به نام خدا

انواع حملات XSS

OWASP Article: Type of Cross-Site Scripting

مقدمه:

در این مقاله به توصیف دسته بندی ها و انواع مختلفِ آسیب پذیریِ XSS و چگونگی ارتباط آن ها با یکدیگر می پردازیم.

دسته بندی های XSS تا قبل از سال ۲۰۰۵ میلادی به دو دسته ی Stored و Reflected تقسیم بندی می شد. اما بعد از آن سال، Amit Klein نوع سومی از آن را با نام DOM Based XSS معرفی کرد.

XSS ذخیره شده یا همان Stored (به عنوان پایدار یا Type I نیز شناخته می شود)

XSS ذخیره شده، زمانی اتفاق می افتد که ورودی های کاربر به نحوی در سرور ذخیره شود. مثل دیتابیس، در پست های فروم، لاگ بازدیدکنندگان، فیلدهای مربوط به بخش نظرها و ... . بعد از آن، قربانی داده های ذخیره شده را بدون پاک سازی و از طریق برنامه ی تحت وب می خواند و آن را توسط مرورگر رندر می کند. با اختراع HTML 5 و سایر تکنولوژی های تحت وب می توانیم داده ها را به صورت دائم در مرورگر قربانی و مثلا از طریق دیتابیس HTML5 ذخیره کنیم و با این روش اصلا نیازی به ارسال داده به سرور نخواهد بود.

tamadonEH@gmail.com
https://twitter.com/tamadonEH

XSS بازتابی یا همان Reflected (به عنوان ناپایدار یا Type II نیز شناخته می شود)

XSS بازتابی زمانی رخ می دهد که ورودی کاربر به صورت فوری توسط برنامه ی تحت وب و در قالب پیام خطا، نتیجه ی جستجو یا هر پاسخی که شامل تمام و یا بخشی از داده هایی که کاربر به عنوان درخواست (request) ارسال می کند و بدون امن سازی ورودی به وسیله ی رندر آن توسط مرورگر بازگردانده شود. در این نوع از حمله نیازی به ذخیره سازی دائمی ورودی نیست. در برخی از موارد ممکن است داده هایی که توسط کاربر تولید می شود، هیچ وقت توسط مرورگر از بین نرود.

DOM Based XSS که به عنوان Type-0 نیز شناخته می شود

[اضافه شده به مقاله:Document Object Model(DOM) یک رابط برنامه نویسی برای سندهای XML و Html است . با استفاده از اینترفیس فوق، نحوه دستیابی و انجام پردازش های لازم در رابطه با سند های XML و Html فراهم می گردد . برنامه نویسان با استفاده از DOM ، قادر به ایجاد یک سند ،حرکت در طول ساختار سند، افزودن ، اصلاح و یا حذف المان های یک سند XML و یا Html می باشند. برنامه ای با نام پارسر، امکان استقرار یک سند XML در حافظه را فراهم می نماید . پس از استقرار سند در حافظه ، اطلاعات مربوطه بکمک DOM ، قابل بازیابی و پردازش خواهد بود.]

بر اساس تعریف Amit Klein -به عنوان کسی که اولین مقاله در خصوص DOM Based XSS را منتشر کرد- این حمله نوعی از XSS است که جریان داده های آلوده ی ورودی را در قسمتی از مرورگر ذخیره می کند. برای مثال سورس داده ها، sinkها و داده هایی که هرگز از مرورگر خارج نمی شوند، در DOM ذخیره می شوند. سورس ها جای اند که داده های مخرب از آن خوانده می شوند برای مثال می تواند URL مربوط به صفحات (مثل document.location.href) و یا یک المان از HTML باشد و sinkها، فرخوانی متدهای حساسی اند که باعث اجرای داده های مخرب می شوند. (مثل document.write)

انواع روش های XSS

اکثر مردم فکر می کنند که این روش ها (Stored, Reflected, DOM XSS) سه روش متفاوت از XSS اند اما در واقع این سه با هم مشترکاتی دارند. ممکن است هر دو روش Stored, Reflected را در DOM داشته باشیم و حتی ممکن است ان دو در حالت Non-DOM رخ دهند. ممکن است درک این موضوع کمی گیج کننده باشد برای همین به نتایج تحقیقی که در سال ۲۰۱۲ صورت گرفت و نتیجه ی آن تعریف دو لغت جدید جهت سامان دهی به این دسته بندی ها بود،می پردازیم:

- XSS سمت سرور
- XSS سمت کلاینت

XSS سمت سرور

XSS سمت سرور زمانی رخ می دهد که یک کاربر نامطمئن داده هایی را وارد کند که شامل پاسخ های HTTP تولید شده توسط سرور باشد. این داده ها ممکن است از یک درخواست (requtest) یا جایی که داده ها را ذخیره می کند، باشد. با این توضیح، دو نوعِ Stored Server XSS و Reflected Server XSS را خواهیم داشت. در این مورد، ورودیِ آسیب پذیر در کدهای سمت سرور بوده و مرورگر تنها پاسخ ها (response) و سایر اسکریپت های معتبری که درون آن وجود دارد را رندر می کند.

XSS سمت کلاینت

XSS سمت کلاینت زمانی است که یک کاربر نامطمئن داده هایی را وارد کند که برای بروز رسانی DOM به وسیله فراخوانی جاوااسکریپت و به صورت ناامن استفاده می شود

فراخوانی جاوا اسکریپت زمانی ناامن است که بتوان آن را به عنوان یک کد جاوااسکریپت معتبر در درون DOM معرفی نمود. این داده ها ممکن است توسط DOM بیاید و یا از طرف سرور ارسال شود. (با استفاده از فراخوانی Ajax یا بارگذاری صفحه) همچنین داده ها ممکن است از یک درخواست (request) و یا محل ذخیره سازی داده ها روی کلاینت یا سرور بیاید. بنابراین هر دو نوع Reflected Client XSS و Stored Client XSS وجود دارند.

با توجه به تعریف جدید، تعریف DOM Based XSS تغییری نمی کند. بنابراین DOM XSS جزو XSSهای سمت کلاینت بوده و داده ها به جای سرور از جایی درون DOM تولید می شوند.

زمانی که یک برنامه آسیب پذیری XSS سمت سرور یا کلاینت دارد به این معناست که هم ممکن است از نوع Stored باشد و هم از نوع Reflected. بر اساس این تعریف می توان ماتریس زیر را رسم نمود

Where untrusted data is used

Data Persistence	XSS	Server	Client
	Stored	Stored Server XSS	Stored Client XSS
	Reflected	Reflected Server XSS	Reflected Client XSS

- DOM Based XSS is a subset of Client XSS (where the data source is from the DOM only)
- Stored vs. Reflected only affects the likelihood of successful attack, not the nature of vulnerability or the most effective defense

.

tamadonEH@gmail.com
https://twitter.com/tamadonEH

راه های پیشنهادی برای دفاع در برابر حملات XSS سمت سرور

XSS سمت سرور به دلیل وجود داده های نامطمئن در پاسخ HTML یا همان HTML Response است. ساده ترین و مطمئن ترین کاری که برای مقابله با حملات سمت سرور می توان انجام داد این است که :

- خروجی سمت سرور را بر اساس حساس-به-محتوا یا همان context-sensetive کدگذاری کنیم.

جزییات بیشتر در خصوص کدگذاری خروجی بر اساس حساس-به-محتوا در موضوع XSS (Cross Site Scripting) Prevention Cheat Sheet بیان شده است.

اعتبارسنجی ورودی ها یا پاک سازی داده ها نیز می تواند جلوی حملات XSS را بگیرد. اما پیاده سازی امن این روش نسبت به روش قبلی سخت تر است.

راه های پیشنهادی برای دفاع در برابر حملات XSS سمت کلاینت

XSS سمت کلاینت زمانی رخ می دهد که داده های نامطمئن جهت به روز رسانی DOM و به وسیله ی یک فراخوانی ناامن جاوااسکریپت مورد استفاده قرار گیرد. ساده ترین و مطمئن ترین کاری که برای مقابله با حملات سمت کلاینت می توان انجام داد این است:

- استفاده از APIهای امن جاوا اسکریپت

هر چند که، اکثر برنامه نویسان نمی دانند کدام API امن و کدام نا امن است. هیچ وقت فکر نکنید که متدهایی که در کتابخانه های محبوب و مورد استقبال جاوا اسکریپت وجود دارند، همگی امن اند. جزییات بیشتر در Unraveling some of the Mysteries around DOM Based XSS، توسط Dave Wichers

اگر از نا امن بودن یک متد در جاوااسکریپت اطلاع دارید، از متد دیگری که امن است، استفاده کنید. در صورتی که به هر دلیلی توانایی انجام این کار را ندارید، قبل از ارسال داده ها به متدهای نا امن جاوا اسکریپ، از کدگذاری خروجی و به صورت حساس-به-محتوا در مرورگر استفاده نمایید.

برای مشاهده جزییات بیشتر به DOM based XSS Prevention Cheat Sheet مراجعه فرمایید.

tamadonEH@gmail.com
https://twitter.com/tamadonEH

منابع

- Cross-site Scripting (XSS)
- Stored XSS
- Reflected XSS
- DOM Based XSS
- XSS (Cross Site Scripting) Prevention Cheat Sheet
- DOM based XSS Prevention Cheat Sheet

تاریخ ساخت: October 29, 2013 یا ۷ آبان ۱۳۹۲

تاریخ تحقیق: August 11, 2014 یا ۲۰ مرداد ۱۳۹۳

/* تصحیح این مقاله، چه در ترجمه و چه در مباحث علمی ، توسط شما دوستان باعث خوشحالی خواهد بود. لطفا آن را با tamadonEH@gmail.com مطرح نمایید.*/

برای مشاهده لیست مقالات کار شده توسط گروه ما به لینک زیر مراجعه فرمایید:

https://github.com/tamadonEH/list/blob/master/list.md

tamadonEH@gmail.com
https://twitter.com/tamadonEH