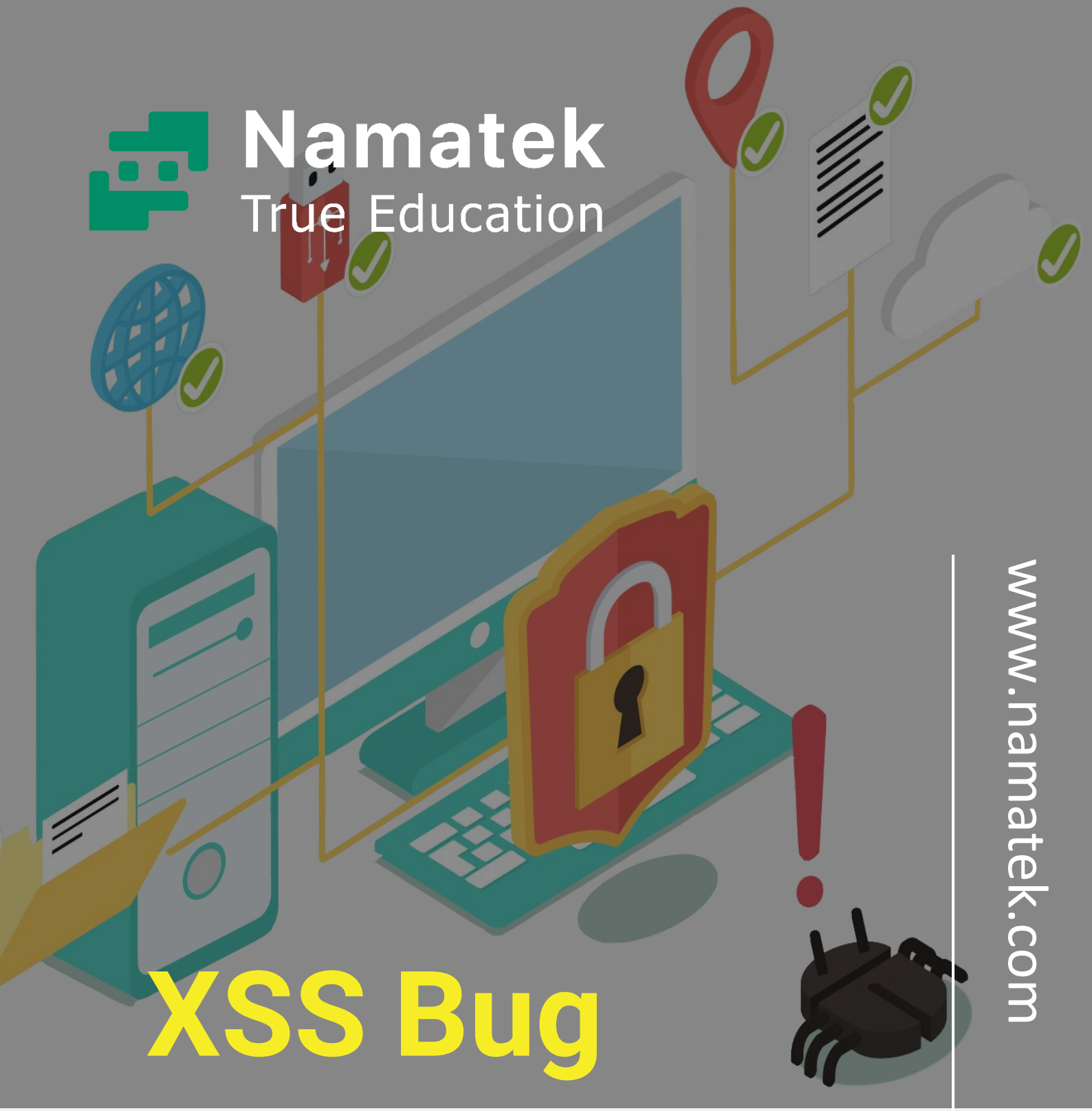
Namatek
True Education



XSS Bug

www.namatek.com

۳ نوع حمله باگ XSS و ۴ روش جلوگیری از آن

فهرست مطالب

امنیت و نگهداری از وب سایت ها در مقابل حملات و سوء استفاده های مختلف مهاجمان و هکرها برای صاحبان کسب و کار ها و کاربران فضای اینترنت بسیار مهم است و همین امر موجب شده است که نیاز به دانستن اینکه باگ XSS چیست را در خود احساس می کنند. برخی از حملات رایج به وب سایت ها از طریق باگ XSS انجام می شود که ناشی از طراحی و پیاده سازی ناامن صفحات و برنامه های وب است و با شناخت کافی می توان به خوبی جلوی آن ها را گرفت.

این مقاله شامل اطلاعات مفید و کاربردی در مورد انواع حملات XSS، نحوه عملکرد این حملات، روش های پیشگیری از آن و ایمن سازی ورودی های وب سایت شماست. با ما همراه باشید.

#۱ باگ XSS چیست؟

باگ XSS مخفف Cross-site scripting است. از آن جا که Cross با حرف یا علامت X نیز شناخته می شود، به همین دلیل از X در ابتدای XSS استفاده شده است تا از تداخل اسمی با CSS که یک زبان نشانه گذاری در طراحی وب است، جلوگیری شود.

XSS در واقع نوعی آسیب پذیری امنیتی است که مهاجمان را قادر می سازد تا اسکریپت های مخرب را به صفحات وب مشاهده شده توسط سایر کاربران تزریق و آن ها را اجرا کنند. حمله XSS هنگامی اتفاق می

افتد که قربانی از صفحه یا برنامه وبی که کد مخرب را اجرا می کند بازدید کند. این صفحه یا برنامه وسیله ای برای تحویل اسکریپت مخرب به مرورگر کاربر است.

برخی از موارد آسیب پذیر در وب که معمولا برای حملات Cross-site Scripting استفاده می شوند، عبارتند از:

- انجمن ها
- صفحات چت
- صفحات وبی که امکان ارسال نظر دارند

یک صفحه یا یک برنامه وب در صورت استفاده از ورودی بررسی نشده، در معرض آسیب پذیری XSS است.

حملات XSS از طریق موراد زیر امکان پذیر است:

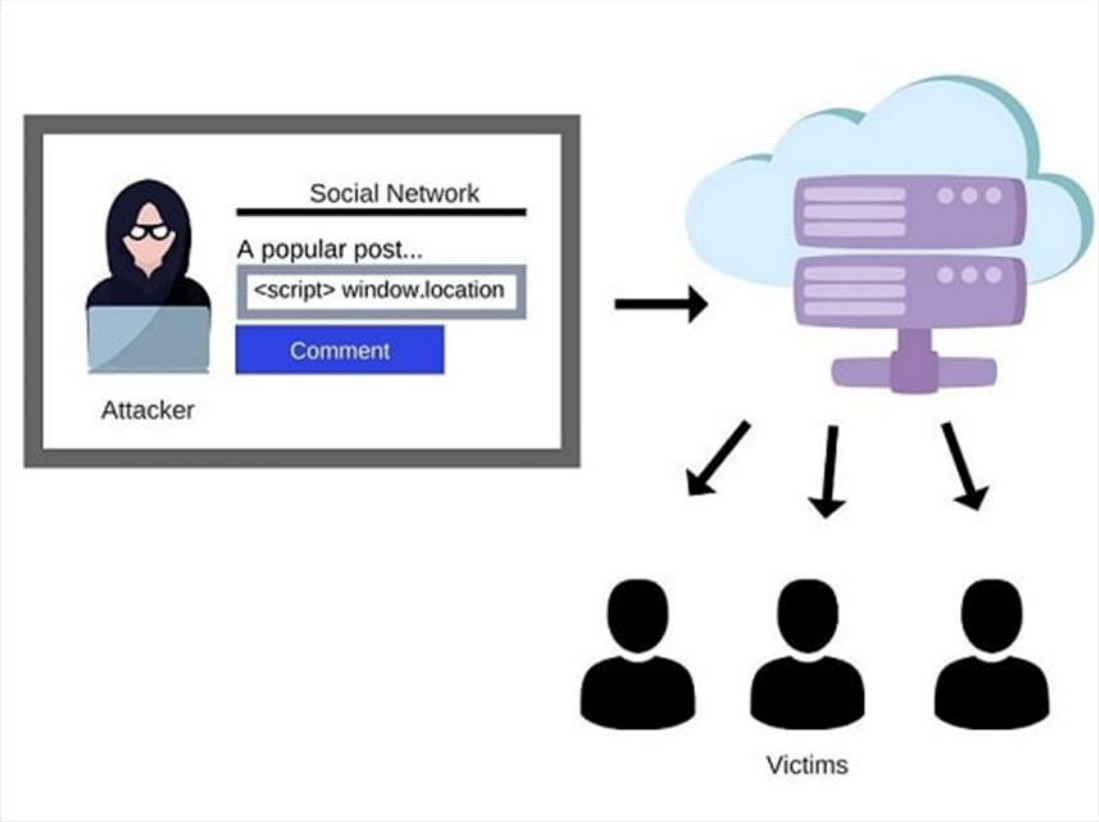
- JavaScript

- VBScript
- ActiveX
- Flash
- CSS

با این حال، مهاجمان بیشتر از JavaScript برای این کار استفاده می کنند. به این دلیل که JavaScript پایه و اساس بخشی از عملکردهای مرورگر است (از JavaScript در نوشتن بخش هایی از مرورگرها استفاده شده است.) اما اثرات باگ XSS چیست؟

XSS می تواند از مزاحمت های کوچک تا خطرهای امنیتی قابل توجه را در بر بگیرد و بسته به حساسیت داده های سایت، میزان آسیب پذیری و در نظر نگرفتن تمهیدات امنیتی توسط مالک وب سایت، می تواند متفاوت باشد.

#۲ باگ XSS چگونه عمل می کند؟



نحوه عملکرد حمله باگ XSS چیست؟

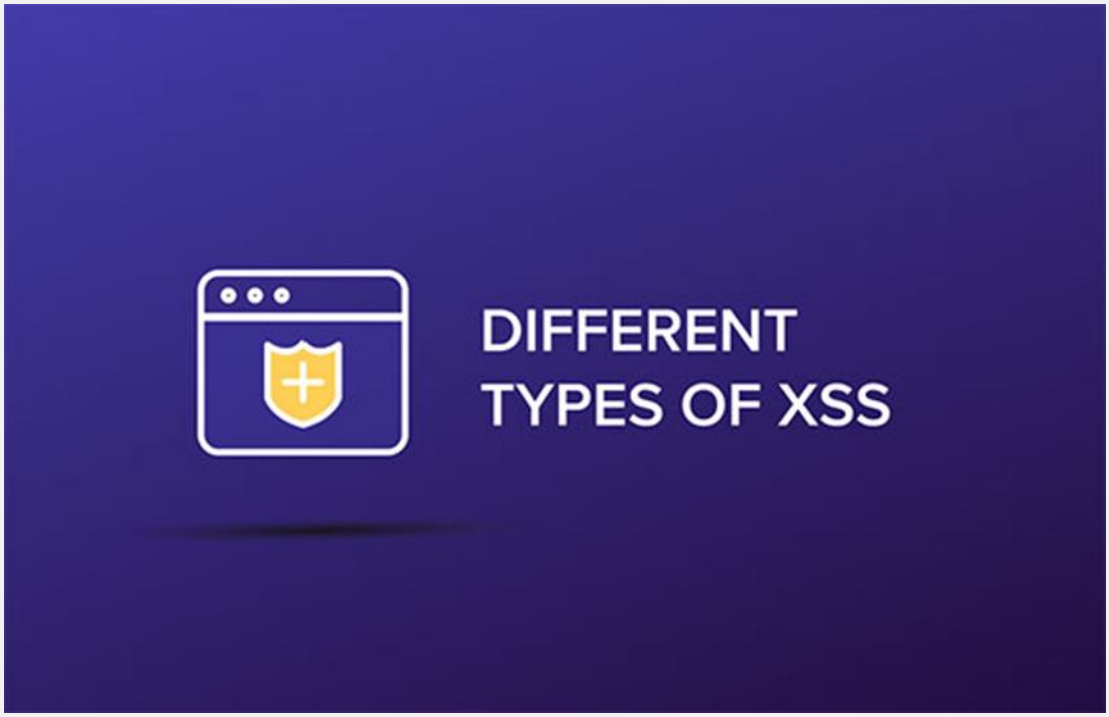
به طور معمول یک حمله معمولی باگ XSS دو مرحله دارد:

۱. برای اجرای کد مخرب JavaScript در مرورگر یک قربانی، ابتدا باید یک مهاجم راهی برای تزریق کد مخرب (payload) به صفحه وب مورد بازدید قربانی پیدا کند.
۲. قربانی باید به صفحه دارای کد مخرب مراجعه کند. اگر حمله متوجه قربانیان خاصی باشد، مهاجم می تواند با استفاده از مهندسی اجتماعی و یا فیشینگ یک URL مخرب برای قربانی ارسال کند.

برای امکان پذیر شدن مرحله ۱، وب سایت آسیب پذیر باید داده های وارد شده توسط کاربر را مستقیما در صفحات خود بگنجاند (مانند ارسال نظر یا دیدگاه در مقالات). از این طریق مهاجم می تواند یک رشته مخرب را وارد کند که در صفحه وب استفاده می شود و توسط مرورگر قربانی به عنوان کد منبع تلقی می شود.

اشکال مختلفی از حملات XSS نیز وجود دارد که در آن مهاجم کاربر را برای بازدید از یک URL فریب می دهد تا کاربر روی پیوند بارگذاری شده توسط او کلیک کند. این پیوند می تواند اطلاعات شخصی و بسیار مهمی نظیر اطلاعات بانکی کاربر را از او دریافت کند.

#۳ انواع حملات باگ XSS



مهاجمان بسته به اهدافشان، می توانند از چندین روش مختلف برای ایجاد حملات باگ XSS استفاده کنند.

۳ نوع اصلی و متداول این حملات عبارتند از:

#۳-۱ بازتاب XSS

بازتاب XSS یا Reflected XSS ساده ترین نوع از حملات Cross-site scripting است و زمانی که یک برنامه داده ای را در یک درخواست HTTP دریافت می کند به وجود می آید.

به عنوان مثال سایت هایی که در آن ها برخی از پردازش ها را از طریق Query string انجام می دهند، در مقابل این نوع از حملات آسیب پذیر هستند. به این صورت که مهاجم می تواند با تغییر المان های داخل

Query stringدر نوار آدرس، محتوای صفحه را تغییر داده و یک لینک به صفحه حاوی اسکریپت های مخرب در آن ایجاد کند.

اگر کاربر از URL ساخته شده توسط مهاجم بازدید کند، اسکریپت مهاجم در در مرورگر کاربر اجرا می شود. در این صورت، اسکریپت می تواند هر عملیاتی را انجام دهد و هر داده ای را که کاربر به آن دسترسی دارد بازیابی کند.

#۳-۲ XSS ذخیره شده

XSS ذخیره شده، Stored XSSیا Persistent XSS که با عنوان second-order XSS نیز شناخته می شود، هنگامی به وجود می آید که برنامه یا صفحه تحت وب، داده ای را از یک منبع غیر قابل اعتماد دریافت می کند و این داده ها را در پاسخ های HTTP خود به روشی ناامن قرار می دهد.

فرض کنید یک وب سایت به کاربران اجازه می دهد که نظرات خود را در مورد پست های وبلاگ ارسال کنند تا برای سایر کاربران نمایش داده شود. در این بین کاربران مهاجم نظرات خود را با درخواست HTTP همراه با کدهای مخرب ارسال و به اصطلاح به صفحه تزریق می کنند. به عنوان مثال برخی از کاربران با استفاده از این روش حمله باگ XSS، با هر بار لود صفحه (بارگذاری) کاربران نهایی را مجبور به دانلود یک برنامه یا فایل مخرب با نام های مستعار و کاربردی می کند.

#۳-۳ XSS مبتنی بر DOM

در ابتدا بهتر است کمی با مفهوم DOM آشنا شویم. DOM یا Document Object Model نمایش سلسله مراتبی مرورگر وب از عناصر موجود در صفحه است.

وب سایت ها می توانند از JavaScript برای دستکاری نمایش بخش ها و اشیای مجود در صفحه و همچنین خصوصیات آن ها استفاده کنند. به عنوان مثال با کلیک کاربر بخشی از صفحه وب مخفی شود.

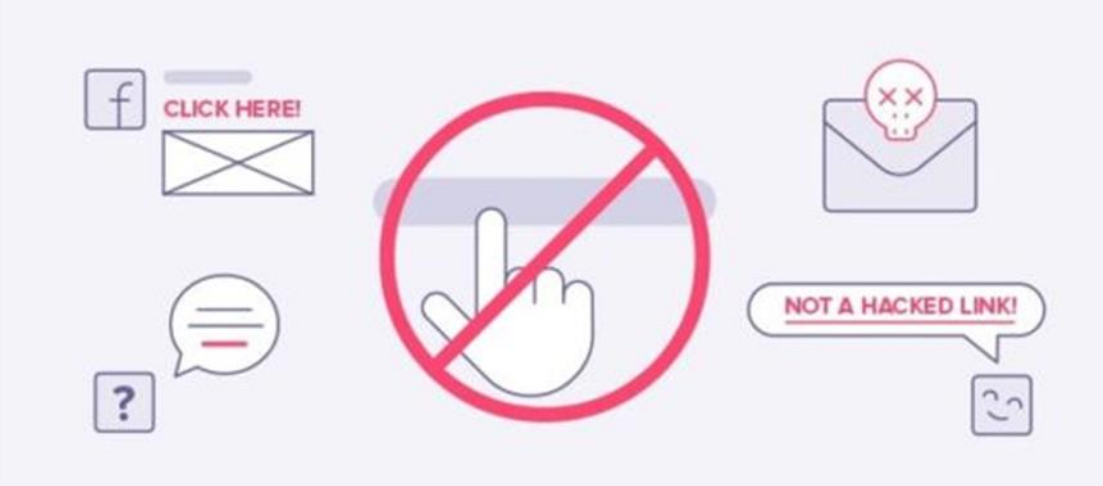
در این روش تگ های HTML به عنوان شیئی در زبان JavaScript در نظر گرفته می شوند که می توان محتوای آن ها را تغییر داده و یا حتی آن ها را حذف کرد.

توجه داشته باشید که DOM به خودی خود مشکلی ایجاد نمی کند؛ چرا که یک بخش جدایی ناپذیر از نحوه کار وب سایت های مدرن است؛ اما به این دلیل که جاوا اسکریپت داده ها را به طور نا امن کنترل می کند، می تواند بستری برای حملات مختلف باشد. بر این اساس، روش اجرای این نوع از حملات باگ XSS چیست؟

آسیب پذیری های XSS مبتنی بر DOM یا DOM-based XSS معمولا زمانی ایجاد می شوند که JavaScript داده ها را از یک منبع قابل کنترل توسط هکر یا مهاجم گرفته و اجرا می کند (مانند یک URL.) این امکان مهاجمان را قادر می سازد تا حساب های کاربران دیگر را بدزدند. از این طریق می توانند به اطلاعات آن ها دست پیدا کرده و یا از حساب آن ها برای انجام کارهای مجرمانه بهره ببرند.

#۴ روش پیشگیری از حملات باگ XSS چیست؟

مهاجمان از روش های مختلفی برای آسیب زدن به وب سایت ها استفاده می کنند. هیچ استراتژی واحدی برای کاهش خطر حمله XSS از طریق سایت وجود ندارد؛ اما مفهوم Cross-Site Scripting به ورودی ناامن و مستقیم کاربر به یک صفحه وب متکی است. بنابراین اگر ورودی های کاربر به درستی پاک سازی شوند، حملات XSS غیرممکن است.



روش های متعددی برای اطمینان از سلامت داده های وارد شده توسط کاربر در وب سایت های شما وجود دارد:

۱. ایجاد فهرست ورودی های مجاز (Allow List Values): با ایجاد یک لیست مجاز و محدود کردن داده های ورودی کاربر، فقط مقادیر شناخته شده و ایمن به سرور ارسال می شوند.
۲. جلوگیری و محدود کردن ارسال داده HTML توسط کاربر (Avoid and Restrict HTML): اگرچه ممکن است ارسال HTML برای محتوای کامل و شکیل مورد نیاز باشد؛ اما باید به کاربران معتمد محدود شود یا از روش های جایگزین برای تولید محتوا مانند Markdown استفاده شود. Markdown یک روش مدرن برای اجرای کد HTML در مقصد با نماد و علائم است. برای مثال اگر از ۲ علامت # در تیتر فایل ورد استفاده کنید، با انتقال آن به وب سایت به تگ H2 (تیتر دوم) تبدیل می شود.
۳. پاک کردن مقادیر ناامن در زمان ارسال (Sanitize Values): وقتی از محتوای تولید شده توسط کاربر در یک صفحه استفاده می کنید، می توانید نویسه های ناامن را با محتوای مورد نظر جایگزین کرده یا پاک کنید.
٤. استفاده از WAF ها (Web Application Firewall): برای جلوگیری از حمله XSS به وب سایت خود می توانید از فایروال ها استفاده کنید. این روش قبل از این که درخواست های مخرب حتی به وب سایت شما برسد، آن ها را رهگیری می کند.