# 10 شیوه رایج حمله و راهکارهای جلوگیری در بستر وب

در این مقاله 10 شیوه رایج حمله و راهکارهای جلوگیری در بستر وب به تفصیل آورده شده است.

این دسته‌بندی‌ها بر اساس دسته‌بندی انجمن OWASP گردآوری شده و سعی شده است با مثال‌هایی ساده، شیوه‌های حمله مشخص و همچنین تفکیک وظایف بین تولیدکننده نرم‌افزار و استفاده‌کننده از نرم‌افزارهای تحت وب شفاف شود. این مقاله سعی دارد به صورت ساده، دانش عمومی‌خوانندگان این مقاله را درباره شیوه‌های حمله و جلوگیری از آن در بستر وب توضیح دهد. اگر شما برنامه‌نویس تحت وب یا تولیدکننده نرم‌افزارهای وبی یا مدیر امنیت سازمان و یا کاربر یک نرم‌افزار وبی هستید، این مقاله می‌تواند برای دانش عمومی‌شما درباره تهدیدات موجود در بستر وب، کمک‌کننده باشد.

شیوه‌های معروف حملات به شرح زیر دسته‌بندی شده است:

- تزریق
- شکسته شدن احراز هویت و مدیریت نشست
- تزریق اسکریپت از طریق سایت (XSS)
- ارجاع مستقیم به اشیاء به صورت ناامن

- پیکربندی اشتباه امنیت
- افشای داده‌های حساس
- از دست رفتن کنترل دسترسی‌ها در سطح توابع
- درخواست‌های تقلبی از طریق سایت (CSRF)
- استفاده از مولفه‌های جانبی با آسیب پذیری‌های شناخته شده
- تغییر مسیرها و ارجاعات نامعتبر

## تزریق

تزریق یا Injection شیوه‌ای است که حمله‌کننده سعی دارد بر اساس مولفه‌های مختلف یک نرم‌افزار وبی، دستورات برنامه‌نویسی را در بخش‌های مختلف سرور وب یا سرور دیتابیس اجرا کند. به صورت خیلی ساده، حمله‌کننده سعی می‌کند از طریق درگاه‌های مختلف ارسال اطلاعات، دستورات اجرایی برای پلتفرم‌های مختلف را ارسال و آن‌ها را روی سرور اجرا کند. نکته مهم درباره این نوع حملات، این است که هدف‌گذاری آنها، اجرای کد بر روی سرورها است و تاثیری بر روی کاربران ندارد.

این نوع حمله به دو بخش تقسیم می‌شود:

- مسیرهای ارسال دستورات اجرایی
- اجرای دستورات در پلتفرم هدف

در رابطه با مسیرهای ارسال دستورات اجرایی به صورت کلی 3 شیوه برای ارسال

دستورات اجرایی به سرور وجود دارد:

- ارسال از طریق متغیرهای تعریف شده در آدرس URL وب سایت (Query Stringها)
- ارسال از طریق عناصر HTMLای تولیدکننده یک فرم مانند جعبه متن‌ها و لیست‌های کشویی و ... که در نرم‌افزارهای تحت وب به‌منظور دریافت اطلاعات از کاربران برای انجام مقاصد خاص مانند ایجاد، ویرایش و یا دریافت اطلاعات، استفاده می‌شوند.
- ارسال از طریق فایل‌هایی که نرم‌افزار آن‌ها را پردازش می‌کند. فایل‌هایی مانند XML که ساختاری محتوایی دارند و برای انتقال اطلاعات به سرور هستند و نرم‌افزار آن‌ها را به منظور دستیابی به اطلاعات پردازش می‌کند.

در رابطه با پلتفرم‌های مختلف اجرای دستورات به صورت کلی این عناصر در سمت سرور مورد هدف هستند:

- سرور وب به منظور انجام عملیات تخریبی بر روی سرور یا به دست‌آوری کنترل سرور و یا اجرا کردن یک ویروس بر روی سرور.
- سرور پایگاه داده به منظور اجرا کردن کدهای پایگاه داده‌ای برای به دست آوردن اطلاعات موجود در آن به صورتی که کنترل‌های امنیتی نرم‌افزار، داده‌های دریافتی را فیلتر نکند.

انجمن OWASP این دسته‌بندی‌ها را با هم‌ترکیب و بر اساس هر کدام از پلتفرم‌ها حملات را نام گذاری کرده که معروف‌ترین آن‌ها به شرح زیر است:

- SQL Injection: حمله‌هایی که سعی در تزریق کدهای مخصوص پایگاه داده دارد.
- Command Injection: حمله‌هایی که سعی در تزریق کدهای مخصوص سیستم عامل برای اجرا بر روی سیستم عامل سرور وب یا سرور پایگاه داده دارد.
- XML Injection: حمله‌هایی که کد در ساختاری معتبر از استاندارد XML نوشته می‌شود و با بارگزاری فایل XML به سرور و پردازش آن در سرور، کدهای تزریق شده، اجرا می‌شوند.
- LDAP Injection: حمله‌هایی که سعی در تزریق کدهای مخصوص پروتکل LDAP بر روی مخزن نگهداری اطلاعات LDAP دارد و می‌خواهد این کدها با دسترسی‌هایی که نرم‌افزار بر روی‌این مخزن دارد، اجرا شود.

تمامی حملات تزریق، سعی در دور زدن ساختارهای امنیتی نرم‌افزار و دسترسی مستقیم به زیرساختی دارد که نرم‌افزار بر روی آن استقرار داده شده است تا بتواند با اختیارات معمولا نامحدودی که خود نرم‌افزار در زیرساخت محل استقرار دارد، اطلاعات مورد نیاز خود و یا کنترل سرورها را به دست آورد.

این نوع حمله را ممکن است هم کاربران شناسایی نشده و هم کاربران شناسایی شده که احراز هویت شده‌اند، انجام دهند. نرم‌افزار نباید برای این نوع حملات به هیچ کاربری با هر سطحی از دسترسی اعتماد کند.

برای تمامی این نوع حملات، قابلیت‌هایی بر روی زبان‌های برنامه‌نویسی و پروتکل‌های تعریف شده بین بخش‌های مختلف وجود دارد تا بتوان از حملات جلوگیری کرد و تولیدکنندگان نرم‌افزار باید از این قابلیت‌ها به صورت کامل استفاده کنند و کسانی که به

عنوان کارشناسان مسئول و ارشد در این شرکت‌ها کار می‌کنند حتما در زمان بازنگری کد در چک لیست ، کدها را از نظر استفاده قابلیت‌ها و عدم وجود فضای نفوذ بررسی کنند. همچنین باید در فاز تست نرم‌افزار تمامی ورودی‌های بیان شده دراین نوع حمله، با هدف مقاومت در برابر درخواست‌های تزریق کد، تست شود تا میزان مقاومت نرم‌افزار دراین نوع حملات بررسی شود.

## شکسته شدن احراز هویت و مدیریت نشست

احراز هویت و مدیریت نشست دو عنصر اصلی در بستر وب هستند. هر کاربری برای استفاده از نرم‌افزار ابتدا باید احراز هویت کند. همچنین بر اساس ماهیت ناهمگام میان کاربر و سرور وب، نیاز است که برای کاربر استفاده‌کننده از نرم‌افزار، نشستی ایجاد شود تا به ازای هر درخواست کاربر، سرور بتواند متوجه شود کدام کاربر درخواست را ارسال کرده است.

حمله‌کنندگان سعی می‌کنند توابع نوشته شده برای احراز هویت را با شیوه‌های مختلف بشکنند. آن‌ها سعی می‌کنند از شیوه‌های مختلف، اطلاعات اشخاص معتبر استفاده‌کننده از نرم‌افزار را به دست آورند. به عنوان مثال آن‌ها از ضعف در کلمات عبور تعریف شده توسط کاربران استفاده می‌کنند. نرم‌افزاری می‌نویسند که برای یک نام کاربری، مجموعه بسیار زیادی ازترکیب‌های مختلف کلمات عبور عامیانه میان کاربران را به سرور ارسال کند تا یکی از آن‌ها توسط سرور درست تشخیص داده شود و بر اساس آن، مشخصات کاربری کاربر معتبر دزدیده شود. شاید باورش سخت باشد اما در حال حاضر نیز هنوز هزاران نفر از کلمه عبور 12345 برای حساب کاربری خود در نرم‌افزارهای وبی

استفاده می‌کنند و این خبری خوب برای هکرهاست. این تنها یک مثال خیلی ساده از این نوع حملات به حساب می‌آید که هدف کلی آنها به دست آوردن مشخصات حساب کاربری کاربران برای ورود به نرم‌افزار است.

نرم‌افزارها باید بتوانند توابع احراز هویت خود را در مقابل انواع این حملات قوی‌تر کنند. سیاست‌گذاری بر روی پیچیدگی کلمات عبور، احراز هویت دو عاملی با استفاده از امضای دیجیتال یا یک بار رمزها (OTP)، استفاده از تصاویر امنیتی برای تشخیص اختلاف بین ماشین و انسان، ساختار های قفل کردن حساب کاربری در زمان‌هایی که کلمه عبور به تعداد خاصی اشتباه وارد می‌شود، استفاده از سوالات امنیتی و ... از جمله نیازمندی‌هایی است که نرم‌افزارهای تحت وب امن باید از آن‌ها پشتیبانی کنند.

در مدیریت نشست، حمله‌کننده به دنبال مشخصات نشست کاربر معتبری است که با نرم‌افزار در حال کار کردن است. این حمله‌کننده می‌تواند از افراد ناشناس و یا کاربران مجاز سیستم باشد که به دنبال اهدافی مانند ارسال درخواست‌های مختلف به سرور و یا انجام کاری به صورت مخفیانه هستند. در دنیای وب، نشست ایجاد شده بین کاربر و سرور از طریق شناسه نشست (SessionID) مدیریت می‌شود. این شناسه نشست توسط سرور ایجاد و برای کاربر بازگردانده می‌شود و مرورگر کاربر به ازای هر درخواست ارسالی به سرور، این شناسه نشست را ارسال می‌کند.

بر اساس تنظیمات نرم‌افزار، این شناسه یا در بخش کوکی‌های مرورگر ذخیره می‌شود و یا به عنوان متغیر در آدرس URL ارسال می‌شود. حمله‌کنندگان به دنبال این هستند تا این شناسه را بدست آورند تا بتوانند خود را به سرور کاربری مجاز تعریف کنند. آن‌ها

برای به دست آوردن این شناسه از روش‌های مختلفی استفاده می‌کنند. برخی شروع به تولید تصادفی شناسه‌ها می‌کنند و با ارسال درخواست به سرور درمی‌یابند که آیا نشستی با این شناسه وجود دارد یا خیر. برخی به دنبال الگوریتم خاصی در تولید شناسه نشست هستند تا بتوانند شناسه‌هایی تولید کنند که از نظر سرور معتبر است. برخی دیگر به دنبال داده‌های حساس درون شناسه نشست هستند. به عنوان مثال اگر نرم‌افزار شناسه‌ها را بر اساس شناسه حساب کاربری کاربر تولید کند و در اختیار مرورگر کاربر قرار دهد با شناسایی‌این شناسه هر درخواستی از هر شخص شناس یا ناشناسی برای سرور معتبر و مربوط به کاربر، مجاز شناخته می‌شود؛ به این ترتیب نشست‌ها شکسته می‌شوند.

نرم‌افزارها باید توابع مدیریت نشست خود را با عناصر مختلفی تقویت کند. به عنوان مثال الگوریتم تولید شناسه باید الگوریتمی‌کاملا تصادفی باشد، شناسه نشست قبل و بعد از احراز هویت یک کاربر باید تغییر کند، اجازه مدیریت تعداد نشست‌های فعال به کاربر یا مدیر امنیتی نرم‌افزار باید در سیستم وجود داشته باشد و کاربر بتواند در هر زمانی نشست خود را خاتمه دهد. نشست‌های ایجادشده باید در یک بازه زمانی معتبر باشند و در صورتیکه کاربر فعالیتی انجام نداد آن نشست توسط نرم‌افزار خاتمه داده شود.
نرم‌افزار باید تنظیماتی را انجام دهد که شناسه نشست بر روی آدرس URL ارسال نشود و در کوکی‌ها نیز توسط کدهای جاوا اسکریپت در دسترس نباشد. همچنین نباید هیچ داده حساسی درون شناسه وجود داشته باشد مگراینکه با الگوریتمی‌رمزنگاری شود که خود رمزنگاری قابل شکستن نباشد واین داده حساس حتما با داده‌هایی که به صورت تصادفی تولید شده‌اند ترکیب شوند.

وظیفه تولیدکنندگان نرم‌افزارها این است که‌این مجموعه قابلیت‌ها را در نرم‌افزار ایجاد

کنند و مدیریت آن را در اختیار مدیران امنیت یا کاربران استفاده‌کننده نرم‌افزارها قرار دهند.

## تزریق اسکریپت از طریق سایت (XSS)

ایده این شیوه حمله بسیار شبیه به حمله تزریق است. در این شیوه نیز سعی می‌شود کدهای اسکریپتی درون نرم‌افزار تزریق شود. ولی تفاوت اصولی‌ای میان آنها وجود دارد که تفاوت در هدف حمله است. در شیوه تزریق هدف، اجرای کد بر روی سرورهای وب یا پایگاه داده بود؛ اما در اینجا هدف، اجرای کد بر روی مرورگر کاربران است.
حمله‌کنندگان سعی می‌کنند کدهایی را در نرم‌افزار ذخیره کنند که هیچ تاثیری در سرورها ندارد و اصولا بر روی سرور قابلیت اجرا ندارد؛ اما وقتی که این دستورات برای درخواست یک کاربر به مرورگر کاربر منتقل می‌شود، مرورگر این اطلاعات را به عنوان اطلاعات قابل اجرا تشخیص می‌دهد و آن را بر روی مرورگر کاربر اجرا می‌کند. هدف از این کار به دست آوردن اطلاعات منتقل شده بر روی مرورگر کاربر یا شناسه نشست و یا مشخصات حساب کاربری کاربر است.

برای مثال فرض کنید که فردی نامه‌ای الکترونیکی برای شما ارسال می‌کند که علاوه بر اطلاعات حاوی کدهای جاوا اسکریپتی به منظور اجرا در مرورگر است. این کد می‌تواند کل اطلاعات دریافت شده از سمت سرور را به آدرس سایتی ناشناس ارسال کند. شما از طریق نرم‌افزار وبی این نامه را باز می‌کنید. با توجه به اینکه اطلاعات سایر نامه‌های الکترونیکی شما نیز همراه با دریافت این نامه از سرور دریافت شده است، این کد اجرا و تمامی‌اطلاعات نامه‌های شما برای سایت ناشناس ارسال می‌شود و اطلاعات

شما کاملا به خطر می‌افتد. حمله‌کنندگان برای حمله از اصول و قواعد پیاده‌سازی شده در مرورگرها مانند HTML، CSS و جاوا اسکریپت استفاده می‌کنند.

این شیوه حمله هم از طریق افراد ناشناس انجام می‌شود و هم از طریق کاربران مجاز درون سیستم. آن‌ها کدهای خود را از طریق متغیرهای آدرس URL، فرم‌های ورود اطلاعات و سیستم‌های همکار نرم‌افزاری به نرم‌افزار تزریق می‌کنند؛ اما در مجموع 2 دسته کلی از این حمله وجود دارد:

- ذخیره شده: در این دسته همانند مثال ذکر شده، کدها درون پایگاه داده نرم‌افزار ذخیره می‌شوند و بعدا برای کاربران با توجه به درخواست‌های دریافت اطلاعات کاربران بازگردانده می‌شوند.
- منعکس شده: در این دسته از حملات کدها درون پایگاه داده ذخیره نمی‌شوند. سعی می‌شود درخواستی نامعتبر بر روی مرورگر کاربر برای سرور ارسال شود و سرور درخواست را رد کند و در پاسخ رد درخواست، کدهای ارسال شده را باز گرداند. به عنوان مثال نرم‌افزار شناسه یک موجودیت را که به صورت عددی است در URL دریافت می‌کند. حمله‌کننده به جای عدد ارسالی، یک کد جاوا اسکریپت در متغیر قرار می‌دهد و آن را ارسال می‌کند. در پاسخ، سرور با توجه به‌اینکه عددی دریافت نکرده است خطا می‌دهد و همراه با پیغام خطای بازگشتی، کدهای ارسال شده را نیز باز می‌گرداند. در این حالت، کدها بر روی مرورگر کاربر هدف اجرا و در اصطلاح، سیستم هک می‌شود.

شیوه‌های مختلفی برای جلوگیری از این نوع حمله وجود دارد؛ اما همه‌این شیوه‌ها بر

اساس یک اصل ساده است و آن این است که هیچ اطلاعات دریافتی از سیستم‌ها یا کاربران مورد اطمینان نیست. تمام وظیفه جلوگیری از این نوع حمله بر عهده تولیدکننده نرم‌افزارست و تولیدکننده باید این اصل را در تمامی‌بخش‌های نرم‌افزار خود رعایت کند.

پایه جلوگیری ازاین نوع حمله Encode کردن داده‌های کاربری است. هر چیزی که به عنوان اطلاعات در نرم‌افزار وجود دارد باید در هنگام نمایش بر روی مرورگر کاربر با ساختار Encode شده نمایش داده شود. تولیدکنندگان نرم‌افزار باید نرم‌افزار را به گونه‌ای تهیه کنند که مرز بین کدهای HTML و جاوا اسکریپت تولیدی خود نرم‌افزار و داده‌های کاربری به صورت شفافی مشخص باشد. ازاین رو باید بر اساس شیوه استفاده از اطلاعات آن‌ها را به 3 شیوه کلی JavaScript، HTMLEncoding Encoding و URL Encoding کد گذاری کنند. این کار باید در تمامی‌فرم‌های نمایش و ورود اطلاعات نرم‌افزار رعایت شود.

## ارجاع مستقیم به اشیاء به صورت ناامن

این نوع حمله از ساده‌ترین ولی کاربردی‌ترین نوع حملات است و برای توضیح آن یک مثال می‌زنم. فرض کنید کاربر معتبری وجود دارد که به نامه‌ای با شناسه 1 دسترسی دارد و به نامه با شناسه 2 دسترسی ندارد. نرم‌افزار به صورت نام امنی برای مشاهده یک نامه به کاربر شناسه، نامه را در آدرس URL دریافت می‌کند و آن را به کاربر نمایش می‌دهد. کاربر با توجه به دسترسی‌ای که دارد ابتدا درخواست مشاهده نامه با شناسه 1 را می‌دهد. بعد از نمایش نامه متوجه می‌شود که در آدرس URL متغیری وجود دارد که

مقدار آن 1 است. کاربراین مقدار را 2 می‌کند و درخواست جدیدی به سرور ارسال می‌نماید. به‌این‌ترتیب کاربر به صورت نا امن ارجاع مستقیم به نامه با شناسه 2 داشته است و اگر نرم‌افزار دسترسی کاربر بر روی‌این نامه را بررسی نکند کاربر به هدف خود دست یافته و اطلاعاتی که به آن دسترسی نداشته را به دست آورده است.

جلوگیری از این نوع حمله کاملا بر عهده شرکت تولیدکننده نرم‌افزارست و دو شیوه برای جلوگیری از این نوع حمله وجود دارد:

- استفاده از ارجاعات غیر مستقیم به اشیاء بر اساس کاربر یا نشست: دراین شیوه، نرم‌افزار شناسه واقعی اشیاء را از طریق الگوریتمی‌به اطلاعات دیگری تبدیل می‌کند و فقط خود نرم‌افزار می‌داند که چه شناسه‌ای تبدیل به چه اطلاعاتی شده است. به عنوان مثال عدد 1 را با یک رمزنگار که وابسته به کاربر یا نشست است رمزنگاری می‌کند و مقدار رمز شده xHTpC32Xt را برای کاربر ارسال می‌کند و در زمان درخواست کاربر انتظار دریافت مقدار رمز شده را دارد و فقط خود نرم‌افزار می‌تواند رمز را باز کند. با توجه به‌اینکه الگوریتم تبدیل داده‌ها به هم را فقط نرم‌افزار در اختیار دارد، هیچ کاربری نمی‌تواند مقداری را به صورت تصادفی تولید و درخواستی برای مشاهده آن دهد.
- بررسی دسترسی: دراین روش پیش از استفاده از هر شی و در اختیار قرار دادن اطلاعات برای کاربر آن، ابتدا بررسی می‌شود که کاربر به آن دسترسی دارد یا خیر. هر نرم‌افزار بر اساس مجموعه ساختارهای دسترسی سیستم اطلاعاتی، مجموعه‌ای از قابلیت‌ها و اشیا را در اختیار کاربران خود قرار می‌دهد. دراین روش نرم‌افزار به ازای هر استفاده از شی بررسی می‌کند که آیا در سیستم اطلاعاتی،

کاربر مجاز به دسترسی شی هست یا خیر.

## پیکربندی اشتباه امنیت

در این نوع حملات مهاجمان سعی می‌کنند به اکانت پیش‌فرض، صفحات استفاده نشده، نقص‌های اصلاح نشده، فایل‌ها و دایرکتوری‌های محافظت نشده و غیره دسترسی پیدا کنند. برای روشن شدن موضوع 2 مثال از حمله می‌زنیم. فرض کنید لیست دایرکتوری‌های بر روی سرور غیر فعال نشده است. مهاجم این لیست را کشف می‌کند و به فایل‌ها دسترسی پیدا می‌کند. به این ترتیب مهاجم به فایل‌های کد کامپایل شده، دسترسی پیدا می‌کند. آن‌ها را decompile می‌کند و کدهای شما در اختیار او قرار می‌گیرد. به این‌ترتیب می‌تواند نواقص کنترل‌های دسترسی موجود در نرم‌افزار را پیدا کند.

فرض کنید پیکربندی سرور نرم‌افزار اجازه می‌دهد در زمان ایجاد خطا در سرور، نرم‌افزار در پاسخ بازگشتی از خطا شرح روال حرکتی در کد تا زمان رویداد خطا (stack track) برای کاربران بازگردانده شود. حقیقت این است که مهاجمان اطلاعات اضافی از پیغام‌های خطا را دوست دارند. آن‌ها از این طریق بدون در اختیار داشتن کدهای شما به کدها دسترسی پیدا می‌کنند. از طریق‌این کدها راه‌های نفوذ به نرم‌افزار را تشخیص می‌دهند و از آن‌ها استفاده می‌کنند.

برای جلوگیری از این نوع حملات برای هر دو گروه، تولیدکننده نرم‌افزار و استفاده‌کننده نرم‌افزار، وظایفی وجود دارد و این دو باید با هم نقشه‌ای از پیکربندی امن تهیه کنند و

آن را در محیط استقرار نرم‌افزار اجرا کنند. فایل‌های قدیمی‌که دیگر استفاده‌ای برای نرم‌افزار ندارد و یا اجازه بازگرداندن شرح روال حرکتی کد تا زمان اجرا و دیگر تنظیمات با هماهنگی این دو بر روی سرور وب و پایگاه داده انجام می‌گیرد.

## افشای داده‌های حساس

در این نوع حمله که توسط افرادی که به داده‌های حساس یا نسخه پشتیبان آن‌ها می‌توانند دسترسی داشته باشند اتفاق می‌افتد، مهاجمانی که به صورت عادی نمی‌توانند به داده‌های رمزگذاری شده دسترسی داشته باشند برای دسترسی به داده‌ها کار دیگری مانند سرقت کلیدها انجام می‌دهند و یا داده‌هایی که در حال انتقال بین بخش‌های مختلف معماری استقرار نرم‌افزار هستند را سرقت می‌کنند. در مجموع باید دانست که هر داده حساسی که از سرور وب و یا از سرور پایگاه داده خارج می‌شود، وارد دنیای نا امنی می‌شود. می‌تواند در مسیر حرکت تا سیستم کاربر یا حتی در خود سیستم کاربر شنود شود. به عنوان مثال فرض کنید یک سایت نمی‌تواند از HTTPS استفاده کند. یک مهاجم ترافیک شبکه را مانیتور می‌کند و کوکی مربوط به Session کاربر را سرقت می‌کند و اقدام به استفاده از این Session برای دسترسی به داده‌های خصوصی کاربر می‌کند. مثال دیگر این است که یک نسخه پشتیبان از پایگاه داده تهیه می‌شود و در مکانی قرار می‌گیرد. اگر داده‌های حساس مانند مشخصات کارت اعتباری کاربران در این پایگاه داده به صورت رمز نشده، ذخیره شود مهاجم با به دست آوردن نسخه پشتیبان به هر روشی می‌تواند به این اطلاعات، دسترسی پیدا کند.

برای جلوگیری از این شیوه حمله، داده‌های حساس چه برای ارسال به کاربر و چه برای

نگهداری در پایگاه داده باید رمزگذاری شوند . همچنین باید داده‌های غیر ضروری حذف شوند، حتی در این مورد نباید به سیستم‌کش شناسه کاربری و رمز عبور مرورگرها اعتماد کرد و باید به گونه‌ای کد نویسی انجام داد که تکمیل اتوماتیک فرم‌ها از جمله فرم احراز هویت کاربر، قابل کش شدن توسط مرورگرها نباشد. برای مسیر انتقال داده‌ها بین سرور و کلاینت و حتی بین سرورهای وب و دیتابیس باید از پروتکل‌های رمزنگاری قوی مانند TLS استفاده کرد.

## از دست رفتن کنترل دسترسی‌ها در سطح توابع

دراین حمله که ممکن است توسط هر کاربری که توانایی ارسال درخواست به نرم‌افزار را دارد روی دهد، مهاجم با تغییر Url یا پارامترهای آن اقدام به اجرای متدی می‌کند که به آن دسترسی ندارد. نرم‌افزارها همیشه از سطوح دسترسی توابع نرم‌افزار حفاظت نمی‌کنند. بعضی زمان‌ها حفاظت بوسیله تنظیمات مدیریت می‌شود و اگراین تنظیمات سیستمی‌درست پیکربندی نشده باشند و یا در برخی از زمان‌ها توسعه دهندگان فراموش کنند که این کار را در کد انجام دهند، راهی برای نفوذ مهاجمان ایجاد می‌شود تا بدون دسترسی توابع نرم‌افزار را فراخوانی کنند.

به عنوان مثال یک صفحه در آدرس Url خود متغیری به نام Action دارد که مقدار آن مشخص‌کننده این است که کاربر درخواست اجرای چه تابعی دارد (این روش در برنامه‌نویسی بسیار شایع است علت آن هم جلوگیری از تکرار کد در سطح نرم‌افزار است) دراین شرایط اگر پیکربندی مناسبی انجام نشده باشد کاربران شناس یا ناشناس که مجوز فراخوانی تابع را ندارند، می‌توانند تابع خاصی را با تغییر در مقدار متغیر

Action فراخوانی و اجرا کنند.

در مورد این نوع حمله وظیفه جلوگیری از نفوذ بر عهده تولیدکننده نرم‌افزارست. تولیدکننده باید تعریف شفاف و ساده‌ای از توابع و دسترسی‌های سوار بر توابع داشته باشد و باید بر اساس این تعریف اقدام به پیاده سازی کدها کند. این پیاده سازی باید در سمت سرور وب موثر باشد و تاثیر بررسی دسترسی‌ها در سمت مرورگر کاربر به تنهایی کفایت نمی‌کند. به عنوان مثال از تاثیرپذیری اکثر تولیدکنندگان نرم‌افزار بر اساس دسترسی‌های کاربر به توابع، اقدام به نمایش یا عدم نمایش یک دکمه در صفحه نمایشی کاربر می‌کنند. درست است که دکمه‌ای نمایش داده نشده است اما کاربری که کمی با دنیای وب آشنا باشد می‌تواند شبیه سازی از حضور این دکمه برای خود را داشته باشد و درخواستی برای اجرای تابع مربوط به دکمه شده به سرور ارسال کند. در این شرایط اگر محدوده اثر بررسی دسترسی‌ها فقط بر روی نمایش یا عدم نمایش دکمه باشد درخواست اجرای تابع کاربر پذیرفته می‌شود. تولیدکننده نرم‌افزار باید به ازای هر تابع درخواستی بررسی دسترسی کاربر به تابع را انجام دهد و ملاک را نباید بر اساس سوابق روال حرکتی کاربر در نرم‌افزار قرار دهد.

## درخواست‌های تقلبی از طریق سایت (CSRF)

این شیوه یکی از هوشمندانه‌ترین و همچنین شایع‌ترین شیوه‌های حمله به نرم‌افزارهای وبی است. به‌این نوع حمله، حمله از تب کناری هم گفته می‌شود. اجازه دهید که با مثالی‌این شیوه را توضیح دهم. این مثال با تغییرات بر روی مثالی است که در توضیح‌این نوع حمله در سایت OWASP آورده شده است.

سایت بانک به کاربران اجازه ارسال درخواست انتقال پول غیر امنی به صورت زیر می‌دهد.

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

این درخواست باعث می‌شود که مبلغ 1500 ریال به حساب مشخص شده واریز شود. مهاجم از این شیوه غیر امن استفاده می‌کند و مقادیر موجود در متغیرهای درون URL را به گونه‌ای تغییر می‌دهد که مبلغی پول به حساب خود واریز شود. این URL تولید شده را به صورت زیر در یک عنصر HTMLای قرار می‌دهد و این را به عنوان محتوای یک ایمیل برای کاربر قربانی ارسال می‌کند.

img>
src=”http://example.com/app/transferFunds?amount=1500&destinatio
</ nAccount=attackersAcct#” width=”0" height=”0"

این عنصر از مرورگر می‌خواهد که تصویر بیان شده در آدرس URL را به کاربر نمایش دهد.

کاربر قربانی مرورگر خود را باز کرده و در تب اول آن وارد سایت بانک خود شده و در تب دوم آن، وارد سایت ایمیل خود شده است. قربانی ایمیل جدید را دریافت می‌کند و آن را باز می‌کند. به محض بازکردن ایمیل توسط قربانی مرورگر قربانی برای نمایش تصویر، درخواستی برای دریافت اطلاعات به آدرس قرار داده شده در عنصر HTMLای

ارسال می‌کند.

با توجه به‌اینکه قربانی در سایت بانک احراز هویت شده و نشست معتبری با سایت بانک خود ایجاد کرده است، مرورگر اطلاعات نشست معتبر کاربر با سایت را نیز همراه با درخواست ارسال می‌کند. سایت بانک این درخواست را با توجه به اینکه نشست معتبری بین سرور و کاربر وجود دارد می‌پذیرد و پول بدون اینکه کاربر متوجه شود از حساب قربانی خارج و به حساب مهاجم منتقل می‌شود. به این‌ترتیب حمله با موفقیت و بدون اینکه مهاجم از مشخصات کاربر قربانی استفاده کرده باشد از تب کناری مرورگر انجام می‌شود.

تمامی‌این شیوه حمله بر اساس ساختار استاندارد مروگرها بر روی سیستم کاربران هستند ولی وظیفه جلوگیری از این نوع حمله کاملا بر عهده تولیدکننده نرم‌افزار است. تولیدکننده نرم‌افزار از زمان احراز هویت کاربر به بعد، به ازای هر درخواست اولیه‌ای که کاربر ارسال می‌نماید ابتدا یک مقدار مرتبط با کاربر ایجاد می‌کند و در پاسخ، درخواست آن را در بخش‌های مختلف از جمله به عنوان یک مقدار مخفی در فرم و یک مقدار در کوکی مرورگر قرار می‌دهد. سپس انتظار دارد درخواست بعدی کاربر حاوی این مقادیر باشد و این مقادیر باید همانی باشد که خود برای درخواست قبلی تولید کرده است و اگر یکی ازاین دو قانون هم برقرار نبود، سرور نرم‌افزار درخواست را برای اجرا رد می‌کند.

# استفاده از مولفه‌های جانبی با آسیب پذیری‌های شناخته شده

تمامی‌تولیدکنندگان نرم‌افزار از مولفه‌ها و ابزارهای جانبی در کنار کد نویسی خود استفاده می‌کنند و برخی از وظایف نرم‌افزار را بر عهده این ابزارها قرار می‌دهد. این ابزارها به عنوان مولفه‌های طرف سوم در تولید نرم‌افزار نام برده می‌شود و به سرعت تولید نرم‌افزار و بالا بردن قابلیت‌های نرم‌افزار کمک می‌کنند. به عنوان مثال JQuery یکی از شناخته شده‌ترین این مولفه‌هاست که به عنوان یک ابزار کمکی به کمک تولیدکنندگان نرم‌افزار می‌آید تا کدهای جاوا اسکریپت کمتری نوشته شود و با عناصر HTMLای صفحه بتوان راحت‌تر و دقیق‌تر کار کرد.

این ابزارهای جانبی همواره در حال توسعه هستند و نسخه‌های مختلفی از آن‌ها به مرور زمان تولید می‌شود؛ اما نکته این است که ابزارها خود می‌توانند دارای باگ‌هایی باشند که باعث ایجاد حفره‌های امنیتی می‌شوند. این حفره‌های امنیتی بعد از کشف، تبدیل به آسیب‌پذیری‌های شناخته شده می‌شوند و اغلب در نسخه‌های بعدی این ابزارها این حفره‌ها بسته و ابزار امن می‌شود. مهاجمان سعی دارند تا نرم‌افزارهایی را پیدا کنند که از نسخه‌های دارای آسیب‌پذیری استفاده می‌کنند و با توجه به وجود این حفره در نسخه استفاده شده از این حفره برای نفوذ به نرم‌افزار بهره ببرند.

وظیفه جلوگیری از این نوع حمله کاملا بر عهده تولیدکننده نرم‌افزار است و تنها راه رفع حفره‌ها این است که ابزار جانبی استفاده شده در نرم‌افزار با نسخه‌ای از این ابزار که

آسیب‌پذیری را رفع کرده است به روز شود و نسخه جدید نرم‌افزار بعد از بروزرسانی مولفه‌های جانبی در تمامی‌سازمان‌های مشتریان، جایگزین نسخه قدیمی شود.

## تغییر مسیرها و ارجاعات نامعتبر

آخرین دسته‌بندی انواع حملات تغییر مسیرها و ارجاعات نامعتبرست که نرم‌افزارها گاهی بر اساس نیازمندی‌های سیستمی‌آن را انجام می‌دهند. اغلب نرم‌افزارها کاربران را برای دسترسی به سایر صفحات به مسیرهای دیگر تغییر مسیر می‌دهند. یا اینکه از ارجاعات داخلی استفاده می‌کنند تا کاربر بتواند فرایند حرکتی انجام یک کار را کامل کند. گاهی اوقات که صفحه هدف در یک پارامتر نامعتبر در آدرس URL مشخص می‌شود به مهاجمین اجازه می‌دهد که صفحه مقصد خود را به جای صفحه داخلی نرم‌افزار انتخاب و کاربر را به سایت خود منتقل کنند.

برای مثال آدرس‌های URLای مانند آدرس زیر دارای‌این راه نفوذ هستند:

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

مهاجم این آدرس را ایجاد و کاربران قربانی سایت را با ترفندهای مختلف به کلیک بر روی این آدرس راضی می‌کند. کاربر فکر می‌کند این آدرس با توجه به دامنه آن، آدرس شناخته شده‌ای برای خودش است پس اعتماد کرده و بر روی آن کلیک می‌کند. کاربر قربانی بعد از ورود به سایت شناخته شده مستقیم به سایت مهاجم منتقل و در آنجا ویروس بر روی سیستم قربانی نصب می‌شود یا اینکه با حملات Fishing اطلاعاتی از کاربر دریافت می‌شود.

وظیفه جلوگیری از این نوع حمله نیز بر عهده تولیدکنندگان نرم‌افزار است. آن‌ها یا باید به صورت ساده این نوع امکانات را نداشته باشند و یا اگر می‌خواهند از این امکان در نرم‌افزار خود استفاده کنند، حتما باید مقدار متغیر را بررسی کنند که ارجاع آن به صفحه‌ای در داخل نرم‌افزار باشد و هر ارجاعی با خارج از نرم‌افزار را رد کنند.

انجمن OWASP علاوه بر معرفی‌این حملات به ازای تک‌تک آن‌ها حالت‌های مختلف حمله را بیان کرده است. همچنین به ازای زبان‌های برنامه‌نویسی و تکنولوژی‌های پایه بر روی هر زبان در وب، راهکارهایی برای کدنویسی امن و جلوگیری از راه‌های نفوذ ارائه داده است. این انجمن همچنین شروع به تولید ابزارهایی کرده که حاوی راهکارهای امن‌سازی شده هستند و می‌توانند به عنوان ابزارهای طرف سوم به کمک تولیدکنندگان نرم‌افزار آمده و به سرعت و دقت امن سازی نرم‌افزارها کمک کنند. آن‌ها حتی راهنماهایی برای توسعه نرم‌افزار و شیوه تفکر امن در فازهای مختلف تولید نرم‌افزار ارائه داده اند. تمامی‌این اطلاعات به صورت کاملا رایگان در وب سایت OWASP در اختیار تمامی استفاده‌کنندگان قرار گرفته است.

**درخواست دموی نرم‌افزارهای دیدگاه**