابل مزایایش سنگینی می‌کند.		✓	✓	۳۴۵

V13.3: الزامات وارسی سرویس‌های وب SOAP

#	توضیح	L1	L2	L3	CWE
۱۳.۳.۱	وارسی کنید که اعتبارسنجی شمای XSD انجام می‌شود یک سند XML به‌درستی شکل گرفته را اطمینان دهد که به دنبالش اعتبارسنجی هر فیلد ورودی انجام می‌شود، قبل از این‌که هرگونه پردازشی روی آن داده انجام شود.	✓	✓	✓	۲۰
۱۳.۳.۲	وارسی کنید که payload پیام با استفاده از WS-Security امضا شده تا از ارتباط قابل اعتماد بین مشتری و سرویس، اطمینان حاصل شود.	✓	✓	✓	۳۴۵

نکته: به‌خاطر مسائل مربوط به حملات XXE در برابر DTD، از اعتبارسنجی DTD نباید استفاده شود، و ارزیابی چارچوب DTD بر اساس الزامات گفته شده در پیکربندی V14، باید غیرفعال شود.

V13.4: GraphQL و دیگر الزامات امنیتی لایه داده سرویس وب

#	توضیح	L1	L2	L3	CWE
۱۳.۴.۱	تأیید کنید که لیست سفید کردن پرس‌وجو یا ترکیبی از محدودکردن عمق و محدودکردن مقدار باید برای جلوگیری از GraphQL یا انکار سرویس (DoS) بیان لایه داده در نتیجه پرس‌وجوهای سنگین و تودرتو استفاده شود. برای سناریوهای پیشرفته‌تر، از تحلیل هزینه پرس‌وجو استفاده می‌شود.		✓	✓	۷۷۰
۱۳.۴.۲	وارسی کنید که GraphQL و یا دیگر منطق‌های مجوزدهی لایه داده باید در لایه منطق کسب‌وکار به‌جای لایه GraphQL پیاده‌سازی شوند.		✓	✓	۲۸۵

منابع

برای اطلاعات بیشتر به منابع زیر رجوع شود:

- OWASP Serverless Top 10
- OWASP Serverless Project
- OWASP Testing Guide 4.0: Configuration and مستقرسازی Management Testing
- OWASP Cross-Site Request Forgery cheat sheet
- OWASP XML External Entity Prevention Cheat Sheet - General Guidance* JSON Web توکن s (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

V14: الزامات وارسی پیکربندی

هدف کنترل

مطمئن شوید که یک برنامه کاربردی وارسی‌شده دارای موارد زیر باشد:

- محیط ساخت امن، تکرارپذیر، خودکارشونده
- کتابخانه شخص ثالث، مدیریت وابستگی و پیکربندی به‌گونه‌ای که مؤلفه‌های منقضی و ناامن در برنامه کاربردی وجود نداشته باشند.
- پیکربندی امن به‌صورت پیش‌فرض، به‌گونه‌ای که مدیران و کاربران باید وضعیت امنیتی پیش‌فرض را تضعیف کنند.

پیکربندی برنامه خارج از جعبه (از دید بیرونی) باید در اینترنت ایمن باشد.

V14.1: ساخت

ساخت خط لوله[۱۶۵] ها پایه و اساس برای امنیتِ تکراری است. هربار که چیزی ناامن کشف می‌شود، می‌توان آن را در کد منبع، ساخت و استقرار اسکریپت‌ها و یا تست‌های خودکار حل کرد. ما به شدت شما را تشویق می‌کنیم به استفاده از ساخت خط لوله‌هایی با امنیت خودکار و کنترل‌های وابستگی که هشدار می‌دهند و ساختن را به‌منظور جلوگیری از مشکلات امنیتی رایج که منجر به استقرار آن‌ها در مرحله تولید می‌شود، متوقف می‌سازند. مراحل دستی ساخت مستقیماً باعث اشتباهات امنیتی اجتناب‌پذیری می‌شود.

همان‌طور که صنعت به یک مدل DevSecOps حرکت می‌کند، مهم است که از دردسترس بودن و یکپارچگی استقرار و پیکربندی برای دستیابی به حالت "خوب شناخته شده" اطمینان حاصل کنیم. در گذشته، اگر یک سیستم هک شده بود، روزها و ماه‌ها طول می‌کشید تا اثبات شود که هیچ نفوذ بیشتری صورت نگرفته است. امروزه، با ظهور زیرساخت‌های تعریف‌شده توسط نرم‌افزار و راه‌اندازی‌های سریع A/B بدون هیچ زمان خاموشی و ساخت کانتینرهای خودکار، ممکن است به‌طور خودکار و مداوم، ساخت و مقاوم‌سازی و استقرار یک جایگزین"خوب شناخته‌شده" را برای هر سیستم آسیب‌دیده انجام داد.

اگر مدل‌های سنتی هنوز مورد استفاده قرار گیرند، اقدامات دستی به منظور محکم‌کردن[۱۶۶] و پشتیبان‌گیری از پیکربندی‌ها باید صورت گیرد تا سیستم‌های آسیب‌دیده سریعاً با سیستم‌های با یکپارچگی بالا و غیرآسیب‌دیده جایگزین شوند.

تصدیق این بخش نیازمند یک سیستم ساخت خودکار و دسترسی به اسکریپت‌های ساخت و نصب دارد.

#	توضیح	L1	L2	L3	CWE

[165] Pipeline
[166] Harden

۱۴.۱.۱	تأیید کنید که فرآیندهای ساخت و استقرار برنامه به روش امن و قابل تکرار انجام شود، مانند CI/CD خودکار، مدیریت پیکربندی خودکار و اسکریپت‌های استقرار خودکار.		✓	✓	
۱۴.۱.۲	تأیید کنید که پرچم‌های کامپایلر به نحوی پیکربندی شده‌اند تا تمامی محافظت‌ها و هشدارهای سرریزی بافر را فعال کنند، شامل تصادفی‌سازی پشته، جلوگیری از اجرای داده و متوقف کردن ساخت اگر یک اشاره‌گر ناامن، حافظه، فرمت رشته، عدد صحیح، یا عملیات رشته یافت شوند.		✓	✓	۱۲۰
۱۴.۱.۳	تأیید کنید که پیکربندی سرویس‌دهنده بر اساس توصیه‌های برنامه سرویس‌دهنده و چارچوب مورد استفاده، امن باشد.		✓	✓	۱۶
۱۴.۱.۴	تأیید کنید که برنامه، پیکربندی و همه وابستگی‌ها قابلیت استقرار مجدد توسط اسکریپت‌های استقرار خودکار را داشته باشند و آن‌ها از یک کتابچه مستندشده و آزمایش‌شده در یک زمان معقول ساخته شده باشند، یا از پشتیبان‌ها در زمان مناسب ترمیم شده باشند.		✓	✓	
۱۴.۱.۵	تأیید کنید که مدیران احرازاصالت شده بتوانند یکپارچگی تمامی پیکربندی‌های امنیتی مرتبط را وارسی کنند تا هرگونه دستکاری را متوجه شوند.			✓	

V14.2: وابستگی‌ها[۱۶۷]

مدیریت وابستگی برای هر عملیاتی بر روی هر نوع برنامه کاربردی حیاتی است. عدم موفقیت در به‌روز نگه‌داشتن وابستگی‌های منسوخ یا ناامن دلیل اصلی بزرگ‌ترین و گران‌ترین حملات تا به امروز است.

نکته: در سطح ۱، انطباق ۱۴.۲.۱ مربوط به مشاهدات یا تشخیص کتابخانه‌ها و مؤلفه‌های سمت مشتری است، نه تحلیل کد ایستای دقیق‌تر زمان ساخت یا تحلیل وابستگی. تکنیک‌های دقیق‌تر در صورت نیاز می‌توانند با یک مصاحبه کشف شوند.

#	توضیح	L1	L2	L3	CWE
۱۴.۲.۱	تأیید کنید تمامی اجزا به‌روز باشند، ترجیحاً به‌وسیله‌ی یک چک‌کننده وابستگی در زمان ساخت یا کامپایل.	✓	✓	✓	۱۰۲۶

[167] Dependency

۱۴.۲.۲	تأیید کنید تمامی ویژگی‌های غیرضروری، مستندات، نمونه‌ها، پیکربندی‌ها حذف شوند، مانند برنامه‌های کاربردی نمونه، مستندات سکو و کاربران پیش‌فرض یا نمونه.	✓	✓	✓	۱۰۰۲
۱۴.۲.۳	تأیید کنید اگر دارایی‌های برنامه، مانند کتابخانه‌های جاوااسکریپت، CSS style sheets یا فونت‌های وب، در یک شبکه توزیع محتوا (CDN) به صورت خارجی یا ارائه دهنده خارجی میزبانی می‌شوند، SRI[۱۶۸] برای اعتبارسنجی یکپارچگی دارایی استفاده شود.	✓	✓	✓	۷۱۴
۱۴.۲.۴	تأیید کنید که مؤلفه‌های شخص ثالث از مخازن از پیش تعریف شده که مورد اعتماد هستند و به‌صورت مداوم نگه‌داری می‌شوند، آمده باشند.		✓	✓	۸۲۹
۱۴.۲.۵	تأیید کنید که یک کاتالوگ موجودی برای تمامی کتابخانه‌های مورداستفاده نگهداری می‌شود.		✓	✓	
۱۴.۲.۵	تأیید کنید که سطح حمله توسط کتابخانه‌های شخص ثالث sandboxing و encapsulating کاهش می‌یابد تا تنها رفتار مورد نیاز را به برنامه کاربردی افشا کند.		✓	✓	۲۶۵

V14.3: الزامات افشای امنیت ناخواسته

پیکربندی‌ها برای مرحله تولید باید ایمن باشد تا در برابر حملات معمول محافظت نماید، مانند کنسول‌های دیباگ (اشکال زدایی) که زمینه را برای حملاتی مانند cross site scripting (XSS) و بارگذاری فایل از راه دور (RFI) فراهم می‌کنند، و برای حذف کردن "آسیب پذیری‌های" کشف اطلاعات ناچیز که در بسیاری از گزارش های آزمایش نفوذ هستند. بسیاری از این مسائل به‌ندرت به‌عنوان یک خطر قابل توجه محسوب می شوند، ولی آن‌ها با آسیب‌پذیری‌های دیگر هم پیوند می‌خورند. اگر این اشکالات به‌طور پیش‌فرض وجود نداشته باشند، قبل از موفق شدن بسیاری از حملات جلوی آن‌ها را می‌گیرد.

#	توضیح	L1	L2	L3	CWE
۱۴.۳.۱	تأیید کنید که پیام‌های خطای سرویس‌دهنده وب یا سرویس‌دهنده برنامه و چارچوب به‌نحوی پیکربندی شده باشند تا پاسخ‌های عملیاتی و پاسخ‌های شخصی‌سازی‌شده را با حذف هرگونه افشای اطلاعات ناخواسته ارائه دهند.	✓	✓	✓	۲۰۹

[168] Subresource Integrity

۱۴.۳.۲	تأیید کنید که حالت‌های دیباگ (اشکال‌زدایی) سرویس‌دهنده وب یا سرویس‌دهنده برنامه و چارچوب برنامه در محیط تولید غیرفعال باشند تا ویژگی‌های دیباگ،کنسول‌های توسعه‌دهنده و افشاگرهای امنیتی ناخواسته غیرفعال باشند.	✓	✓	✓	۴۹۷
۱۴.۳.۳	تأیید کنید که سرآیندهای HTTP یا هر قسمت از پاسخ HTTP هیچ‌گونه جزئیاتی از اطلاعات نسخه مؤلفه‌های سیستم را افشا نکنند.	✓	✓	✓	۲۰۰

V14.4: الزامات سرآیند های امنیتی HTTP

#	توضیح	L1	L2	L3	CWE
۱۴.۴.۱	تأیید کنید که هر پاسخ HTTP دارای سرآیندی تحت عنوان content type باشد که یک مجموعه کاراکتر ایمن (به‌عنوان مثال، UTF-8،ISO 8859-1) را مشخص کند.	✓	✓	✓	۱۷۳
۱۴.۴.۲	تأیید کنید که تمامی پاسخ‌های API شامل Content-Disposition: attachment; filename="api.json" باشند (یا هرگونه نام فایل مناسب برای نوع محتوای خود)	✓	✓	✓	۱۱۶
۱۴.۴.۳	تأیید کنید که به‌کارگیری سیاست امنیت محتوا (CSPv2) کمک می‌کند که تأثیر حملات XSS مانند آسیب‌پذیری‌های تزریق HTML، DOM، JSON و JavaScript کاهش یابد.	✓	✓	✓	۱۰۲۱
۱۴.۴.۴	تأیید کنید که تمامی پاسخ‌ها دارای سرآیند X-Content-Type-Options: nosniff باشد.	✓	✓	✓	۱۱۶
۱۴.۴.۵	تأیید کنید که سرآیند HTTP Strict Transport Security در تمامی پاسخ‌ها و در تمامی زیردامنه ها قرار داده شوند. مانند Strict-Transport-Security: max-age=15724800;.	✓	✓	✓	۵۲۳
۱۴.۴.۶	تأیید کنید که سرآیند مناسب "Referrer-Policy" قرار داده شود. مانند "no-referrer" یا "same-origin".	✓	✓	✓	۱۱۶
۱۴.۴.۷	تأیید کنید که سرآیند مناسب X-Frame-Options یا Content-Security-Policy: frame-ancestors برای سایت‌هایی که محتوای آن‌ها نباید از سایت‌های شخص ثالث تعبیه شود، قرار داده شود.	✓	✓	✓	۳۴۶

V14.5: الزامات اعتبارسنجی سرآیند درخواست HTTP

#	توضیح	L1	L2	L3	CWE
۱۴.۵.۱	تأیید کنید که سرویس‌دهنده برنامه تنها متدهایی از HTTP را قبول می‌کند که درحال استفاده توسط برنامه یا API باشد، شامل pre-flight OPTIONS.	✓	✓	✓	۷۴۹
۱۴.۵.۲	تأیید کنید منشاء سرآیند Origin برای احرازاصالت یا تصمیم‌گیری برای کنترل دسترسی استفاده نمی‌شود، چون این سرآیند می‌تواند به‌آسانی توسط مهاجم تغییر یابد.	✓	✓	✓	۳۴۶
۱۴.۵.۳	تأیید کنید که سرآیند Access-Control-Allow-Origin CORS[169] از یک لیست سفید برای دامنه های مورد اطمینان استفاده می‌کند و با آن لیست تطبیق می‌دهد و از مقدار null برای origin پشتیبانی نمی‌کند.	✓	✓	✓	۳۴۶
۱۴.۵.۴	تأیید کنید که سرآیندهای HTTP توسط پروکسی های معتبر یا دستگاه‌های SSO اضافه شوند، مانند یک توکن bearer که توسط برنامه تأیید شده است.		✓	✓	۳۰۶

منابع

برای اطلاعات بیشتر به منابع زیر رجوع شود:

- OWASP Testing Guide 4.0: Testing for HTTP Verb Tampering
- Adding Content-Disposition to API responses helps prevent many attacks based on misunderstanding on the MIME type between client and server, and the "filename" option specifically helps prevent Reflected File Download attacks.
- Content Security Policy Cheat Sheet
- Exploiting CORS misconfiguration for BitCoins and Bounties
- OWASP Testing Guide 4.0: Configuration and مستقرسازی Management Testing
- Sandboxing third party components

[169] Cross-domain resource sharing

پیوست A: واژه‌نامه

- **2FA** - احرازاصالت دوعامله (2FA) سطح دوم احرازاصالت را برای ورود به حساب کاربری اضافه می‌کند.
- **Address Space Layout Randomization (ASLR)** - تکنیکی که اکسپلویت اشکالات خرابی حافظه را مشکل می‌سازد.
- **Application Security** - امنیت در سطح برنامه کاربردی، به‌جای تمرکز برروی سیستم‌عامل یا شبکه‌های متصل، برروی تحلیل اجزایی تمرکز می‌کند که لایه‌ی برنامه کاربردی از مدل مرجع OSI را مورد تهدید قرار می‌دهند.
- **Application Security Verification** - ارزیابی فنی یک برنامه با استاندارد OWASP ASVS.
- **Application Security Verification Report** - گزارشی که نتایج کلی و تحلیل‌های موردپشتیبانی تولیدشده توسط تأییدکننده برای یک برنامه خاص را مستند می‌کند.
- **Authentication** - وارسی یک هویت ادعاشده توسط کاربر برنامه کاربردی.
- **Automated Verification** - استفاده از ابزارهای خودکار (ابزارهای تحلیل پویا، ابزارهای تحلیل ایستا یا هر دو) که امضای آسیب‌پذیری‌ها برای یافتن مشکلات استفاده می‌کند.
- **Black box testing** - یک روش تست نرم‌افزار است که عملکرد یک برنامه را بدون در نظر گرفتن ساختار های داخلی یا طرز کار آن، بررسی می‌کند.
- **Component** - یک واحد تشکیل‌شده از کد که با واسط‌های دیسک و شبکه مرتبط است و با دیگر اجزا ارتباط برقرار می‌کند.
- **Cross-Site Scripting** (XSS) - یک آسیب پذیری امنیتی که معمولاً در برنامه های وب یافت می‌شود و باعث تزریق اسکریپت‌های سمت کاربر در محتوا می‌شود.
- **Cryptographic module** - سخت‌افزار، نرم‌افزار و/یا سفت‌افزار که الگوریتم‌های رمزنگاری را پیاده‌سازی کرده و/یا کلیدهای رمزنگاری را تولید کند.
- **CWE** - CWE[170] یک لیست توسعه‌یافته از نقاط ضعف امنیتی نرم‌افزار است و به‌عنوان یک زبان مشترک و معیار اندازه‌گیری برای ابزارهای امنیتی نرم‌افزاری و همچنین به‌عنوان مبنایی برای شناسایی ضعف، کاهش و جلوگیری عمل می‌کند.
- **DAST** - DAST[171] فن‌آوری‌هایی است که برای شناسایی شرایطی که حاکی از وجود آسیب‌پذیری امنیتی در یک نرم‌افزار در حالت اجرا، طراحی شده‌اند.
- **Design Verification** - ارزیابی فنی معماری امنیتی یک برنامه کاربردی

---

[170] Common Weakness Enumeration
[171] Dynamic application security testing

- **Dynamic Verification** - استفاده از ابزارهای خودکار که از امضاهای آسیب‌پذیری‌ها برای یافتن مشکلات در حین اجرای یک برنامه استفاده می‌کنند.
- **Globally Unique Identifier** (GUID) - یک شماره مرجع منحصر به فرد که به‌عنوان یک شناسه در نرم‌افزار استفاده می‌شود.
- **Hyper Text Transfer Protocol** (HTTPS) - یک پروتکل برنامه کاربردی برای سیستم‌های اطلاعاتی توزیع‌شده، مشارکتی و فرارسانه‌ای است. این پروتکل مبنای ارتباطات داده برای وب جهانی است.
- **Hardcoded keys** - کلیدهای رمزنگاری که در فایل سیستم، کد، کامنت‌ها یا فایل‌ها ذخیره می‌شوند.
- **Input Validation** – کانونی‌سازی و اعتبارسنجی ورودی غیرقابل اعتماد کاربر.
- **Malicious Code** - کدی که در یک برنامه کاربردی در طول توسعه آن نادانسته توسط مالک برنامه قرار داده شده است و سیاست امنیتی مورد نظر برنامه را رد می‌کند. همانند بدافزارها از قبیل ویروس یا کرم نیست!
- **Malware** - کد اجرایی که بدون آگاهی کاربر برنامه کاربردی یا مدیر در برنامه حین اجرای برنامه به آن وارد می‌شود.
- **Open Web Application Security Project** (OWASP) - پروژه امنیت وب اپلیکیشن یک جامعه آزاد و باز در جهان است که بر بهبود امنیت نرم‌افزار کاربردی تمرکز دارد. مأموریت ما این است که امنیت برنامه را قابل مشاهده کنیم تا افراد و سازمان‌ها بتوانند تصمیمات آگاهانه در مورد ریسک‌های امنیتی برنامه ها بگیرند. به لینک زیر رجوع کنید: https://www.owasp.org/
- **Personally Identifiable Information** (PII) - اطلاعاتی است که می‌تواند به تنهایی یا با سایر اطلاعات برای شناسایی، تماس یا یافتن یک فرد یا شناسایی یک شخص در زمینه مورد استفاده قرار گیرد.
- **PIE** – Position-independent executable (PIE) - یک قسمت از کد ماشین است که در جایی در حافظه اصلی قرار می گیرد و به درستی بدون در نظر گرفتن آدرس مطلق آن اجرا می‌شود.
- **PKI** – Public Key Infrastructure (PKI) - آرایشی است که کلیدهای عمومی را با هویت‌های مربوط به اشخاص مرتبط می‌کند. این ارتباط توسط یک پروسه ثبت و صدور گواهی‌نامه، توسط مرجع صدور گواهی‌نامه (CA) صورت می‌گیرد.
- **SAST** – مجموعه‌ای از فن‌آوری‌های طراحی‌شده برای تحلیل کد منبع برنامه، بایت‌کد و فایل‌های دودویی برای برنامه‌نویسی و شرایط طراحی است که نشان‌دهنده آسیب‌پذیری‌های امنیتی است. راه‌حل‌های SAST یک برنامه کاربردی را از "داخل" در یک حالت غیراجرا تحلیل می‌کند.
- **SDLC** - چرخه زندگی توسعه نرم‌افزار.
- **Security Architecture** - انتزاعی از طراحی یک برنامه کاربردی که شناسایی و توصیف می‌کند که کجا و چگونه کنترل‌های امنیتی استفاده می‌شود، و همچنین موقعیت و حساسیت داده‌های کاربر و برنامه را شناسایی و توصیف می‌کند.
- **Security Configuration** - پیکربندی زمان اجرا یک برنامه کاربردی که بر نحوه استفاده از کنترل‌های امنیتی تأثیر می‌گذارد.

- **Security Control** - یک تابع یا جزء که یک بررسی امنیتی را انجام می‌دهد (مثلا بررسی کنترل دسترسی) یا هنگامی‌که نتیجه یک اثر امنیتی فراخوانده می‌شود (برای مثال ایجاد یک رکورد حسابرسی).
- **SQL Injection (SQLi)** - یک تکنیک تزریق کد که برای حمله به برنامه های کاربردی تحت هدایت داده استفاده می‌شود، که در آن عبارت SQL مخرب در ورودی وارد می‌شود.
- **SSO Authentication** - زمانی رخ می‌دهد که یک کاربر وارد یک برنامه می‌شود و سپس به‌طور خودکار وارد برنامه های دیگر می‌شود بدون آنکه مجدداً احرازاصالت نماید. برای مثال هنگامی‌که وارد گوگل می‌شوید، وقتی میخواهید به سرویس‌های دیگر گوگل مثل Youtube، Google Docs و یا Gmail دسترسی پیدا کنید، به صورت خودکار وارد میشوید.
- **Threat Modeling** - یک روش متشکل از توسعه معماری‌های امنیتی به‌طور فزاینده‌ای برای شناسایی عوامل تهدید، مناطق امنیتی، کنترل‌های امنیتی و مهمات فنی و تجاری است.
- **Transport Layer Security** – پروتکل‌های رمزنگاری که امنیت ارتباط را بر روی یک اتصال شبکه تأمین می‌کنند.
- **URI/URL/URL fragments** - Uniform Resource Identifier یک رشته از کاراکترهایی است که برای شناسایی یک نام یا یک منبع وب استفاده می‌شود. Uniform Resource Locator به‌عنوان یک مرجع به منبع استفاده می‌شود.
- **Verifier** - فرد یا تیمی که در حال بررسی یک درخواست با الزامات OWASP ASVS است.
- **Whitelist** - لیستی از داده‌ها یا عملیات مجاز، به‌عنوان مثال یک لیست از کاراکترهایی که برای انجام اعتبار ورودی مجاز هستند.
- **X.509 Certificate** - یک گواهی X.509 یک گواهی دیجیتالی است که از استاندارد PKI که به‌صورت گسترده و بین‌المللی پذیرفته شده است استفاده می‌کند، برای تأیید این‌که کلید عمومی متعلق به کاربر، کامپیوتر، یا سرویسی است که در درون گواهی‌نامه قرار دارد.

پیوست B: مراجع

پروژه‌های OWASP زیر برای کاربران این استاندارد مفید خواهند بود:

OWASP Core Projects

1. OWASP Top 10 Project: https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project
2. OWASP Testing Guide: https://www.owasp.org/index.php/OWASP_Testing_Project
3. OWASP Proactive Controls: https://www.owasp.org/index.php/OWASP_Proactive_Controls
4. OWASP Security Knowledge Framework: https://www.owasp.org/index.php/OWASP_Security_Knowledge_Framework
5. OWASP Software Assurance Maturity Model (SAMM): https://www.owasp.org/index.php/OWASP_SAMM_Project

Mobile Security Related Projects

1. OWASP Mobile Security Project: https://www.owasp.org/index.php/OWASP_Mobile_Security_Project
2. OWASP Mobile Top 10 Risks: https://www.owasp.org/index.php/Projects/OWASP_Mobile_Security_Project_-_Top_Ten_Mobile_Risks
3. OWASP Mobile Security Testing Guide: https://www.owasp.org/index.php/OWASP_Mobile_Security_Testing_Guide

OWASP Internet of Things related projects

1. OWASP Internet of Things Project: https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project

OWASP Serverless projects

1. OWASP Serverless Project: https://www.owasp.org/index.php/OWASP_Serverless_Top_10_Project

Others

Similarly, the following web sites are most likely to be useful to users/adopters of this standard

1. SecLists Github: https://github.com/danielmiessler/SecLists
2. MITRE Common Weakness Enumeration: https://cwe.mitre.org/
3. PCI Security Standards Council: https://www.pcisecuritystandards.org
4. PCI Data Security Standard (DSS) v3.2.1 Requirements and Security Assessment Procedures: https://www.pcisecuritystandards.org/documents/PCI_DSS_v3-2-1.pdf
5. PCI Software Security Framework - Secure Software Requirements and Assessment Procedures: https://www.pcisecuritystandards.org/documents/PCI-Secure-Software-Standard-v1_0.pdf
6. PCI Secure Software Lifecycle (Secure SLC) Requirements and Assessment Procedures: https://www.pcisecuritystandards.org/documents/PCI-Secure-SLC-Standard-v1_0.pdf

پیوست C: الزامات وارسی اینترنت اشیاء

این بخش در ابتدا در شاخه اصلی بود، اما با کارهایی که تیم OWASP IoT انجام داده است، منطقی نیست که دو استاندارد مختلف در مورد موضوع نگهداری شود. برای نسخه 4.0، این بخش را به پیوست انتقال می‌دهیم و از همه کسانی که به این بخش نیاز دارند می‌خواهیم که از این پیوست به‌جای OWASP IoT شاخه اصلی استفاده کنند.

هدف کنترل

دستگاه‌های embedded/IoT باید:

- کنترل‌های امنیتی که در دستگاه است باید هم سطح با سرویس‌دهنده پیدا شوند با مجبور کردن کنترل‌های امنیتی در محیط قابل اعتماد.
- اطلاعات حساس ذخیره شده بر روی دستگاه باید به نحوی امن با استفاده از ذخیره‌سازی پشتیبان سخت‌افزاری مانند عناصر امن انجام شود.
- تمام اطلاعات حساس منتقل‌شده از دستگاه باید از امنیت لایه حمل و نقل استفاده کنند.

الزامات وارسی امنیت

#	توضیح	L1	L2	L3	Since
C.1	تأیید کنید که رابط‌های لایه اشکال‌زدایی برنامه مانند USB، UART و سایر انتقال‌دهندگان سریال، غیرفعال یا توسط گذرواژه پیچیده محافظت می‌شوند.	✓	✓	✓	4.0
C.2	تأیید کنید که کلیدهای رمزنگاری و گواهی‌نامه‌ها برای هر دستگاه شخصی منحصربه‌فرد هستند.	✓	✓	✓	4.0
C.3	تأیید کنید که کنترل‌های حفاظت از حافظه مانند ASLR و DEP توسط سیستم‌عامل جاسازی‌شده/IoT فعال شده باشد (اگر قابل اجرا است).	✓	✓	✓	4.0
C.4	اطمینان حاصل کنید که رابط‌های اشکال‌زدائی در تراشه مانند JTAG یا SWD غیرفعال هستند و یا سازوکار حفاظت در دسترس فعال و مناسب پیکربندی شده است.	✓	✓	✓	4.0
C.5	اطمینان حاصل کنید که اجرای قابل اعتماد پیاده‌سازی و فعال شده باشد، اگر در SoC یا CPU دستگاه موجود است.	✓	✓	✓	4.0

C.6	اطمینان حاصل کنید که داده‌های حساس، کلیدهای خصوصی و گواهی‌نامه‌ها به‌صورت ایمن در یک عنصر امن، TPM، TEE[172] ذخیره می‌شوند یا با استفاده از رمزنگاری قوی محافظت می‌شوند.	✓	✓	✓	4.0
C.7	تأیید کنید که برنامه‌های سفت‌افزار از داده در حال انتقال با استفاده از امنیت لایه انتقال، محافظت می‌کند.	✓	✓	✓	4.0
C.8	تأیید کنید که برنامه‌های سفت‌افزار امضای دیجیتال اتصالات سرویس‌دهنده را تأیید می‌کنند.	✓	✓	✓	4.0
C.9	تأیید کنید که ارتباطات بی‌سیم دو طرفه احرازاصالت می‌شوند.	✓	✓	✓	4.0
C.10	تأیید کنید که ارتباطات بی‌سیم از طریق کانال رمزنگارشده ارسال می‌شوند.	✓	✓	✓	4.0
C.11	تأیید کنید که هرگونه استفاده از توابع ممنوع C توسط توابع معادل امن جایگزین شده‌اند.	✓	✓	✓	4.0
C.12	تأیید کنید که هر سفت‌افزار یک نرم‌افزار متشکل از کاتالوگی از اجزای شخص ثالث، نسخه‌بندی و آسیب‌پذیری‌های منتشر شده را نگه می‌دارد.	✓	✓	✓	4.0
C.13	تأیید کنید که تمامی کدها شامل دودویی‌های شخص ثالث، کتابخانه‌ها و چارچوب‌ها از اطلاعات محرمانه هاردکدشده بازبینی شده‌اند. (backdoors)	✓	✓	✓	4.0
C.14	تأیید کنید که اجزای برنامه و سفت‌افزار به تزریق دستورات سیستم‌عاملی با به‌کارگیری دستورات shell، اسکریپت‌ها واکنش نشان نمی‌دهد.	✓	✓	✓	4.0
C.15	تأیید کنید که برنامه‌های سفت‌افزار امضای دیجیتال را به سرویس‌دهنده‌های مورد اعتماد مرتبط کرده‌اند.		✓	✓	4.0
C.16	تأیید کنید که قابلیت‌های مقاومت در برابر دستکاری و/یا شناسایی دستکاری وجود داشته باشند.		✓	✓	4.0
C.17	تأیید کنید که هر فن‌آوری حفاظت از مالکیت معنوی ارائه شده توسط سازنده تراشه فعال است.		✓	✓	4.0
C.18	تأیید کنید که کنترل‌های امنیتی برای جلوگیری از مهندسی معکوس کردن سفت‌افزار وجود دارد (به‌عنوان مثال، حذف نمادهای اشکال زدایی).		✓	✓	4.0

[172] Trusted Execution Environment

C.19	تأیید کنید که دستگاه قبل از اقدام به بوت شدن، امضای تصویر[173] آن را تصدیق کند.		✓	✓	4.0
C.20	تأیید کنید که فرآیند به‌روزرسانی سفت‌افزار به حملات زمان بررسی در مقابل زمان استفاده آسیب‌پذیر نیست.		✓	✓	4.0
C.21	تأیید کنید که دستگاه از امضای کد استفاده می‌کند و قبل از نصب، فایل‌های ارتقاء سیستم‌عامل را تأیید می‌کند.		✓	✓	4.0
C.22	تأیید کنید که دستگاه را نمی‌توان به نسخه های قدیمی (anti-rollback) از سفت‌افزار معتبر کاهش داد.		✓	✓	4.0
C.23	استفاده رمزنگارانه امن از تولیدکننده اعداد شبه‌تصادفی را بر روی دستگاه تأیید کنید (برای مثال، از تولیدکننده‌های ارائه شده توسط خود تراشه برای تولید اعداد تصادفی استفاده شود).		✓	✓	4.0
C.24	تأیید کنید که سیستم‌عامل می‌تواند به‌روزرسانی خودکار سیستم‌عامل بر اساس یک برنامه از پیش تعریف‌شده انجام دهد.		✓	✓	4.0
C.25	تأیید کنید که دستگاه پس از تشخیص دستکاری یا دریافت پیام نامعتبر، سیستم‌عامل و اطلاعات حساس را پاک می‌کند.			✓	4.0
C.26	تأیید کنید که تنها از میکروکنترلرهایی استفاده می‌شود که از قابلیت غیرفعال‌کردن رابط‌های اشکال‌زدایی پشتیبانی می‌کنند.			✓	4.0
C.27	تأیید کنید که تنها از میکروکنترلرهایی استفاده می‌شود که محافظت اساسی ازحملات decapping و side channel را ارائه می‌دهند.			✓	4.0
C.28	تأیید کنید که ردیاب‌های حساس در لایه های بیرونی مدار برد قرار نگرفته باشند.			✓	4.0
C.29	تأیید کنید که ارتباط بین تراشه‌ها رمزگذاری شده است (به‌عنوان مثال، ارتباطات Main board با Daughter board).			✓	4.0
C.30	تأیید کنید که دستگاه از امضای کد استفاده می‌کند و قبل از اجرا کد را تأیید می‌کند.			✓	4.0
C.31	اطمینان حاصل کنید که اطلاعات حساس که در حافظه نگه‌داری می شوند، به زودی به صفر رونویسی می‌شوند، به محض آن‌که دیگر لازم نباشد.			✓	4.0

[173] Image

C.32	اطمینان حاصل کنید که برنامه‌های سفت‌افزار از کانتینرهای هسته برای جداسازی بین برنامه‌ها استفاده می‌کنند.			✓	4.0
C.33	تأیید کنید که پرچم‌های امنیتی کامپایلر مانند fPIE, -fstack-protector-all, -Wl,-z,noexecstack, -Wl,-z,noexecheap -Wl,-z,noexecheap برای ساختن سفت‌افزار پیکربندی شده باشند.			✓	4.0
C.34	تأیید کنید که میکروکنترلرها با حفاظت از کد پیکربندی شده باشند (در صورت وجود).			✓	4.0

منابع

برای اطلاعات بیشتر به منابع زیر رجوع شود:

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