# **T10 ده ‌‌‌آسیب‌پذیری بحرانی امنیت API از منظر OWASP – 2019**

|  |  |
| --- | --- |
| **API1: مجوزدهی نادرست در سطح اشیا** | APIها معمولا توابع مدیریت کننده شناسه‌های اشیا را در معرض دید قرار داده و سطح حمله[[1]](#footnote-1) گسترده ای را برای نقض کنترل دسترسی ایجاد می‌نمایند. کنترل‌های مجوزدهی در سطح اشیا بایستی در کلیه توابعی که با گرفتن ورودی از کاربر به منابع داده دسترسی دارند پیاده‌سازی شود. |
| **API2: احرازهویت نادرست کاربر** | مکانیزم‌های احرازهویت غالبا به درستی پیاده‌سازی نشده و سبب دسترسی مهاجمین به توکن‌های احرازهویت و ربایش موقت یا دائمی هویت سایر کاربران با استفاده از نقایص این مکانیزم‌ها می شوند. نقض توانایی سیستم در شناسایی کلاینت یا کاربر، منجر به نقض امنیت API خواهد شد. |
| **API3: افشای مفرط داده** | با بکارگیری سرویس‌‌های عمومی API، توسعه دهندگان عملا تمامی ویژگی‌‌های اشیا را بدون درنظر گرفتن حساسیت تک تک آنها و صرفا با تکیه بر فیلترینگ داده پیش از نمایش به کاربر، توسط کلاینت، در معرض دید عموم قرار می‌دهند. |
| **API4: کمبود منابع و نبود محدودیت نرخ در ارسال درخواست** | معمولا APIها هیچ محدودیتی بر اندازه یا تعداد منابع درخواستی توسط کلاینت یا کاربر اعمال نمی‌نمایند. این موضوع نه تنها با تاثیرگذاری منفی بر عملکرد سرور API می‌تواند منجر به حمله رد سرویس (DoS) شود، بلکه در را برای نقض احرازهویت از طریق حملاتی نظیر Force Brute نیز باز می‌گذارد. |
| **API5: مجوزدهی نادرست در سطح توابع** | مکانیزم‌‌های پیچیده کنترل دسترسی با سلسله مراتب، گروه‌‌ها و نقش‌‌های متفاوت و مرز نامشخص بین توابع عادی و مدیریتی سبب بروز نقایص مجوزدهی می‌شوند. با بهره برداری از این آسیب‌پذیری‌‌ها مهاجمین به منابع سایر کاربران و یا توابع مدیریتی دست خواهند یافت. |
| **API6: تخصیص جمعی** | پیوند دادن داده ارائه شده توسط کلاینت (نظیر اشیا JSON) با مدل‌‌های داده بدون فیلترکردن مناسب آنها بر مبنای یک لیست سفید می‌تواند منجر به تخصیص جمعی شود. با تشخیص ویژگی‌‌های اشیا، کاوش سایر توابع، خواندن مستندات یا ارائه ویژگی‌‌های اضافی برای اشیا در بدنه درخواست‌‌ها، مهاجم می‌تواند ویژگی‌‌هایی از اشیا که برای وی مجاز نیست را دستکاری نماید. |
| **API7: پیکربندی امنیتی نادرست** | پیکربندی امنیتی نادرست پیامدی از بکارگیری پیکربندی ناایمن پیشفرض، پیکربندی ناقص یا غیرمتمرکز، فضای ذخیره سازی ابری باز و محافظت نشده، سرایندهای HTTP با پیکربندی نادرست، متدهای غیرضروری HTTP، خط مشی‌‌های سهل انگارانه برای اشتراک گذاری منابع متقابل (CORS) و پیامهای خطای تفصیلی و مشروح می‌باشد. |
| **API8: ‌‌‌آسیب‌پذیری‌‌های تزریق** | آسیب‌پذیری‌‌های مبتنی بر تزریق نظیر SQL، NoSQL، تزریق دستور و ... زمانی رخ می‌دهند که داده‌ی نامطمئن بعنوان بخشی از یک دستور یا پرس و جو به مفسر تحویل داده شود. این داده مخرب می‌تواند مفسر را وادار به اجرای دستوری ناخواسته یا دسترسی غیرمجاز به داده‌‌ها نماید. |
| **API9: مدیریت نادرست دارایی‌‌ها** | APIها معمولا توابع بیشتری را نسبت به وب اپلیکیشن‌‌های سنتی در معرض دید قرار می‌دهند که این موضوع اهمیت مستندسازی مناسب و بروز را دوچندان می‌نماید. داشتن فهرستی از میزبان‌‌ها و نسخه‌‌های بکارگرفته شده API نقش مهمی در رفع ‌‌‌آسیب‌پذیری‌‌های مرتبط با نسخ قدیمی API و توابع مرتبط با debugging ایفا می‌کند. |
| **API10: پایش و نظارت ناکافی** | پایش و نظارت ناکافی در کنار عدم وجود فرایند پاسخ دهی به وقایع[[2]](#footnote-2) یا پیاده‌سازی ناقص آن به مهاجم امکان تثبیت دسترسی، حمله به سایر سیستم‌‌ها و استخراج/نابودسازی داده‌‌ها را می‌دهد. مطالعات انجام شده بیانگر آن است که زمان آگاهی یافتن از نفوذ انجام شده به طور میانگین بیش از 200 روز پس از انجام نفوذ بوده و تشخیص آن نیز بجای آنکه توسط فرایندهای درونی پایش و نظارت باشد توسط شرکت‌‌های ثالث صورت می‌پذیرد. |

1. Attack Surface [↑](#footnote-ref-1)
2. Incident Response [↑](#footnote-ref-2)