OWASP

Open Web Application Security Project

استاندارد بررسی امنیت نرم‌افزار نسخه ۳.۰.۱

سپاسگزاری

درباره استاندارد

استاندارد بررسی امنیت نرم‌افزار، فهرستی از نیازمندی‌های امنیت نرم‌افزار یا آزمون‌هایی است که می‌تواند توسط معمارها، توسعه‌دهنده‌ها، کارشناسان امنیت و حتی مشتریانی که می‌خواهند تعریف خود را از یک برنامه امن بازگو کنند، استفاده شود.

ترجمه استاندارد

این استاندارد به سفارش مرکز ماهر ایران توسط آزمایشگاه تخصصی آپا دانشگاه فردوسی مشهد تهیه و ترجمه شده است، که می‌تواند به‌عنوان مرجع بررسی، در فرآیند آزمون نفوذپذیری مورد استفاده قرار گیرد.

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

pourali@cert.um.ac.ir

مجوز و حق کپی‌رایت

نسخه ۳.۰، ۲۰۱۶

رهبران پروژه	نویسندگان ارشد	همکاران و بازنگران
Andrew van der Stock	Jim Manico	Abhinav Sejpal
Daniel Cuthbert		Anthony Weems
		Ari Kesaniemi
		Boy Baukema
		Colin Watson
سرپرست مترجمان	مترجمان	Cristinel Dumtiru
Sajjad Pourali	Mohammad Kahani	David Ryan
	Mahdi Bagheri	Francois-Eric Guyomarc’h
	Saeed KhademiDoroh	Gary Robinson
	Sajjad Pourali	Glenn Ten Cate
		James Holland
		Kelby Ludwig
		Nathan Sportsman
		Martin Knobloch
		Raoul Endres
		Ravishankar S
		Riccardo Ten Cate
		Robert Erbes
		Roberto Martelloni
		Ryan Dewhurst
		Stephen de Varies
		Steven van der Baan

نسخه ۲.۰، ۲۰۱۴

رهبران پروژه	نویسندگان ارشد	همکاران و بازنگران
Daniel Cuthbert	Andrew van der Stock	Antonio Fontes
Sahba Kazerooni	Krishna Raja	Archangel Cuison
		Ari Kesaniemi
		Boy Baukema
		Colin Watson
سرپرست مترجمان	مترجمان	Dr Emin Tatli
Sajjad Pourali	Abbas Javan Jafari	Etienne Stalmans
	Sajjad Pourali	Evan Gaustad
		Jeff Sergeant
		Jerome Athias
		Jim Manico
		Mait Peekma
		Pekka Sillanpaa
		Safuat Hamdy
		Scott Luc
		Sebastien Deleersmyder

نسخه ۱.۰، ۲۰۰۹

رهبران پروژه	نویسندگان ارشد	همکاران و بازنگران

Andrew van der Stock	Jim Manico	Mike Boberski
Barry Boyd		Jeff Williams
Bedirhan Urgun		Dave Wichers
Colin Watson		
Dan Cornell		
Dave Hausladen		
Dave van Stein		
Dr. Sarbari Gupta		
Dr. Thomas Braun		
Eoin Keary		
Gaurang Shah		
George Lawless		
Jeff LOSapio		
Jeremiah Grossman		
John Martin		
John Steven		
Ken Huang		
Ketan Dilipkumar Vyas		
Liz Fong Shouvik Bardhan		
Mandeep khera		
Matt Persson		
Nam Nguyen		
Paul Douthit		
Pierre Parrend		
Richard Campbell		
Scott Matsumoto		
Stan Wisseman		
Stephen de Vries		
Steve Coyle		
Terrie Diaz		
Theodore Winograd		

پیشگفتار

به نسخه سوم از استاندارد بررسی امنیت نرم‌افزار (ASVS) خوش آمدید. ASVS به‌صورت جامعه‌محور، سعی کرده است چارچوبی[1] مناسب برای نیازمندی‌های امنیتی و همچنین کنترل‌های امنیتی ایجاد کند. این موارد بیشتر بر قانونمند کردن نیازمندی‌های کنترل امنیت تمرکز دارد. درواقع موارد استفاده اصلی از این استاندارد، هنگام طراحی، توسعه و آزمایش برنامه‌های تحت وب می‌باشد.

نسخه سوم ASVS از اهمیت بالایی در جامعه و دریافت بازخورد از صنعت برخوردار است. ما نیاز زیادی به وجود تجربیات واقعی یا تجربیاتی شبیه به دنیای حقیقی را حس کردیم. این کار باعث می‌شود تا تمامی افراد تازه وارد به این استاندارد، بتوانند برنامه‌ریزی مناسبی برای استفاده هرچه بهتر از آن داشته باشند. همچنین این کار باعث کمک به شرکت‌های موجود برای استفاده از تجربیات دیگران نیز می‌باشد.

طبق انتظار ما، به احتمال زیاد هرگز توافق صددرصدی بر سر این استاندارد نخواهد بود. تجزیه و تحلیل ریسک همیشه تا حدی ذهنی است که در صورت تلاش برای تعمیم، می‌تواند در یک اندازه متناسب برای تمام استانداردها باشد و این خود باعث ایجاد چالش می‌شود. با این حال، امیدواریم که آخرین به‌روزرسانی‌های انجام شده در این نسخه گامی در راستای مسیر درست باشد، و همچنین با تمامی مفاهیم معرفی شده در استاندارد صنعتی منافاتی نداشته باشد.

چه موارد جدیدی در نسخه ۳,۰ وجود دارند؟

در این نسخه، برای کاربردی‌تر شدن استاندارد، چندین بخش جدید از جمله پیکربندی، سرویس‌های وب، برنامه‌های مدرن (سمت مشتری)[2] و همچنین دستگاه‌های اینترنت اشیا اضافه شده است.

سعی بر این بوده است که در این استاندارد موارد تکراری حذف شوند. به‌عنوان مثال، اطمینان حاصل شود که یک توسعه‌دهنده موبایل نیازی به چندین بار آزمایش دوباره موارد مشابه نداشته باشد.

همچنین یک نقشه جامع برای فرهنگ لغت CWE طراحی کردیم. نقشه CWE می‌تواند برای شناسایی اطلاعاتی ازجمله احتمال سواستفاده (exploit)، نتیجه موفق یک exploit و همچنین، به‌طور گسترده در مورد درک بهتر این‌که اگر کنترل‌های امنیتی استفاده نشود یا به‌طور مؤثر استفاده نشده باشد و با ضعف همراه باشد، چه چیزی می‌تواند به سمت نادرستی حرکت کند، صحبت می‌کند.

درنهایت به جامعه رسیدیم و جلسات نظرسنجی و بازنگری را در AppSec EU 2015 (امنیت نرم‌افزار اروپا ۲۰۱۵) و جلسه کاری نهایی را در AppSec USA 2015 (امنیت نرم‌افزار آمریکا) برگزار کردیم تا این مستند درون خود، شمار زیادی از بازخوردهای اجتماعی را شامل شود. در طول بازنگری‌ها اگر تغییرات کنترلی قابل توجهی رخ می‌داد، کنترل جدید را ایجاد می‌کردیم و نسخه قبلی را به‌عنوان منسوخ‌شده و قدیمی می‌شناختیم. این کار را عمداً انجام دادیم تا مجدداً از آن نسخه‌های قدیمی استفاده نشود، چرا که نسخه‌های قدیمی خودشان باعث سردرگمی خواهند شد. همچنین یک نقشه کامل از تغییرات رخ داده در صفحه ضمیمه الف قرار دادیم.
روی هم رفته، این نسخه شامل یکی از تغییرات بزرگ در طول تاریخ استاندارد می‌باشد. امید است نسخه به‌روز استاندارد را پیدا کنید و به همان درستی که در تصور ما است از آن استفاده کنید.

[1] Framework
[2] Client side

استفاده از استاندارد بررسی امنیت برنامه

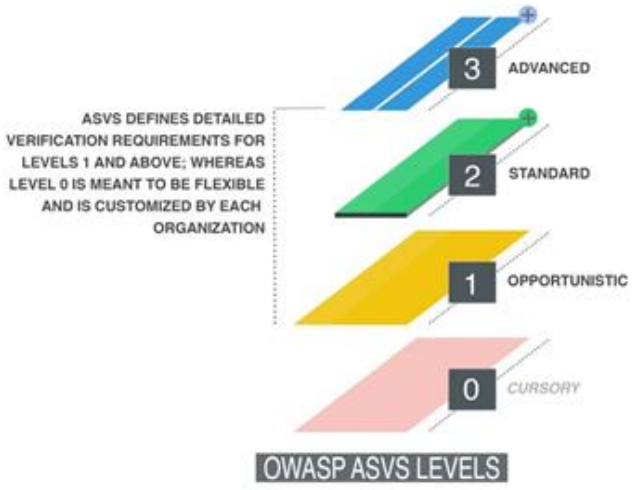
ASVS دو هدف مهم به شرح زیر دارد:

- به سازمان‌ها در راستای توسعه یک برنامه امن و حفاظت از آن کمک کند.
- به سرویس‌های امنیتی، فروشندگان ابزار امنیتی و همچنین مصرف‌کنندگان اجازه دهد تا نیازها و پیشنهادهای خود را مطرح کنند.

سطوح بررسی امنیتی برنامه

این استاندارد در سه سطح امنیتی به شرح زیر تعریف شده است که هر سطح نسبت به سطح قبلی موارد بیشتری را مورد بررسی قرارمی‌دهد.

- سطح اول ASVS برای تمامی نرم‌افزارها است.
- سطح دوم ASVS برای برنامه‌هایی است که اطلاعات حساسی دارند و طبیعتاً نیازمند حفاظت از اطلاعاتشان هستند.
- سطح سوم ASVS برای برنامه‌هایی است که بیشترین حساسیت را دارند – برنامه‌هایی که هر روزه در حال تراکنش‌های فراوان هستند. (ممکن است اطلاعات حساس پزشکی باشد یا هر برنامه دیگری که نیاز به بالاترین سطح امنیت دارد.)



چطور از این استاندارد استفاده کنیم؟

یکی از بهترین روش‌های استفاده از استاندارد این است که به کمک آن یک چک لیست[1] مخصوص برنامه یا سازمان خود طراحی کنید که بتوانید با کمک آن از امنیت برنامه یا سازمان خود مطمئن شوید. طراحی این چک لیست ASVS به شما کمک می‌کند تا تمرکز بیشتری بر روی نیازمندی‌های برنامه و محیط کاری خود داشته باشید.

سطح اول: فرصت طلبی

یک نرم‌افزار به شرطی به سطح اول ASVS می‌رسد که بتواند به‌طور مناسب از آسیب‌پذیری‌های گوناگون حفظ شود. این آسیب‌پذیری‌ها خیلی ساده کشف می‌شوند و در لیست ده آسیب‌پذیری برتر چک لیست OWASP یا دیگر چک لیست‌های امنیتی ذکر شده‌اند.

سطح اول معمولاً برای برنامه‌هایی است که به سطوح ساده اما صحیحی از کنترل نیاز دارند، یا برای ارایه یک روش تجزیه و تحلیل سریع برنامه‌های سازمانی، یا کمک به توسعه یک لیست اولویت‌بندی شده از نیازمندی‌های امنیتی به‌عنوان رفع بخشی از نیازمندی‌های برنامه‌ها مناسب است. سطح اول از کنترل‌ها می‌تواند هم به‌صورت خودکار (با استفاده از ابزار) و هم به‌صورت ساده و دستی، بدون دسترسی به کد منبع[2] تضمین شود. همچنین باید بدانید که سطح اول از نظر ما حداقل امنیتی است که هر برنامه‌ای باید رعایت کند.

در اغلب اوقات تهدید برنامه‌ها توسط مهاجمانی صورت می‌گیرد که از تکنیک‌های ساده و کم تلاش برای شناسایی و بهره‌برداری از آسیب‌پذیری‌ها استفاده می‌کنند و این موضوع در مقایسه با یک مهاجم مصمم که با انرژی بسیار زیاد برای هدف خود به‌طور متمرکز تلاش می‌کند، مورد بررسی است. اگر شما داده و اطلاعات مهمی داشته باشید به‌ندرت پیش می‌آید که در همین سطح اول از امنیت برنامه خود رضایت کافی داشته باشید.

سطح دوم : استاندارد

یک نرم‌افزار به‌شرطی به سطح دوم ASVS می‌رسد که بتواند از خود به خوبی در برابر خطرات ناشی از نرم‌افزارهای امروزی محافظت کند.

سطح دوم این اطمینان را به ما می‌دهد که کنترل‌های استفاده شده درون برنامه به‌موقع و مؤثر خواهد بود. سطح دوم معمولاً برای برنامه‌هایی مناسب است که معاملات مهم کسب‌وکار را شامل می‌شوند، از جمله آن‌هایی که پردازش اطلاعات مهم پزشکی، تجارت‌ها و کارکردها یا سایر پردازش‌های حساس را اجرا می‌کنند.

معمولاً تهدیدات برنامه‌های سطح دوم دارای آسیب‌پذیری و نقاط ضعفی است که معمولاً کشف و بهره‌برداری از آن‌ها توسط مهاجمان ماهر اتفاق می‌افتد، که استفاده از ابزار خاص و تکنیک‌هایی که بر روی آن‌ها مسلط هستند، آن‌را امکان‌پذیر می‌سازد.

سطح سوم : پیشرفته

سطح سوم ASVS در حال حاضر بالاترین سطح امنیتی از نظر ما می‌باشد. این سطح به‌طور معمول برای برنامه‌هایی مورد نیاز است که نیازمند سطح قابل توجهی از بررسی امنیتی هستند، همانند برنامه‌هایی که در حوزه‌های نظامی، بهداشت و ایمنی، زیرساخت‌های حیاتی و غیره استفاده می‌شوند.

[1] Checklist
[2] Source code

سازمان‌ها ممکن است سطح سوم ASVS را برای برنامه‌هایی که عملکردی حیاتی برای سازمان انجام می‌دهند استفاده کنند، مخصوصاً در جایی که شکست تأثیر مستقیم بر روی کارکرد سازمان یا حتی بقای سازمان بگذارد.
مثال راهنما برای کاربرد سطح سوم ASVS در اینجا آمده است. یک نرم‌افزار به شرطی به سطح سوم ASVS می‌رسد که نه تنها به‌طور مناسب از خود در برابر خطرات و آسیب‌پذیری‌های پیشرفته برنامه‌های امروزی دفاع کند، بلکه بتواند قوانین طراحی امنیتی مناسب را رعایت کند و در خود بگنجاند.

یک برنامه در سطح سوم ASVS نیازمند عمیق‌ترین تجزیه و تحلیل، معماری، برنامه‌نویسی، و آزمایش است که شامل تمام سطوح قبلی نیز می‌باشد. این برنامه امن به‌طور معقول ماژولار[1] می‌شود (به‌منظور راحتی، به‌عنوان مثال انعطاف‌پذیری، مقیاس‌پذیری و مهم‌تر از همه لایه‌های امنیتی) و هرکدام از این ماژول‌ها (که خود آن یا نمونه از آن توسط ارتباط شبکه از هم جدا می‌شوند) مسئولیت‌های امنیتی خود را برعهده دارند، (مسئولیت دفاع در عمیق‌ترین لایه‌ها) که باید به‌خوبی مستند شود. این مسئولیت‌ها شامل کنترل‌هایی برای اطمینان از محرمانه بودن (مانند رمزنگاری[2])، یکپارچگی (مانند معاملات و بررسی اعتبار ورودی[3])، در دسترس بودن (مانند به‌خوبی مدیریت کردن بار[4] سیستم)، احراز هویت (شامل احراز هویت بین چندین سیستم)، حسابرسی (همانند ثبت گزارش)، مجوزدهی و عدم تخلف می‌باشند.

پیاده‌سازی ASVS در عمل

پشت تهدیدات مختلف، انگیزه‌های مختلفی است. برخی از صنایع دارای اطلاعات و فناوری‌های منحصربه‌فرد و نیازمندی‌های منطبق با استانداردهای خاص می‌باشند.

در اینجا ما راهنمایی‌های مربوط به صنعت را در ارتباط با سطوح توصیه شده ASVS ارائه می‌دهیم. اگرچه برخی از معیارها منحصر به فرد هستند، اما تفاوت‌هایی در تهدیدها برای هر صنعت وجود دارد. موضوع مشترک در همه بخش‌های صنعت این است که مهاجمان فرصت طلب به دنبال برنامه‌هایی هستند که به‌راحتی آسیب‌پذیری آن‌ها را پیدا و از آن‌ها بهره‌برداری کنند، و این دلیل وجود سطح اول ASVS می‌باشد که باید در تمامی برنامه‌ها استفاده شود. این پیشنهاد یک نقطه شروع، به‌عنوان ساده‌ترین راه برای یافتن خطرات است. سازمان‌ها به‌شدت تشویق می‌شوند تا دیدگاهی عمیق بر ویژگی‌های ریسک‌پذیر منحصربه‌فرد خود مخصوصاً بر اساس ماهیت کسب‌وکار داشته باشند. در انتها، قسمت اصلی سطح سوم ASVS است و از آن برای مواردی استفاده می‌شود که ممکن است به امنیت انسان‌ها آسیب برسانند، یا زمانی‌که نقض کامل نرم‌افزار به‌شدت بر سازمان تأثیر بگذارد.

[1] Modulized
[2] Encryption
[3] Input validation
[4] Load

صنعت	مشخصات تهدید	توصیه‌های سطح اول L1	توصیه‌های سطح دوم L2	توصیه‌های سطح سوم L3
بیمه و مالی	اگر چه این بخش با تلاش مهاجمان فرصت‌طلب همراه است، اما اغلب به‌عنوان یک هدف ارزشمند توسط مهاجمان با انگیزه نیز مورد بررسی قرار می‌گیرد. چون این حملات بیشتر با انگیزه مالی رخ می‌دهد. معمولاً مهاجمان به‌دنبال اطلاعات حساس یا حساب کاربری هستند، که می‌توانند کلاه‌برداری کنند و یا به‌طور مستقیم با روش انتقال پول از طریق برنامه‌های کابردی، از این اطلاعات سوءاستفاده نمایند. تکنیک‌های این بخش اغلب شامل دزدیده شدن اعتبارنامه‌ها[1]، حمله به برنامه‌های کاربردی و مهندسی اجتماعی است. برخی مطابق ملاحظات عمده شامل استانداردهای امنیت داده‌ها در صنعت پرداخت کارت (PCIDSS)، Gramm Leech Act و Sarbanes-OxleyAct (SOX) عمل می‌کنند.	تمامی برنامه‌هایی که به شبکه دسترسی دارند.	برنامه هایی که دارای اطلاعات حساسی همچون شماره کارت بانکی و اطلاعات شخصی می‌باشند که می‌توانند مقدار مشخصی پول را از مسیر مشخصی انتقال دهند. به‌عنوان مثال: (یک): انتقال پول بین حساب‌های یک موسسه یا (دو): یک نوع کند و محدودتر در انتقال پول به‌عنوان مثال (ACH) یا (سه): انتقال از طریق سیم مثلاً در هارد دیسک‌ها که با محدودیت در یک بازه زمانی همراه است.	برنامه‌هایی که دارای اطلاعات حساس زیادی هستند یا اجازه انتقال اطلاعات با سرعت زیاد را دارند یا مبالغ هنگفت (مثلاً انتقال سیمی) و/یا انتقال مبالغ هنگفت به‌صورت یک تراکنش تنها یا به عنوان یک دسته از انتقال‌های کوچک‌تر
تولید، تخصصی، حمل و نقل، تکنولوژی، خدمات رفاهی، زیرساخت و	این صنایع ممکن است وجه‌های اشتراک زیادی نداشته باشند اما تهدیدکنندگانی که به احتمال بیشتر به این بخش از سازمان حمله می‌کنند، متمایل به اجرای حمله‌هایی با منابع، مهارت‌ها و زمان بیشتری می‌باشند. سیستم‌ها یا اطلاعات حساس را نمی‌توان به‌سادگی مکان‌یابی کرد و باید	تمامی برنامه هایی که به شبکه دسترسی دارند	برنامه‌هایی که شامل اطلاعات داخلی یا اطلاعاتی از کارمندانی که ممکن است در مهندسی اجتماعی مورد نفوذ واقع شوند. برنامه‌هایی که شامل	برنامه‌هایی که اطلاعات خیلی مهمی درون خود دارند، اسرار تجاری، اسرار دولتی (به‌عنوان مثال در ایالات متحده ممکن است هر چیز محرمانه‌ای به این

[1] Credentials

دفاع	توجه داشت که نیاز به نفوذ داخلی و تکنیک‌های مهندسی اجتماعی دارند. حمله‌ها ممکن است از نوع داخلی، خارجی و یا ترکیبی از این دو باشند. هدفشان ممکن است شامل دسترسی به مالکیت فکری برای مزیتی استراتژیک یا تکنولوژیک باشد. همچنین ما نمی‌خواهیم بیش از حد به حمله‌کنندگان از نظر سوءاستفاده عملکرد برنامه یا رفتار آن در سیستم‌های حساس، توجه کنیم. بیشتر حمله‌کنندگان به‌دنبال اطلاعات حساسی هستند که به‌صورت مستقیم یا غیرمستقیم از پرداخت اطلاعات و اطلاعات شخصی قابل تعریف، منفعت ببرند. اطلاعات اغلب برای شناسایی سارق، پرداخت‌های تقلبی، یا انواع تخلف می‌تواند مورد استفاده قرار گیرد.		مالکیت فکری یا رمزهای تجاری غیرضروری اما مهم می‌باشند.	گونه باشد یا حتی سطح محرمانه بالاتری داشته باشد) که برای بقا و موفقیت سازمان بسیار اهمیت دارد. برنامه هایی که کاربردهای خیلی حساس (همانند حمل‌ونقل، تجهیزات تولیدی، سیستم‌های کنترلی همانند موشک ردیاب) را کنترل می‌کنند. یا سیستم‌هایی که باعث به خطر افتادن زندگی بشریت می‌شوند.
مراقبت‌های پزشکی	اکثر حمله‌کنندگان به‌دنبال اطلاعات و داده‌های حساسی هستند که می‌توانند به‌طور مسقیم یا غیرمستقیم از آن‌ها سود ببرند تا اطلاعات شخصی و اطلاعات مربوط به پرداخت را به‌دست آورند. اغلب از این داده‌ها به‌منظور شناسایی سارق/سارقین، پرداخت‌های تقلبی یا انواع مختلف عملیات تقلبی استفاده می‌شود. برای بخش مراقبت‌های پزشکی ایالات متحده، بیمه سلامت قابل انتقال، HIPAA، امنیت، قوانین نقص اطلاع‌رسانی و قانون امنیت بیمار: http://www.hhs.gov/ocr/privacy/	تمامی برنامه هایی که به شبکه دسترسی دارند	برنامه‌هایی که در حوزه اطلاعات پزشکی حساسیت متوسط و یا حتی کم دارند همانند (اطلاعات محافظت شده‌ی بهداشت)، اطلاعات شخصی قابل شناسایی یا اطلاعات پرداختی	برنامه‌های مورد استفاده برای کنترل ابزار یا تجهیزات پزشکی یا پرونده‌های پزشکی که ممکن است زندگی انسان‌ها را به خطر بیندازد. پرداخت و سیستم های POS (Point Of Sale) که شامل مقادیر زیادی از تراکنش داده‌ای که بتواند مورد سوءاستفاده و کلاه‌برداری قرار گیرد. این شامل هر رابط مدیرتی برای این برنامه‌ها می‌باشد.
خرده‌فروشی،	بسیاری از مهاجمان در این بخش،	تمامی برنامه هایی	مناسب برای	پرداخت و

غذا، مهمان‌نوازی	تاکتیک‌های فرصت‌طلبانه‌ی smash & grab را به‌کار می‌برند. اگرچه یک سری تهدیدات مشخصی از حملات خاص روی برنامه‌های شناخته شده‌ای که شامل اطلاعات پرداخت، تراکنش‌های مالی انجام شده یا اطلاعات قابل شناسایی شخصی وجود دارد. هر چند نسبت به تهدیدهای ذکر شده در بالا، کمتر مشابه هستند، امکان تهدیدهای به‌مراتب بیشرفته‌تر برای دزدیدن مالیت فکری، دستاورد رقابتی، هوش، یا مزیت به‌دست آمده با سازمان هدف یا مذاکرات همراه تجارت؛ که منجر به حمله به این بخش صنعت شود، وجود دارد.	که به شبکه دسترسی دارند	برنامه‌های تجاری، اطلاعات کاتالوگ، اطلاعات شرکت‌های بزرگ داخلی و برنامه‌های با اطلاعات کاربری محدود (به‌عنوان مثال اطلاعات تماس). برنامه‌هایی با مقدار پرداختی داده یا عملیات تصویه حساب کم و متوسط	سیستم‌های POS (Point Of Sale) که شامل مقادیر زیادی از تراکنش داده‌ای که بتواند مورد سوءاستفاده و کلاه‌برداری قرار گیرد. این شامل هر رابط مدیریتی برای این برنامه می‌باشد. برنامه‌هایی با حجم زیادی از اطلاعات حساس مانند تمام اعداد کارت اعتباری، نام اختصاصی، اعداد امنیتی اجتماعی و غیره

موارد مطالعاتی

مورد مطالعاتی ۱: به‌عنوان یک راهنمای آزمون امنیتی

در یک دانشگاه خصوصی در شهر یوتای ایالات متحده آمریکا، تیم دانشگاهی Red از OWASP ASVS به‌عنوان راهنما برای اجرای آزمون‌های نفوذپذیری برنامه استفاده می‌کنند. از OWASP ASVS در تمام فرآیند آزمون نفوذپذیری، از برنامه‌ریزی اولیه و نشست‌های هدف گرفته، تا راهنمای فعالیت‌های آزمون و روش چارچوب‌بندی یافته‌های نهایی برای ارائه گزارش به مشتری استفاده شده است. همچنین، تیم Red آموزش‌هایی برای تیمی که از ASVS استفاده می‌کنند، سازمان‌دهی می‌کند.

تیم Red، آزمون نفوذپذیری شبکه و برنامه برای دپارتمان را به‌عنوان بخشی از استراتژی امنیت اطلاعات کلی دانشگاه انجام می‌دهد. در حین اولین جلسات برنامه‌ریزی، مشتریان اغلب اجازه آزمون برنامه را به تیم‌های دانشجویی نمی‌دهند. با معرفی ASVS به سهامداران و توضیح این موضوع که فعالیت‌های آزمون توسط این استاندارد هدایت می‌شوند، بسیاری از نگرانی‌ها به سرعت بر طرف می‌شوند. بنابراین ASVS برای کمک به تعیین میزان زمان و تلاش لازم برای یک آزمون استفاده می‌شود. از طریق استفاده از سطوح بررسی از پیش تعریف شده ASVS، تیم Red آزمون مبتنی بر ریسک را توضیح می‌دهد. با این کار به مشتری‌ها، سهامداران و خود تیم کمک می‌کند که جهت بررسی مناسب برای برنامه مورد نظر به توافق برسند.

هنگامی که آزمون شروع شود، تیم Red از ASVS برای سازماندهی فعالیت‌ها و همچنین برای تقسیم حجم کاری استفاده می‌کند. مدیران پروژه‌ی تیم، از طریق دنبال‌کردن این که کدام نیازمندی‌های بررسی شده‌اند و کدام‌ها منتظر آزمون هستند، به‌راحتی می‌توانند روند آزمون را مشاهده کنند. این کار باعث بهبود ارتباط مشتری‌ها شده و به مدیران پروژه قدرت مدیریت بهتر منابع را می‌دهد. به دلیل این که تیم Red عمدتاً از دانشجویان تشکیل شده است، بیشتر اعضای تیم باید به‌طور هم‌زمان به چندین کار مربوط به درس‌هایشان نیز رسیدگی کنند. وظایف خوش تعریفی که براساس نیازمندی‌های بررسی شخصی یا تمامی

دسته‌بندی‌ها می‌باشند، به اعضای تیم کمک می‌کنند که بدانند برای آزمون به چه چیزهایی نیاز دارند و به آن‌ها اجازه می‌دهد که تخمین دقیقی برای فرآیند تکمیل یک آزمون ارائه کنند. همچنین فرآیند گزارش‌دهی از سازماندهی واضح ASVS بهره می‌گیرد، هنگامی‌که اعضای تیم می‌توانند قبل از حرکت به وظیفه بعدی، شرح یک یافته را بنویسند، به‌طور مؤثری حجم زیادی از گزارش را همزمان با آزمون نفوذ انجام می‌دهند.

تیم Red گزارش نهایی را حول ASVS سازماندهی کرده و وضعیت هر نیازمندی را بررسی و گزارش کرده و جزئیات دقیق متناسب را فراهم می‌کند. این گزارش به مشتری‌ها و سهامداران، یک ایده خوب از جایی که برنامه‌شان توسط استاندارد سنجیده شده است را می‌دهد. همچنین این برای پیگیری‌های تعاملی بسیار با ارزش است، به این علت که به آن‌ها اجازه می‌دهد ببینند که امنیت چگونه به مرور زمان بهبود یافته یا از دست رفته است. به‌علاوه، سهامدارانی که مشتاق چگونگی شکل‌گیری برنامه به فرم تک دسته یا چند دسته‌ای هستند، به‌راحتی می‌توانند آن اطلاعات را به‌دلیل گزارش بسیار نزدیک و هم‌تراز ASVS، درک کنند. همچنین، سازماندهی روشن ASVS، آموزش اعضای جدید تیم را آسان‌تر کرده است تا بتوانند به‌راحتی گزارش‌نویسی‌هایی انجام دهندکه با فرمت گزارش‌های قبل مقایسه می‌شود.

نهایتاً آموزش تیم Red پس از اتخاذ ASVS بهبود پیدا کرده است. قبلاً، آموزش‌های هفته‌ای بر یک موضوع منتخب توسط رهبر تیم یا مدیر پروژه متمرکز شده بودند. این‌ها بر اساس اعضای تیم و نیازشان انتخاب شده‌اند. آموزش مبتنی بر این معیار، پتانسیل گسترده کردن مهارت های اعضای تیم را دارد، اما لزوماً به فعالیت‌های هسته تیم Red مرتبط نیست. به‌عبارت دیگر، در آزمون نفوذپذیری به دستاورد مهم‌تری نرسیدند. پس از اتخاذ ASVS، آموزش تیم اکنون بر چگونگی نیازمندی‌های بررسی آزمون فردی متمرکز می‌شود. این موضوع منجر به یک پیشرفت بزرگ در توانمندی‌های قابل اندازه‌گیری اعضای تیم و کیفیت گزارش‌های نهایی شده است.

مورد مطالعاتی۲: به عنوان SDLC امن

یک استارت‌آپ، به‌دنبال حمایت از تحلیل‌های BigData برای مؤسسات مالی است که متوجه شده‌اند برای دستیابی به فرآیند Metadata ی مالی در دست یافته‌هایشان، امنیت باید در اولویت قرار گیرد. در این مثال، استارت‌آپ، استفاده از ASVS را به‌عنوان اساس چرخه زندگی پیشرفت امن متحرک‌شان انتخاب کرده است.

این استارت‌آپ از ASVS برای تولید epic ها و از مواردی برای مسائل امنیتی عملیاتی استفاده می‌کند. به‌عنوان مثال، چطور عملیات ورود به بهترین نحو پیاده‌سازی شود. این استارت‌آپ از ASVS به شیوه متفاوت‌تری استفاده می‌کند، در واقع در ASVS نظاره کرده و نیازمندی‌هایی که برای sprint کنونی مناسب‌تر است را برمی‌دارد. همچنین، در صورتی‌که یک نیازمندی عملیاتی باشد یا اگر به‌عنوان محدودیتی برای استفاده از موارد غیرعملیاتی تلقی شود، آن‌ها را در یک sprint backlog جمع می‌کند. به‌عنوان مثال، انتخاب مجموع TOTP با احراز هویت دو عامله، همراه با سیاست‌های رمز عبور و تنظیم‌کننده سرویس وبی که به‌عنوان تشخیص حمله جستجوی کورکورانه[1] و سازوکار ممنوعیت رخ می‌دهد. در sprint های آینده، نیازمندی‌های اضافه‌ای بر اساس "فقط در لحظه"[2] و "شما به آن نیاز نخواهید داشت" انتخاب خواهند شد.

توسعه‌دهندگان از ASVS به‌عنوان یک چک‌لیست بازبینی استفاده می‌کنند که اطمینان حاصل می‌کند که کد ناامن در آن‌ها بررسی نشده است. همچنین، برنامه‌های گذشته‌نگر برای بروز مشکل برای توسعه‌دهندگانی که یک ویژگی جدید را بررسی

[1] Brute force
[2] Just in time

کرده‌اند، اطمینان حاصل می‌کند که آن‌ها احتمالاً ASVS مورد نیاز را مورد بررسی قرار داده‌اند و اگر هر چیزی را بهبود بخشند، منجر به کاهش sprint در آینده می‌شود.

ارزیابی نرم‌افزار باعث دستیابی به سطح بررسی شده است

OWASP مبتنی بر بررسی‌های ASVS وعلائم اعتماد[1]

OWASP به‌عنوان یک فروشنده خنثی و غیرتجاری هیچ فروشنده، بررسی‌کننده و یا برنامه‌ای را بررسی نمی‌کند. تمامی این گواهی‌های ضمانت، علائم اعتماد یا گواهی‌ها به‌صورت رسمی توسط OWASP ثبت و یا بررسی نمی‌شوند؛ بنابراین یک سازمان مبتنی بر این دیدگاه لازم است نسبت به اعتماد به شخص ثالث یا علامت اعتماد بیان‌گر بررسی ASVS محتاط باشد. این باعث منع سازمان‌ها از پیشنهاد چنین سرویس‌های تضمینی نمی‌شود، از این نظر که آن‌ها ادعای بررسی OWASP رسمی را انجام نمی‌دهند.

راهنمایی برای سازمان‌های صادرکننده گواهی

استاندارد بررسی امنیتی برنامه می‌تواند به‌عنوان یک بررسی متن‌باز برای این برنامه استفاده شود که شامل دسترسی باز و بدون محدودیت به منابع اصلی و کلیدی مانند معماران و توسعه‌دهندگان، مستندات پروژه، کد منبع، دسترسی احراز هویت شده به سیستم‌های آزمایشی (شامل دسترسی به حداقل یک حساب کاربری در هر نقش)، مخصوصاً برای بررسی‌های L2 و L3 می‌باشد.

از لحاظ تاریخی، آزمون نفوذپذیری و مرور کدهای امن شامل مسائل "با استثنا[2]" هستند که تنها مسائلی که با شکست روبه‌رو شده‌اند در گزارش آخر ظاهر می‌شوند. یک سازمان بررسی‌کننده باید شامل هر گزارشی از محدوده بررسی (مخصوصاً اگر یک جزء کلیدی خارج از محدوده باشد مانند احراز هویت SSO)، خلاصه‌ای از یافته‌های بررسی‌ها شامل آزمون‌های قبول‌شده و شکست‌خورده (با نشانه‌هایی واضح برای نشان دادن چگونگی بر طرف کردن آزمون‌های شکست‌خورده) باشد.

نگه‌داری برگه‌های کاری دقیق، تصاویر، فیلم‌ها و اسکریپت[3]هایی که با اطمینان و مکرراً از یک مسأله استفاده می‌کنند، سوابق الکترونیکی نحوه آزمون (به‌عنوان مثال ردیابی پروکسی‌ها) و همچنین یادداشت‌های مرتبط (مثلاً یک لیست پاک کردن) به‌عنوان تمرین صنعت استاندارد در نظر گرفته می‌شوند. این‌که یک ابزار اجرا شود و فقط شکست‌ها را گزارش دهد، کافی نیست؛ این‌ها شواهد کافی مبنی بر این‌که همه‌ی مسائل در یک سطح بررسی‌شده آزمون شده‌اند، فراهم نمی‌کند. در صورت اختلاف، باید شواهد حمایتی کافی وجود داشته باشد که نشان دهد هر الزام بررسی شده‌ای قطعاً آزمون شده است.

نقش ابزارهای آزمون نفوذ خودکار

ابزارهای آزمون نفوذ خودکار به ارائه بیشترین پوشش‌دهی ممکن و عمل‌کردن بیشترین پارامترهای قابل امکان با متنوع‌ترین نوع ورودی مخرب، ترغیب شده‌اند.

کامل کردن بررسی ASVS، تنها با استفاده از آزمون نفوذ خودکار ممکن نیست. در حالی‌که بیشتر نیازمندی‌های L1 می‌توانند با استفاده از آزمون‌های خودکار اعمال شود، مجموع نیازمندی‌های دیگر به آزمون نفوذ خودکار متمایل هستند.

[1] Trust marks
[2] Exception
[3] Script

لطفاً توجه شود که مرزهای بین آزمون‌های دستی و خودکار با توجه به اهمیت صنعت امنیت برنامه از بین رفته است. ابزارهای خودکار اغلب توسط متخصصین به‌صورت دستی تنظیم می‌شوند و آزمون‌کننده‌های دستی معمولاً طیف وسیعی از ابزارهای خودکار را به‌کار می‌برند.

نقش آزمون نفوذ

فراهم کردن یک آزمون نفوذ دستی و بررسی تمامی مسائل L1 بدون نیاز به دسترسی به کد منبع ممکن است، اما این منجر به عمل نمی‌شود. L2 نیازمند دسترسی به توسعه‌دهندگان، مستندات، متن کد و دسترسی احراز هویت شده به سیستم اصلی است. پوشش آزمون نفوذ گسترده در L۳، ازآن‌جا که بیشتر مسائل افزوده درگیر بازبینی تنظیمات سیستم، بازبینی کد مخرب، مدل‌سازی تهدید، و دیگر دست‌یافته‌های تستی غیرقابل نفوذ می‌باشند، امکان‌پذیر نخواهد بود.

به‌عنوان راهنمای معماری امنیتی دقیق

یکی از رایج‌ترین کاربردهای استاندارد بررسی امنیت برنامه‌ها، استفاده به‌عنوان منبعی برای معماری‌های امنیتی است. دو چارچوب بزرگ معماری امنیتی شامل SABSA و TOGAF فاقد اطلاعات کافی مورد نیاز برای تکمیل بررسی‌های معماری امنیتی برنامه‌ها هستند. ASVS می‌تواند برای پر کردن آن شکاف‌ها استفاده شود، به این صورت که به معماران امنیتی اجازه انتخاب کنترل‌های بهتر برای مشکلات رایج مانند الگوهای محافظت از اطلاعات و راهبردهای بررسی ورودی را می‌دهد.

به‌عنوان جایگزینی برای چک‌لیست‌های از رده خارج شده کدگذاری امن

خیلی از سازمان‌ها از اتخاذ ASVS می‌توانند با انتخاب یکی از سه سطح، یا با چند شاخه کردن ASVS و تغییر چیزی که لازمه‌ی هر سطح ریسک برنامه در مسیر دامنه‌ی خاص می‌باشد، منفعت ببرند. ازآنجایی‌که ردیابی نگه داشته می‌شود، از این نوع چند شاخه کردن استقبال می‌شود، بنابراین اگر یک برنامه نیازمندی ۴,۱ را بگذراند، این برای کپی‌های خراب به معنی استاندارد تکامل یافته است.

به‌عنوان راهنمای آزمون‌های یکپارچه‌سازی و واحد خودکار

ASVS طوری طراحی شده است که با وجود تنها استثنای نیازمندی‌های معماری و کد مخرب، کاملاً قابل آزمون باشد. با ساختن واحد و آزمون‌های یکپارچه‌ای که برای فاز مرتبط و خاص و موارد سوءاستفاده آزمون می‌شوند، برنامه تقریباً با هر ساختاری به‌صورت خودکار بررسی می‌شود. به‌عنوان مثال، آزمون‌های اضافه برای مجموعه آزمون برای کنترل‌کننده ورودی، آزمون پارامتر نام کاربری برای نام‌های کاربری رایج، شمارش حساب‌های کاربری، جستجوی کورکورانه، LDAP و تزریق SQL و XSS[1] می‌تواند طراحی شود. به‌طور مشابه، یک آزمون بر روی پارامتر رمز عبور باید شامل رمز عبورهای رایج، طول رمز عبور، تزریق بایت تهی، حذف پارامتر، XSS، شمارش حساب کاربری و غیره شود.

[1] SQL / XSS injection

به‌عنوان آموزش توسعه امن

ASVS همچنین می‌تواند برای تعریف مشخصه‌های نرم‌افزار امن استفاده شود. خیلی از دوره‌های "کدنویسی امن" در واقع دوره‌های هک اخلاقی با نکاتی از کدنویسی هستند، این هیچ کمکی به توسعه‌دهندگان نمی‌کند. دوره‌های توسعه امن می‌توانند از ASVS، با یک تمرکز قوی بر کنترل فعالانه مستند در ASVS، به‌جای این‌که به‌عنوان مثال از ۱۰ عدد از بدترین چیزهایی که نباید انجام شوند، استفاده کنند.

پروژه‌های OWASP با استفاده از ASVS

چارچوب آگاهی امنیتی

http://www.owasp.org/index.php/OWASP_Security_Knowledge_Framework

آموزش توسعه‌دهندگان در نوشتن کد امن – SKF یک برنامه وب مبتنی بر پایتون متن باز است که از استاندارد بررسی امنیت برنامه OWASP برای آموزش شما و تیم‌تان در کدنویسی امن به روش طراحی، استفاده می‌کند.

OWASP Zed Attack Proxy Project

http://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

پروکسی حمله زد OWASP (ZAP)، یک ابزار آزمون نفوذ یکپارچه‌ی ساده است که برای یافتن آسیب‌پذیری‌های برنامه‌های وب استفاده می‌شود و برای استفاده مردم با یک طیف وسیعی از تجربه‌های امنیتی طراحی شده است؛ بنابراین برای توسعه‌دهندگانی که نسبت به آزمون نفوذ، تازه‌کار هستند کاملاً ایده‌آل است. ZAP، پویشگرهای خودکار و یک مجموعه از ابزارهایی که به شما اجازه یافتن آسیب‌پذیری امنیتی دستی می‌دهد را فراهم می‌کند.

OWASP Cornucopia

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

OWASP بهم ریخته، سازوکاری است به شکل کارت‌های بازی که برای کمک به تیم توسعه‌دهنده نرم‌افزار تشکیل شده است و مشخص‌کننده نیازمندی‌های امنیتی در فرآیندهای توسعه رسمی، قراردادی و سریع‌العمل است و همچنین از نظر زبان، پایه و اساس و از نظر تکنولوژی به فرم اگنوستیک و غیر قابل اطمینان است. این‌گونه بهم ریختگی بر اساس ساختار OWASP عملیاتی مبتنی بر کدگذاری امن، راهنمای مرجع سبک، ملاحظات دوچندان بخش‌های استاندارد برنامه بررسی استاندارد OWASP، راهنمای آزمون OWASP و اصول توسعه امن David Rook می‌باشد.

نیازمندی‌های بررسی دقیق

بررسی ۱. معماری، طراحی و مدل‌سازی تهدید

بررسی ۲. احراز هویت

بررسی ۳. مدیریت نشست

بررسی ۴. کنترل دسترسی

بررسی ۵. مدیریت و رسیدگی به ورودی مخرب

بررسی ۷. رمزنگاری در حالت REST

بررسی ۸. مدیریت خطا و ثبت گزارش

بررسی ۹. محافظت از اطلاعات

بررسی ۱۰. ارتباطات

بررسی ۱۱. تنظیمات امنیت HTTP

بررسی ۱۳. کنترل‌های مخرب

بررسی ۱۵. منطق تجارت

بررسی ۱۶. فایل و منابع

بررسی ۱۷. موبایل

بررسی ۱۸. سرویس‌های وب (جدید برای ۳,۰)

بررسی ۱۹. پیکربندی (جدید برای ۳,۰)

بررسی ۲۰. اینترنت اشیاء (IoT)

بررسی ۱: معماری، طراحی و نیازمندی‌های بررسی مدل‌سازی تهدید

هدف کنترل

اطمینان حاصل کنید که برنامه بررسی‌شده نیازمندی‌های با درجه بالای زیر را برآورده می‌کند:

- در سطح ۱، مؤلفه‌های[1] برنامه مشخص شده و برای حضورشان در این برنامه، دلیلی وجود داشته باشد.
- در سطح ۲، معماری تعریف شده است که کد برنامه به معماری تعریف شده پایبند باشد.
- در سطح ۳، معماری و طراحی به‌کار رفته در جای مناسب قرار گرفته باشد و به‌درستی و به‌طور مؤثری استفاده شده باشد.

توجه: این بخش در نسخه ۳,۰ دوباره معرفی شده است ولی کنترل‌های معماری آن با نسخه ۱,۰ از ASVS یکی است.

نیازمندی‌ها

#	توضیحات	۱	۲	۳	از نسخه
۱,۱	بررسی کنید که تمامی مؤلفه‌های برنامه شناسایی شده و مورد نیاز می‌باشند.	✓	✓	✓	۱,۰
۱,۲	بررسی کنید که تمامی مؤلفه‌ها مانند کتابخانه‌ها، ماژول‌ها، سیستم‌های خارجی که جزئی از برنامه نیستند ولی برنامه مبتنی برآن عمل می‌کند، شناسایی می‌شود.		✓	✓	۱,۰
۱,۳	بررسی کنید که معماری سطح بالا[2] برای برنامه تعریف می‌شود.		✓	✓	۱,۰
۱,۴	بررسی کنید که تمامی مؤلفه‌های برنامه براساس توابع تجاری و یا توابع امنیتی که ارائه می‌کنند، تعریف می‌شوند.			✓	۱,۰
۱,۵	بررسی کنید تمامی مؤلفه‌هایی که جزئی از برنامه نیستند ولی برنامه مبتنی برآن عمل می‌کند، براساس توابع و یا توابع امنیتی که ارائه می‌کنند، تعریف می‌شوند.			✓	۱,۰
۱,۶	بررسی کنید که مدل تهدیدی برای برنامه هدف تعریف و تمامی خطرات ناشی از جاسوسی[3]، دستکاری[4]، تخلف و دزدی[5]، افشای اطلاعات، عدم‌پذیرش سرویس[6] و بالابردن سطح دسترسی[7] پوشش داده می‌شود.			✓	۱,۰
۱,۷	بررسی کنید که تمامی کنترل‌های امنیتی (شامل کتابخانه‌هایی که سرویس‌های امنیتی خارجی را فراخوانی می‌کنند) پیاده‌سازی متمرکز دارند.		✓	✓	۳,۰
۱,۸	بررسی کنید که تمامی مؤلفه‌هایی که توسط کنترل امنیتی تعریف شده‌اند، مانند تقسیم‌بندی شبکه[8]، قوانین دیواره آتش و یا فضای ابری مبتنی بر گروه‌های امنیتی، از هم جدا می‌شود.		✓	✓	۳,۰

[1] Components
[2] High-level
[3] Spoofing
[4] Tampering
[5] Repudation
[6] DoS
[7] Elevation of privilege(STRIDE)
[8] Network segmentation

#	توضیحات	۱	۲	۳	از نسخه
۱,۹	بررسی کنید که برنامه لایه‌های اطلاعات، نمایشی و کنترلی را از یکدیگر تفکیک نماید، به‌طوری‌که بتوان تصمیم‌های امنیتی را در سیستم‌های مورد اعتماد اجرا کرد.		✓	✓	۳,۰
۱,۱۰	بررسی کنید که منطق تجاری حساس، کلیدهای مخفی یا اطلاعات اختصاصی در کد سمت مشتری[1]، وجود ندارد.		✓	✓	۳,۰
۱,۱۱	بررسی کنید که تمامی اجزای برنامه ،کتابخانه‌ها، ماژول‌ها، چارچوب‌ها، پلت‌فرم[2] و سیستم‌های عامل، فاقد آسیب‌پذیری می‌باشند.		✓	✓	۳,۰,۱

منابع

برای اطلاعات بیشتر به منابع زیر مراجعه کنید:

- Threat Modeling Cheat Sheet
  https://www.owasp.org/index.php/Application_Security_Architecture_Cheat_Sheet
- Attack Surface Analysis Cheat Sheet
- https://www.owasp.org/index.php/Attack_Surface_Analysis_Cheat_Sheet

[1] Client side
[2] platform

بررسی ۲: نیازمندی‌های بررسی احراز هویت

هدف کنترل

احراز هویت عمل بررسی و تأیید یک فرد یا چیزی است که ادعای هویت صحیح می‌نماید. اطمینان حاصل نمایید که برنامه تأییدشده موارد زیر را رعایت می‌نماید:

- هویت دیجیتالی فرستنده را بررسی کند.
- از این اطمینان حاصل کند که فقط افراد مجاز توانایی احراز هویت داشته باشند و گواهینامه‌ها به‌طور امن منتقل می‌شوند.

نیازمندی‌ها

#	توضیحات	۱	۲	۳	از نسخه
۲,۱	بررسی کنید که تمامی صفحات و منابع به‌صورت پیش‌فرض ملزم به احراز هویت می‌باشند، البته به‌جز آن‌هایی‌که به‌صورت عمومی در نظر گرفته شده‌اند (اصل وساطت کامل[1]).	✓	✓	✓	۱,۰
۲,۲	بررسی کنید که برنامه به‌صورت خودکار گواهینامه‌ها، فیلدهای مخفی[2]، آرگومان‌های URL، درخواست‌های Ajax و لیست‌های ورودی را پر نمی‌کند؛ که این اشاره به متن ساده[3]، کلمه عبور برگشت‌پذیر و قابل رمزگشایی دارد.	✓	✓	✓	۳,۱
۲,۴	بررسی کنید که تمامی کنترل‌های احراز هویت در سمت سرویس‌دهنده[4] انجام می‌شود.	✓	✓	✓	۱,۰
۲,۶	بررسی کنید که تمامی کنترل‌های احراز هویتی به‌صورت امن با شکست مواجه می‌شوند. همچنین از این‌که مهاجم نمی‌تواند ورود پیدا کند، اطمینان حاصل می‌شود.	✓	✓	✓	۱,۰
۲,۷	بررسی کنید که فیلدهای ورودی رمز عبور اجازه استفاده passphrase را داده و یا به آن تشویق می‌کنند. همچنین از مدیران رمز عبور به علت استفاده از عبارت عبور طولانی و یا وارد کردن رمز عبور خیلی پیچیده، جلوگیری نمی‌شود.	✓	✓	✓	۳,۰,۱
۲,۸	بررسی کنید که تمامی توابع احراز هویت حساب کاربری (به‌عنوان مثال، به‌روزرسانی مشخصات، رمز عبور فراموش شده، توکن غیرفعال و یا گم‌شده، میزکار کمکی یا IVR) که ممکن است مجدداً به حساب کاربری دسترسی پیدا کنند، حداقل به‌اندازه‌ی سازوکار احراز هویت اصلی نسبت به حمله‌کنندگان، مقاوم باشند.	✓	✓	✓	۲,۰
۲,۹	بررسی کنید که عملیات تغییر رمز عبور شامل رمز عبور قبلی، رمز عبور جدید و بررسی رمز عبور باشد.	✓	✓	✓	۱,۰

[1] Principal of complete mediation
[2] Hidden fields
[3] Plain text
[4] Server side

#	توضیحات	۱	۲	۳	از نسخه
۲,۱۲	بررسی کنید که تمامی تصمیمات احراز هویت را بدون ذخیره‌سازی شناسه‌های حساس نشست یا رمزهای عبور، بتوان ثبت کرد. این باید شامل درخواست‌هایی با metadata مرتبط که برای رسیدگی‌های امنیتی مورد نیاز است، باشد.		✓	✓	۳,۰,۱
۲,۱۳	بررسی کنید که رمزهای عبور حساب کاربری، به صورت یک‌طرفه هش‌شده و عامل کاری کافی برای مقابله با جستجوی کورکورانه، حمله‌های ریکاوری رمز عبور هش‌شده، وجود دارد.		✓	✓	۳,۰,۱
۲,۱۶	بررسی کنید که گواهینامه‌ها از طریق یک لینک رمزنگاری شده مناسب منتقل می‌شوند و تمامی صفحات و توابعی که لازم است کاربر برایشان گواهینامه‌ها را وارد کند، از لینکی رمزنگاری شده باشد.	✓	✓	✓	۳,۰
۲,۱۷	بررسی کنید که عملیات رمز عبور فراموش شده و همچنین دیگر مسیرهای بازگردانی رمز عبور، رمز عبور فعلی را نمایان نکنند، به‌علاوه، رمز عبور جدید به‌صورت متن رمز شده برای کاربر فرستاده شود.	✓	✓	✓	۲,۰
۲,۱۸	بررسی کنید که جمع‌آوری اطلاعات از طریق ورود، بازنشانی رمز عبور و یا عملیات حساب کاربری فراموش‌شده ممکن نباشد.	✓	✓	✓	۲,۰
۲,۱۹	بررسی کنید که رمزهای عبور پیش‌فرض برای چارچوب برنامه و هیچ‌یک از مؤلفه‌های مورد استفاده توسط برنامه وجود نداشته باشد (مثلاً "admin/password")	✓	✓	✓	۲,۰
۲,۲۰	بررسی کنید که سازوکار ضد اتوماسیون برای جلوگیری از آزمون گواهینامه نقض‌شده[1]، جستجوی کورکورانه و حمله account lockout اعمال شده باشد.	✓	✓	✓	۳,۰,۱
۲,۲۱	بررسی کنید که تمامی گواهینامه[2]های احراز هویت برای دسترسی به سرویس‌های خارج از برنامه، رمزنگاری و در مکانی محافظت‌شده ذخیره می‌شوند.		✓	✓	۲,۰
۲,۲۲	بررسی کنید که رمز عبور فراموش‌شده و مسیرهای بازگردانی دیگر، از یک TOTP و یا soft token دیگر، mobile push یا دیگر سازوکارهای بازگردانی آفلاین استفاده می‌کنند. استفاده از یک مقدار تصادفی در پست الکترونیکی یا پیامک، باید به عنوان آخرین مراجعه در نظر گرفته شود و باید دانست که ضعیف تلقی می‌شود.	✓	✓	✓	۳,۰,۱
۲,۲۳	بررسی کنید که تحریم حساب کاربری به دو وضعیت قفل نرم و قفل سخت تقسیم می‌شود و این دو متقابلاً منحصربه‌فرد نیستند. اگر یک حساب کاربری به‌علت حمله جستجوی فراگیر به صورت موقتی دچار تحریم قفل نرم شود، باعث بازنشانی وضعیت قفل سخت نمی‌شود.		✓	✓	۳,۰
۲,۲۴	بررسی کنید که اگر به سؤالاتی بر پایه دانش مشترک نیاز باشد (همچنان به‌عنوان "سؤال‌های مخفی")، این سؤالات قوانین حریم خصوصی را نقض نمی‌کنند و به‌اندازه کافی برای محافظت از حساب‌های کاربری در برابر بازگردانی مخرب، قوی هستند.	✓	✓	✓	۳,۰,۱
۲,۲۵	بررسی کنید که برنامه‌های با ارزش بالا به‌نحوی قابل تنظیم باشند که بتوان از استفاده از تعداد قابل تنظیمی از رمزهای عبور قبلی، جلوگیری کرد.		✓	✓	۳,۱
۲,۲۶	بررسی کنید که برای تراکنش‌های با ارزش بالا، احراز هویت مجدد مبتنی بر ریسک، امضای دو عامله یا تراکنش محور اعمال شود.		✓	✓	۳,۰,۱

[1] Breached credential testing
[2] credential

#	توضیحات	۱	۲	۳	از نسخه
۲,۲۷	بررسی کنید که اقداماتی برای جلوگیری کاربر از استفاده از رمزهای عبور رایج و عبارات عبور ضعیف، اعمال شود.	✓	✓	✓	۳,۰
۲,۲۸	بررسی کنید که تمامی چالش‌های احراز هویت، خواه موفق، خواه مردود، باید پاسخی در زمان پاسخ‌دهی متوسط مشابه داشته باشند.			✓	۳,۰
۲,۲۹	بررسی کنید که رمزها، کلیدهای API، رمزهای عبور، در کد منبع و یا مخزن[1]های کد منبع آنلاین وجود نداشته باشند.			✓	۳,۰
۲,۳۱	بررسی کنید که کاربران بتوانند در موارد زیر ثبت‌نام و ازآن‌ها استفاده کنند. بررسی TOTP، سازوکار احراز هویت دو عامله، سازوکار احراز هویت بیومتریک (Touch ID یا موارد مشابه)، سازوکار احراز هویت چند عامله مشابه که در برابر افشای گواهینامه تک‌عامله محافظت می‌کند.		✓	✓	۳,۰,۱
۲,۳۲	بررسی کنید که واسطه[2]های مدیریتی، در دسترس اشخاص غیر قابل اطمینان نباشند.	✓	✓	✓	۳,۰
۲,۳۳	بررسی کنید که اگر از از نظر سیاست مبتنی بر خطر ممنوعیتی وجود نداشته باشد، برنامه با مدیران از نظر رمز عبور شخص ثالث و مرورگر، سازگار باشد.	✓	✓	✓	۳,۰,۱

منابع

برای اطلاعات بیشتر به منابع زیر مراجعه کنید:

- OWASP Testing Guide 4.0: Testing for Authentication
  https://www.owasp.org/index.php/Testing_for_authentication
- Password storage cheat sheet
  https://www.owasp.org/index.php/Password_Storage_Cheat_sheet
- Forgot password cheat sheet
  https://www.owasp.org/index.php/Forgot_Password_Cheat_Sheet
- Choosing and Using Security Questions at
  http://www.owasp.org/index.php/Choosing_and_Security_Questions_Cheat_Sheet

[1] Repository
[2] Interface

بررسی ۳: نیازمندی‌های بررسی مدیریت نشست

هدف کنترل

یکی از اساسی‌ترین مؤلفه‌های هر برنامه مبتنی بر وب، سازوکاری است که وضعیت کاربری که با آن تعامل دارد را کنترل و نگه‌داری می‌کند. این مهم، به مدیریت نشست ارجاع می‌دهد و به‌عنوان مجموعه‌ای از کنترل‌هایی که برتعامل تمام وضعیتی بین کاربر و برنامه مبتنی بر وب حکم فرمایی می‌کنند، تعریف می‌شود.

از این‌که برنامه بررسی‌شده، نیازمندی‌های با درجه بالای زیر را برآورده کند اطمینان حاصل شود که:

- نشست‌ها نسبت به هر شخص، منحصربه‌فرد باشند و نتوان آن‌ها را حدس زد و یا به اشتراک گذاشت.
- نشست‌ها هنگامی‌که دیگر به جلسات نیازی نباشد و یا مهلت زمان عدم فعالیت گذشته باشد، نامعتبر شوند.

نیازمندی‌ها

#	توضیحات	۱	۲	۳	از نسخه
۳,۱	بررسی کنید که مدیر نشست سفارشی[1] وجود نداشته باشد و یا این که مدیر نشست سفارشی در برابر حملات مدیریت نشست رایج، آسیب‌پذیر نباشد.	✓	✓	✓	۱,۰
۳,۲	بررسی کنید که نشست هنگامی که کاربر خارج شود، فاقد اعتبار باشد	✓	✓	✓	۱,۰
۳,۳	بررسی کنید که نشست‌ها پس از مدت زمان مشخصی از عدم فعالیت، مهلت‌شان تمام شود.	✓	✓	✓	۱,۰
۳,۴	بررسی کنید که جلسات بعد از حداکثر یک دوره زمانی قابل تنظیم مدیریتی، فارغ از فعالیت یا عدم فعالیت مهلت‌شان تمام می‌شود. (an absolute timeout)		✓	✓	۱,۰
۳,۵	بررسی کنید که تمامی صفحاتی که نیاز به احراز هویت دارند، دارای دسترسی آسان و قابل رؤیت به عملیات خروج باشند.	✓	✓	✓	۱,۰
۳,۶	بررسی کنید که ID جلسات هرگز در URLها، پیغام‌های خطا یا ثبت گزارشات آشکار نمی‌شوند. این شامل بررسی است که برنامه از دوباره‌نویسی کوکی‌های نشست، پشتیبانی نکند.	✓	✓	✓	۱,۰
۳,۷	بررسی کنید که تمامی احراز هویت‌های موفق و درخواست‌های احراز هویت مجدد، منجر به تولید نشست‌های جدید همراه با ID نشست می‌شود.	✓	✓	✓	۱,۰
۳,۱۰	بررسی کنید که ID های نشست‌هایی که توسط چارچوب برنامه تولید می‌شوند، توسط برنامه به‌عنوان فعال شناخته می‌شوند.		✓	✓	۱,۰
۳,۱۱	بررسی کنید که ID های نشست به اندازه کافی طولانی، تصادفی و به‌صورت یکتا مبتنی بر نشست فعال صحیح می باشند	✓	✓	✓	۱,۰

[1] Custom

#	توضیحات	۱	۲	۳	از نسخه
۳,۱۲	بررسی کنید که ID های نشستی که در کوکی‌ها ذخیره می‌شوند، دارای مسیری باشند که مقدار محدودکننده‌ای مناسب برای برنامه داشته باشند و همچنین توکن‌های نشست احراز هویت به مشخصه‌های "HttpOnly" و "secure" تنظیم شده باشند.	✓	✓	✓	۳,۰
۳,۱۶	بررسی کنید که برنامه‌های با ارزش بالا، تعداد جلسات همزمان فعال را محدود می‌کنند.			✓	۳,۰
۳,۱۷	بررسی کنید که لیست نشست فعال، در نمایش حساب کاربری یا مورد مشابهی از هرکاربر، نمایش داده شود. این کاربر باید قادر به خاتمه دادن هر نشست فعالی باشد.		✓	✓	۳,۰
۳,۱۸	بررسی کنید که برای برنامه‌های با ارزش بالا، امکان خاتمه دادن تمام جلسات فعال دیگر، البته بعد از فرآیند تغییر رمز عبور موفق برای کاربر فراهم شود.			✓	۳,۰

منابع

برای اطلاعات بیشتر به منابع زیر مراجعه کنید:

- OWASP Testing Giude 4.0: Session Management Testing
  https://www.owasp.org/index.php/Testing_for_Session_Management
- OWASP Session Management Cheat Sheet
  https://www.owasp.org/index.php/Session_Managment_Cheat_Sheet

بررسی ۴: دسترسی به نیازمندی‌های بررسی کنترل

هدف کنترل

مجوز به مفهوم اجازه دسترسی به منابع، آن هم فقط برای افرادی که مجوز استفاده از این منابع را دارند، می‌باشد. اطمینان حاصل شود که برنامه نیازمندی‌های با درجه بالای زیر را برآورده کند:

- اشخاص دارای دسترسی به منابع، برای انجام هر عملی، دارای گواهینامه‌های معتبر باشند.
- نقش‌ها و دسترسی‌های اختصاص داده شده به کاربران به‌درستی تعریف شده باشند.
- مجوز و نقش Metadata از دستکاری یا بازپخش محافظت شده باشد.

نیازمندی‌ها

#	توضیحات	۱	۲	۳	از نسخه
۴,۱	بررسی کنید که قانون کمترین حق امتیاز وجود داشته باشد – کاربران فقط برای دسترسی به موارد زیر نیاز به مجوز خاص داشته باشند: توابع دسترسی، فایل‌های اطلاعاتی، URL ها، کنترل‌کننده‌ها، سرویس‌ها و دیگر منابع	✓	✓	✓	۱,۰
۴,۴	بررسی کنید که دسترسی به گزارش‌های حساس به‌صورت محافظت شده باشد به طوری که تنها اشیاء یا اطلاعات مجاز، برای کاربر قابل دسترسی باشد (به‌عنوان مثال، محافظت در برابر دستکاری یک پارامتر قابل مشاهده توسط کاربران یا تغییر حساب کاربری یک کاربر دیگر)	✓	✓	✓	۱,۰
۴,۵	بررسی کنید که برنامه نباید اجازه اکتشاف یا افشای metadata ی فایل یا دایرکتوری مانندپوشه‌های Thumbs.db، DS_Store، git. یا svn. را بدهد.	✓	✓	✓	۱,۰
۴,۸	بررسی کنید که کنترل‌های دسترسی به‌صورت امن با شکست مواجه شوند.	✓	✓	✓	۱,۰
۴,۹	بررسی کنید که همان قوانین کنترلی دسترسی که توسط لایه ارائه، نشان داده می‌شوند در سمت سرویس‌دهنده اعمال شوند.	✓	✓	✓	۱,۰
۴,۱۰	بررسی کنید که تمام صفات کاربران و داده‌ها و اطلاعات سیاسی که توسط کنترل‌های دسترسی استفاده می‌شوند، توسط کاربران نهایی دست‌کاری نمی‌شوند، مگر آن‌هایی که به‌صورت خاص اجازه داشته باشند.		✓	✓	۱,۰
۴,۱۱	بررسی کنید که سازوکار متمرکزی (شامل کتابخانه‌هایی که سرویس‌های مجاز خارجی را فراخوانی می‌کنند) برای محافظت از دسترسی به هر نوع منبع محافظت‌شده، وجود داشته باشد.			✓	۱,۰
۴,۱۲	بررسی کنید که تمامی تصمیمات کنترل دسترسی را بتوان ثبت کرد و همچنین تمام تصمیمات مردود نیز ثبت شوند.		✓	✓	۲,۰
۴,۱۳	بررسی کنید که برنامه یا چارچوب، از توکن‌های ضد CSRF تصادفی استفاده کنند و سازوکار محافظت از تراکنش دیگری نیز داشته باشند.	✓	✓	✓	۲,۰

#	توضیحات	۱	۲	۳	از نسخه
۴,۱۴	بررسی کنید که سیستم بتواند از دسترسی کلی و یا پیوسته به اطلاعات، منابع یا توابع امن، محافظت کند. به‌عنوان مثال، استفاده از یک حاکم منابع را در نظر بگیرید که تعداد ویرایش در هر ساعت را محدود می‌کند یا از پاک کردن یک پایگاه‌داده کلی توسط یک کاربر، جلوگیری می‌کند.		✓	✓	۲,۰
۴,۱۵	بررسی کنید که برنامه دارای مجوز افزوده‌ای (به‌عنوان مثال، مجوز سازگار شونده یا افزایشی) برای سیستم‌های با مقدار ضعیف و یا قدرت تفکیک‌پذیری وظایف برای برنامه‌هایی با مقادیر بالا به‌منظور اعمال کنترل‌های ضد تقلب برای هر ریسک برنامه و تقلب گذشته‌اش، باشد.		✓	✓	۳,۰
۴,۱۶	بررسی کنید که برنامه مجوز حساس به متن[1] را به‌درستی اعمال کند، به‌طوری‌که اجازه دست‌کاری غیرمجاز به‌معنی دست‌کاری پارامتر داده نشود.	✓	✓	✓	۳,۰

منابع

برای اطلاعات بیشتر به منابع زیر مراجعه کنید:

- OWASP Testing Guide 4.0: Authorization
  https://www.owasp.org/index.php/Testing_for_Authorization
- OWASP Cheat Sheet: Access Control
  https://www.owasp.org/index.php/Access_Control_Cheat_Sheet

[1] Context sensitive authorisation

بررسی ۵ : بررسی نیازمندی‌های مدیریت ورودی‌های مخرب

هدف کنترل

رایج‌ترین ضعف امنیتی برنامه وب این است که قبل از استفاده ورودی‌های مشتری یا محیط، معتبرسازی نمی‌شود. این ضعف تقریباً منجر به آسیب‌پذیری‌های عمده در برنامه‌های وب می‌شود که می‌توان نمونه‌هایی مانند cross site scripting، تزریق SQL، تزریق مفسر[1]، حملات محلی / Unicode، حملات به File System و سرریز بافر[2] را نام برد.

باید مطمئن شد که یک برنامه بررسی‌شده نیازمندی‌های با درجه بالای زیر را برآورده می‌کند:

- تمام ورودی‌ها درست و متناسب با هدف مورد نظر هستند.
- داده‌های یک موجودیت یا یک مشتری خارجی هرگز نباید مورد اعتماد قرار گیرند و باید بر اساس ضوابط مدیریت شوند.

نیازمندی‌ها

#	توضیحات	۱	۲	۳	از نسخه
۵,۱	بررسی کنید که محیط زمان اجرا مستعد به سرریزی بافر نیست یا این‌که کنترل‌های امنیتی باعث جلوگیری از سرریزی بافر می‌شوند.	✓	✓	✓	۱,۰
۵,۳	بررسی کنید که اشتباهات اعتبارسنجی ورودی‌های سمت سرویس‌دهنده منجر به رد درخواست شده و ثبت می‌گردند.	✓	✓	✓	۱,۰
۵,۵	بررسی کنید که روال‌های معتبرسازی ورودی‌ها در سمت سرویس‌دهنده اعمال می‌شوند.	✓	✓	✓	۱,۰
۵,۶	بررسی کنید که یک کنترل معتبر ورودی در برنامه، برای هر نوع داده پذیرفته شده استفاده می‌شود.			✓	۱,۰
۵,۱۰	بررسی کنید که همه Query های SQL، HQL، OSQL، NOSQL و روش‌های ذخیره شده و فراخوانی فرآیندهای ذخیره شده توسط دستورات آماده شده و یا پارامترسازی Query محافظت می‌شود و بنابراین مستعد به حمله تزریق SQL نیست.	✓	✓	✓	۲,۰
۵,۱۱	بررسی کنید که برنامه حساس به تزریق LDAP نیست، یا این‌که کنترل‌های امنیتی از LDAP Injection جلوگیری می‌کنند.	✓	✓	✓	۲,۰
۵,۱۲	بررسی کنید که برنامه حساس به تزریق دستورات سیستم‌عامل نیست یا این‌که کنترل‌های امنیتی از تزریق دستورات سیستم‌عامل جلوگیری می‌کنند.	✓	✓	✓	۲,۰

[1] Interpreter injection
[2] Buffer overflow

#	توضیحات	۱	۲	۳	از نسخه
۵,۱۳	بررسی کنید برنامه هنگام استفاده از محتویات و دسترسی به مسیرهای فایل‌ها، مستعد به Remote File Inclusion (RFI) و یا Local File Inclusion (LFI) نیست.	✓	✓	✓	۳,۰
۵,۱۴	بررسی کنید که این برنامه مستعد به حملات معمولی XML مانند، دست‌کاری XPath Query، حملات XML External Entity و حملات تزریق XML نباشد.	✓	✓	✓	۲,۰
۵,۱۵	بررسی کنید که تمام متغیرهای رشته‌ای که در HTML یا دیگر کدهای مشتریان وب قرار داده می‌شوند یا به‌صورت دستی رمزگذاری شده‌اند، یا از قالب‌هایی استفاده می‌کنند که به‌طور خودکار رمزگذاری می‌کنند تا اطمینان حاصل کنید که برنامه مستعد به حملات Reflected، Stored و DOM از نوع XSS نباشد.	✓	✓	✓	۳,۰
۵,۱۶	بررسی کنید که چارچوب برنامه امکان اختصاص‌دهی خودکار چند پارامتر (انقیاد خودکار متغیرها) از درخواست ورودی به یک مدل را فراهم می‌نماید، همچنین مطمئن شوید که متغیرهای حساس امنیتی مثل accountBalance، role و password از انقیادهای خودکار مخرب[1] محافظت شده‌اند.		✓	✓	۳,۰
۵,۱۷	بررسی کنید که برنامه دارای دفاع از حملات پارامترهای HTTP است، به‌خصوص اگر چارچوب برنامه هیچ تمایزی در مورد منبع پارامترهای درخواستی ندارد. (Post، Get، کوکی، سرآیند[2]، محیط و غیره)		✓	✓	۲,۰
۵,۱۸	بررسی کنید که اعتبارسنجی سمت مشتری به‌عنوان خط دوم دفاع به همراه اعتبارسنجی سمت سرویس‌دهنده مورد استفاده قرار گیرد.		✓	✓	۳,۰
۵,۱۹	بررسی کنید که همه داده‌های ورودی معتبر هستند، نه تنها متغیرهای فرم HTML، بلکه تمام منابع ورودی مانند فراخوانی‌های REST، Query های پارامترها، سرآیندهای HTTP، کوکی‌ها، فایل‌های Batch، خوراک‌های RSS و غیره. این اعتبارسنجی را می‌توان توسط اعتبارسنجی مثبت (لیست سفید)، و اعتبار سنجی میانی فهرست خاکستری (حذف رشته‌های شناخته شده مخرب)، یا رد ورودی‌های مخرب (لیست سیاه) انجام داد.		✓	✓	۳,۰
۵,۲۱	بررسی کنید که داده‌های بدون ساختار، برای اجرای اقدامات ایمنی عمومی مانند کاراکترهای مجاز، و چیزهایی که به‌طور بالقوه در زمینه داده شده مضر هستند، باید از آن‌ها محافظت شود (مثلاً نام‌های طبیعی با Unicode یا آپوستروف، مثل نام‌های ねこ یا O'Hara).		✓	✓	۳,۰
۵,۲۲	بررسی کنید که فایل‌های HTML غیرمطمئن از ویرایش‌گر WYSIWYG و ویرایش‌گرهای مشابه، به‌صورت درست توسط یک فایل HTML پاک‌سازی کننده، پاکسازی شده‌اند. همچنین این فایل‌ها باید بر اساس اعتبارسنجی ورودی و رمزنگاری به‌صورت درست به‌کار گرفته شوند.	✓	✓	✓	۳,۰
۵,۲۳	بررسی کنید که برنامه در چه جاهایی از auto-escaping استفاده می‌کند و در چه جاهایی auto-escaping غیرفعال است. خروجی باید به‌صورت دستی رمزگذاری یا پاک‌سازی شوند، تا از XSS جلوگیری گردد.		✓	✓	۳,۰

[1] Malicious automatic binding
[2] Header

#	توضیحات	۱	۲	۳	از نسخه
۵,۲۴	بررسی کنید که داده‌ها از یک DOM به یکی دیگر منتقل می‌شود. این تبدیل از متدهای جاوا اسکریپت مانند innerText یا .val استفاده می‌کند.		✓	✓	۳,۰
۵,۲۵	بررسی کنید که هنگام تجزیه JSON در مرورگرها، JSON.parse جهت تجزیه JSON برای مشتری استفاده می‌شود. از ()eval برای تجزیه JSON برای مشتری استفاده نکنید.		✓	✓	۳,۰
۵,۲۶	بررسی کنید که داده های احراز هویت مانند DOM یک مرورگر، بعد از پایان نشست از محل ذخیره‌سازی مشتری حذف شوند.		✓	✓	۳,۰

منابع

برای اطلاعات بیشتر به منابع زیر مراجعه کنید:

- OWASP Testinh Guide 4.0: Input Validation Testing
https://www.owasp.org/index.php/Testing_for_Input_Validation
- OWASP Cheat Sheet: Input Validation
https://www.owasp.org/index.php/Input_Validation_Cheat_Sheet
- OWASP Testing Guide 4.0: Testing for Http Parameter Pullution
https://www.owasp.org/index.php/Testing_for_HTTP_Parameter_pollution_%28OTG-INPVAL-004%29
- OWASP LDAP Injection Cheat Sheet
https://www.owasp.org/index.php/LDAP_Injection_Prevention_Cheat_Sheet
- OWASP Testing Guide 4.0: Client Side Testing
https://www.owasp.org/index.php/Client_Side_Testing
- OWASP Cross Site Scripting Prevention Cheat Sheet
https://www.owasp.org/index.php/XSS_%28Cross_Site_Scripting%29_Prevention_Cheat_Sheet
- OWASP Java Encoding Project
https://www.owasp.org/index.php/OWASP_Java_Encoder_Project

برای اطلاعات بیشتر در مورد auto-escaping مراجعه کنید:

- کاهش XSS توسط Automatic Context-Aware Escaping در Template Systems
http://googleonlinesecurity.blogspot.com/2009/03/reducing-xss-by-way-of- automatic.html
- AngularJS Strict Contextual Escaping https://docs.angularjs.org/api/ng/service/$sce
https://cwe.mitre.org/data/definitions/915.html

بررسی ۶: بررسی نیازمندی‌های کدگذاری و بی‌اثرسازی خروجی

این بخش در بررسی ۵ استاندارد بررسی امنیت اپلیکیشن ۲,۰ آورده شده است. الزامات ASVS ۵,۱۶ به رمزگذاری خروجی متنی به جهت جلوگیری از Cross Site Scripting اشاره دارد.

بررسی ۷: رمزنگاری در بررسی نیازمندی‌های راکد

هدف کنترل

اطمینان حاصل کنید که یک برنامه تأیید شده شرایط زیر را به بهترین نحو برآورده می‌کند:

- تمام ماژول‌های رمزنگاری به نحوی امن از کار می‌افتند و خطاها به‌درستی مدیریت شده‌اند.
- به‌هنگام نیاز به Randomness، یک تولید کننده مناسب عدد تصادفی استفاده می‌شود.
- دسترسی به کلیدها به‌صورت امن مدیریت می‌شود.

نیازمندی‌ها

#	توضیحات	۱	۲	۳	از نسخه
۷,۲	بررسی کنید که تمام ماژول‌های رمزنگاری به‌نحوی ایمن از کار می‌افتند و خطاها به‌گونه‌ای مدیریت می‌شوند که Padding Oracle فعال نمی‌گردد.	✓	✓	✓	۱,۰
۷,۶	بررسی کنید که تمامی اعداد تصادفی، اسامی تصادفی فایل‌ها، GUID های تصادفی و رشته‌های تصادفی توسط تولیدکننده عدد تصادفی مورد تأیید ماژول رمزنگاری شده تولید می‌شوند، به‌گونه‌ای که این مقادیر تصادفی توسط حمله‌کننده قابل حدس زدن نباشد.		✓	✓	۱,۰
۷,۷	بررسی کنید که الگوریتم‌های رمزنگاری استفاده شده علیه FIPS 140-2 یا یک استاندارد معادل اعتبارسنجی شده باشند.	✓	✓	✓	۱,۰
۷,۸	بررسی کنید که ماژول‌های رمزنگاری در حالت تأییدشده خود با توجه به سیاست‌های امنیتی منتشر شده خود عمل می‌کنند.			✓	۱,۰
۷,۹	بررسی کنید که یک روش صریح برای چگونگی مدیریت کلیدهای رمزنگاری (به‌عنوان مثال، تولید، توزیع، لغو و منقضی‌شده) وجود دارد. بررسی کنید که این چرخه حیات کلید به‌درستی اجرا بشود.		✓	✓	۱,۰
۷,۱۱	بررسی کنید که تمام مصرف‌کنندگان خدمات رمزنگاری دسترسی مستقیم به مواد کلید ندارند. فرآیندهای رمزنگاری را ایزوله کنید، از جمله رمزهای اصلی، همچنین استفاده از یک قفل مجازی یا فیزیکی سخت افزاری (HSM) را در نظر بگیرید.			✓	۳,۰,۱
۷,۱۲	بررسی کنید که اطلاعات حساس و شخصی قابل شناسایی در حالت راکد و در حال انتقال رمزنگاری شده باشند.		✓	✓	۳,۰
۷,۱۳	بررسی کنید که کلمه‌های عبور یا مواد کلید واقع در حافظه، به محض عدم نیاز به آن با صفر جایگزین می‌شوند تا از حملات Memory Dumping جلوگیری شود.		✓	✓	۳,۰,۱

#	توضیحات	۱	۲	۳	از نسخه
۷,۱۴	بررسی کنید که تمام کلیدها و کلمه‌های عبور قابل تعویض هستند و در زمان نصب تولید یا جایگزین می‌شوند.		✓	✓	۳,۰
۷,۱۵	بررسی کنید که اعداد تصادفی با بی‌نظمی (آنتروپی) مناسب تولید شوند، حتی زمانی‌که اپلیکیشن تحت بار بالا بوده و یا در شرایطی است که کارکرد آن کاهش پیدا کرده باشد.			✓	۳,۰

منابع

برای اطلاعات بیشتر به منابع زیر مراجعه کنید:

- OWASP Testing Guide 4.0: Testing for weak Cryptography
https://www.owasp.org/index.php/Testing_for_weak_Cryptography
- OWASP Cheat Sheet: Cryptography Storage
https://www.owasp.org/index.php/Cryptographic_Storage_Cheat_Sheet

بررسی ۸: بررسی نیازمندی‌های مدیریت خطا و ثبت گزارش

هدف کنترل

هدف اولیه مدیریت خطا و ثبت گزارش، ارائه یک واکنش مفید توسط کاربر، مدیران و تیم واکنش حادثه است. هدف ایجاد مقادیر زیاد از گزارش نیست، بلکه تولید گزارش‌هایی با کیفیت بالا می‌باشد که بیش از Noise های نامربوط دارای سیگنال باشند.

گزارش‌های با کیفیت بالا اغلب حاوی اطلاعات حساس هستند و باید تحت قوانین یا دستورالعمل‌های حفظ حریم خصوصی محافظت شوند. که باید شامل موارد زیر باشد:

- عدم جمع‌آوری یا ثبت اطلاعات حساس، اگر به‌طور خاص مورد نیاز نباشند.
- اطمینان از مدیریت امن و محافظت‌شده داده‌های ثبت شده بر اساس اهمیت و طبقه‌بندی آن‌ها.
- اطمینان از این‌که که گزارشات برای همیشه نیستند، بلکه یک دوره مشخص دارند که باید تا حد امکان کوتاه باشد.

اگر گزارشات حاوی اطلاعات خصوصی یا حساس باشند که تعریف آن از یک کشور به کشور دیگر متفاوت است، گزارشات خود به حساس‌ترین اطلاعات برنامه تبدیل می‌شوند که برای حمله‌کنندگان درجایگاه خود بسیار جذاب خواهند بود.

نیازمندی‌ها

#	توضیحات	۱	۲	۳	از نسخه
۸,۱	بررسی کنید که برنامه پیام‌های خطا یا stack trace های حاوی اطلاعات حساسی که بتواند به مهاجم کمک کند را نمایش ندهد، از جمله شناسه نشست، نسخه نرم‌افزار یا framework و اطلاعات شخصی.	✓	✓	✓	۱,۰
۸,۲	بررسی کنید که منطق کنترل خطا در کنترل‌های امنیتی به‌طور پیش‌فرض دسترسی را رد می‌کند.		✓	✓	۱,۰
۸,۳	بررسی کنید که کنترل‌های ثبت‌کردن، از قابلیت ثبت موفق برخوردار باشد، به‌خصوص هنگام حوادثی که مرتبط با امنیت هستند.		✓	✓	۱,۰
۸,۴	بررسی کنید که گزارشات هر حادثه شامل اطلاعات ضروری برای بررسی تحقیق زمان رخداد آن باشد.		✓	✓	۱,۰
۸,۵	بررسی کنید که تمام حوادثی که حاوی اطلاعات غیر قابل اعتماد هستند به‌عنوان کد در نرم‌افزار در نظر گرفته شده برای مشاهده گزارش سیستم اجرا نخواهند شد.		✓	✓	۱,۰
۸,۶	بررسی کنید که گزارش‌ها از دسترسی و تغییر غیرمجاز محافظت می‌شوند.		✓	✓	۱,۰
۸,۷	بررسی کنید که برنامه، داده‌های حساس را آن‌گونه که در سیاست‌های قوانین محلی تعریف شده است ثبت نکند، مانند داده‌های حساسی که با ریسک همراه هستند و یا داده‌های حساس احراز هویت که می‌توانند به حمله‌کننده کمک کنند و در شناسه‌های نشست کاربر آمده‌اند، همچنین رمزهای عبور، Hash ها و API Token ها.		✓	✓	۳,۰
۸,۸	بررسی کنید که تمام نشانه‌های غیر قابل چاپ و جدا کننده‌های زمینه به‌درستی در رویداد، به‌منظور جلوگیری از تزریق گزارش، رمزگذاری شده‌اند.			✓	۲,۰
۸,۹	بررسی کنید که گزارش از منابع قابل اعتماد و غیرقابل اعتماد در ورودی قابل تشخیص است.	✓	✓	✓	۲,۰

#	توضیحات	۱	۲	۳	از نسخه
۸,۱۰	بررسی کنید که یک audit log یا موارد مشابه آن، قابلیت جابه‌جایی کلید را فراهم می‌سازند.			✓	۳,۰
۸,۱۱	بررسی کنید که رویدادهای امنیتی دارای بررسی یکپارچگی یا کنترل برای جلوگیری از دست‌کاری‌های غیرمجاز است.			✓	۳,۰
۸,۱۲	بررسی کنید که رویدادها در یک پارتیشن متفاوت از برنامه همراه با چرخش مناسب رویدادها در حال اجرا ذخیره می‌شوند.			✓	۳,۰
۸,۱۳	بررسی کنید که منابع با منطقه زمانی صحیح هماهنگ‌سازی شده‌اند.	✓	✓	✓	۳,۰,۱

منابع

برای اطلاعات بیشتر به منابع زیر مراجعه کنید:

- OWASP Testing Guide 4.0 content: Testing for Error Handling
  https://www.owasp.org/index.php/Testing_for_Error_Handling

بررسی ۹: بررسی نیازمندی‌های محافظت داده

هدف کنترل

سه عنصر کلیدی برای محافظت از داده‌ها وجود دارد: محرمانگی، اصالت و دسترس‌پذیری (CIA). این استاندارد بر این اساس است که حفاظت از داده‌ها بر روی یک سیستم قابل اعتماد که تقویت شده است، مانند یک سرویس‌دهنده اجرا شده و دارای حمایت‌های کافی است.

برنامه‌های کاربردی باید فرض کنند که تمام دستگاه‌های کاربر به نوعی به خطر افتاده‌اند. هنگامی‌که نرم‌افزاری در حال انتقال یا نگه‌داری اطلاعات حساس در دستگاه‌های ناامن مانند رایانه‌ها، تلفن‌ها و تبلت‌های مشترک است، این نرم‌افزار مسئول اطمینان از رمزگذاری داده‌های ذخیره شده در این دستگاه‌ها است و این‌که نمی‌توان به‌راحتی به آن‌ها دسترسی پیدا کرد، یا آن‌ها را تغییر داد.

اطمینان حاصل کنید که یک برنامه تأیید شده نیازمندی‌های سطح بالای زیر را برآورده می‌کند:

محرمانگی: داده‌ها باید از مشاهدات غیرمجاز و یا افشا در هنگام ذخیره‌سازی و انتقال محافظت شوند.

اصالت: داده‌ها باید از این‌که توسط مهاجمین به‌صورت مخرب ایجاد شده، تغییر کرده و یا حذف شوند، محافظت گردند.

دسترس‌پذیری: در صورت نیاز، داده‌ها باید برای کاربران مجاز در دسترس باشند.

نیازمندی‌ها

#	توضیحات	۱	۲	۳	از نسخه
۹,۱	بررسی کنید که همه فرم‌هایی که حاوی اطلاعات حساس هستند، caching را در سمت مشتری غیرفعال کرده‌اند، از جمله قابلیت‌های autocomplete.	✓	✓	✓	۱,۰
۹,۲	بررسی کنید که فهرستی از اطلاعات حساس پردازش شده توسط برنامه مشخص شود و سیاستی تعبیه شود که تعیین نماید که این داده‌ها باید چگونه کنترل شده، رمزگذاری شده و چگونه اصول حفاظت داده روی آن‌ها اعمال شود.			✓	۱,۰
۹,۳	بررسی کنید که همه اطلاعات حساس در قسمت پیام یا متن HTTP به سرویس‌دهنده ارسال می‌شوند. (به‌عنوان مثال، پارامترهای URL هرگز برای ارسال اطلاعات حساس استفاده نمی‌شوند)	✓	✓	✓	۱,۰
۹,۴	بررسی کنید که برنامه سرآیندهای کافی ضد cache را تعبیه نماید، به‌طوری‌که هر گونه اطلاعات حساس و شخصی نمایش داده شده توسط برنامه یا وارد شده توسط کاربر، نباید توسط مرورگرهای امروزی روی دیسک سخت cache شود.	✓	✓	✓	۳,۰,۱
۹,۵	بررسی کنید که در سرویس‌دهنده، تمام نسخه‌های ذخیره شده یا موقت داده‌های حساس از دسترسی غیرمجاز محافظت شده باشد، یا پس از این‌که کاربر مجاز به آن‌ها دسترسی پیدا کرد، این داده‌ها پاک شوند.		✓	✓	۱,۰

#	توضیحات	۱	۲	۳	از نسخه
۹,۶	بررسی کنید که در پایان سیاست حفظ و نگه‌داری مورد نیاز، یک روش برای حذف هر نوع اطلاعات حساس از برنامه وجود دارد.			✓	۱,۰
۹,۷	بررسی کنید که برنامه در یک درخواست تعداد پارامترهایی همچون فیلدهای پنهان، متغیرهای Ajax، کوکی‌ها و مقادیر سرآیندها را به حداقل می‌رساند.		✓	✓	۲,۰
۹,۸	بررسی کنید که برنامه توانایی تشخیص و هشدار به تعداد غیرعادی درخواست برای جمع‌آوری داده‌ها مانند screen scraping را دارد.			✓	۲,۰
۹,۹	بررسی کنید که داده ذخیره شده در ذخیره‌سازی سمت مشتری (مانند ذخیره‌سازی محلی HTML5، ذخیره‌سازی نشست، IndexedDB، کوکی‌های معمولی یا کوکی‌های فلش) شامل اطلاعات حساس یا PII نباشند.	✓	✓	✓	۳,۰,۱
۹,۱۰	بررسی کنید که دسترسی به داده‌ها، اگر که داده مربوطه تحت اصول حفاظت داده جمع‌آوری شده باشد و یا در ثبت گزارش دسترسی لازم را پیدا کرده باشد، حتماً ثبت شود.			✓	۳,۰
۹,۱۱	بررسی کنید که اطلاعات حساس واقع در حافظه، به محض عدم نیاز با صفر جایگزین شوند تا از حملات Memory Dumping جلوگیری گردد.			✓	۳,۰,۱

منابع

برای اطلاعات بیشتر به منابع زیر مراجعه کنید:

- امکان استفاده از https://securityheaders.io به جهت چک کردن امنیت و سرآیندهای ضد cache را بررسی کنید.

- OWASP Secure Headers Project:
https://www.owasp.org/index.php/OWASP_Secure_Headers_Project

- User Privacy Protection Cheat Sheet
https://www.owasp.org/index.php/User_Privacy_Protection_Cheat_Sheet

بررسی ۱۰: بررسی نیازمندی‌های امنیت ارتباطات

هدف کنترل

اطمینان حاصل کنید که یک برنامه بررسی شده نیازمندی‌های سطح بالای زیر را برآورده می‌کند:

- این‌که TLS هنگام انتقال داده‌های حساس مورد استفاده قرار گیرد.
- این‌که الگوریتم‌ها و رمزگذاری‌های قوی همواره مورد استفاده قرار گیرند.

نیازمندی‌ها

#	توضیحات	۱	۲	۳	از نسخه
۱۰,۱	بررسی کنید که یک مسیر همواره می‌تواند از یک CA مورد اعتماد به هر یک از سرویس‌دهنده‌های گواهی‌دهنده TLS طراحی گردد، به طوری‌که گواهی هر سرویس‌دهنده معتبر باشد.	✓	✓	✓	۱,۰
۱۰,۳	بررسی کنید که TLS برای تمامی اتصالات (شامل اتصالات خارجی و backend) که اعتبارسنجی شده یا شامل داده‌های حساس یا توابع هستند، استفاده می‌شود و دارای پروتکل‌های ناامن و غیر رمزشده نیست. اطمینان حاصل کنید که محتمل‌ترین حالت الگوریتم استفاده گردد.	✓	✓	✓	۳,۰
۱۰,۴	بررسی کنید که خطاهای اتصال TLS backend ثبت شده‌اند.			✓	۱,۰
۱۰,۵	بررسی کنید که مسیرهای گواهی ساخته شده، و برای گواهینامه‌های همه مشتری‌ها با استفاده از سازوکار لنگرهای اعتماد و برگردانی اطلاعات تأیید شده‌اند.			✓	۱,۰
۱۰,۶	بررسی کنید که تمامی اتصالات به سیستم‌های خارجی که شامل اطلاعات حساس می‌باشد، احراز هویت می‌شوند.		✓	✓	۱,۰
۱۰,۸	بررسی کنید که یک پیاده‌سازی از استاندارد TLS وجود داشته و توسط برنامه، در حالت عملکردی بررسی شده به‌کار گرفته می‌شود.			✓	۱,۰
۱۰,۱۰	بررسی کنید که برچسب‌زنی عمومی کلید گواهی TLS (HPKP) برای تولید و پشتیبانی از کلیدهای عمومی پیاده‌سازی شده است. به جهت کسب اطلاعات بیشتر به منابع زیر مراجعه کنید.		✓	✓	۳,۰۱
۱۰,۱۱	بررسی کنید که سرآیندهای HTTP Strict Transport Security، برای همه زیردامنه‌ها و برای همه درخواست‌ها مانند Strict-Transport-Security:max-age=15724800;includeSubdomains شامل می‌شود.	✓	✓	✓	۳,۰

#	توضیحات	۱	۲	۳	از نسخه
۱۰,۱۲	بررسی کنید که URL وب‌سایت تولیدکننده که نسبت به آن ثبت‌نام صورت گرفته است در لیست دامنه‌های Strict Transport Security که توسط فروشندگان مرورگر پشتیبانی می‌شود، از پیش بارگذاری شده باشد. لطفاً منابع زیر را مشاهده نمایید.			✓	۳,۰
۱۰,۱۳	بررسی کنید که پنهان‌سازی کامل ارسالات جهت مقابله با حمله‌کنندگانی که ترافیک را ضبط می‌کنند، تنظیم شده باشد.	✓	✓	✓	۳,۰
۱۰,۱۴	بررسی کنید که بازگردانی مناسب گواهی‌ها مانند Online Certificate Status Protocol (OSCP) فعال شده و تنظیم شده باشند.	✓	✓	✓	۳,۰
۱۰,۱۵	بررسی کنید که فقط الگوریتم‌ها و رمزگذاری‌های قوی و پروتکل‌های تأیید شده در ساختار خوشه‌ای گواهی‌ها مورد استفاده قرار گرفته باشند، شامل ریشه و واسط گواهی‌دهنده شما.	✓	✓	✓	۳,۰
۱۰,۱۶	بررسی کنید که تنظیمات TLS با عملکرد کنونی هم‌جهت است، به‌خصوص که در یک تنظیمات عمومی، رمزگذاری‌های عمومی و الگوریتم‌ها نیز ناامن‌تر می‌شوند.	✓	✓	✓	۳,۰

منابع

برای اطلاعات بیشتر به منابع زیر مراجعه کنید:

- **OWASP – برگه تقلب TLS**:

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

- **یادداشتی بر "حالات مورد تأیید TLS"**. در گذشته، ASVS به استاندارد FIPS 140-2 آمریکا گفته می‌شد، اما به‌عنوان یک استاندارد جهانی، با استفاده از استانداردهای آمریکا، این امر می‌تواند دشوار، متناقض یا گیج‌کننده باشد. یک روش بهتر برای رسیدن به انطباق با ۱۰،۸، بررسی راهنمایی‌هایی مانند (https://wiki.mozilla.org/Security/Server_Side_TLS) است. همچنین تولید تنظیمات شناخته شده https://mozilla.github.io/server-side-tls/ssl-config-generator/ و نیز با استفاده از ابزارهای سنجش TLS شناخته شده مانند sslyze، اسکنرهای مختلف آسیب‌پذیری یا خدمات ارزیابی آنلاین TLS معتبر برای به‌دست آوردن سطح مطلوب امنیت است. به‌طور کلی، عدم انطباق برای این بخش مشاهده می‌گردد، استفاده از رمزها، الگوریتم‌ها، پروتکل‌های SSL قدیمی یا ناامن و رمزگذاری‌های قدیمی ضعیف می‌باشد که منجر به نقص در محرمانگی می‌گردد.
- **پیوند گواهی:** برای اطلاعات بیشتر لطفا به لینک زیر مراجعه کنید:

https://tools.ietf.org/html/rfc7469.

- Production and backup keys is business continuity - see https://noncombatant.org/2015/05/01/about-http-public-key-pinning/
- OWASP Certificate Pinnig Cheat Sheet https://www.owasp.org/index.php/Pinning_Cheat_Sheet
- OWASP Certificate and Public Key Pinning https://www.owasp.org/index.php/Certificate_and_Public_Key_Pinning
- Timr of first use (TOFU) Pinnig https://developer.mozilla.org/en/docs/Web/Security/Public_Key_Pinning
- HTTP Strict Transport Security https://www.chromium.org/hsts

بررسی ۱۱: بررسی نیازمندی‌های تنظیمات امنیتی HTTP

هدف کنترل

اطمینان حاصل کنید که یک برنامه بررسی شده نیازمندی‌های سطح بالای زیر را برآورده می‌کند:

- سرویس‌دهنده برنامه به طور مناسب از تنظیمات پیش‌فرض ایجاد شده است.
- پاسخ‌های HTTP حاوی یک کاراکتر ایمن در محتوای سرآیند است.

نیازمندی‌ها

#	توضیحات	۱	۲	۳	از نسخه
۱۱,۱	بررسی کنید که برنامه فقط یک مجموعه متشکل از روش‌های درخواست HTTP مورد نیاز مانند GET و POST را قبول می‌کند و روش‌های استفاده نشده (مانند PUT،TRACE و DELETE) به‌صورت صریح مسدود شده باشند.	✓	✓	✓	۱,۰
۱۱,۲	بررسی کنید که هر پاسخ HTTP دارای یک نوع محتوا در سرآیند خود باشد که مشخص کند مجموعه کاراکتری این عبارت از چه جنسی است. (به‌عنوان مثال، UTF-8، ISO 8859-1)	✓	✓	✓	۱,۰
۱۱,۳	بررسی کنید که سرآیندهای HTTP توسط یک پروکسی مطمئن یا دستگاه‌های SSO، مانند یک نشان‌گر حامل، توسط برنامه احراز هویت شده باشد.		✓	✓	۲,۰
۱۱,۴	بررسی کنید که یک سرآیند مناسب X-FRAME-OPTIONS برای سایت‌هایی که محتوای آن نباید در یک X-Frame شخص ثالث مشاهده شود، استفاده گردد.		✓	✓	۳,۰۱
۱۱,۵	بررسی کنید که سرآیند HTTP یا هر بخشی از پاسخ HTTP اطلاعات جزئی مربوط به نسخه اجزای سیستم را نشان نمی‌دهند.	✓	✓	✓	۲,۰
۱۱,۶	بررسی کنید که تمام پاسخ‌های API شامل X-Content- Type-Options: nosniff و "Content-Disposition: attachment; filename="api.json باشند. (یا سایر اسم فایل‌ها و نوع محتواهای مناسب).	✓	✓	✓	۳,۰
۱۱,۷	بررسی کنید که یک سیاست امنیت محتوا (CSPv2) وجود دارد که آسیب‌پذیری‌های معمول DOM، XSS، JSON و جاوا اسکریپت را کاهش می‌دهد.	✓	✓	✓	۳,۰۱
۱۱,۸	بررسی کنید که تنظیمات X-XSS-Protection: 1؛ mode = block header برای فعال کردن فیلترهای XSS انعکاسی وجود دارد.	✓	✓	✓	۳,۰

منابع

برای اطلاعات بیشتر به منابع زیر مراجعه کنید:

- آموزش آزمون OWASP ۴,۰: تستی برای HTTP Verb Tampering

https://www.owasp.org/index.php/Testing_for_HTTP_Verb_Tampering_%28OTG-INPVAL-003%29

- اضافه کردن Content-Disposition به پاسخ های API کمک می کند تا از حملات بسیاری بر اساس سوءتفاهم در نوع MIME بین کاربر و سرویس‌دهنده جلوگیری کند. همچنین گزینه "نام فایل" به‌طور خاص کمک می‌کند تا از حملات Reflected File Download جلوگیری شود.

https://www.blackhat.com/docs/eu-14/materials/eu-14-Hafif-Reflected-File-Download-A-New-Web-Attack-Vector.pdf

https://www.owasp.org/index.php?title=Content_Security_Policy_Cheat_Sheet&setl ang=en

- OWASP User Privacy Protection Cheat Sheet

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

- امکان استفاده از https://securityheaders.io جهت چک کردن امنیت و سرآیندهای ضد caching را بررسی کنید.
- OWASP Secure Headers Project

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

بررسی ۱۲: بررسی نیازمندی‌های پیکربندی امنیتی

این بخش به بررسی ۱۱ در استاندارد بررسی امنیت برنامه ۲,۰ منتقل شد.

بررسی ۱۳: بررسی نیازمندی‌های کنترل‌های مخرب

هدف کنترل

اطمینان حاصل کنید که یک برنامه بررسی شده نیازمندی‌های سطح بالای زیر را برآورده می‌کند:

- فعالیت مخرب به‌صورت ایمن و درست مدیریت شود تا بقیه برنامه را تحت‌تأثیر قرار ندهد.
- بمب‌های زمان‌دار یا سایر حمله‌های زمان‌دار داخل آن‌ها وجود نداشته باشد.
- به مقاصد مخرب و یا مجوز داده نشده "phone home" نکنید.
- برنامه‌های کاربردی back door، easter eggs، حمله‌های salami و یا خطاهای منطقی که توسط حمله‌کننده قابل کنترل است، نداشته باشد.

کد مخرب بسیار نادر است و شناسایی آن نیز مشکل است. بررسی خط به خط کد می‌تواند به جستجوی بمب‌های منطقی کمک کند، اما حتی خبره‌ترین اشخاص که کدها را بررسی می‌کنند، نیز حتی اگر بدانند که چنین کدهایی وجود دارد، به سختی می‌توانند کدهای مخرب را پیدا نمایند. این بخش بدون دسترسی به کد منبع امکان‌پذیر نیست، ازجمله همه کتابخانه‌های شخص ثالث.

نیازمندی‌ها

#	توضیحات	۱	۲	۳	از نسخه
۱۳,۱	بررسی کنید که تمام فعالیت‌های مخرب به‌طور مناسب جعبه شنی[1] شده، محدود و یا جدا شده تا از حملات به سایر برنامه ها جلوگیری شود.			✓	۲,۰
۱۳,۲	بررسی کنید که کد منبع و تا جای ممکن همه کتابخانه‌های شخص ثالث، دارای back door، easter egg و خطاهای منطقی در احراز هویت، کنترل دسترسی، تأیید ورودی و منطق تجاری انتقال‌های مالی با مبالغ بالا، نباشند.			✓	۳,۰۱

منابع

برای اطلاعات بیشتر به منابع زیر مراجعه کنید:

- http://www.dwheeler.com/essays/apple-goto-fail.html

[1] Sandbox

بررسی ۱۴: بررسی نیازمندی‌های امنیت داخلی

این بخش به بررسی ۱۳ در استاندارد بررسی امنیت برنامه ۲,۰ منتقل شد.

بررسی ۱۵: بررسی نیازمندی‌های منطق کسب و کار

هدف کنترل

اطمینان حاصل کنید که یک برنامه بررسی شده نیازمندی‌های سطح بالای زیر را برآورده می‌کند:

- جریان منطق کسب‌وکار، پیوسته و به‌ترتیب است.
- منطق کسب‌وکار شامل محدودیت‌هایی برای شناسایی و جلوگیری از حملات خودکار مانند انتقال مداوم و اندک پول نقد و یا اضافه کردن یک میلیون نفر از دوستان در یک زمان و غیره است.
- جریان‌های منطقی کسب‌وکار با ارزش، موارد سوءاستفاده و عناصر فعال مخرب را مورد توجه قرار داده و نیز در برابر spoofing، tampering، repudiation، information disclosure و حملات privilege مقاوم باشد.

نیازمندی‌ها

#	توضیحات	۱	۲	۳	از نسخه
۱۵,۱	بررسی کنید که برنامه تنها پردازش منطق کسب و کار را به‌صورت مراحل و گام‌های متوالی پردازش می‌کند، و تمامی گام‌ها در زمان واقعی انسانی پردازش شوند نه خارج از نوبت و با پرش از روی گام‌ها و پردازش گام‌های یک کاربر دیگر یا تراکنش‌هایی که خیلی سریع بررسی شده باشند.		✓	✓	۲,۰
۱۵,۲	بررسی کنید که برنامه دارای محدودیت‌های تجاری است و به‌درستی بر اساس هر کاربر اجرا می‌شود، همچنین دارای سیستم هشداردهی قابل تنظیم و خودکار باشد که واکنش‌های خودکار در برابر حملات غیرمعمول و خودکار از خود نشان دهد.		✓	✓	۲,۰

منابع

برای اطلاعات بیشتر به منابع زیر مراجعه کنید:

- OWASP Testing Guide 4.0: Business Logic Testing
https://www.owasp.org/index.php/Testing_for_business_logic

- OWASP Cheat Sheet
https://www.owasp.org/index.php/Business_Logic_Security_Cheat_Sheet

بررسی ۱۶: بررسی نیازمندی‌های فایل‌ها و منابع

هدف کنترل

اطمینان حاصل کنید که یک برنامه بررسی شده نیازمندی‌های سطح بالای زیر را برآورده می‌کند:

- داده های فایل غیر قابل اطمینان باید بر اساس نوع به طریق امن مدیریت شوند.
- دریافتی‌ها از منابع غیر مطمئن در بیرون از شاخه اصلی برنامه و با دسترسی‌های محدود ذخیره شود.

نیازمندی‌ها

#	توضیحات	۱	۲	۳	از نسخه
۱۶,۱	بررسی کنید که URL های Redirects و Forwards فقط اجازه انتقال به مقصد های موجود در لیست سفید را داشته باشد و یا هنگام redirect به مقصدهای نامطمئن هشداری را نشان دهد.	✓	✓	✓	۲,۰
۱۶,۲	بررسی کنید که داده های فایل غیر قابل اطمینان داده شده به برنامه مستقیما با دستورات I/O فایل مورد استفاده قرار نگیرند، به خصوص برای محافظت در برابر آسیب‌پذیری‌های path traversal، local file include، file MIME type، reflective file download و تزریق دستورات سیستم‌عامل.	✓	✓	✓	۳,۰,۱
۱۶,۳	بررسی کنید که فایل‌های به دست آمده از منابع نامعتبر به صورت درحال انتظار برای تایید باشند. و توسط اسکنرهای آنتی ویروس اسکن شده‌اند تا از آپلود محتوای مخرب شناخته شده جلوگیری شود.	✓	✓	✓	۲,۰
۱۶,۴	بررسی کنید که داده های غیرمعتمد در داخل inclusion، class loader و یا قابلیت‌های انعکاسی استفاده نشوند تا از آسیب پذیری های اجرای کد به صورت محلی و از راه دور جلوگیری شود.	✓	✓	✓	۳,۱
۱۶,۵	بررسی کنید که داده‌های نامطمئن در داخل به اشتراک‌گذاری منابع cross-domain (CORS) استفاده نشوند تا از حملات arbitrary remote content جلوگیری شود.	✓	✓	✓	۲,۰
۱۶,۶	بررسی کنید که فایل‌های دریافت شده از منابع نامطمئن خارج از webroot ذخیره شده و دسترسی‌های محدودی داشته باشند و ترجیحا با بررسی‌های دقیق.		✓	✓	۳,۰
۱۶,۷	بررسی کنید که سرور وب یا برنامه به صورت پیش فرض پیکربندی شده است تا دسترسی به منابع از راه دور یا سیستم های خارج از وب یا سرور برنامه را رد کند.		✓	✓	۲,۰
۱۶,۸	بررسی کنید که برنامه و کد آن داده‌های بارگذاری شده از منابع نامطمئن را اجرا نکند.	✓	✓	✓	۳,۰
۱۶,۹	بررسی کنید که تکنولوژی های جانبی پشتیبانی نشده، ناامن یا مفقود شده از قبیل پلاگین های NSAPI، فلش، Shockwave، Active-X، Silverlight، NACL یا Java Appletهای سمت مشتری استفاده نشوند.	✓	✓	✓	۳,۰,۱

منابع

برای اطلاعات بیشتر به منابع زیر مراجعه کنید:

- مدیریت پسوندهای فایل‌ها برای اطلاعات حساس

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

- Reflective File Download (Oren Hatif)

https://www.trustwave.com/Resources/SpiderLabs-Blog/Reflected-File-Download--- A-New-Web-Attack-Vector/

بررسی ۱۷: نیازمندی‌های بررسی موبایل

هدف کنترل

این قسمت شامل کنترل های مخصوص نرم افزار های موبایل می باشد . این کنترل ها از نسخه ۲,۰ غیر تکراری شده اند ، بنابراین باید به طور پیوسته با سایر قسمت های مرتبط سطوح بررسی ASVS درنظر گرفته شود .

برنامه‌های موبایل باید:

- با اجرا کردن کنترل های امنیتی در محیط های مطمئن، کنترل های امنیتی هم سطح با موبایل های مشتری که در سرویس دهنده موجود هستند داشته باشند .
- ذخیره داده های حساس بر روی دستگاه باید به روشی امن انجام شود.
- انتقال تمام داده های حساس برنامه باید با درنظر گرفتن لایه انتقال امن، انجام شود.

نیازمندی‌ها

#	توضیحات	۱	۲	۳	از نسخه
۱۷,۱	بررسی کنید که مقدار های شناسه ای که در سیستم ذخیره می شوند یا توسط سایر برنامه ها قابل استفاده می باشند، مانند شماره های UDID یا IMEI به عنوان توکن های احراز هویت مورد استفاده قرار نمی گیرند .	✓	✓	✓	۲,۰
۱۷,۲	بررسی کنید که برنامه موبایل اطلاعات حساس و مهم را روی منابع مشترک رمزگذاری نشده دستگاه ذخیره نمی کند. (مانند کارت حافظه (SD Card) یا پوشه‌های به اشتراک گذاشته شده)	✓	✓	✓	۲,۰
۱۷,۳	بررسی کنید که اطلاعات حساس حتی در محیط های محافظت شده سیستم از جمله key chains به صورت غیر محافظت شده روی دستگاه ذخیره نمی شوند.	✓	✓	✓	۲,۰
۱۷,۴	بررسی کنید که کلید های مخفی[1]، توکن‌های API یا کلمه های عبور به صورت پویا در برنامه های موبایل ایجاد می شوند.	✓	✓	✓	۲,۰
۱۷,۵	بررسی کنید که برنامه موبایل از نشر اطلاعات حساس جلوگیری می کند مانند عکس های ذخیره شده از صفحه نمایش برنامه در حال اجرا که ممکن است از طریق برنامه هایی که در پشت صحنه در حال اجرا هستند استفاده شود یا اطلاعات حساسی را در کنسول[2] بنویسد.		✓	✓	۲,۰

[1] Secret keys
[2] Console

- OWASP Mobile Security Project:
https://www.owasp.org/index.php/OWASP_Mobile_Security_Project
- iOS Developer Cheat Sheet:

#	توضیحات	۱	۲	۳	از نسخه
۱۷,۶	وارسی کنید که برنامه کمترین مجوز ها را برای کارایی و دسترسی به منابع درخواست می کند.		✓	✓	۲,۰
۱۷,۷	وارسی کنید که کد حساس برنامه در حافظه به صورت غیر قابل پیش‌بینی چیده می‌شود(مانند ASLR)	✓	✓	✓	۲,۰
۱۷,۸	وارسی کنید که تکنیک های ممانعت از دیباگ کردن وجود داشته باشد به گونه‌ای که برای بازداشتن یا به تأخیر انداختن حمله کنندگان احتمالی از تزریق دیباگرها به برنامه موبایل مفید باشند.(مانند GDB)			✓	۲,۰
۱۷,۹	وارسی کنید که برنامه فعالیت‌های حساس، هدف‌های برنامه یا content providers را در اختیار سایر برنامه های موبایل در همان دستگاه برای اکسپلویت نمی گذارد.	✓	✓	✓	۲,۰
۱۷,۱۰	وارسی کنید که اطلاعات حساس نگهداری شده در حافظه هنگامی که دیگر مورد نیاز نیستند با صفر جایگزین می‌شوند تا حمله های memory dumping کاهش پیدا کند.		✓	✓	۳,۰,۱
۱۷,۱۱	وارسی کنید که برنامه برای فعالیت ها، هدف‌های برنامه یا content providers صادر شده، ورودی ها را اعتبار سنجی می کند.	✓	✓	✓	۳,۰,۱

منابع

برای اطلاعات بیشتر به منابع زیر مراجعه کنید:

- OWASP Mobile Security Project:
https://www.owasp.org/index.php/OWASP_Mobile_Security_Project
- iOS Developer Cheat Sheet:
https://www.owasp.org/index.php/IOS_Developer_Cheat_Sheet

بررسی ۱۸ : نیازمندی‌های بررسی سرویس های وب

هدف کنترل

مطمئن شوید که برنامه‌های بررسی شده‌ای که از سرویس های مبتنی بر وب مانند SOAP یا RESTful استفاده می‌کنند دارای :

- احراز هویت مناسب ، مدیریت نشست و مجوز از همه سرویس های وب باشد.
- اعتبار سنجی ورودی ها از همه مؤلفه هایی که از سطح اعتماد پایینتر به سطح اعتماد بالاتر منتقل می‌شوند، باشد.
- قابلیت همکاری اولیه با لایه سرویس وب SOAP برای بالا بردن استفاده از API، باشد.

نیازمندی‌ها

#	توضیحات	۱	۲	۳	از نسخه
۱۸,۱	بررسی کنید که روش کدگذاری یکسان در سمت سرویس دهنده و سرویس گیرنده استفاه شده باشد.	✓	✓	✓	۳,۰
۱۸,۲	بررسی کنید که دسترسی به مدیریت و راهبری توابع در برنامه سرویس وب به مدیر های سرویس وب محدود می‌شود .	✓	✓	✓	۳,۰
۱۸,۳	بررسی کنید که شمای XML و JSON وجود دارد و پیش از پذیرش ورودی بررسی می شود.	✓	✓	✓	۳,۰
۱۸,۴	بررسی کنید که همه ورودی ها به حجمی مناسب محدود می شوند.	✓	✓	✓	۳,۰
۱۸,۵	بررسی کنیدکه سرویس های وب مبتنی بر SOAP با حداقل مشخصات سازمان Interoperability Web Services سازگار است. این اساساً رمزنگاری TLS معنی می دهد.	✓	✓	✓	۳,۰,۱
۱۸,۶	استفاده از احراز هویت و مجوز[1] مبنی بر نشست را بررسی کنید. لطفاً برای راهنمایی بیشتر به بخش‌های ۳،۲ و ۴ مراجعه کنید. هم چنین از استفاده از کلید های API ایستا و مشابه خودداری کنید.	✓	✓	✓	۳,۰
۱۸,۷	بررسی کنید که سرویس REST با استفاده ازحداقل یکی از موارد زیر در برابر Cross Site Request Forgery محافظت می شود: چک کردن مبدأ ها ، دوبار ارسال کردن الگوی کوکی ، CSRF nonces و چک کردن ارجاع دهنده ها	✓	✓	✓	۳,۰,۱

[1] Authorization

#	توضیحات	۱	۲	۳	از نسخه
۱۸,۸	بررسی کنید که سرویس REST برای اینکه همان نوع محتوا ورودی قابل انتظار باشد به صراحت آنرا بررسی می کند، از جمله application/xml یا application/json		✓	✓	۳,۰
۱۸,۹	بررسی کنید که جهت انتقال امن بین سرویس دهنده و سرویس گیرنده، پیام از سازوکار های امضای JSON و امنیت WS برای درخواست های SOAP استفاده می کند.		✓	✓	۳,۰,۱
۱۸,۱۰	بررسی کنید که موارد دیگر و راه‌های دسترسی با امنیت کمتر وجود نداشته باشد .		✓	✓	۳,۰

منابع

برای اطلاعات بیشتر به منابع زیر مراجعه کنید:

- OWASP Teting Guide 4.0: Configuring and Developing and Deployment Management Testing
  https://www.owasp.org/index.php/Testing_for_configuration_management
- OWASP Cross-site Request Forgery Cheat Sheet
  https://www.owasp.org/index.php/Cross- Site_Request_Forgery_(CSRF)_Prevention_Cheat_Sheet
- JSON Web Tokens (and Signing)
  https://jwt.io/

بررسی ۱۹ :نیازمندی‌های بررسی پیکربندی

هدف کنترل

اطمینان حاصل کنید که برنامه بررسی شده دارای :

- کتابخانه‌ها و بسترهای به‌روز باشد.
- پیکربندی امن به صورت پیش‌فرض باشد.
- مستحکم سازی کافی باشد؛ به طوری‌که تغییرات اعمال شده نسبت به پیکربندی پیش‌فرض توسط کاربر، سیستم‌های اساسی را در معرض ایجاد اشکالات یا ضعف‌های امنیتی بی‌مورد قرار ندهد.

نیازمندی‌های

#	توضیحات	۱	۲	۳	از نسخه
۱۹,۱	بررسی کنید که همه مؤلفه‌ها با پیکربندی‌ها و نسخه هایی با امنیت مناسب به روز می شوند. این به‌روزرسانی باید شامل حذف پیکربندی‌ها و پوشه‌های غیر ضروری از جمله : برنامه‌های نمونه، بسترهای مستندسازی و کاربران پیش‌فرض یا نمونه باشد.	✓	✓	✓	۳,۰
۱۹,۲	بررسی کنید که ارتباط میان مؤلفه ها،مانند ارتباط بین سرور برنامه و سرور پایگاه داده کدگذاری می شود. به‌خصوص هنگامی که مؤلفه ها در مخازن[1] و یا سیستم‌های متفاوتی قرار دارند.		✓	✓	۳,۰
۱۹,۳	بررسی کنید که ارتباطات میان مؤلفه ها مانند ارتباطات میان سرور برنامه و سرور پایگاه داده توأم با احراز هویت می‌باشد که از حساب با حداقل مجوز[2] مورد نیاز استفاده می کند.		✓	✓	۳,۰
۱۹,۴	بررسی کنید که نصب همه برنامه‌ها در جعبه شن[3] قرار گرفته، نگه داری یا ایزوله شده است تا حمله مهاجمان به سایر برنامه‌ها را به تأخیر بیندازد و یا مانع آن‌ها شود.		✓	✓	۳,۰
۱۹,۵	بررسی کنید که فرآیند تولید و نصب برنامه‌ها به روش قابل تکرار و امن صورت می‌گیرد مانند اتوماسیون CI/CD و مدیریت پیکربندی خودکار شده.		✓	✓	۳,۰
۱۹,۶	بررسی کنید که مدیرهای دارای مجوز، قابلیت بررسی بی عیبی و درستی امنیت			✓	۳,۰

[1] Repository
[2] Privilege
[3] Sand box

#	توضیحات	۱	۲	۳	از نسخه
	همه ی پیکربندی‌های مربوطه برای شناسایی دستکاری را دارند.				
۱۹,۷	بررسی کنید همه مؤلفه های برنامه به امضا می رسند.			✓	۳,۰
۱۹,۸	بررسی کنید که مؤلفه های third-party از مخازن معتبر می آیند			✓	۳,۰
۱۹,۹	بررسی کنید که همه ی پرچم های امنیتی[1] فرآیند های تولید برای زبان سطح ماشین فعال می شوند مانند : ASLR، DEP و بررسی های امنیتی.			✓	۳,۰
۱۹,۱۰	بررسی کنید که همه ی دارایی های برنامه توسط خود برنامه میزبانی می‌شوند ؛ به عنوان مثال کتابخانه های javascript، CSS stylesheets و فونت‌های وب بجای اینکه به یک CDN یا عامل خارجی وابسته باشند توسط برنامه میزبانی می‌شوند.			✓	۳,۰,۱

منابع

برای اطلاعات بیشتر به منابع زیر مراجعه کنید:

- OWASP Testing Guide 4.0: Comfiguration and Deployment Management Testing
  https://www.owasp.org/index.php/Testing_for_configuration_management

[1] Security flags

پیوست الف : چه اتفاقی برای نیازمندی‌های زیر افتاده است ...؟

# اصلی	توضیحات	وضعیت	حذف‌شده	دلیل
۲,۳	بررسی کنید که اگر از حداکثر تعداد دفعات تلاش برای احراز هویت گذشته باشد، حساب برای دوره‌ای با مدت زمان کافی قفل شود تا از حملات جست‌وجوی کورکورانه ممانعت شود.	از رده خارج شده	۲,۰	نیازمندی پیچیده‌تری جایگزین شده است. (نیازمندی ۲,۲۰)
۲,۵	بررسی کنیدکه تمامی کنترل‌های احراز هویت (شامل کتابخانه‌هایی که سرویس‌های خارجی احراز هویت را فراخوانی می‌کنند) یک پیاده‌سازی مرکزی دارند.	ادغام شده	۳,۰	برای شامل شدن همه کنترل‌های امنیت به‌صورت عمومی در نیازمندی ۱,۱۰ بررسی می‌شود.
۲,۱۰	بررسی کنید که قبل از این‌که به هرگونه عملیات حساس مخصوص برنامه‌ها مجوز داده شود، به احراز هویت دوباره نیاز داشته باشند.	از رده خارج شده	۲,۰	احراز هویت دوباره به‌ندرت مشاهده می‌شد که تصمیم گرفتیم این کنترل را حذف کنیم.
۲,۱۱	بررسی کنید که بعد از یک دوره زمانی قابل تنظیم اجرایی، اعتبارنامه‌های احراز هویت باطل می‌شود.	از رده خارج شده	۲,۰	تعلیق‌های مطلق و انقضاء اعتبارنامه به دلیل مؤثر نبودن کنترل حذف شدند.
۲,۱۴	بررسی کنید که همه اعتبارنامه‌های احراز هویت برای دسترسی به سرویس‌های خارج از برنامه کدگذاری می‌شوند و در محل محافظت شده (نه در کد منبع) دخیره می‌شوند.	به‌روزرسانی شده	۲,۰	به نیازمندی ۲,۲۱ ارتقا یافته است.
۲,۱۵	بررسی کنید که همه‌ی کدهایی که از کنترل احراز هویت‌ها استفاده یا پیاده‌سازی می‌کنند از هیچ کد مخربی تأثیر نمی‌گیرند.	جابجا شده	۲,۰	به بررسی ۱۳ کد مخرب جابجا شده است.
۲,۳۰	بررسی کنید که آیا برنامه اجازه احراز هویت را به کاربران می‌دهد. برنامه‌ها از یک سازوکار احراز هویت امن اثبات شده استفاده می‌کنند.	از رده خارج شده	۳,۰,۱	برای آزمایش شدن بسیار مبهم بود، در واقع خلاصه‌ای از همه نیازمندی‌های بررسی ۲ می‌باشد.
۳,۸	بررسی کنید که شناسه نشست در طول احراز هویت دوباره تغییر داده می‌شود.	به‌روزرسانی شده	۳,۰	به نیازمندی ۳,۷ گردش یافته است
۳,۹	بررسی کنید که شناسه نشست به هنگام خروج از سیستم تغییر می‌یابد یا پاک می‌شود.	به‌روزرسانی شده	۳,۰	به نیازمندی ۳,۷ گردش یافته است
۳,۱۳	بررسی کنید که همه‌ی پیاده‌سازی‌های کدی یا کنترل‌هایی که برای مدیریت نشست استفاده می‌شود از هیچ کد مخربی تأثیر نمی‌یابد.	جابجا شده	۲,۰	به بررسی ۱۳ کد مخرب جابجا شده است.
۳,۱۴	بررسی کنید که توکن نشست‌های احراز هویت شده که از کوکی‌ها استفاده می‌کنند، با استفاده از HttpOnly محافظت می‌شوند.	به‌روزرسانی شده	۳,۰	به نیازمندی ۳,۱۳ جابجا شده است.

# اصلی	توضیحات	وضعیت	حذف‌شده	دلیل
۳,۱۵	بررسی کنید که توکن نشست‌های احراز هویت شده که از کوکی‌ها استفاده می‌کنند با فعال‌سازی ویژگی secure محافظت می‌شوند.	به‌روزرسانی شده	۳,۰	به نیازمندی ۳,۱۳ جابجا شده است.
۴,۲	بررسی کنید که کاربرها می‌توانند تنها به URL های امنی دسترسی پیدا کنند که برای هر کدام از آن‌ها مجوزهای خاصی دارند.	به‌روزرسانی شده	۳,۰	به نیازمندی ۴,۱ گردش یافته است
۴,۳	بررسی کنید که کاربرها می‌توانند تنها به فایل‌های اطلاعاتی دسترسی پیدا کنند که برای هر کدام از آن‌ها مجوزهای خاصی دارند.	به‌روزرسانی شده	۳,۰	به نیازمندی ۴,۱ گردش یافته است
۴,۱۳	بررسی کنید که محدودیت‌هایی که برای ورودی‌ها و دسترسی‌های تحمیل شده تجاری روی برنامه وجود دارند به‌نحوی قابل عبور نباشند. (مانند معامله‌های روزانه محدود یا توالی‌های امور مهم)	جابجا شده	۳,۰	به بررسی ۱۵ منطق تجارت جابجا شده است
۴,۱۵	بررسی کنید که همه‌ی کدهایی که کنترل دسترسی‌ها را استفاده یا پیاده‌سازی می‌کنند از هیچ کد مخربی تأثیر نمی‌یابند.	جابجا شده	۲,۰	به بررسی ۱۳ کنترل بدافزارها جابجا شده است.
۵,۲	بررسی کنید که الگو های اعتبارسنجی مثبت مشخص می‌شوند و به همه ورودی‌ها اعمال می‌گردند.	از رده خارج شده	۲,۰	به‌دلیل این‌که پیاده‌سازی به‌خصوص به‌صورت رایگان برای ورودی‌های متنی بسیار مشکل بود، حذف شده است.
۵,۴	بررسی کنید که یک مجموعه حروف مانند UTF-8 برای همه منابع ورودی مشخص می‌شود.	از رده خارج شده	۳,۰	به دلیل پیاده‌سازی بسیار مشکل اکثر زبان‌ها، حذف شده است.
۵,۷	بررسی کنید که همه‌ی ورودی‌های با اعتبار ناموفق ثبت می‌شوند.	از رده خارج شده	۳,۰	به دلیل ایجاد گزارش‌های بیهوده که مورد چشم‌پوشی واقع می‌شد، حذف شده است.
۵,۸	بررسی کنید که همه داده‌های ورودی مقدم بر اعتبارسنجی برای کدگشاهای سطح پایین یا مفسرها استفاده می‌شوند.	از رده خارج شده	۳,۰	از نوع ۱ تکنولوژی حذف شده است و برای چارچوب‌های مدرن مشکلی نخواهد داشت.
۵,۹	بررسی کنید که همه کنترل‌های اعتبارسنجی‌های ورودی‌ها توسط هیچ کد مخربی تأثیر نمی‌یابند.	جابجا شده	۲,۰	به بررسی ۱۳ کنترل بدافزارها جابجا شده است .
۵,۱۴	بررسی کنید که محیط زمان اجرا به XML injections آسیب‌پذیر و حساس نمی‌باشد و یا کنترل‌های امنیت از XML injections جلوگیری می‌کنند.	ادغام شده	۳,۰	با نیازمندی ۵,۱۳ ادغام شده است
۵,۱۵	نیازمندی خالی	حذف شده	۳,۰	این نیازمندی هرگز وجود نداشته است.
۵,۱۹	بررسی کنید که برای کدگذاری هر نوع خروجی که توسط برنامه اعمال	ادغام شده	۳,۰	برای شامل شدن همه کنترل های امنیت بصورت

# اصلی	توضیحات	وضعیت	حذف‌شده	دلیل
	می‌شود، کنترل امنیت برای خروجی در مقصد مورد نظر وجود دارد.			عمومی در نیازمندی ۱,۱۰ بررسی شده است.
۷,۱	بررسی کنید که همه توابع رمزنگاری شده که برای حفاظت رمزها از کاربر استفاده می‌شود در سمت سرویس‌دهنده پیاده‌سازی می‌شوند.	از رده خارج شده	۳,۰	بسیاری از پاسخگویی‌های مدرن و برنامه‌های موبایل در طراحی خود این مورد را رعایت کرده‌اند.
۷,۳	بررسی کنید هرگونه شاه رازی در برابر دسترسی غیرمجاز محافظت می‌شود. (یک شاه راز یک اعتبارنامه برنامه است که به‌عنوان متن اصلی روی دیسک ذخیره می‌شود که برای حفاظت از دسترسی به اطلاعات پیکربندی امنیت مورد استفاده قرار می‌گیرد.)	جابجا شده	۳,۰	به نیازمندی ۲,۲۹ جابجا شده است.
۷,۴	بررسی کنید که چکیده‌سازهای رمز عبور به‌هنگام ایجاد شدن، نمک زده می‌شوند.	جابجا شده	۲,۰	به نیازمندی ۲,۱۳ جابجا شده است.
۷,۵	بررسی کنید که ماژول‌های ناموفق رمزنگاری شده ثبت می‌شوند.	از رده خارج شده	۲,۰	ایجاد گزارش‌های غیرضروری که هرگز بررسی نمی‌شوند ضد تولید محسوب می‌شوند.
۷,۱۰	بررسی کنید که همه کدهایی که از ماژول رمزنگاری شده پشتیبانی یا استفاده می‌کنند از کدهای مخرب تأثیر نمی‌یابند.	جابجا شده	۲,۰	به بررسی ۱۳ جابجا شده است.
۸,۲	بررسی کنید مدیریت همه خطاها روی دستگاهی قابل اعتماد انجام می‌شود.		۳,۰	از رده خارج شده
۸,۳	بررسی کنید که همه کنترل رویدادنگاری‌ها روی سرویس‌دهنده پیاده‌سازی می‌شوند.	جابجا شده	۳,۰	به بررسی ۱۳ کنترل بدافزارها جابجا شده است.
۸,۹	بررسی کنید که پیاده‌سازی رویدادنگاری که در سطح برنامه انجام می‌شود توسط نرم‌افزار استفاده می‌شود.	جابجا شده	۳,۰	بیشتر به یک کنترل عام معماری تبدیل شده است
۸,۱۱	بررسی کنید که یک ابزار تحلیل گزارش وجود دارد که این امکان را به تحلیل‌گر می‌دهد که رویدادهای ثبت شده را جستجو کند. این جستجو مبتنی بر ترکیبات معیارهای جستجو در همه زمینه‌ها، در فرمت ضبط رویدادها ،که توسط سیستم پشتیبانی می‌شود، است.	از رده خارج شده	۳,۰	به‌دلیل عدم نیاز برای نرم‌افزار امن حذف شده است .
۸,۱۲	بررسی کنید که همه‌ی کدهایی که مدیریت خطا وکنترل گزارش را استفاده یا پیاده‌سازی می‌کنند از هیچ کد مخربی تأثیر نمی‌یابند.	جابجا شده	۲,۰	برای شامل شدن همه کنترل‌های امنیت به‌صورت عمومی در نیازمندی ۱,۱۳ بررسی شده است.
۸,۱۵	بررسی کنید که رویدادنگاری قبل از اجرای تراکنش انجام می‌شود. اگر رویدادنگاری ناموفق بود (به‌عنوان مثال به‌دلیل تکمیل بودن ظرفیت دیسک یا مجوزهای ناکافی) برنامه بی‌خطر متوقف می‌شود. این برای زمانی است که درستی و عدم تخلف واجب است.	از رده خارج شده	۳,۰	به‌دلیل ریزبینانه شدن کنترل که تنها در درصد پایینی از برنامه‌ها کاربرد دارد، حذف شده است.

# اصلی	توضیحات	وضعیت	حذف‌شده	دلیل
۱۰,۲	بررسی کنید که اتصالات TLS ناموفق به اتصالات HTTP ناامن تنزل نمی‌کنند.	ادغام شده	۳,۰	با نیازمندی ۱۰,۳ ادغام شده است.
۱۰,۷	بررسی کنید که تمام اتصالات به سیستم خارجی که شامل اطلاعات حساس و توابع می‌باشد از یک حساب که به داشتن حداقل مجوز ضروری برای برنامه تنظیم می‌شود، استفاده کنند تا به درستی عمل کنند.			
۱۰,۹	بررسی کنید که کدگذاری کاراکترهای مخصوص برای همه اتصالات مشخص شده باشد مانند UTF-8			
۱۱,۱	از رده خارج شده			
۱۱,۴	از رده خارج شده			
۱۱,۵	از رده خارج شده			
۱۱,۶	از رده خارج شده			
۱۱,۷	از رده خارج شده			
۱۱,۸	از رده خارج شده			
۱۱,۴	از رده خارج شده			
۱۳,۱	از رده خارج شده			
۱۳,۲	از رده خارج شده			
۱۳,۳	از رده خارج شده			
۱۳,۴	از رده خارج شده			
۱۳,۵	از رده خارج شده			
۱۳,۶	از رده خارج شده			
۱۳,۷	از رده خارج شده			
۱۳,۸	از رده خارج شده			
۱۳,۹	از رده خارج شده			
۱۵,۱- ۱۵,۷ ۱۵,۹	بخش منطق تجاری	ادغام شده	۳,۰	اکثر بخش ۱۵ به ۱۵,۸ و ۱۵,۱۰ ادغام شده است
۱۵,۱۱	بررسی کنید که برنامه خطرات مربوط به جاسوسی، دستکاری کردن، انکار کردن، افشای اطلاعات و بالابردن سطح امتیاز (STIDE) را پوشش می‌دهد.	تکراری شده	۳,۰	نیازمندی تکراری. در نیازمندی ۱,۶ به آن پرداخته شده است.
۱۶,۴	بررسی کنید که پارامترهای به‌دست آمده از منابع نامعتبر پیش از کانناليزه شدن و اعتبارسنجی ورودی‌ها در دستکاری نام فایل‌ها ، نام مسیرها یا هر گونه شیء سیستم‌فایل‌ها مورد استفاده قرار نمی‌گیرند، تا از حمله‌های گنجاندن	جابجا شده	۳,۰	به نیازمندی ۱۶,۲ جابجا شده است.

# اصلی	توضیحات	وضعیت	حذف‌شده	دلیل
	فایل محلی جلوگیری شود.			
۱۷,۱	بررسی کنید که مشتری گواهی‌نامه‌های SSL را اعتبارسنجی می کند.	از رده خارج شده	۳,۰	نیازمندی تکراری. قبلاً نیازمندی عمومی در بررسی ۱۰ به آن پرداخته شده است.
۱۷,۷	از رده خارج شده			
۱۷,۸	از رده خارج شده			
۱۷,۱۰	از رده خارج شده			
۱۷,۱۱	از رده خارج شده			
۱۷,۱۲	از رده خارج شده			
۱۷,۱۳	از رده خارج شده			
۱۷,۱۴	از رده خارج شده			
۱۷,۱۵	از رده خارج شده			
۱۷,۱۶	از رده خارج شده			
۱۷,۱۷	از رده خارج شده			
۱۷,۱۸	از رده خارج شده			
۱۷,۱۹	از رده خارج شده			
۱۷,۲۰	از رده خارج شده			
۱۷,۲۱	از رده خارج شده			
۱۷,۲۲	از رده خارج شده			
۱۲,۲۳	از رده خارج شده			
۱۷,۲۴	از رده خارج شده			

پیوست ب: واژه‌نامه

- **کنترل دسترسی:** سازوکاری است برای محدود کردن دسترسی به فایل ها ، توابع ارجاع داده شده ، آدرس های اینترنتی و داده ها مبنی بر هویت کاربران یا گروه هایی که هر کدام به آن ها تعلق دارند.
- **تصادفی سازی چیدمان فضای آدرس(ASLR)** : روشی برای کمک به محافظت در برابر حمله های سرریز بافر
- **امنیت برنامه** : امنیت سطح برنامه به عنوان مثال بجای تمرکز بر روی سیستم اعمال اساسی یا شبکه های متصل، روی تحلیل مؤلفه هایی که لایه برنامه مدل مرجع متصل سیستم های باز (OSI Model)را در بر می گیرد، تمرکز می کند .
- **بررسی امنیت برنامه** : بررسی فنی یک برنامه در برابر OWASP ASVS
- **گزارش بررسی امنیت برنامه** : گزارشی است که به طور کلی نتایج و تحلیل های بعد از آن که توسط تایید کننده برای یک برنامه مورد نظر بدست آمده را مستند می کند .
- **احراز هویت** : بررسی هویت درخواست شده ی کاربر برنامه
- **بررسی خودکار:** استفاده از ابزار های خودکار(چه ابزار های تحلیل پویا، چه ابزار های تحلیل ایستا و یا هر دو) که برای پیدا کردن مشکلات از امضای آسیب پذیری استفاده می کند.
- **درهای پشتی(back doors)** : یک نوع کد مخرب که به دسترسی غیر مجاز به برنامه مجوز می دهد.
- **فهرست سیاه:** فهرستی از داده ها یا عملیات ها که مجوز ندارند مانند : فهرستی از کاراکترهایی که به عنوان ورودی مجوز ندارند.
- **Cascading Style Sheets (CSS)** :یک زبان ورق های تعریف (style sheets) که برای توصیف نمایش معنایی مستندات نوشته شده به زبان نشانه گذاری شده مانند HTML استفاده می شود.
- **مرجع صدور گواهی(CA)** : نهادی که گواهی دیجیتالی صادر می کند.
- **امنیت ارتباط** : محافظت اطلاعات برنامه هنگامی که اطلاعات میان مؤلفه های یک برنامه ، میان مشتری و سرور و میان سیستم های خارجی و برنامه تبادل می شود.
- **مؤلفه** : واحد کاملی از کد ، به همراه دیسک مربوط و واسط های شبکه که با سایر مؤلفه ها ارتباط برقرار می کند.
- **Cross-Site Scripting (XSS)** : آسیب پذیری امنیتی که معمولا دربرنامه تحت وب پیدا می شوند که اجازه تزریق نبشته(script) های سمت مشتری را به محتوا می دهد.
- **ماژول رمز نگاری شده** : سخت افزار، نرم افزار ، یا firmware که الگوریتم های رمزنگاری شده را پیاده سازی و یا کلید های رمز نگاری شده را تولید می کند.
- **حمله منع خدمت (DoS)** : با تعداد درخواست های بیشتر از توانایی کنترل برنامه برای برنامه سیلاب سازی می کند.
- **بررسی طراحی** : بررسی فنی معماری امنیت برنامه
- **بررسی پویا** : استفاده از ابزار خودکار که برای پیدا کردن مشکلات برنامه در حال اجرا از امضای های آسیپ پذیری استفاده می کند.
- **Eastern Egg** : یک نوع کد مخرب که تا زمانی که مجموعه ورودی خاص از سوی کاربر وارد نشود ، فعال نمی شود.
- **سیستم های خارجی** : برنامه یا ویروس سمت سرور که بخشی از برنامه محسوب نمی شود .

- **FIPS 140-2**: استانداردی که می تواند به عنوان پایه برای بررسی و طراحی و پیاده سازی ماژول های رمز نگاری شده مورد استفاده قرار بگیرد.
- **شناسه منحصر به فرد به صورت سراسری(GUID)** : شماره ارجاعی منحصر به فردی که به عنوان شناسه در نرم افزار استفاده می شود.
- **زبانHyperText Markup(HTML)**: زبان نشانه گذاری اصلی برای خلق صفحه های وب و سایر اطلاعاتی که در یک مرورگر وب نمایش داده می شود.
- **پروتکل انتقال ابر متنی(HTTP)**: یک پروتکل برنامه برای سیستم های اطلاعاتی توزیع شده ، مشارکتی و ابررسانه. این پروتکل پایه و اساس ارتباط اطلاعاتی برای تار جهان گستر(وب) می باشد.
- **اعتبار سنجی ورودی ها** : اعتبارسنجی وکاننالیزه کردن ورودی های نامعتبر کاربر
- **پروتکل راهنمایی سبک دسترسی(LDAP)** : یک پروتکل برنامه برای دسترسی و نگهداری سرویس های اطلاعاتی راهنمایی توزیع شده روی شبکه
- **کد مخرب** : کدی که به برنامه در حال توسعه به صورت نامشخص به صاحب برنامه معرفی می شود که سیاست امنیت مورد نظر برنامه را دور می زند. کد مخرب مشابه بدافزار (مانند ویروس یا کرم) عمل نمی کند.
- **بدافزار:** کد قابل اجرایی که به برنامه در حال اجرا بدون اطلاع کاربر و یا مدیر برنامه معرفی می شود.
- **پروژه امنیت برنامه باز وب** : OWASP یک جامعه باز و رایگان جهانی می باشد که روی بهبود امنیت نرم افزار برنامه ها تمرکز کرده است. ماموریت ما این است که امنیت برنامه را قابل دیدن کنیم. بنابراین مردم یا سازمان ها می توانند تصمیمات آگاهانه درباره خطرهای امنیت برنامه بگیرند.
- **کد گذاری خروجی** : اعتبار سنجی خروجی برنامه به مرورگر های وب و سیستم های خارجی
- **اطلاعات شخص قابل شناسایی** : اطلاعاتی که می توان به خودی خود یا باسایر اطلاعات برای شناسایی، برقراری تماس یا مشخص کردن محل یک نفر یا شناسایی فرد در متن استفاده کرد.
- **اعتبار سنجی مثبت** : به فهرست سفید مراجعه کنید.
- **معماری امنیت** : تجریدی از طراحی یک برنامه است که کجا وچگونگی استفاده شدن از کنترل های امنیت و هم چنین محل و حساسیت هم داده های یک برنامه و هم داده های یک کاربر را شناسایی و توصیف می کند.
- **پیکربندی امنیت** : پیکر بندی زمان اجرای یک برنامه که روی چگونگی استفاده شدن کنترل های امنیت تاثیر می گذارد.
- **کنترل امنیت** : یک تابع یا مؤلفه که بررسی امنیتی را انجام می دهد (مانند بررسی امنیت یک دسترسی) یا زمانی که فراخوانی می شود یک تاثیر امنیتی را نتیجه می دهد(مانند ایجاد یک سابقه ممیزی)
- **تزریق SQL:** روش تزریق که درصورتی که عبارتهای SQL مخرب به یک ورودی وارد شوند. برای حمله داده مبنا برنامه ها استفاده می شود.
- **بررسی ایستا** : استفاده از ابزار خودکار که برای پیدا کردن مشکلات در کد منبع برنامه از امضای های آسیپ پذیری استفاده می کند.
- **هدف بررسی** : اگر شما در حال انجام بررسی برنامه طبق نیازمندی‌های OWASP ASVS هستید ، این بررسی یک برنامه مورد نظر خواهد بود. این برنامه ، هدف بررسی یا ToV نامیده می شود.

- **مدل سازی تهدید:** روشی که شامل توسعه به طور فزاینده ی معماری های اصلاح شده امنیت برای شناسایی عاملان تهدید ، مناطق امنیت ، کنترل های امنیت و دارایی های مهم تجاری و فنی می باشد .
- **امنیت لایه ی تبادل :** پروتکل های رمزنگاری شده که امنیت ارتباط روی اینترنت را فراهم می کند.
- **قطعه های URL/URI/URL:** یک شناسه منبع یکنواخت رشته ای از کاراکترهاست که برای شناسایی اسم یا منبع یک وب استفاده می شود. یک یابنده منبع یکنواخت که عملا به عنوان یک ارجاع دهنده به یک منبع استفاده می شود.
- **آزمایش پذیرش کاربر(UAT)** : یک محیط آزمایش که بطور سنتی مانند یک محیط تولید جایی که همه آزمایشات نرم افزارها قبل از اجرا صورت می گیرد ، عمل می کند.
- **بررسی کننده :** فرد یا تیمی که در حال بررسی نیازمندی‌های OWASP می باشد.
- **فهرست سفید :** فهرستی از عملیات ها و داده های مجاز مانند فهرستی از کاراکترهایی که مجوز اعتبارسنجی ورودی ها را دارند.
- **XML**: یک زبان نشانه گذاری شده که مجموعه از از قوانین را برای کدگذاری مستندات مشخص می کند.

پیوست ج: منابع

به احتمال زیاد پروژه‌های اوسپ زیر برای کاربران و متقاضیان این استاندارد مفید باشد :

- راهنمای آزمایش اوسپ

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

- راهنمای مرور کد اوسپ

http://www.owasp.org/index.php/Category:OWASP_Code_Review_Project

- برگه های تقلب اوسپ

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

- کنترل های پیشگیرانه اوسپ

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

- ۱۰ مورد برتر اوسپ

https://www.owasp.org/index.php/Top_10_2013-Top_10

- ۱۰ مورد برتر برای موبایل اوسپ

https://www.owasp.org/index.php/Projects/OWASP_Mobile_Security_Project_-_Top_Ten_Mobile_Risks

به طور مشابه، به احتمال زیاد وب سایت های زیر برای کاربران و متقاضیان این استاندارد مفید باشد :

- ضعف های شمارشی معمولMITRE

http://cwe.mitre.org/

- شورای استانداره های امنیت PCI

https://www.pcisecuritystandards.org

- نیازمندی‌های نسخه ۳ استاندارد امنیت داده ها(DSS) PCI و روش های بررسی امنیت

https://www.pcisecuritystandards.org/documents/PCI_DSS_v3.pdf

پیوست د: نقشه برداری های استانداردها

PCI DSS 6.5 از ۱۰ مورد برتر اوسپ در سال های ۲۰۰۷/۲۰۰۴ به همراه تعمیم های فرآیند های اخیر نتیجه می شود. ASVS مجموعه ای اکید از ۱۰ مورد برتر سال ۲۰۱۳ اوسپ(۱۵۴ مورد تا ۱ مورد). بنابراین تمام موضوعاتی که توسط ۱۰ مورد برتر اوسپ و PCI DSS 6.5.x پوشش داده می شوند توسط کنترل نیازمندی‌های مناسب تر ASVS مدیریت می شوند. به عنوان مثال "احراز هویت نقض شده و مدیریت نشست" دقیقا در بررسی ۲ احراز هویت و بررسی۳ مدیریت نشست در نقشه قرار می گیرند.

اگرچه سطح ۲ بررسی اکثر نیازمندی‌های PCI DSS 6.5 بجز ۶،۵،۳ و ۶،۵،۴ را آدرس دهی می کند، نقشه برداری کامل توسط سطح ۳ بررسی بدست آمده است. موضوعات پردازشی از جمله PCI DSS 6.5.6 توسط ASVS پوشش داده نمی شود.

PCI-DSS 3.0	ASVS 3.0	توضیحات
6.5.1اشکالات تزریق بخصوص تزریق SQL. همچنین تزریق دستور سیستم عامل ، LDAP و اشکالات تزریق XPath را هم علاوه بر سایر اشکالات تزریق در نظر بگیرید.	۵.۱۱،۵.۱۲،۵.۱۳،۸.۱۴،۱۶.۲	نقشه برداری دقیق
۶،۵،۲ بافر سرریز می شود	۵،۱	نقشه برداری دقیق
۶،۵،۳ فضای رمزنگاری شده ناامن	تمامی بررسی ۷	نقشه برداری جامع از سطح ۱ به بعد
۶،۵،۴ ارتباطات نا امن	تمامی بررسی ۱۰	نقشه برداری جامع از سطح ۱ به بعد
مدیریت نامناسب خطا ها	۳.۶،۷.۲،۸.۱،۸.۲	نفشه برداری دقیق
۶،۵،۵ نبشته سایت تقلبی	۵.۱۶،۵.۲۰،۵.۲۱،۵.۲۴،۵.۲۵،۵.۲۶،۵.۲۷،۱۱.۴،۱۱ .۱۵	ASVS ، XSS را با برجسته سازی پیچیدگی حفاظت XSS به خصوص برای برنامه هایی که از آن استفاده می کنند به تعدادی نیازمندی‌های تجزیه می کند.
کنترل دسترسی نامناسب(از جمله ارجاع های شیء مستقیم ناامن ، ناموفق بودن در محدود کردن کنترل URL ، گذر گاه دایرکتوری و ناموفق بودن در محدود کردن کنترل کاربر به توابع).	تمامی بررسی ۴	نقشه برداری جامع از سطح ۱ به بعد

PCI-DSS 3.0	ASVS 3.0	توضیحات
۶,۵,۹ جعل درخواست سایت تقلبی (CSRF)	۴,۱۳	نقشه برداری دقیق.ASVS حفاظت از CSRF را به عنوان جنبه کنترل دسترسی درنظر میگیرد.
۶,۵,۱۰ احراز هویت نقض شده و مدیریت نشست	تمامی وراسی ۲ و بررسی ۳	نقشه برداری جامع از سطح ۱ به بعد