



OWASP

Open Web Application
Security Project

استاندارد بررسی امنیت نرم افزار نسخه ۳.۰.۱

۵	سپاسگزاری
۵	درباره‌ی استاندارد
۵	حق کپی رایت و مجوز
۸	پیشگفتار
۸	افزونه‌های جدید این نسخه
۹	استفاده از استاندارد بررسی امنیت نرم‌افزار
۹	مراحل بررسی امنیت نرم‌افزار
۱۰	چطور از این استاندارد استفاده کنیم
۱۱	پیاده‌سازی عملی ASVS
۱۴	موارد مطالعاتی
۱۴	مورد اول : تحت عنوان راهنمای تست امنیتی
۱۵	مورد دوم : تحت عنوان یک SDLC امن
۱۶	ارزیابی نرم‌افزار به یک سطح بررسی رسید
۱۶	OWASP در مقابل گواهینامه‌های ASVS و trustMarks
۱۶	راهنمایی برای سازمان های صادرکننده گواهینامه
۱۶	نقش ابزار تست نفوذ خودکار
۱۷	نقش تست نفوذ
۱۷	به عنوان راهنمای معماری امنیتی دقیق
۱۷	به عنوان یک جایگزین مناسب برای checklist های امنیتی
۱۷	به عنوان یک راهنما برای واحد خودکار و تست های یکپارچه
۱۸	به عنوان تمرین توسعه امن
۱۸	پروژه های OWASP با استفاده از ASVS
۱۸	Framework دانش امنیتی
۱۸	OWASP ZED ATTACK PROXY
۱۸	OWASP Cornucopia
۱۹	نیازمندی های بررسی دقیق
۲۰	بررسی-اول : نیازمندی های بررسی معماری، طراحی و مدل تهدیدی
۲۰	هدف کنترل
۲۰	نیازمندی ها
۲۱	منابع
۲۲	بررسی-دوم : نیازمندی های بررسی احراز هویت
۲۲	هدف کنترل
۲۲	نیازمندی ها
۲۴	منابع
۲۵	بررسی-سوم : نیازمندی های بررسی مدیریت نشست

۲۵	هدف کنترل
۲۵	نیازمندی ها
۲۶	منابع
بررسی-چهارم : نیازمندی های بررسی کنترل سطح دسترسی	
۲۷	هدف کنترل
۲۷	نیازمندی ها
۲۸	منابع
بررسی-پنجم : نیازمندی های بررسی کنترل ورودی های مخرب	
۲۹	هدف کنترل
۲۹	نیازمندی ها
۳۱	منابع
بررسی-ششم : نیازمندی های بررسی کدگذاری و بی اثر سازی خروجی	
بررسی-هفتم : نیازمندی های بررسی رمزنگاری در حالت REST	
۳۳	هدف کنترل
۳۳	نیازمندی ها
۳۴	منابع
بررسی-هشتم : نیازمندی های بررسی مدیریت خطاها و ثبت گزارش	
۳۵	هدف کنترل
۳۵	نیازمندی ها
۳۶	منابع
بررسی-نهم : نیازمندی های بررسی محافظت داده ها	
۳۷	هدف کنترل
۳۷	نیازمندی ها
۳۸	منابع
بررسی-دهم : نیازمندی های بررسی امنیت ارتباطات	
۳۹	هدف کنترل
۴۰	نیازمندی ها
۴۱	منابع
بررسی-یازدهم : نیازمندی های بررسی پیکربندی امنیتی HTTP	
۴۲	هدف کنترل
۴۲	نیازمندی ها
۴۳	منابع
بررسی-دوازدهم : نیازمندی های بررسی پیکربندی امنیتی	
بررسی-سیزدهم : نیازمندی های بررسی کنترل مخرب	
۴۵	

۴۵	هدف کنترل
۴۵	نیازمندی ها
۴۵	منابع

۴۶ بررسی-چهاردهم : نیازمندی های بررسی امنیت داخلی

۴۷	بررسی-پانزدهم : نیازمندی های بررسی منطق تجارت
۴۷	هدف کنترل
۴۷	نیازمندی ها
۴۷	منابع

۴۸	بررسی-شانزدهم : نیازمندی های بررسی فایل ها و منابع
۴۸	هدف کنترل
۴۸	نیازمندی ها
۴۹	منابع

۵۰	بررسی-هفدهم : نیازمندی های بررسی موبایل
۵۰	هدف کنترل
۵۰	نیازمندی ها
۵۱	منابع

۵۲	بررسی-هجدهم : نیازمندی های بررسی سرویس های وب
۵۲	هدف کنترل
۵۲	نیازمندی ها
۵۳	منابع

۵۴	بررسی-نوزدهم : نیازمندی های بررسی پیکربندی
۵۴	هدف کنترل
۵۴	نیازمندی ها
۵۵	منابع

۵۶ ضمیمه الف :چه اتفاقی برای این نیازمندی های افتاده است؟

۶۱ ضمیمه ب : واژه نامه

۶۴	ضمیمه پ : منابع
۶۵	ضمیمه ت : نقشه های استاندارد

سپاسگزاری

درباره استاندارد

استاندارد بررسی امنیت نرم افزار، یک لیست از نیازمندی های امنیت نرم افزار یا تست هایی است که میتوان توسط معمارها، توسعه دهنده ها، کارشناس های 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 می تواند بعنوان مرجع بررسی، در فرایند آزمون نفوذپذیری استفاده گردد.



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 کپی رایت



کپی رایت 2008-2016 سازمان OWASP.

این مستند تحت مجوز [Creative Commons AttributionShareAlike4.0](https://creativecommons.org/licenses/by-sa/4.0/legalcode) منتشر شده است. شما می توانید به صورت رایگان این مستند را به اشتراک بگذارید یا از آن استفاده تجاری و غیرتجاری کنید اما با قید منبع. برای اطلاعات بیشتر لطفا به سایت زیر مراجعه کنید:

<https://creativecommons.org/licenses/by-sa/4.0/legalcode>

نسخه ۳.۰، ۲۰۱۶

همکاران و بازنگران	نویسندگان ارشد	رهبران پروژه
Abhinav Sejal	Jim Manico	Andrew van der Stock
Anthony Weems		Daniel Cuthbert
Ari Kesaniemi		
Boy Baukema		
Colin Watson		
Cristinel Dumtiru	مترجمان	سرپرست مترجمان
David Ryan	Mohammad Kahani	Sajjad Pourali
Francois-Eric Guyomarc'h	Mahdi Bagheri	
Gary Robinson	Saeed KhademiDoroh	
Glenn Ten Cate	Sajjad Pourali	
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

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

همکاران و بازنگران	نویسندگان ارشد	رهبران پروژه
Andrew van der Stock	Jim Manico	Mike Boberski
Barry Boyd		Jeff Williams
Bedirhan Urgan		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 نرم افزار (ASVS) خوش آمدید. ASVS به طور جامع محور، سعی کرده چهارچوبی¹ مناسب برای نیازمندی های امنیتی و همچنین کنترل های امنیتی ایجاد کند. این موارد بیشتر بر قانونمند کردن نیازمندی های کنترل امنیت تمرکز دارد، درواقع موارد استفاده اصلی از آن، هنگام طراحی، توسعه و آزمایش برنامه های تحت وب می باشد.

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 می شود تا تمامی افراد تازه وارد به این استاندارد، بتوانند برنامه ریزی مناسبی برای استفاده هرچه بهتر از آن داشته باشند، همچنین این کار باعث کمک به شرکت های موجود برای استفاده از تجربیات دیگران نیز می باشد.

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 استاندارد ها باشد و این خود باعث ایجاد چالش می شود. با این حال، ما امیدواریم که آخرین به روز رسانی های 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 3.0 هست؟

در این نسخه برای کاربردی تر شدن استاندارد چندین بخش جدید از جمله پیکربندی، سرویس های وب، برنامه های مدرن (سمت مشتری)² و همچنین دستگاه های اینترنت اشیا اضافه شده است. مخصوصا در حوزه برنامه های مدرنی که معمولا برنامه های پاسخگو با گستردگی html5 به عنوان front-end یا سرویس گیرنده موبایل مجموعه ای مشترک از سرویس های وب RESTful، استفاده کننده روش SAML برای احراز هویت می باشد.

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

همچنین یک نقشه جامع برای فرهنگ لغت CWE طراحی کردیم. نقشه CWE می تواند برای شناسایی اطلاعاتی از جمله احتمال exploit، نتیجه موفق یک exploit و همچنین به طور گسترده در مورد درک بهتر این که اگر کنترل های امنیتی استفاده نشود یا به طور موثر استفاده نشده باشد و با ضعف همراه باشد، چه چیزی می تواند به سمت نادرستی حرکت کند، صحبت می کند.

در نهایت به جامعه رسیدیم و جلسات نظرسنجی و بازنگری را در AppSec EU 2015 (امنیت نرم افزار اروپا 2015) و جلسه کاری نهایی را در AppSec USA 2015 (امنیت نرم افزار امریکا) برگزار کردیم تا این مستند درون خود، شمار زیادی از بازخورد های اجتماعی را شامل شود. در طول بازنگری ها اگر تغییرات کنترلی قابل توجهی رخ می داد، کنترل جدید را ایجاد می کردیم و نسخه قبلی را به عنوان منسوخ شده و قدیمی می شناختیم. این کار را عمدا انجام دادیم تا مجددا از آن نسخه های قدیمی استفاده نشود چرا که نسخه های قدیمی خودشان باعث سردرگمی خواهند شد. همچنین یک نقشه کامل از تغییرات رخ داده در صفحه ضمیمه الف قرار دادیم.

¹ Framework

² Client side

روی هم رفته، این نسخه شامل یکی از تغییرات بزرگ در طول تاریخ استاندارد می باشد. امید است نسخه بروز استاندارد را پیدا کنید و به همان درستی که در تصور ماست از آن استفاده کنید.

استفاده از استاندارد بررسی امنیت برنامه

ASVS دو هدف مهم به شرح زیر دارد:

- به سازمان ها در راستای توسعه یک برنامه امن و حفاظت از آن کمک کند.
- به سرویس های امنیتی، فروشندگان ابزار امنیتی و همچنین مصرف کنندگان اجازه دهد تا نیازها و پیشنهادات 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 قرار می دهد.

- سطح اول ASVS برای تمامی نرم افزارها می باشد.
- سطح دوم ASVS برای برنامه هایی است که اطلاعات حساسی دارند و طبیعتاً نیاز به حفاظت از اطلاعاتشان دارند.
- سطح سوم ASVS برای برنامه هایی است که بیشترین حساسیت را دارند - برنامه هایی که هر روزه در حال تراکنش های 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 روش های استفاده از استاندارد این است که به کمک آن یک چک لیست^۱ مخصوص برنامه یا سازمان خود طراحی کنید که بتوانید به کمک آن از امنیت برنامه یا سازمان خود مطمئن شوید. طراحی کردن این چک لیست ASVS به شما کمک می کند تا تمرکز بیشتر بر روی نیازمندی های برنامه و محیط کاری خود داشته باشید.

سطح اول : فرصت طلبی

یک نرم افزار به شرطی به سطح اول ASVS میرسد که بتواند به طور مناسب از آسیب پذیری های گوناگون حفظ شود. این آسیب پذیری ها خیلی ساده کشف می شوند و در لیست ده آسیب برتر چک لیست OWASP یا دیگر چک لیست های امنیتی ذکر شده اند.

سطح اول معمولا برای برنامه هایی که به سطوح ساده اما صحیحی از کنترل نیاز دارند، یا برای ارایه یک روش تجزیه و تحلیل سریع برنامه های سازمانی ، یا کمک به توسعه یک لیست اولویت بندی شده از نیازمندی های امنیتی به عنوان رفع بخشی از نیازمندی های برنامه ها مناسب است. سطح اول از کنترل ها می تواند به صورت خودکار (با استفاده از ابزار) و هم به صورت ساده و دستی، بدون دسترسی به کد منبع^۲ تضمین شود. همچنین باید بدانید که سطح اول از نظر ما حداقل امنیتی است که هر برنامه ای باید رعایت کند.

در اغلب اوقات تهدید برنامه ها توسط مهاجمانی است که از تکنیک های ساده و کم تلاش برای شناسایی و بهره برداری از آسیب پذیری ها استفاده می کنند و این موضوع در مقایسه با یک مهاجم مصمم که با انرژی بسیار زیاد برای هدف خود به طور متمرکز تلاش می کند، مورد بررسی است. اگر شما داده و اطلاعات مهمی داشته باشید به ندرت پیش می آید که در همین سطح اول از امنیت برنامه خود رضایت کافی داشته باشید.

سطح دوم : استاندارد

یک نرم افزار به شرطی به سطح دوم ASVS می رسد که بتواند از خود به خوبی در برابر خطرات ناشی از نرم افزارهای امروزی محافظت کند.

سطح دوم این اطمینان را به ما می دهد که کنترل های استفاده شده درون برنامه به موقع و موثر خواهد بود. سطح دوم معمولا برای برنامه هایی مناسب است که معاملات مهم کسب و کار را شامل می شوند، از جمله آن هایی که پردازش اطلاعات مهم پزشکی ، تجارت ها و کارکردها یا سایر پردازش های حساس را اجرا می کنند.

معمولا تهدیدات برنامه های سطح دوم دارای آسیب پذیری و نقاط ضعفی است که معمولا کشف و بهره برداری از آن ها توسط مهاجمان ماهر اتفاق می افتد که استفاده از ابزار خاص و تکنیک هایی که بر روی آنها مسلط هستند آن را امکان پذیر می کند.

سطح سوم : پیشرفته

سطح سوم ASVS در حال حاضر بالاترین سطح امنیتی از نظر ماست. این سطح به طور معمول برای برنامه هایی مورد نیاز است که نیازمند سطح قابل توجهی از بررسی امنیتی می باشند. همانند برنامه هایی که در حوزه های نظامی، بهداشت و ایمنی، زیرساخت های حیاتی و غیره استفاده می شوند.

¹ Checklist

² Source code

سازمان ها ممکن است سطح سوم ASVS را برای برنامه هایی که عملکردی حیاتی برای سازمان انجام می دهند استفاده کنند، مخصوصا در جایی که شکست تاثیر مستقیم بر روی کارکرد سازمان یا حتی بقای سازمان بگذارد. مثال راهنما برای کاربرد سطح سوم ASVS در اینجا آمده است. یک نرم افزار به شرطی به سطح سوم ASVS میرسد که نه تنها به طور مناسب از خود در برابر خطرات و آسیب پذیری های پیشرفته برنامه های امروزی دفاع کند، بلکه بتواند قوانین طراحی امنیتی مناسب را رعایت کند و در خود بگنجانند.

یک برنامه در سطح سوم ASVS نیازمند عمیق ترین تجزیه و تحلیل، معماری، برنامه نویسی، و آزمایش می باشد که شامل تمام سطوح قبلی نیز می باشد. این برنامه امن به طور معقول ماژولار^۱ می شود (به منظور راحتی، به عنوان مثال انعطاف پذیری، مقیاس پذیری و مهم تر از همه لایه های امنیتی) و هرکدام از این ماژول ها (که خود آن یا نمونه از آن توسط ارتباط شبکه از هم جدا می شوند) مسئولیت های امنیتی خود را برعهده دارد، (مسئولیت دفاع در عمیق ترین لایه ها) که باید به خوبی مستند شود. این مسئولیت ها شامل کنترل هایی برای اطمینان از محرمانه بودن (مانند رمزنگاری^۲)، یکپارچگی (مانند معاملات و بررسی اعتبار ورودی^۳)، در دسترس بودن (مانند به خوبی مدیریت کردن بار^۴ سیستم)، احراز هویت (شامل احراز هویت بین چندین سیستم)، حسابرسی (همانند ثبت گزارش)، مجوز دهی و عدم تخلف می باشند.

پیاده سازی ASVS در عمل

پشت تهدیدات مختلف، انگیزه های مختلفی است. برخی از صنایع دارای اطلاعات و فناوری های منحصر به فرد و نیازمندی های منطبق با استاندارد های خاص میباشند.

در اینجا ما راهنمایی های مربوط به صنعت را در ارتباط با سطوح توصیه شده ASVS ارائه می دهیم. اگرچه برخی از معیار ها منحصر به فرد هستند اما تفاوت هایی در تهدید ها برای هر صنعت وجود دارد، موضوع مشترک در همه بخش های صنعت این است که مهاجمان فرصت طلب به دنبال برنامه هایی هستند که به راحتی آسیب پذیری آن ها را پیدا و بهره برداری کنند، و این دلیل وجود سطح اول ASVS می باشد که باید در تمامی برنامه ها استفاده شود. این پیشنهاد یک نقطه شروع به عنوان ساده ترین راه برای یافتن خطرات است. سازمان ها به شدت تشویق می شوند تا دیدگاهی عمیق بر ویژگی های ریسک پذیر منحصر به فرد خود مخصوصا بر اساس ماهیت کسب و کار داشته باشند. در انتها، قسمت اصلی سطح سوم ASVS است و از آن برای مواردی استفاده می شود که ممکن است به امنیت انسان ها آسیب برساند یا زمانی که نقض کامل نرم افزار به شدت بر سازمان تاثیر بگذارد.

¹ Modulized

² Encryption

³ Input validation

⁴ Load

صنعت	مشخصات تهدید	توصیه های سطح اول L1	توصیه های سطح دوم L2	توصیه های سطح سوم L3
بیمه و مالی	<p>اگرچه این بخش با تلاش مهاجمان فرصت طلب همراه است، اغلب به عنوان یک هدف ارزشمند توسط مهاجمان با انگیزه نیز مورد بررسی قرار میگیرد. چون این حملات اغلب با انگیزه مالی رخ میدهد. معمولاً مهاجمان به دنبال اطلاعات حساس یا حساب کاربری هستند که میتوانند کلاهبرداری و یا به طور مستقیم با روش انتقال پول از طریق برنامه های کاربردی سوء استفاده کنند.</p> <p>تکنیک ها اغلب شامل دزدیده شدن اعتبارنامه ها¹، حملات به برنامه های کاربردی و مهندسی اجتماعی برخی مطابق ملاحظات عمده شامل استانداردهای امنیت داده ها در صنعت پرداخت کارت (PCIDSS)، Gramm Leach Act و Sarbanes-Oxley Act (SOX) عمل میکنند.</p>	<p>تمامی برنامه هایی که به شبکه دسترسی دارند.</p>	<p>برنامه هایی که دارای اطلاعات حساسی همچون شماره کارت بانکی و اطلاعات شخصی می باشند که می تواند مقداری مشخصی پول را از مسیر مشخصی انتقال دهد.</p> <p>به طور مثال : (یک) : انتقال پول بین حساب های یک موسسه یا (دو) : یک نوع کند و محدود تر در انتقال پول به عنوان مثال (ACH) یا (سه) : انتقال از طریق سیم مثلاً در هارد دیسک ها که با محدودیت در یک بازه زمانی همراه است</p>	<p>برنامه هایی 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 به صورت یک تراکنش تنها یا به عنوان یک دسته از انتقال های کوچکتر</p>
تولید، تخصصی، حمل و نقل، تکنولوژی، خدمات رفاهی، زیرساخت و دفاع	<p>این صنایع ممکن است وجه های اشتراک زیادی نداشته باشند اما تهدیدکنندگانی که به احتمال بیشتر به این بخش از سازمان حمله میکنند، متمایل به اجرای حمله هایی با منابع، مهارت ها و زمان بیشتری می باشند. سیستم ها یا اطلاعات حساس را نمی توان به سادگی مکان یابی کرد و باید توجه داشت که نیاز به نفوذ داخلی و تکنیک های مهندسی اجتماعی دارند. حمله ها 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 نمی خواهیم بیش از حد به حمله کنندگان از نظر سوء استفاده عملکرد برنامه یا رفتار آن در سیستم های حساس، توجه کنیم.</p> <p>بیشتر حمله کنندگان به دنبال اطلاعات حساسی هستند که به صورت مستقیم یا غیر مستقیم از پرداخت اطلاعات و اطلاعات شخصی قابل تعریف، منفعت ببرند. اطلاعات اغلب برای شناسایی سارق، پرداخت های تقلبی، یا انواع تخلف می تواند مورد استفاده قرار گیرد.</p>	<p>تمامی برنامه هایی که به شبکه دسترسی دارند</p>	<p>برنامه هایی که شامل اطلاعات داخلی یا اطلاعاتی از کارمندی که ممکن است در مهندسی اجتماعی مورد نفوذ واقع شوند. برنامه هایی که شامل مالکیت فکری یا رمز های تجاری غیر ضروری اما مهم می باشند.</p>	<p>برنامه هایی که اطلاعات خیلی مهمی درون خود دارند، اسرار تجاری، اسرار دولتی (به طور مثال در ایالات متحده ممکن است هر چیز محرمانه ای به این گونه باشد یا حتی سطح محرمانه بالاتری داشته باشد) که برای بقا و موفقیت سازمان بسیار اهمیت دارد. برنامه هایی که کاربرد های خیلی حساس (همانند حمل و نقل، تجهیزات تولیدی، سیستم های کنترلی همانند موشک ردیاب) را کنترل میکنند یا سیستم هایی که باعث به خطر افتادن زندگی بشریت میشود.</p>

¹ Credentials

توصیه های سطح سوم L3	توصیه های سطح دوم L2	توصیه های سطح اول L1	مشخصات تهدید	صنعت
<p>برنامه های مورد استفاده برای کنترل ابزار یا تجهیزات پزشکی یا پرونده های پزشکی که ممکن است زندگی انسان ها را به خطر بیندازد. پرداخت و سیستم های POS (Point of Sale) که شامل مقادیر زیادی از تراکنش داده ای که بتواند مورد سوء استفاده و کلاه برداری قرار گیرد. این شامل هر رابط مدیریتی برای این برنامه ها می باشد.</p>	<p>برنامه هایی که در حوزه اطلاعات پزشکی حساسیت متوسط و یا حتی کم دارند همانند (اطلاعات محافظت شده ی بهداشت)، اطلاعات شخصی قابل شناسایی یا اطلاعات پرداختی</p>	<p>تمامی برنامه هایی که به شبکه دسترسی دارند</p>	<p>اکثر حمله کنندگان به دنبال اطلاعات و داده های حساسی هستند که میتوانند به طور مسقیم یا غیر مستقیم از آنها سود ببرند تا اطلاعات شخصی و اطلاعات مربوط به پرداخت را به دست آورند. اغلب از این داده ها به منظور شناسایی سارق/سارقین، پرداخت های تقلبی یا انواع مختلف عملیات تقلبی استفاده میشود.</p> <p>برای بخش مراقبت های پزشکی ایالات متحده، بیمه سلامت قابل انتقال، HIPAA، امنیت، قوانین نقص اطلاع رسانی و قانون امنیت بیمار :</p> <p>http://www.hhs.gov/ocr/privacy/</p>	<p>مراقبت های پزشکی</p>
<p>پرداخت و سیستم های POS (Point Of Sale) که شامل مقادیر زیادی از تراکنش داده ای که بتواند مورد سوء استفاده و کلاهبرداری قرار گیرد. این شامل هر رابط مدیریتی برای این برنامه می باشد. برنامه هایی با حجم زیادی از اطلاعات حساس مانند تمام اعداد کارت اعتباری، نام اختصاصی، اعداد امنیتی اجتماعی و غیره</p>	<p>مناسب برای برنامه های تجاری، اطلاعات کاتالوگ، اطلاعات شرکت های بزرگ داخلی و برنامه های با اطلاعات کاربری محدود(به عنوان مثال اطلاعات تماس). برنامه هایی با مقدار پرداختی داده یا عملیات توصیه حساب کم و متوسط</p>	<p>تمامی برنامه هایی که به شبکه دسترسی دارند</p>	<p>خیلی از مهاجمان در این بخش، تاکتیک های فرصت طلبانه smash & grab را به کار می برند. اگرچه یک سری تهدیدات مشخصی از حملات خاص روی برنامه های شناخته شده ای که شامل اطلاعات پرداخت، تراکنش های مالی انجام شده یا اطلاعات قابل شناسایی شخصی وجود دارد. هر چند نسبت به تهدید ها ذکر شده در بالا، کمتر مشابه هستند، امکان تهدید های به مراتب پیشرفته تر برای دزدیدن مالیت فکری، دستاورد رقابتی، هوش، یا مزیت بدست آمده با سازمان هدف یا مذاکرات همراه تجارت؛ که منجر به حمله به این بخش صنعت شود، وجود دارد.</p>	<p>خرده فروشی، غذا، مهمان نوازی</p>

مورد های مطالعاتی

مورد مطالعاتی ۱: به عنوان یک راهنمای تست امنیتی

در یک دانشگاه خصوصی در شهر یوتای ایالات متحده آمریکا، تیم دانشگاهی Red از OWASP ASVS به عنوان راهنما برای اجرای تست های نفوذ برنامه استفاده می کنند. از OWASP ASVS در تمام فرایند تست نفوذ، از برنامه ریزی اولیه و نشست های هدف گرفته، تا راهنمای فعالیت های تست کردن و روش چهارچوب بندی یافته های نهایی برای ارائه گزارش به مشتری استفاده شده است. همچنین، تیم Red آموزش هایی برای تیمی که از ASVS استفاده می کنند، سازمان دهی می کند.

تیم Red تست نفوذ شبکه و برنامه برای دپارتمان را به عنوان بخشی از استراتژی امنیت اطلاعات کلی دانشگاه انجام می دهد. در حین اولین جلسات برنامه ریزی، مشتریان اغلب اجازه تست برنامه را به تیم های دانشجویی نمی دهند. با معرفی ASVS به سهامداران و توضیح این موضوع که فعالیت های تست توسط این استاندارد هدایت می شوند، خیلی از نگرانی ها به سرعت بر طرف می شوند. بنابراین ASVS برای کمک به تعیین میزان زمان و تلاش لازم برای یک تست استفاده می شود. از طریق استفاده از سطوح بررسی از پیش تعریف شده ASVS، تیم Red تست مبتنی بر ریسک را توضیح می دهد. این به مشتری ها، سهامداران و خود تیم کمک می کند که جهت بررسی مناسب برای برنامه مورد نظر، به توافق برسند.

هنگامی که تست شروع شود، تیم Red از ASVS برای سازماندهی فعالیت ها و همچنین برای تقسیم حجم کاری استفاده می کند. مدیران پروژه ی تیم، از طریق دنبال کردن این که کدام نیازمندی های بررسی تست شده اند و کدام ها منتظر تست شدن هستند، به راحتی می توانند روند تست را مشاهده کنند. این کار باعث بهبود ارتباط مشتری ها شده و به مدیران پروژه قدرت مدیریت بهتر منابع را می دهد. به دلیل این که تیم Red عمدتاً از دانشجویان تشکیل شده، بیشتر اعضای تیم باید به طور هم زمان به چندین کار مربوط به درس هایشان نیز رسیدگی کنند. وظایف خوش تعریفی که براساس نیازمندی های بررسی شخصی یا تمامی دسته بندی ها می باشند، به اعضای تیم کمک می کند که بدانند برای تست به چه چیز هایی نیاز دارند و به آن ها اجازه می دهد که تخمین دقیقی برای فرایند تکمیل یک تست، ارائه کنند. همچنین فرآیند گزارش دهی از سازماندهی واضح ASVS بهره می گیرد، هنگامی که اعضای تیم می توانند قبل از حرکت به وظیفه بعدی، شرح یک یافته را بنویسند، به طور موثری حجم زیادی از گزارش را همزمان با تست نفوذ انجام می دهند.

تیم Red گزارش نهایی را حول ASVS سازماندهی کرده و وضعیت هر نیازمندی را بررسی و گزارش کرده و جزئیات دقیق متناسب را فراهم می کند. این گزارش به مشتری ها و سهامداران، یک ایده خوب از جایی که برنامه شان توسط استاندارد سنجیده شده، می دهد. همچنین این برای پیگیری های تعاملی بسیار با ارزش است به این علت که به آن ها اجازه می دهد ببینند امنیت چگونه به مرور زمان بهبود یافته یا از دست رفته است. بعلاوه، سهامدارانی که مشتاق چگونگی شکل گیری برنامه به فرم تک دسته یا چند دسته ای هستند، براحتی می توانند آن اطلاعات را به دلیل گزارش بسیار نزدیک و هم تراز ASVS، درک کنند. همچنین، سازماندهی روشن ASVS، آموزش اعضای جدید تیم را آسان تر کرده تا بتوانند به راحتی گزارش نویسی هایی انجام دهند که با فرمت گزارش های قبل مقایسه می شود.

نهایتاً آموزش تیم Red پس از اتخاذ ASVS بهبود پیدا کرده است. قبلاً، آموزش های هفته ای بر یک موضوع منتخب توسط رهبر تیم یا مدیر پروژه متمرکز شده بودند. این ها بر اساس اعضای تیم و نیازشان انتخاب شده اند. آموزش مبتنی بر این معیار، پتانسیل گسترده کردن مهارت های اعضای تیم را دارد، اما لزوماً به فعالیت های هسته تیم Red مرتبط نیست. به عبارت دیگر، در تست نفوذ به دستاورد مهم تری نرسیدند. پس از اتخاذ ASVS، آموزش تیم اکنون بر چگونگی نیازمندی های بررسی تست فردی متمرکز می شود. این موضوع منجر به یک پیشرفت بزرگ در توانمندی های قابل اندازه گیری اعضای تیم و کیفیت گزارش های نهایی شده است.

مورد مطالعاتی 2: به عنوان SDLC امن

یک استارت آپ، به دنبال حمایت از تحلیل های BigData برای موسسات مالی است که متوجه شده اند برای دستیابی به فرایند Metadata ی مالی در دست یافته هایشان، امنیت باید در راس اهمیت قرار گیرد. در این مثال، استارت آپ، استفاده از ASVS را به عنوان اساس چرخه زندگی پیشرفت امن متحرک شان انتخاب کرده است.

این استارت آپ از ASVS برای تولید epic ها و از مواردی برای مسائل امنیتی عملیاتی استفاده می کند به عنوان مثال چطور عملیات ورود به بهترین نحو پیاده سازی شود. این استارت آپ از ASVS به شیوه متفاوت تری استفاده می کند که در واقع در ASVS نظاره کرده و نیازمندی های ی که برای sprint کنونی مناسب تر است را بر می دارد. همچنین، در صورتی که یک نیازمندی عملیاتی باشد یا اگر به عنوان محدودیتی برای استفاده از موارد غیر عملیاتی تلقی شود، آن ها را در یک sprint backlog جمع می کند. به عنوان مثال، انتخاب مجموع TOTP با احراز هویت دو عامله، همراه با سیاست های رمز عبور و تنظیم کننده سرویس وبی که به عنوان تشخیص حمله جستجوی فراگیر¹ و مکانیزم ممنوعیت رخ میدهد. در sprint های آینده، نیازمندی های اضافی ای مبتنی بر اساس "فقط در لحظه"² و "شما به آن نیاز نخواهید داشت" انتخاب خواهند شد.

توسعه دهندگان از ASVS به عنوان یک چک لیست بازبینی استفاده می کنند که اطمینان حاصل می کند که کد ناامن در آنها بررسی نشده است، همچنین، برنامه های گذشته نگر برای بروز مشکل برای توسعه دهندگانی که یک ویژگی جدید را بررسی کرده اند، اطمینان حاصل می کند که آنها احتمالاً ASVS مورد نیاز را مورد بررسی قرار داده اند و اگر هر چیزی را بهبود بخشند، منجر به کاهش sprint در آینده شود.

در انتها، توسعه دهندگان از ASVS به عنوان بخشی از واحد امنیت بررسی خودکار و انواع تست یکپارچه، مصارف تستی، سوء استفاده و موارد تست fuzzing استفاده می کنند. هدف این است که ریسک روش آشنایی³ "تست نفوذ نهایی" که باعث بهبود پرهزینه هنگام به تولید رساندن نقاط عطف ساخته ها می شود. همانطور که ساخته های جدید می توانند بعد از هر sprint ای نمایش داده شوند، تکیه بر یک فعالیت به تنهایی کافی نمی باشد، و بنابراین با خودکار کردن سازمان تست شان، نباید مساله مهمی با وجود یک تست نفوذگر ماهر با هفته ها تجربه تست برنامه وجود داشته باشد.

¹ Brute force

² Just in time

³ Waterfall methodology

ارزیابی نرم افزار باعث دستیابی به سطح بررسی شده است

OWASP مبتنی بر بررسی های ASVS و علائم اعتماد¹

OWASP به عنوان یک فروشنده خنثی و غیر تجاری هیچ فروشنده، بررسی کننده و یا برنامه ای را بررسی نمی کند. تمامی این گواهی های ضمانت، علائم اعتماد یا گواهی ها به صورت رسمی توسط OWASP بررسی، ثبت و یا بررسی نمی شوند؛ 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 سازمان ها از پیشنهاد چنین سرویس های تضمینی نمی شود، از آن نظر که آنها ادعای بررسی OWASP رسمی را انجام نمی دهند.

راهنمایی برای سازمان های صادرکننده گواهی

استاندارد بررسی امنیتی برنامه می تواند به عنوان یک بررسی متن باز برای این برنامه استفاده شود که شامل دسترسی باز و بدون محدودیت به منابع اصلی و کلیدی مانند معماران و توسعه دهندگان، مستندات پروژه، کد منبع، دسترسی احراز هویت شده به سیستم های آزمایشی (شامل دسترسی به حداقل یک حساب کاربری در هر نقش)، مخصوصاً برای بررسی های L2 و L3 می باشد.

از لحاظ تاریخی، تست نفوذ و مرورهای کد امن شامل مسائل "با استثناء"² هستند که تنها مسائلی که با شکست رو به رو شده اند در گزارش آخر ظاهر می شوند. یک سازمان بررسی کننده باید شامل هر گزارشی از محدوده بررسی (مخصوصاً اگر یک جزء کلیدی خارج از محدوده باشد مانند احراز هویت SSO)، یک خلاصه ای از یافته های بررسی ها شامل تست های قبول شده و شکست خورده (با نشانه هایی واضح برای نشان دادن چگونگی بر طرف کردن تست های شکست خورده) باشد.

نگهداری برگه های کاری دقیق، تصاویر، فیلم ها و اسکرپت³ هایی که با اطمینان و مکرراً از یک مساله استفاده کند، سوابق الکترونیکی نحوه تست کردن (به عنوان مثال ردیابی پروکسی ها) و همچنین یادداشت های مرتبط (مثلاً یک لیست پاک کردن) به عنوان تمرین صنعت استاندارد در نظر گرفته می شوند. این که یک ابزار اجرا شود و فقط شکست ها را گزارش دهد، کافی نیست؛ این ها شواهدی کافی مبنی بر این که همه ی مسائل در یک سطح بررسی شده تست شده اند، فراهم نمی کند. در صورت اختلاف، باید شواهد حمایتی کافی وجود داشته باشد که نشان دهد هر الزام بررسی شده ای قطعاً تست شده است.

نقش ابزار های تست نفوذ اتوماتیک

ابزارهای تست نفوذ اتوماتیک به ارائه بیشترین پوشش دهی ممکن و عمل کردن بیشترین پارامترهای قابل امکان با متنوع ترین نوع ورودی مخرب، ترغیب شده اند.

کامل کردن بررسی ASVS، تنها با استفاده از تست نفوذ اتوماتیک ممکن نیست. در حالی که بیشتر نیازمندی های در L1 می تواند با استفاده از تست های اتوماتیک اعمال شود، مجموع نیازمندی های دیگر به تست نفوذ اتوماتیک متمایل هستند.

¹ Trust marks

² Exemption

³ Script

لطفا توجه شود که مرزهای بین تست های دستی و اتوماتیک با توجه به اهمیت صنعت امنیت برنامه از بین رفته است. ابزارهای اتوماتیک اغلب توسط متخصصین به صورت دستی تنظیم می شوند و تست کننده های دستی معمولا طیف وسیعی از ابزار های اتوماتیک را به کار می برند.

نقش تست نفوذ

فراهم کردن یک تست نفوذ دستی و بررسی تمامی مسائل L1 بدون نیاز به دسترسی به کد منبع ممکن است اما این منجر به عمل نمی شود. L2 نیازمند دسترسی به توسعه دهندگان، مستندات، متن کد و دسترسی احراز هویت شده به سیستم اصلی است. پوشش تستی نفوذ گسترده در L3 از آنجا که بیشتر مسائل افزوده درگیر بازبینی تنظیمات سیستم، بازبینی کد مخرب، مدل سازی تهدید، و دیگر دست یافته های تستی غیر قابل نفوذ می باشند، امکان پذیر نخواهد بود.

به عنوان راهنمای معماری امنیتی دقیق

یکی از رایج ترین کاربردهای استاندارد بررسی امنیت برنامه ها، استفاده به عنوان منبعی برای معماری های امنیتی است. دو چهارچوب بزرگ معماری امنیتی شامل SABSA و TOGAF فاقد اطلاعات کافی مورد نیاز برای تکمیل بررسی های معماری امنیتی برنامه ها هستند. ASVS می تواند برای پر کردن آن شکاف ها استفاده شود به این صورت که به معماران امنیتی اجازه انتخاب کنترل های بهتر برای مشکلات رایج مانند الگوهای محافظت از اطلاعات و راهبرد های بررسی ورودی را می دهد.

به عنوان جایگزینی برای چک لیست های از رده خارج شده کدگذاری امن

خیلی از سازمان ها از اتخاذ ASVS می توانند با انتخاب یکی از سه سطح، یا با چند شاخه کردن ASVS و تغییر چیزی که لازمه ی هر سطح ریسک برنامه در مسیر دامنه ی خاص می باشد، منفعت ببرند. از آنجایی که ردیابی نگه داشته می شود، از این نوع چند شاخه کردن استقبال می شود، بنابراین اگر یک برنامه نیازمندی 4.1 را بگذرانند، این برای کپی های خراب به معنی استاندارد تکامل یافته است.

به عنوان راهنمای تست های یکپارچه سازی و واحد اتوماتیک

ASVS طوری طراحی شده است که با وجود تنها استثنای نیازمندی های معماری و کد مخرب کاملا قابل تست باشد. با ساختن واحد و تست های یکپارچه ای که برای فاز مرتبط و خاص و موارد سوء استفاده تست می شوند، برنامه تقریبا با هر ساختاری به صورت خودکار بررسی می شود. به عنوان مثال، تست های اضافه برای مجموعه آزمون برای کنترل کننده ورودی، تست پارامتر نام کاربری برای نام های کاربری رایج، شمارش حساب های کاربری، جستجوی فراگیر، LDAP و تزریق SQL و XSS¹ می تواند طراحی شود. به طور مشابه، یک تست 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

به عنوان آموزش توسعه امن

ASVS همچنین می تواند برای تعریف مشخصه های نرم افزار امن استفاده شود. خیلی از دوره های "کد گذاری امن" در واقع دوره های هک اخلاقی با لکه ای از نکات کد گذاری هستند این هیچ کمکی به توسعه دهندگان نمی کند. دوره های توسعه امن می توانند از ASVS، با یک تمرکز قوی بر کنترل فعالانه مستند در ASVS به جای این که به عنوان مثال از 10 عدد از بدترین چیز هایی که نباید انجام شوند، استفاده کنند.

پروژه های OWASP با استفاده از ASVS

چهار چوب آگاهی امنیتی

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

آموزش توسعه دهندگان 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 می کند.

OWASP Zed Attack Proxy Project

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

پروکسی حمله زد OWASP (ZAP)، یک ابزار تست نفوذ یکپارچه ی ساده است که برای یافتن آسیب پذیری های برنامه های وب استفاده می شود. برای استفاده مردم با یک طیف وسیع از تجربه امنیتی طراحی شده است؛ بنابراین برای توسعه دهندگان و تست کنندگان عملیاتی که نسبت به تست نفوذ، تازه کار هستند کاملاً ایده آل است. ZAP، اسکنر های اتوماتیک و یک مجموعه از ابزار هایی که به شما اجازه یافتن آسیب پذیری امنیتی دستی را می دهد، فراهم می کند.

OWASP Cornucopia

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

OWASP بهم ریخته، مکانیزمی است به شکل کارت های بازی که برای کمک به تیم توسعه دهنده نرم افزار تشکیل شده و مشخص کننده نیازمندی های امنیتی در فرایند های توسعه رسمی، قراردادی و سریع العمل است و همچنین از نظر زبان، پایه و اساس و از نظر تکنولوژی به فرم اگنوستیک و غیر قابل اطمینان است. این طور بهم ریختگی بر اساس ساختار OWASP عملیاتی مبتنی بر کدگذاری امن، راهنمای مرجع سبک، ملاحظات دوچندان بخش های استاندارد برنامه بررسی استاندارد OWASP، راهنمای تست OWASP و اصول توسعه امن David Rook می باشد.

نیازمندی های بررسی دقیق

بررسی ۱. معماری، طراحی و مدل سازی تهدید

بررسی ۲. احراز هویت

بررسی ۳. مدیریت نشست

بررسی ۴. کنترل دسترسی

بررسی ۵. مدیریت و رسیدگی به ورودی مخرب

بررسی ۷. رمز نگاری 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. کنترل های مخرب

بررسی ۱۵. منطق تجارت

بررسی ۱۶. فایل و منابع

بررسی ۱۷. موبایل

بررسی ۱۸. سرویس های وب (جدید برای ۳.۰)

بررسی ۱۹. پیکربندی (جدید برای ۳.۰)

بررسی ۲۰. اینترنت اشیا (IoT)

بررسی ۱: معماری، طراحی و نیازمندی های بررسی مدل سازی تهدید

هدف کنترل

- اطمینان حاصل کنید که برنامه بررسی شده نیازمندی های با درجه بالای زیر را برآورده می کند :
- در سطح ۱، مولفه های^۱ 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 به کار رفته در جای مناسب قرار گرفته باشد و به درستی و به طور موثری استفاده شده باشد.

توجه: این بخش در نسخه 3.0 دوباره معرفی شده است ولی کنترل های معماری آن با نسخه 1.0 از ASVS یکی است.

نیازمندی ها

#	توضیحات	۱	۲	۳	از نسخه
۱.۱	بررسی کنید که تمامی مولفه های برنامه شناسایی شده و مورد نیاز می باشند.	✓	✓	✓	۱.۰
۱.۲	بررسی کنید که تمامی مولفه ها مانند کتابخانه ها، ماژول ها، سیستم های خارجی که جزئی از برنامه نیستند ولی برنامه مبتنی بر آن عمل می کند، شناسایی می شود.		✓	✓	۱.۰
۱.۳	بررسی کنید که معماری سطح بالا ^۲ برای برنامه تعریف می شود.		✓	✓	۱.۰
۱.۴	بررسی کنید که تمامی مولفه های برنامه براساس توابع تجاری و یا توابع امنیتی که ارائه می کنند، تعریف می شوند.			✓	۱.۰
۱.۵	بررسی کنید تمامی مولفه هایی که جزئی از برنامه نیستند ولی برنامه مبتنی بر آن عمل می کند براساس توابع و یا توابع امنیتی که ارائه می کنند، تعریف می شوند.			✓	۱.۰
۱.۶	بررسی کنید که مدل تهدیدی برای برنامه هدف تعریف و تمامی خطرات ناشی از جاسوسی ^۳ ، دستکاری ^۴ ، تخلف و دزدی ^۵ ، افشاء اطلاعات، عدم پذیرش سرویس ^۶ و بالا بردن سطح دسترسی ^۷ پوشش داده می شود.			✓	۱.۰
۱.۷	بررسی کنید که تمامی کنترل های امنیتی (شامل کتابخانه هایی که سرویس های امنیتی خارجی را فراخوانی می کنند) پیاده سازی متمرکز دارند.		✓	✓	۳.۰
۱.۸	بررسی کنید که تمامی مولفه هایی که توسط کنترل امنیتی تعریف شده، مانند تقسیم بندی شبکه ^۸ ، قوانین دیوار آتش و یا فضای ابری مبتنی بر گروه های امنیتی، از هم جدا می شود.		✓	✓	۳.۰

¹ Components

² High-level

³ Spoofing

⁴ Tampering

⁵ Repudiation

⁶ DoS

⁷ Elevation of privilege (STRIDE)

⁸ Network segmentation

#	توضیحات	۱	۲	۳	از نسخه
۱.۹	بررسی کنید که برنامه لایه های اطلاعات، نمایشی و کنترلی را از یکدیگر تفکیک نماید به طوری که بتوان تصمیم های امنیتی را در سیستم های مورد اعتماد اجرا کرد.	✓	✓	✓	۳.۰
۱.۱۰	بررسی کنید که منطق تجاری حساس، کلید های مخفی یا اطلاعات اختصاصی در کد سمت مشتری ^۱ ، وجود ندارد.	✓	✓	✓	۳.۰
۱.۱۱	بررسی کنید که تمامی اجزای برنامه، کتابخانه ها، ماژول ها، چهارچوب ها، پلتفرم ^۲ و سیستم های عامل عاری از آسیب پذیری می باشند.	✓	✓	✓	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

^۱ Client side

^۲ platform

بررسی ۲: نیازمندی های بررسی احراز هویت

هدف کنترل

- احراز هویت عمل بررسی و تایید یک فرد یا چیزی است که ادعای هویت صحیح می نماید. اطمینان حاصل نمایید که برنامه تایید شده موارد زیر را رعایت می نماید:
- هویت دیجیتالی فرستنده را بررسی کند.
 - از این اطمینان حاصل کند که فقط افراد مجاز توانایی احراز هویت داشته باشند و گواهینامه ها به طور امن منتقل می شوند.

نیازمندی ها

#	توضیحات	۱	۲	۳	از نسخه
۲.۱	بررسی کنید که تمامی صفحات و منابع به صورت پیش فرض ملزم به احراز هویت می باشند البته به جز آنهایی که به صورت عمومی در نظر گرفته شده اند(اصل وساطت کامل) ^۱ .	✓	✓	✓	۱.۰
۲.۲	بررسی کنید که برنامه به صورت اتوماتیک گواهینامه ها، فیلدهای مخفی ^۲ ، آرگومان های URL، درخواست های Ajax و لیست های ورودی را پر نمی کند؛ که این اشاره به متن ساده ^۳ ، کلمه عبور برگشت پذیر و قابل رمز گشایی دارد.	✓	✓	✓	۳.۱
۲.۴	بررسی کنید که تمامی کنترل های احراز هویت در سمت سرور ^۴ انجام می شود.	✓	✓	✓	۱.۰
۲.۶	بررسی کنید که تمامی کنترل های احراز هویتی به صورت امن با شکست مواجه می شوند. همچنین از این که مهاجم نمی تواند ورود پیدا کند، اطمینان حاصل می شود.	✓	✓	✓	۱.۰
۲.۷	بررسی کنید که فیلد های ورودی رمز عبور اجازه استفاده passphrase را داده و یا به آن تشویق می کند. همچنین از مدیران رمز عبور به علت استفاده از عبارت عبور طولانی و یا وارد کردن رمز عبور خیلی پیچیده، جلوگیری نمی شود.	✓	✓	✓	۳.۰.۱
۲.۸	بررسی کنید که تمامی توابع احراز هویت حساب کاربری(به عنوان مثال، بروزرسانی مشخصات، رمز عبور فراموش شده، توکن غیر فعال و یا گم شده، میزکار کمکی IVRL) که ممکن است مجدداً به حساب کاربری دسترسی پیدا کنند، حداقل به اندازه ی مکانیزم احراز هویت اصلی نسبت به حمله کنندگان، 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 ذخیره سازی شناسه های حساس نشست یا رمزهای عبور، بتوان ثبت کرد. این باید شامل درخواست هایی با metadata مرتبط که برای رسیدگی های امنیتی مورد نیاز است، باشد.	✓	✓		۳.۰.۱
۲.۱۳	بررسی کنید که رمز های عبور حساب کاربری، به صورت یک طرفه هش شده و عامل کاری کافی برای مقابله با جستجوی فراگیر، حمله های ریکاوری رمز عبور هش ^۱ شده، وجود دارد.	✓	✓		۳.۰.۱
۲.۱۶	بررسی کنید که گواهینامه ها از طریق یک لینک رمزنگاری شده مناسب منتقل می شوند و تمامی صفحات و توابعی که لازم است کاربر برایشان گواهینامه ها را وارد کند، از لینکی رمز نگاری شده باشد.	✓	✓	✓	۳.۰
۲.۱۷	بررسی کنید که عملیات رمز عبور فراموش شده و همچنین دیگر مسیر های بازگردانی رمز عبور، رمز عبور فعلی را نمایان نکنند، بعلاوه، رمز عبور جدید به صورت متن رمز شده برای کاربر فرستاده شود.	✓	✓	✓	۲.۰
۲.۱۸	بررسی کنید که جمع آوری اطلاعات از طریق ورود، بازنشانی رمز عبور و یا عملیات حساب کاربری فراموش شده ممکن نباشد.	✓	✓	✓	۲.۰
۲.۱۹	بررسی کنید که رمزهای عبور پیش فرض برای چهارچوب برنامه و هیچ یک از مولفه های مورد استفاده توسط برنامه وجود نداشته باشد (مثلا "admin/password")	✓	✓	✓	۲.۰
۲.۲۰	بررسی کنید که مکانیزم ضد اتوماسیون برای جلوگیری از تست گواهینامه نقض شده ^۲ ، جستجوی فراگیر ^۳ و حمله account lockout اعمال شده باشد.	✓	✓	✓	۳.۰.۱
۲.۲۱	بررسی کنید که تمامی گواهینامه ^۴ های احراز هویت برای دسترسی به سرویس های خارج از برنامه ، رمزنگاری و در مکانی محافظت شده ذخیره می شوند.	✓	✓		۲.۰
۲.۲۲	بررسی کنید که رمز عبور فراموش شده و مسیر های بازگردانی دیگر، از یک TOTP و یا soft token دیگر، mobile push یا دیگر مکانیزم های بازگردانی آفلاین استفاده می کنند. استفاده ی از یک مقدار تصادفی در پست الکترونیکی یا پیامک، باید به عنوان آخرین مراجعه در نظر گرفته شود و باید دانست که ضعیف تلقی می شود.	✓	✓	✓	۳.۰.۱
۲.۲۳	بررسی کنید که تحریم حساب کاربری به دو وضعیت قفل نرم و قفل سخت تقسیم می شود و این دو متقابلا منحصر به فرد نیستند. اگر یک حساب کاربری به علت حمله جستجوی فراگیر به صورت موقتی دچار تحریم قفل نرم شود، باعث بازنشانی وضعیت قفل سخت نمی شود.	✓	✓		۳.۰
۲.۲۴	بررسی کنید که اگر به سوالاتی بر پایه دانش مشترک، نیاز باشد(همچنین به عنوان "سوال های مخفی")، این سوالات قوانین حریم خصوصی را نقض نمی کنند و به اندازه کافی برای محافظت از حساب های کاربری در برابر بازگردانی مخرب، قوی هستند.	✓	✓	✓	۳.۰.۱
۲.۲۵	بررسی کنید که برنامه های با ارزش بالا به نحوی قابل تنظیم باشند که بتوان از استفاده از تعداد قابل تنظیمی از رمزهای عبور قبلی، جلوگیری کرد.	✓	✓		۳.۱

¹ Hash

² Breached credential testing

³ Brute

⁴ credential

#	توضیحات	۱	۲	۳	از نسخه
۲.۲۶	بررسی کنید که برای تراکنش های با ارزش بالا، احراز هویت مجدد مبتنی بر ریسک، امضای دو عامله یا تراکنش محور اعمال شود.		✓	✓	۳.۰.۱
۲.۲۷	بررسی کنید که اقداماتی برای جلوگیری کاربر از استفاده از رمز های عبور رایج و عبارات عبور ضعیف، اعمال شود.	✓	✓	✓	۳.۰
۲.۲۸	بررسی کنید که تمامی چالش های احراز هویت، خواه موفق، خواه مردود، باید پاسخی در زمان پاسخ دهی متوسط مشابه داشته باشند.			✓	۳.۰
۲.۲۹	بررسی کنید که رمز ها، کلید های API، رمز های عبور، در کد منبع و یا مخزن ^۱ های کد منبع آنلاین وجود نداشته باشند.			✓	۳.۰
۲.۳۱	بررسی کنید که کاربران بتوانند در موارد زیر ثبت نام و از آنها استفاده کنند: بررسی TOTP، مکانیزم احراز هویت دو عامله، مکانیزم احراز هویت بیومتریک (Touch ID) یا موارد مشابه)، مکانیزم احراز هویت چند عامله مشابه که در برابر افشاء گواهینامه تک عامله محافظت می کند.		✓	✓	۳.۰.۱
۲.۳۲	بررسی کنید که واسطه ^۲ های مدیریتی، در دسترس اشخاص غیر قابل اطمینان نباشند.	✓	✓	✓	۳.۰
۲.۳۳	بررسی کنید که اگر 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: نیازمندی های بررسی مدیریت نشست

هدف کنترل

یکی از اساسی ترین مولفه های هر برنامه مبتنی بر وب، مکانیزمی است که وضعیت کاربری که با آن تعامل دارد را کنترل و نگهداری می کند. این مهم، به مدیریت نشست ارجاع می دهد و به عنوان مجموعه ای از کنترل هایی که بر تعامل تمام وضعیتی بین کاربر و برنامه مبتنی بر وب حکم فرمایی می کنند، تعریف می شود.

از این که برنامه بررسی شده، نیازمندی های با درجه بالای زیر را برآورده کند اطمینان حاصل شود که:

- نشست ها نسبت به هر شخص، منحصر به فرد باشند و نتوان آنها را حدس زد و یا به اشتراک گذاشت.
- نشست ها هنگامی که دیگر به جلسات نیازی نباشد و یا مهلت زمان عدم فعالیت گذشته باشد، نامعتبر شوند.

نیازمندی ها

#	توضیحات	۱	۲	۳	از نسخه
۳.۱	بررسی کنید که مدیر نشست سفارشی ^۱ وجود نداشته باشد و یا این که مدیر نشست سفارشی در برابر حملات مدیریت نشست رایج، آسیب پذیر نباشد.	✓	✓	✓	۱.۰
۳.۲	بررسی کنید که نشست هنگامی که کاربر خارج شود، فاقد اعتبار باشد	✓	✓	✓	۱.۰
۳.۳	بررسی کنید که نشست پس از مدت زمان مشخصی از عدم فعالیت، مهلت شان 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 می شود. (an absolute timeout)	✓	✓		۱.۰
۳.۵	بررسی کنید که تمامی صفحاتی که نیاز به احراز هویت دارند، دارای دسترسی آسان و قابل رویت به عملیات خروج باشند.	✓	✓	✓	۱.۰
۳.۶	بررسی کنید که ID جلسات هرگز در URL ها، پیغام های خطا یا ثبت گزارشات آشکار نمی شوند. این شامل بررسی ای است که برنامه از دوباره نویسی کوکی های نشست، پشتیبانی نکند.	✓	✓	✓	۱.۰
۳.۷	بررسی کنید که تمامی احراز هویت های موفق و در خواست های احراز هویت مجدد، منجر به تولید نشست های جدید همراه با ID نشست می شود.	✓	✓	✓	۱.۰

¹ Custom

#	توضیحات	۱	۲	۳	از نسخه
۳.۱۰	بررسی کنید که ID های نشست هایی که توسط چهارچوب برنامه تولید می شوند، توسط برنامه به عنوان فعال شناخته می شوند.		✓	✓	۱.۰
۳.۱۱	بررسی کنید که ID های نشست به اندازه کافی طولانی، تصادفی و به صورت یکتا مبتنی بر نشست فعال صحیح می باشند	✓	✓	✓	۱.۰
۳.۱۲	بررسی کنید که ID های نشست که در کوکی ها ذخیره می شوند، دارای مسیری باشند که مقدار محدود کننده ای مناسب برای برنامه داشته باشند و همچنین توکن های نشست احراز هویت به مشخصه های "HttpOnly" و "secure" تنظیم شده باشند.	✓	✓	✓	۳.۰
۳.۱۶	بررسی کنید که برنامه های با ارزش بالا، تعداد جلسات همزمان فعال را محدود می کنند.			✓	۳.۰
۳.۱۷	بررسی کنید که لیست نشست فعال، در نمایش حساب کاربری یا مورد مشابهی از هر کاربر، نمایش داده شود. این کاربر باید قادر به خاتمه دادن هر نشست فعالی باشد.		✓	✓	۳.۰
۳.۱۸	بررسی کنید که برای برنامه های با ارزش بالا، امکان خاتمه دادن تمام جلسات فعال دیگر، البته بعد از فرایند تغییر رمز عبور موفق برای کاربر فراهم شود.			✓	۳.۰

منابع

برای اطلاعات بیشتر به منابع زیر مراجعه کنید:

- OWASP Testing Guide 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_Management_Cheat_Sheet

بررسی ۴: دسترسی به نیازمندی های بررسی کنترل

هدف کنترل

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

- اشخاص دارای دسترسی به منابع، برای انجام هر عملی، دارای گواهینامه های معتبر باشند.
- نقش ها و دسترسی های اختصاص داده شده به کاربران به درستی تعریف شده باشند.
- مجوز و نقش Metadata از دستکاری یا باز پخش محافظت شده باشد.

نیازمندی ها

#	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، فایل های اطلاعاتی، URL ها، کنترل کننده ها، سرویس ها و دیگر منابع	✓	✓	✓	۱.۰
۴.۴	بررسی کنید که دسترسی به گزارش های حساس به صورت محافظت شده باشد به طوری که تنها اشیاء یا اطلاعات مجاز، برای کاربر قابل دسترسی باشد (به عنوان 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 افشاء metadata ی فایل یا دایرکتوری مانند پوشه های .svn، .git، .DS_Store، Thumbs.db را بدهد.	✓	✓	✓	۱.۰
۴.۸	بررسی کنید که کنترل های دسترسی به صورت امن با شکست مواجه شوند.	✓	✓	✓	۱.۰
۴.۹	بررسی کنید که همان قوانین کنترلی دسترسی که توسط لایه ارائه، نشان داده می شوند در سمت سرور اعمال شوند.	✓	✓	✓	۱.۰
۴.۱۰	بررسی کنید که تمام صفات کاربران و داده ها و اطلاعات سیاسی که توسط کنترل های دسترسی استفاده می شوند، توسط کاربران نهایی دستکاری نمی شوند مگر آنهایی که به صورت خاص اجازه داشته باشند.	✓	✓		۱.۰
۴.۱۱	بررسی کنید که مکانیزمی متمرکز (شامل کتابخانه هایی که سرویس های مجاز خارجی را فراخوانی می کنند) برای محافظت از دسترسی به هر نوع منبع محافظت شده، 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

#	توضیحات	۱	۲	۳	از نسخه
۴.۱۳	بررسی کنید که برنامه یا چهارچوب، از توکن های ضد CSRF تصادفی استفاده کنند و مکانیزم محافظت از تراکنش دیگری نیز داشته باشند.	✓	✓	✓	۲.۰
۴.۱۴	بررسی کنید که سیستم بتواند از دسترسی کلی و یا پیوسته به اطلاعات، منابع یا توابع امن، محافظت کند. به عنوان مثال، استفاده از یک حاکم منابع را در نظر بگیرید که تعداد ویرایش در هر ساعت را محدود می کند یا از پاک کردن یک پایگاه داده کلی توسط یک کاربر، جلوگیری می کند.		✓	✓	۲.۰
۴.۱۵	بررسی کنید که برنامه دارای مجوز افزوده ای (به عنوان مثال مجوز سازگار شونده یا افزایشی) برای سیستم های با مقدار ضعیف و یا قدرت تفکیک پذیری وظایف برای برنامه هایی با مقادیر بالا به منظور اعمال کنترل های ضد تقلب برای هر ریسک برنامه و تقلب گذشته اش، باشد.		✓	✓	۳.۰
۴.۱۶	بررسی کنید که برنامه مجوز حساس به متن ^۱ را به درستی اعمال کند به طوری که اجازه دستکاری غیر مجاز به معنی 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](https://www.owasp.org/index.php/Testing_for_Authorization)
- OWASP Cheat Sheet: Access Control
[https://www.owasp.org/index.php/Access Control Cheat Sheet](https://www.owasp.org/index.php/Access_Control_Cheat_Sheet)

¹ Context sensitive authorisation

بررسی ۵: بررسی نیازمندی های مدیریت ورودی های مخرب

هدف کنترل

رایج ترین ضعف امنیتی برنامه وب این است که قبل از استفاده ورودی های مشتری یا محیط معتبر سازی نمی شود. این ضعف تقریباً منجر به آسیب پذیری های عمده در برنامه های وب می شود که میتوان نمونه هایی مانند cross site scripting، تزریق SQL، تزریق مفسر^۱، حملات محلی / Unicode، حملات به File System و سرریز بافر^۲ را نام برد.

باید مطمئن شد که یک برنامه بررسی شده نیازمندی های با درجه بالای زیر را برآورده می کند:

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

نیازمندی ها

#	توضیحات	۱	۲	۳	از نسخه
۵.۱	بررسی کنید که محیط زمان اجرا مستعد به سرریز بافر نیست یا اینکه کنترل های امنیتی باعث جلوگیری از سرریز بافر می شوند.	✓	✓	✓	۱.۰
۵.۳	بررسی کنید که اشتباهات اعتبار سنجی ورودی های سمت سرویس دهنده منجر به رد درخواست شده و ثبت می گردند.	✓	✓	✓	۱.۰
۵.۵	بررسی کنید که روال ها معتبرسازی ورودی ها در سمت سرویس دهنده اعمال می شوند.	✓	✓	✓	۱.۰
۵.۶	بررسی کنید که یک کنترل معتبر ورودی در برنامه، برای هر نوع داده پذیرفته شده استفاده می شود.	✓			۱.۰
۵.۱۰	بررسی کنید که همه Query های SQL، OSQL، HQL، NOSQL و روشهای ذخیره شده و فراخوانی فرآیندهای ذخیره شده توسط دستورات آماده شده و یا پارامترسازی Query محافظت می شود و بنابراین مستعد به حمله تزریق SQL نیست.	✓	✓	✓	۲.۰
۵.۱۱	بررسی کنید که برنامه حساس به تزریق LDAP نیست، یا اینکه کنترل های امنیتی از LDAP Injection جلوگیری می کنند.	✓	✓	✓	۲.۰
۵.۱۲	بررسی کنید که برنامه حساس به تزریق دستورات سیستم عامل نیست یا اینکه کنترل های امنیتی از تزریق دستورات سیستم عامل جلوگیری می کنند.	✓	✓	✓	۲.۰

^۱ Interpreter injection

^۲ 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 رشته ای که در HTML یا دیگر کدهای مشتریان وب قرار داده می شوند یا به صورت دستی رمزگذاری شده‌اند یا از قالب‌هایی استفاده می کنند که به طور خودکار رمزگذاری می کنند 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 اختصاص دهی خودکار چند پارامتر (انقیاد خودکار متغیرها) از درخواست ورودی به یک مدل را فراهم می نماید، همچنین مطمئن شوید که متغیرهای حساس امنیتی مثل accountBalance، role و password از انقیادهای خودکار مخرب ^۱ محافظت شده‌اند.	✓	✓		۳.۰
۵.۱۷	بررسی کنید که برنامه دارای دفاع از حملات پارامترهای HTTP است، به خصوص اگر چارچوب برنامه هیچ تمایزی در مورد منبع پارامترهای درخواستی ندارد. (Post، Get، کوکی، سرآیند ^۲ ، محیط و غیره)	✓	✓		۲.۰
۵.۱۸	بررسی کنید که اعتبارسنجی سمت مشتری به عنوان خط دوم دفاع به همراه اعتبارسنجی سمت سرور 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، کوئری‌های پارامترها، سرآیندهای HTTP، کوکی‌ها، فایل‌های Batch، خوراکی‌های RSS و غیره. این اعتبارسنجی را میتوان توسط اعتبارسنجی مثبت (لیست سفید)، و اعتبار سنجی میانی فهرست خاکستری (حذف رشته های شناخته شده مخرب)، یا رد ورودی های مخرب (لیست سیاه) انجام داد.	✓	✓		۳.۰
۵.۲۱	بررسی کنید که داده های بدون ساختار، برای اجرای اقدامات ایمنی عمومی مانند کاراکترهای مجاز، و چیزهایی که به طور بالقوه در زمینه داده شده مضر هستند، باید از آنها محافظت شود (مثلا نامهای طبیعی با Unicode یا آپوستروف، مثل نام های ね یا O'Hara).	✓	✓		۳.۰
۵.۲۲	بررسی کنید که فایل های HTML غیر مطمئن از ادیتورهای WYSIWYG و ادیتورهای مشابه، به صورت درست توسط یک فایل HTML پاکسازی کننده، پاکسازی شده‌اند. همچنین این فایل‌ها باید بر اساس اعتبارسنجی ورودی و رمزنگاری به صورت درست به کار گرفته شوند.	✓	✓	✓	۳.۰
۵.۲۳	بررسی کنید که برنامه در چه جاهایی از auto-escaping استفاده می کند و در چه جاهایی auto-escaping غیر فعال است. خروجی باید به صورت دستی رمز گذاری 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 به جهت تجزیه JSON برای مشتری استفاده می‌شود. از eval() برای تجزیه JSON برای مشتری استفاده نکنید.	✓	✓		۳.۰
۵.۲۶	بررسی کنید که داده‌های 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://docs.angularjs.org/api/ng/service/$sce)
<https://cwe.mitre.org/data/definitions/915.html>

بررسی ۶: بررسی نیازمندی های کدگذاری و بی اثر سازی خروجی

این بخش در بررسی 5 استاندارد بررسی امنیت اپلیکیشن 2.0 آورده شده است. الزامات ASVS 5.16 به رمزگذاری خروجی متنی به جهت جلوگیری از Cross Site Scripting اشاره دارد.

بررسی ۷: رمزنگاری در بررسی نیازمندی های راکد

هدف کنترل

اطمینان حاصل کنید که یک برنامه تایید شده شرایط زیر را به بهترین نحو برآورده می کند:

- تمام ماژول های رمزنگاری به نحوی امن از کار می افتند و خطاها به درستی مدیریت شده اند.
- به هنگام نیاز به Randomness، یک تولید کننده مناسب عدد تصادفی استفاده می شود.
- دسترسی به کلیدها به صورت امن مدیریت می شود.

نیازمندی ها

#	توضیحات	۱	۲	۳	از نسخه
۷.۲	بررسی کنید که تمام ماژول های رمزنگاری به نحوی ایمن از کار می افتند و خطاها به گونه ای مدیریت می شوند که Padding Oracle فعال نمی گردد.	✓	✓	✓	۱.۰
۷.۶	بررسی کنید که تمامی اعداد تصادفی، اسامی تصادفی فایل ها، GUID های تصادفی و رشته های تصادفی توسط تولید کننده عدد تصادفی مورد تایید ماژول رمزنگاری شده تولید می شوند، به گونه ای که این مقادیر تصادفی توسط حمله کننده قابل حدس زدن نباشد.		✓	✓	۱.۰
۷.۷	بررسی کنید که الگوریتم های رمزنگاری استفاده شده علیه FIPS 140-2 یا یک استاندارد معادل اعتبار سنجی شده باشند.	✓	✓	✓	۱.۰
۷.۸	بررسی کنید که ماژول های رمزنگاری در حالت تایید شده خود با توجه به سیاست های امنیتی منتشر شده خود عمل می کنند.			✓	۱.۰
۷.۹	بررسی کنید که یک روش صریح برای چگونگی مدیریت کلیدهای رمزنگاری (به عنوان مثال تولید، توزیع، لغو و منقضی شده) وجود دارد. بررسی کنید که این چرخه حیات کلید به درستی اجرا بشود.		✓	✓	۱.۰
۷.۱۱	بررسی کنید که تمام مصرف کنندگان 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

#	توضیحات	۱	۲	۳	از نسخه
۷.۱۳	بررسی کنید که پسورها 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
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 زمانی که 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 نیازمندی های 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 گزارش هایی با کیفیت بالا می باشد که بیش از Noise های نامربوط دارای سیگنال باشند.

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

- عدم جمع آوری یا ثبت اطلاعات حساس اگر به طور خاص مورد نیاز نباشند.
 - اطمینان از مدیریت امن و محافظت شده داده های ثبت شده بر اساس اهمیت و طبقه بندی آن ها.
 - 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 حساس ترین اطلاعات برنامه تبدیل می شوند که برای حمله کنندگان در جایگاه خود بسیار جذاب خواهند بود.

نیازمندی ها

#	توضیحات	۱	۲	۳	از نسخه
۸.۱	بررسی کنید که برنامه پیام های خطا یا stack trace های حاوی اطلاعات حساسی که بتواند به مهاجم کمک کند را نمایش ندهد، از جمله شناسه نشست، نسخه نرم افزار یا framework و اطلاعات شخصی.	✓	✓	✓	۱.۰
۸.۲	بررسی کنید که منطق کنترل خطا در کنترل های امنیتی به طور پیش فرض دسترسی را رد می کند.		✓	✓	۱.۰
۸.۳	بررسی کنید که کنترل های ثبت کردن، از قابلیت ثبت موفق برخوردار باشد، به خصوص 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 به عنوان کد در نرم افزار در نظر گرفته شده برای مشاهده گزارش سیستم اجرا نخواهند شد.		✓	✓	۱.۰
۸.۶	بررسی کنید که گزارش ها از دسترسی و تغییر غیر مجاز محافظت می شوند.		✓	✓	۱.۰
۸.۷	بررسی کنید که برنامه، داده های حساس را آنگونه که در سیاست های قوانین محلی تعریف شده است ثبت نکند، مانند داده های حساسی که با ریسک همراه هستند و یا داده های حساس احراز هویت که می توانند به حمله کننده کمک کنند و در شناسه های نشست کاربر آمده اند، همچنین رمزهای عبور، Hash ها و API Token ها.		✓	✓	۳.۰
۸.۸	بررسی کنید که تمام نشانه های غیر قابل چاپ و جدا کننده های زمینه به درستی در رویداد به منظور جلوگیری از تزریق گزارش رمزگذاری شده اند.			✓	۲.۰

#	توضیحات	۱	۲	۳	از نسخه
۸.۹	بررسی کنید که گزارش از منابع قابل اعتماد و غیر قابل اعتماد در ورودی قابل تشخیص است.	✓	✓	✓	۲.۰
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 هماهنگ سازی 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 نیازمندی های محافظت داده

هدف کنترل

سه عنصر کلیدی برای محافظت از داده ها وجود دارد: محرمانگی، اصالت و دسترسی پذیری (CIA). این استاندارد بر این اساس است که حفاظت از داده ها بر روی یک سیستم قابل اعتماد که تقویت شده، مانند یک سرور اجرا شده و دارای حمایت های کافی است.

برنامه های کاربردی باید فرض کنند که تمام دستگاه های کاربر به نوعی به خطر افتاده اند. هنگامی که نرم افزاری در حال انتقال یا نگهداری اطلاعات حساس در دستگاه های نامن مانند رایانه ها، تلفن ها و تبلت های مشترک است، این نرم افزار مسئول اطمینان از رمزگذاری داده های ذخیره شده در این دستگاه ها است و اینکه نمی توان به راحتی به آن ها دسترسی پیدا کرد، یا آن ها را تغییر داد.

اطمینان حاصل کنید که یک برنامه تایید شده نیازمندی ها سطح بالای زیر را برآورده می کند:
محرمانگی: داده ها باید از مشاهدات غیر مجاز و یا افشا در هنگام ذخیره سازی و انتقال محافظت شود.
اصالت: داده ها باید از اینکه توسط مهاجمین به صورت مخرب ایجاد شده، تغییر کرده و یا حذف شوند، محافظت گردند.
دسترسی پذیری: در صورت نیاز، داده ها باید برای کاربران مجاز در دسترس باشند.

نیازمندی ها

#	توضیحات	۱	۲	۳	از نسخه
۹.۱	بررسی کنید که همه فرم هایی که حاوی اطلاعات حساس هستند، caching را در سمت مشتری غیرفعال کرده اند، از جمله قابلیت های 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 داده ها باید چگونه کنترل شده، رمزگذاری شده و چگونه اصول حفاظت داده روی آن ها اعمال شود.		✓	✓	۱.۰
۹.۳	بررسی کنید که همه اطلاعات حساس در قسمت پیام یا متن HTTP به سرور ارسال می شوند. (به عنوان مثال، پارامترهای URL هرگز برای ارسال اطلاعات حساس استفاده نمی شوند)	✓	✓	✓	۱.۰
۹.۴	بررسی کنید که برنامه سرآیندهای کافی ضد cache را تعبیه نماید، به طوری که هر گونه اطلاعات حساس و شخصی نمایش داده شده توسط برنامه یا وارد شده توسط کاربر، نباید توسط مرورگرهای امروزی روی هارد cache شود.	✓	✓	✓	۳.۰.۱
۹.۵	بررسی کنید که در سرور، تمام نسخه های ذخیره شده یا موقت داده های حساس از دسترسی غیرمجاز محافظت شده باشد، یا پس از اینکه کاربر مجاز به آن ها دسترسی پیدا کرد، این داده ها 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، کوکی ها 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 ذخیره سازی محلی HTML5، ذخیره سازی نشست، IndexedDB، کوکی رهای معمولی یا کوکی های 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، حتما ثبت بشود.			✓	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 نیازمندی های امنیت ارتباطات

هدف کنترل

اطمینان حاصل کنید که یک برنامه بررسی شده نیازمندی های سطح بالای زیر را برآورده می کند:

- اینکه TLS هنگام انتقال داده های حساس مورد استفاده قرار گیرد.
- اینکه الگوریتم ها و رمزگذاری های قوی همواره مورد استفاده قرار گیرند.

نیازمندی ها

#	توضیحات	۱	۲	۳	از نسخه
۱۰.۱	بررسی کنید که یک مسیر همواره می تواند از یک CA مورد اعتماد به هر یک از سرورهای گواهی دهنده TLS طراحی کرد و به طوریکه گواهی هر سرور معتبر باشد.	✓	✓	✓	۱.۰
۱۰.۳	بررسی کنید که TLS برای تمامی اتصالات (شامل اتصالات خارجی و backend) که اعتبارسنجی شده یا شامل داده های حساس یا توابع هستند، استفاده می شود و دارای پروتکل های ناامن و غیر رمز شده نیست. اطمینان حاصل کنید که محتمل ترین حالت الگوریتم استفاده گردد.	✓	✓	✓	۳.۰
۱۰.۴	بررسی کنید که خطاهای اتصال TLS backend ثبت شده اند.	✓			۱.۰
۱۰.۵	بررسی کنید که مسیرهای گواهی ساخته شده، و برای گواهینامه های همه مشتری ها با استفاده از مکانیزم لنگرهای اعتماد و برگردانی اطلاعات تایید شده اند.	✓			۱.۰
۱۰.۶	بررسی کنید که تمامی اتصالات به سیستم های خارجی که شامل اطلاعات حساس می باشد، احراز هویت می شوند.	✓	✓		۱.۰
۱۰.۸	بررسی کنید که یک پیاده سازی از استاندارد TLS وجود داشته و توسط برنامه، در حالت عملکردی بررسی شده به کار گرفته می شود.	✓			۱.۰
۱۰.۱۰	بررسی کنید که برچسب زنی عمومی کلید گواهی TLS (HPKP) برای تولید و پشتیبانی از کلیدهای عمومی پیاده سازی شده است. به جهت کسب اطلاعات بیشتر به منابع زیر مراجعه کنید.	✓	✓		۳.۰.۱
۱۰.۱۱	بررسی کنید که سرآیندهای HTTP Strict Transport Security، برای همه زیردامنه ها و برای همه درخواست ها مانند Strict-Transport-Security: max-age=15724800; includeSubdomains شامل می شود.	✓	✓	✓	۳.۰
۱۰.۱۲	بررسی کنید که URL وبسایت تولید کننده که نسبت به آن ثبت نام صورت گرفته است در لیست دامنه های Strict Transport Security که توسط فروشندگان	✓			۳.۰

#	توضیحات	۱	۲	۳	از نسخه
	مرورگر پشتیبانی می‌شود، از پیش بارگذاری شده باشد. لطفا منابع زیر را مشاهده نمایید.				
۱۰.۱۳	بررسی کنید که پنهان‌سازی کامل ارسالات به جهت مقابله با حمله‌کنندگانی که ترافیک ما 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 تایید 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، به خصوص که در یک تنظیمات عمومی، رمزگذاری‌های عمومی و الگوریتم‌ها نیز ناامن تر 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 تایید TLS". در گذشته، ASVS به استاندارد FIPS 140-2 آمریکا گفته می‌شد، اما به عنوان یک استاندارد جهانی، با استفاده از استانداردهای آمریکا، این امر می‌تواند دشوار، متناقض یا گیج کننده باشد. یک روش بهتر برای رسیدن به انطباق با 10.8، بررسی راهنمایی‌هایی مانند https://wiki.mozilla.org/Security/Server_Side_TLS می‌باشد، همچنین تولید تنظیمات شناخته شده مانند <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 به دست آوردن سطح مطلوب امنیت است. به طور 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 لطفاً 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 نیازمندی های تنظیمات امنیتی HTTP

هدف کنترل

- اطمینان حاصل کنید که یک برنامه بررسی شده نیازمندی های سطح بالای زیر را برآورده می کند:
- سرور برنامه به طور مناسب از تنظیمات پیش فرض ایجاد شده است.
 - پاسخ های HTTP حاوی یک کاراکتر ایمن در محتوا سرآیند است.

نیازمندی ها

#	توضیحات	۱	۲	۳	از نسخه
۱۱.۱	بررسی کنید که برنامه فقط یک مجموعه متشکل از روش های درخواست HTTP مورد نیاز مانند GET و POST را قبول می کند و روش های استفاده نشده (مانند PUT, TRACE و DELETE) به صورت 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. (به عنوان مثال، ISO 8859-1, UTF-8)	✓	✓	✓	۱.۰
۱۱.۳	بررسی کنید که سرآیندهای HTTP توسط یک پروکسی مطمئن یا دستگاه های 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 سایت هایی که محتوای آن نباید در یک X-Frame شخص ثالث مشاهده شود، استفاده گردد.	✓	✓		۳.۰.۱
۱۱.۵	بررسی کنید که سرآیند HTTP یا هر بخشی از پاسخ HTTP اطلاعات جزئی مربوط به نسخه اجزاء سیستم را نشان نمی دهند.	✓	✓	✓	۲.۰
۱۱.۶	بررسی کنید که تمام پاسخ های API شامل X-Content-Type-Options: nosniff و Content-Disposition: attachment; filename="api.json" باشند. (یا سایر اسم فایل ها و نوع محتواهای مناسب).	✓	✓	✓	۳.۰
۱۱.۷	بررسی کنید که یک سیاست امنیت محتوا (CSPV2) وجود دارد که آسیب پذیری های معمول XSS، DOM و جاوا اسکریپت را کاهش می دهد.	✓	✓	✓	۳.۰.۱
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 تست OWASP 4.0: 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 سوء تفاهم در نوع MIME بین کاربر و سرور جلوگیری کند. همچنین گزینه "نام فایل" به طور خاص کمک می کند 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&setlang=en
- OWASP User Privacy Protection Cheat Sheet
https://www.owasp.org/index.php/User_Privacy_Protection_Cheat_Sheet
- امکان استفاده از <https://securityheaders.io> به 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 نیازمندی های پیکر بندی امنیتی

این بخش به بررسی ۱۱ در استاندارد بررسی امنیت برنامه ۲.۰ منتقل شد.

بررسی ۱۳: بررسی نیازمندی های کنترل های مخرب

هدف کنترل

- اطمینان حاصل کنید که یک برنامه بررسی شده نیازمندی های سطح بالای زیر را برآورده می کند:
- فعالیت مخرب به صورت ایمن و درست مدیریت شود تا بقیه برنامه را تحت تاثیر قرار ندهد.
- بمب های زمان دار یا سایر حمله های زمان دار داخل آن ها وجود نداشته باشد.
- به مقاصد مخرب و یا مجوز داده نشده "phone home" نکنید.
- برنامه های کاربردی back door, easter eggs, حمله های salami و یا خطاهای منطقی که توسط حمله کننده قابل کنترل است، نداشته باشد.

کد مخرب بسیار نادر است و شناسایی آن نیز مشکل است. بررسی خط به خط کد می تواند به جستجوی بمب های منطقی کمک کند، اما حتی خبره ترین اشخاص که کدها را بررسی می کنند، نیز حتی اگر بدانند که چنین کدهایی وجود دارد، به سختی می توانند کدهای مخرب را پیدا نمایند. این بخش بدون دسترسی به کد منبع امکان پذیر نیست، از جمله همه کتابخانه های شخص ثالث.

نیازمندی ها

#	توضیحات	۱	۲	۳	از نسخه
۱۳.۱	بررسی کنید که تمام فعالیت های مخرب به طور مناسب جعبه شنی ^۱ 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 کتابخانه های شخص ثالث، دارای back door, easter egg و خطاهای منطقی در احراز هویت، کنترل دسترسی، تایید ورودی و منطق تجاری انتقال های 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 نیازمندی های امنیت داخلی

این بخش به بررسی ۱۳ در استاندارد بررسی امنیت برنامه ۲.۰ منتقل شد.

بررسی ۱۵: بررسی نیازمندی های منطق کسب و کار

هدف کنترل

اطمینان حاصل کنید که یک برنامه بررسی شده نیازمندی های سطح بالای زیر را برآورده می کند:

- جریان منطق کسب و کار، پیوسته و به ترتیب است.
- منطق کسب و کار شامل محدودیت هایی 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 کسب و کار با ارزش، موارد سوء استفاده و عناصر فعال مخرب را مورد توجه قرار داده و نیز در برابر جریانهایی مانند repudiation, tampering, spoofing, information disclosure و حملات privilege مقاوم باشد.

نیازمندی ها

#	توضیحات	۱	۲	۳	از نسخه
۱۵.۱	بررسی کنید که برنامه تنها پردازش منطق کسب و کار را به صورت مراحل و گام-های متوالی پردازش می کند، و تمامی گام ها در زمان واقعی انسانی پردازش شوند نه خارج از نوبت و با پرش از روی گام ها و پردازش گام های یک کاربر دیگر یا تراکنش هایی که خیلی سریع بررسی شده باشند.		✓	✓	۲.۰
۱۵.۲	بررسی کنید که برنامه دارای محدودیت های تجاری است و به درستی بر اساس هر کاربر اجرا می شود، همچنین دارای سیستم هشداردهی قابل تنظیم و خودکار باشد که واکنش های خودکار در برابر حملات غیر معمول 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 نیازمندی های فایلها و منابع

هدف کنترل

- اطمینان حاصل کنید که یک برنامه بررسی شده نیازمندی های سطح بالای زیر را برآورده می کند:
- داده های فایل غیر قابل اطمینان باید بر اساس نوع به طریق امن مدیریت شوند.
 - دریافتی ها از منابع غیر مطمئن در بیرون از شاخه اصلی برنامه و با دسترسی های محدود ذخیره شود.

نیازمندی ها

#	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 هشدار را نشان دهد.	✓	✓	✓	۲.۰
۱۶.۲	بررسی کنید که داده های فایل غیر قابل اطمینان داده شده به برنامه مستقیماً با دستورات I/O فایل مورد استفاده قرار نگیرند، به خصوص برای محافظت در برابر آسیب پذیری های path traversal، local file include، file MIME type، reflective file download و تزریق دستورات سیستم عامل.	✓	✓	✓	۳.۰.۱
۱۶.۳	بررسی کنید که فایل های به دست آمده از منابع نامعتبر به صورت در حال انتظار برای تایید باشند. و توسط اسکنرهای آنتی ویروس اسکن شده اند تا از آپلود محتوای مخرب شناخته شده جلوگیری شود.	✓	✓	✓	۲.۰
۱۶.۴	بررسی کنید که داده های غیرمعمد در داخل class loader، inclusion و یا قابلیت های 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 داده های نامطمئن در داخل به اشتراک گذاری منابع cross-domain (CORS) استفاده نشوند تا از حملات arbitrary remote content جلوگیری شود.	✓	✓	✓	۲.۰
۱۶.۶	بررسی کنید که فایل های دریافت شده از منابع نامطمئن خارج از webroot ذخیره شده و دسترسی های محدودی داشته باشند و ترجیحاً با بررسی های 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 داده های 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 Applet های سمت مشتری استفاده نشوند.	✓	✓	✓	۳.۰.۱

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

بررسی ۱۷: نیازمندی های 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 در نظر گرفته شود . برنامه های 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 در نظر گرفتن لایه انتقال امن، انجام شود.

نیازمندی ها

#	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 پوشه های 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 ^۱ ، توکن های 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 ^۲ بنویسد.	✓	✓		۲.۰

^۱ Secret keys

^۲ 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: نیازمندی های بررسی سرویس های وب

هدف کنترل

- مطمئن شوید که برنامه های بررسی شده ای که از سرویس های مبتنی بر وب مانند SOAP یا RESTful استفاده می کنند 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 پایین تر به سطح اعتماد بالاتر منتقل می شوند، باشد.
 - قابلیت همکاری اولیه با لایه سرویس وب SOAP برای بالا بردن استفاده از API، باشد.

نیازمندی ها

#	توضیحات	۱	۲	۳	از نسخه
۱۸.۱	بررسی کنید که روش کدگذاری یکسان در سمت سرویس دهنده و سرویس گیرنده استفاده 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 می شود .	✓	✓	✓	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 کنید که 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 ^۱ مبنی بر نشست را بررسی کنید. لطفاً برای راهنمایی بیشتر به بخش های 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 از حداقل 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 مکانیزم های امضای JSON و امنیت WS برای درخواست های SOAP استفاده می کند.	✓	✓	✓	۳.۰.۱
۱۸.۱۰	بررسی کنید که موارد دیگر و راه های 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](https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)_Prevention_Cheat_Sheet)
- JSON Web Tokens (and Signing)
<https://jwt.io/>

بررسی ۱۹: نیازمندی های بررسی پیکربندی

هدف کنترل

- اطمینان حاصل کنید که برنامه بررسی شده دارای :
- کتابخانه ها و بسته های به روز باشد.
 - پیکربندی امن به صورت پیش فرض باشد.
 - مستحکم سازی کافی باشد؛ به طوریکه تغییرات اعمال شده نسبت به پیکربندی پیش فرض توسط کاربر ، سیستم های اساسی را در معرض ایجاد اشکالات یا ضعف های امنیتی بی مورد قرار ندهد.

نیازمندی های

#	توضیحات	۱	۲	۳	از نسخه
۱۹.۱	بررسی کنید که همه مؤلفه ها با پیکربندی ها و نسخه هایی با امنیت مناسب به روز می شوند. این بروز رسانی باید شامل حذف پیکربندی ها و پوشه های غیر ضروری از جمله : برنامه های نمونه، بسته های مستندسازی و کاربران پیش فرض 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. به خصوص هنگامی که مؤلفه ها در مخازن ^۱ و یا سیستم های 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 می باشد که از حساب با حداقل مجوز ^۲ مورد نیاز استفاده می کند.	✓	✓	✓	۳.۰
۱۹.۴	بررسی کنید که نصب همه برنامه ها در جعبه شنی ^۳ قرار گرفته ، نگه داری یا ایزوله شده است تا حمله مهاجمان به سایر برنامه ها را به تأخیر بیندازد و یا مانع آن ها شود.	✓	✓	✓	۳.۰
۱۹.۵	بررسی کنید که فرآیند تولید و نصب برنامه ها به روش قابل تکرار و امن صورت می گیرد مانند اتوماسیون CI/CD و مدیریت پیکربندی خودکار شده.	✓	✓	✓	۳.۰

^۱ Repository

^۲ Privilege

^۳ Sand box

#	توضیحات	۱	۲	۳	از نسخه
۱۹.۶	بررسی کنید که مدیرهای دارای مجوز، قابلیت بررسی بی عیبی و درستی امنیت همه ی پیکربندی های مربوطه برای شناسایی دستکاری را دارند.			✓	۳.۰
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 ^۱ 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 می شوند ؛ به عنوان مثال کتابخانه های javascript، CSS stylesheets و فونت های وب بجای اینکه به یک CDN یا عامل خارجی وابسته باشند توسط برنامه میزبانی می شوند.			✓	۳.۰.۱

منابع

برای اطلاعات بیشتر به منابع زیر مراجعه کنید:

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

¹ Security flags

پیوست الف : چه اتفاقی برای نیازمندی ها زیر افتاده است ...؟

# اصلی	توضیحات	وضعیت	حذف شده	علت
۲.۳	بررسی کنید که اگر از ماکسیمم تعداد دفعات تلاش برای احراز هویت گذشته باشد، حساب برای دوره ای با مدت زمان کافی قفل شود تا از حملات جستجوی فراگیر ممانعت شود.	از رده خارج شده	۲.۰	نیازمندی پیچیده تری جایگزین شده است. (نیازمندی ۲.۲۰)
۲.۵	بررسی کنید که تمامی کنترل های احراز هویت (شامل کتابخانه هایی که سرویس های خارجی احراز هویت را فراخوانی می کنند). یک پیاده سازی مرکزی دارند.	ادغام شده	۳.۰	برای شامل شدن همه کنترل های امنیت بصورت عمومی در نیازمندی ۱.۱۰ بررسی می شود.
۲.۱۰	بررسی کنید که قبل از اینکه هرگونه عملیات حساس مخصوص برنامه ها مجوز داشته باشد، احراز هویت دوباره نیاز دارد.	از رده خارج شده	۲.۰	احراز هویت دوباره به ندرت مشاهده می شد که تصمیم گرفتیم این کنترل را حذف کنیم
۲.۱۱	بررسی کنید که بعد از یک دوره زمانی قابل تنظیم اجرایی، اعتبارنامه های احراز هویت باطل می شوند.	از رده خارج شده	۲.۰	تعلیق های مطلق و انقضاء اعتبارنامه به دلیل موثر نبودن کنترل حذف شدند.
۲.۱۴	بررسی کنید که همه اعتبارنامه های احراز هویت برای دسترسی به سرویس های خارج از برنامه کدگذاری می شوند و در محل محافظت شده (نه در کد منبع) ذخیره می شوند.	به روز رسانی شده	۲.۰	به نیازمندی ۲.۲۱ ارتقا یافته است
۲.۱۵	بررسی کنید که همه ی کد هایی که از کنترل احراز هویت ها استفاده یا پیاده سازی می کنند از هیچ کد مخربی تاثیر نمی یابند.	جایجا شده	۲.۰	به بررسی ۱۳ کد مخرب جایجا شده است
۲.۳۰	بررسی کنید که آیا برنامه اجازه احراز هویت به کاربران می دهد. برنامه ها از یک مکانیزم احراز هویت امن اثبات شده استفاده می کند.	از رده خارج شده	۳.۰.۱	برای آزمایش شدن بسیار مبهم بود ، در واقع خلاصه ای از همه نیازمندی های بررسی ۲ می باشد.
۳.۸	بررسی کنید که شناسه نشست در طول احراز هویت دوباره تغییر داده می شود.	بروز رسانی شده	۳.۰	به نیازمندی ۳.۷ گردش یافته است
۳.۹	بررسی کنید که شناسه نشست به هنگام خروج از سیستم تغییر می یابد یا پاک می شود.	بروز رسانی شده	۳.۰	به نیازمندی ۳.۷ گردش یافته است
۳.۱۳	بررسی کنید که همه ی پیاده سازی های کدی یا کنترل هایی که برای مدیریت	جایجا شده	۲.۰	به بررسی ۱۳ کد مخرب

# اصلی	توضیحات	وضعیت	حذف شده	علت
	نشست استفاده می‌شود از هیچ کد مخربی تأثیر نمی‌یابد.			جابجا شده است.
۳.۱۴	بررسی کنید که توکن نشست های احراز هویت شده که از کوکی ها استفاده می‌کنند با استفاده از HttpOnly محافظت می‌شوند.	بروز رسانی شده	۳.۰	به نیازمندی ۳.۱۳ جابجا شده است.
۳.۱۵	بررسی کنید که توکن نشست های احراز هویت شده که از کوکی ها استفاده می‌کنند با فعالسازی ویژگی secure محافظت می‌شوند.	به روز رسانی 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 مجوز های خاصی دارند.	به روز رسانی 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 مجوز های خاصی دارند.	به روز رسانی شده	۳.۰	به نیازمندی ۴.۱ گردش یافته است
۴.۱۳	بررسی کنید که محدودیت‌هایی که برای ورودی ها و دسترسی های تحمیل شده تجاری روی برنامه وجود دارند به نحوی قابل عبور نباشند. (مانند معامله های روزانه محدود یا توالی های امور مهم)	جابجا شده	۳.۰	به بررسی ۱۵ منطق تجارت جابجا شده است
۴.۱۵	بررسی کنید که همه ی کد هایی که کنترل دسترسی ها را استفاده یا پیاده سازی 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 ورودی ها اعمال می‌گردند.	از رده خارج شده	۲.۰	به دلیل اینکه پیاده سازی به خصوص به صورت رایگان ورودی های 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 پیاده سازی بسیار مشکل اکثر زبان ها حذف شده است
۵.۷	بررسی کنید که همه ی ورودی های با اعتبار ناموفق ثبت می‌شوند.	از رده خارج شده	۳.۰	به دلیل ایجاد گزارش های بهبوده که مورد چشم پوشی واقع می‌شد حذف شده است.
۵.۸	بررسی کنید که همه داده های ورودی مقدم بر اعتبار سنجی برای کدگشاهای پایین دست یا مفسر ها کانالیزه می‌شوند.	از رده خارج شده	۳.۰	از نوع ۱ تکنولوژی تَشپ حذف شده است و برای چارچوب های مدرن مشکلی نخواهد داشت.
۵.۹	بررسی کنید که همه کنترل های اعتبار سنجی های ورودی ها توسط هیچ کد مخربی تأثیر نمی‌یابند.	جابجا شده	۲.۰	به بررسی ۱۳ کنترل بدافزار ها جابجا شده است .
۵.۱۴	بررسی کنید که محیط زمان اجرا به XML injections آسیب پذیر و حساس	ادغام شده	۳.۰	با نیازمندی ۵.۱۳ ادغام شده است

# اصلی	توضیحات	وضعیت	حذف شده	علت
	نمی باشد و یا کنترل های امنیت از XML injections جلوگیری می کنند.			
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 می شود، کنترل امنیت برای برای خروجی در مقصد مورد نظر وجود دارد.	ادغام شده	۳.۰	برای شامل شدن همه کنترل های امنیت بصورت عمومی در نیازمندی ۱.۱۰، بررسی شده است.
۷.۱	بررسی کنید که همه توابع رمز نگاری شده که برای حفاظت رمز ها از کاربر استفاده می شود سمت سرور پیاده سازی می شوند.	از رده خارج شده	۳.۰	بسیاری از پاسخگویی های مدرن و برنامه های موبایل در طراحی خود این مورد را رعایت کرده اند.
۷.۳	بررسی کنید هرگونه شاه رازی در برابر دسترسی غیر مجاز محافظت می شود. (یک شاه راز یک اعتبار نامه برنامه است که به عنوان متن اصلی روی دیسک ذخیره می شود که برای حفاظت از دسترسی به اطلاعات پیکربندی امنیت مورد استفاده قرار می گیرد).	جابجا شده	۳.۰	به نیازمندی ۲.۲۹ جابجا شده است.
۷.۴	بررسی کنید که چکیده ساز های رمز عبور به هنگام ایجاد شدن، نمک زده می شوند.	جابجا شده	۲.۰	به نیازمندی ۲.۱۳ جابجا شده است.
۷.۵	بررسی کنید که ماژول های ناموفق رمز نگاری شده ثبت می شوند.	از رده خارج شده	۲.۰	ایجاد گزارش های غیر ضروری که هرگز بررسی نمی شوند ضد تولید محسوب می شوند.
۷.۱۰	بررسی کنید که همه کد هایی که از از ماژول رمزنگاری شده پشتیبانی یا استفاده می کنند از کد های مخرب تاثیر نمی یابند.	جابجا شده	۲.۰	به بررسی ۱۳ جابجا شده است.
۸.۲	بررسی کنید مدیریت همه خطاها روی دستگاهی معتمد انجام می شود.		۳.۰	از رده خارج شده
۸.۳	بررسی کنید که همه کنترل رویدادنگاری ها روی سرور پیاده سازی می شوند.	جابجا شده	۳.۰	به بررسی ۱۳ کنترل بدافزار ها جابجا شده است.
۸.۹	بررسی کنید که پیاده سازی رویدادنگاری که در سطح برنامه انجام می شود توسط نرم افزار استفاده می شود.	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 تحلیل گر می دهد که رویداد های ثبت شده را جستجو کند. این جستجو مبنی بر ترکیبات معیار های جستجو در همه زمینه ها در فرمت ضبط رویداد ها که توسط سیستم پشتیبانی می شود می باشد.	از رده خارج شده	۳.۰	به دلیل عدم نیاز برای نرم افزار امن حذف شده است.
۸.۱۲	بررسی کنید که همه ی کد هایی که مدیریت خطا و کنترل گزارش را استفاده یا پیاده سازی می کنند از هیچ کد مخربی تاثیر نمی یابند.	جابجا شده	۲.۰	برای شامل شدن همه کنترل های امنیت بصورت

# اصلی	توضیحات	وضعیت	حذف شده	علت
	بررسی کنید که رویدادنگاری قبل از اجرای تراکنش انجام می شود. اگر رویداد نگاری ناموفق بود (به عنوان مثال به دلیل تکمیل بودن ظرفیت دیسک یا مجوز های ناکافی) برنامه بی خطر متوقف می شود. این برای زمانی است که درستی و عدم تخلف واجب است.	از رده خارج شده	۳.۰	عمومی در نیازمندی ۱.۱۳ بررسی شده است. به دلیل ریزبینانه شدن کنترل که تنها در درصد پائینی از برنامه ها کاربرد دارد، حذف شده است.
۱۰.۲	بررسی کنید اتصالات TLS ناموفق به اتصالات HTTP ناامن تنزل نمی کند.	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 می باشد از یک حساب که به داشتن حداقل مجوز ضروری برای برنامه تنظیم می شود استفاده کند تا به درستی عمل کند.			
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 می دهد.	تکراری شده	۳.۰	نیازمندی تکراری . در نیازمندی ۱۶ به آن پرداخته

# اصلی	توضیحات	وضعیت	حذف شده	علت
	بررسی کنید که پارامترهای بدست آمده از منابع نا معتبر پیش از کانالیزه شدن و اعتبار سنجی ورودی ها در دستکاری نام فایل ها ، نام مسیر ها یا هر گونه شی سیستم فایل ها مورد استفاده قرار نمی گیرند تا از حمله های گنجاندن فایل محلی جلوگیری شود.	جایجا شده	۳.۰	به نیازمندی ۱۶.۲ جایجا شده است.
۱۶.۴				
۱۷.۱	بررسی کنید که مشتری گواهی های SSL را اعتبار سنجی می کند	از رده خارج شده	۳.۰	نیازمندی تکراری قبلا 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 : واژه نامه

- **کنترل دسترسی:** مکانیزمی 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 مولفه 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 مولفه های یک برنامه ، میان مشتری و سرور و میان سیستم های خارجی و برنامه تبادل می شود.
- **مولفه :** واحد کاملی از کد ، به همراه دیسک مربوط و واسط های شبکه که با سایر مولفه ها ارتباط برقرار می کند.
- **Cross-Site Scripting (XSS):** آسیب پذیری امنیتی که معمولاً در برنامه 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 آسیب 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**: استاندارد می تواند به عنوان پایه برای بررسی و طراحی و پیاده سازی مژول 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.
- **زبان HyperText 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 و کانالیزه 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 با سایر 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** : تجربیدی از طراحی یک برنامه است که کجا و چگونه 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** : پیکربندی زمان اجرای یک برنامه که روی چگونگی استفاده شدن کنترل های امنیت تاثیر می گذارد.
- **کنترل امنیت** : یک تابع یا مولفه 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 در صورتی 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 آسیب پذیری استفاده می کند.
- **هدف بررسی** : اگر شما در حال انجام بررسی برنامه طبق نیازمندی های 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 عملاً 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 نیازمندی های 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 پروژه های 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
- 10 مورد برتر اوسپ
https://www.owasp.org/index.php/Top_10_2013-Top_10
- 10 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 معمول MITRE
<http://cwe.mitre.org/>
- شورای استاندارد های امنیت PCI
<https://www.pcisecuritystandards.org>
- نیازمندی های نسخه 3 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 10 مورد برتر اوسپ در سال های 2007/2004 به همراه تعمیم های فرآیند های اخیر نتیجه می شود. ASVS مجموعه ای اکید از 10 مورد برتر سال 2013 اوسپ(154 مورد تا 1 مورد). بنابراین تمام موضوعاتی که توسط 10 مورد برتر اوسپ و PCI DSS 6.5.x پوشش داده می شوند توسط کنترل نیازمندی های مناسب تر ASVS مدیریت می شوند. به عنوان مثال "احراز هویت نقض شده و مدیریت نشست "دقیقا در بررسی 2 احراز هویت و بررسی 3 مدیریت نشست در نقشه قرار می گیرند. اگرچه سطح 2 بررسی اکثر نیازمندی های PCI DSS 6.5 بجز 6.5.3 و 6.5.4 را آدرس دهی می کند، نقشه برداری کامل توسط سطح 3 بررسی بدست آمده است. موضوعات پردازی از جمله PCI DSS 6.5.6 توسط ASVS پوشش داده نمی شود.

توضیحات	ASVS 3.0	PCI-DSS 3.0
نقشه برداری دقیق	۵.۱۱،۵.۱۲،۵.۱۳،۸.۱۴،۱۶.۲	6.5.1 اشکالات تزریق بخصوص تزریق SQL. همچنین تزریق دستور سیستم عامل ، LDAP و اشکالات تزریق XPath را هم علاوه بر سایر اشکالات تزریق در نظر بگیرید.
نقشه برداری دقیق	۵.۱	6.5.2 بافر سرریز می شود
نقشه برداری جامع از سطح 1 به بعد	تمامی بررسی ۷	6.5.3 فضای رمزنگاری شده ناامن
نقشه برداری جامع از سطح 1 به بعد	تمامی بررسی ۱۰	6.5.4 ارتباطات نا امن
نقشه برداری دقیق	۳.۶،۷.۲،۸.۱،۸.۲	مدیریت نامناسب خطا ها
ASVS ، XSS را با برجسته سازی پیچیدگی حفاظت XSS به خصوص برای برنامه هایی که از آن استفاده می کنند به تعدادی نیازمندی های تجزیه می کند.	۵.۱۶،۵.۲۰،۵.۲۱،۵.۲۴،۵.۲۵،۵.۲۶،۵.۲۷،۱۱.۴،۱۱.۱۵	6.5.5 نبشته سایت تقلبی
نقشه برداری جامع از سطح 1 به بعد	تمامی بررسی ۴	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).

توضیحات	ASVS 3.0	PCI-DSS 3.0
نقشه برداری دقیق. ASVS حفاظت از CSRF را به عنوان جنبه کنترل دسترسی در نظر میگیرد.	4.13	6.5.9 جعل درخواست سایت تقلبی (CSRF)
نقشه برداری جامع از سطح 1 به بعد	تمامی وراسی 2 و بررسی 3	6.5.10 احراز هویت نقض شده و مدیریت نشست