GO-TO CVE

CVE-2023-4357

XEE on WebBrowsers

Week 4

Author : Ali Soltani

سلام خوش اومدین به چهارمین قسمت از برنامه هفتگی GO-TO CVE . این هفته به بررسی آسیب‌پذیری Chromeمی‌پردازیم که من بعد از بررسی‌اش واقعا لذت بردم. این آسیب‌پذیری با CVSS Score: 8.8 وAV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H یک XEE است که در ادامه به بررسی کامل این آسیب‌پذیری می‌پردازیم. با ما همراه باشید ....

درباره Chrome

مرورگر گوگل کروم یکی از محبوب‌ترین و پرکاربردترین مرورگرهای وب در جهان است که توسط شرکت گوگل توسعه داده شده است. این مرورگر با استفاده از موتور رندرینگ Blink و موتور جاوااسکریپتV۸، تجربه‌ی کاربری سریع و پایداری را فراهم می‌کند. با این حال، مانند هر نرم‌افزار پیچیده‌ای، کروم نیز ممکن است دارای آسیب‌پذیری‌های امنیتی می‌باشد که می‌تواند امنیت کاربران را به خطر بیندازد.

آسیب‌پذیری‌های امنیتی و تاثیرات آن‌ها :

آسیب‌پذیری‌ها در مرورگر کروم می‌توانند به شکل‌های مختلفی بروز کنند، از جمله:

- حملات اجرای کد از راه دور (Remote Code Execution - RCE) مهاجمان می‌توانند با سوءاستفاده از یک آسیب‌پذیریRCE، کد مخرب خود را بر روی دستگاه قربانی اجرا کنند. این نوع حملات می‌تواند منجر به سرقت اطلاعات شخصی، دسترسی به محتوای فایل‌ها نصب بدافزارها و دسترسی غیرمجاز به سیستم شود.
- حملات تزریق کد (Code Injection) از طریق آسیب‌پذیری‌های تزریق کد، مهاجمان می‌توانند اسکریپت‌های مخرب را در صفحات وب قرار دهند، که به آن‌ها امکان دسترسی به اطلاعات حساس کاربران مانند کوکی‌ها و اطلاعات لاگین را می‌دهد.
- حملات عبور از مکانیزم‌های امنیتی (Security Mechanism Bypass) برخی از آسیب‌پذیری‌ها می‌توانند باعث شوند که مکانیزم‌های امنیتی مانند Same-Origin Policy (SOP)، Content Security Policy (CSP)، و دیگر قابلیت‌های محافظتی مرورگر به درستی عمل نکنند.

مکانیزم‌های امنیتی کروم :

مرورگر کروم به منظور حفاظت از کاربران خود، از چندین مکانیزم امنیتی پیشرفته استفاده می‌کند:

- Sandboxing این تکنیک باعث می‌شود که هر تب مرورگر در یک فرآیند جداگانه و ایزوله اجرا شود، بنابراین اگر یک تب مورد حمله قرار گیرد، تاثیر آن به سایر تب‌ها و سیستم عامل کاربر گسترش نمی‌یابد.
- Site Isolation این ویژگی، محتواهای هر سایت را در فرآیندهای جداگانه اجرا می‌کند تا دسترسی بین سایت‌ها محدود شود، حتی اگر یکی از سایت‌ها مخرب باشد.
- Automatic Updates گوگل کروم به‌طور خودکار به‌روزرسانی‌های امنیتی را دریافت و نصب می‌کند تا کاربران همواره از آخرین اصلاحات امنیتی بهره‌مند شوند.
- Safe Browsing این ویژگی کاربران را از سایت‌های مخرب و فیشینگ هشدار می‌دهد و آن‌ها را از بازدید این سایت‌ها باز می‌دارد.

اهمیت به‌روزرسانی مرورگرها :

با توجه به اینکه آسیب‌پذیری‌های جدید به طور مداوم کشف می‌شوند، اهمیت به‌روزرسانی مرورگر کروم برای محافظت از کاربران در برابر تهدیدات جدید بسیار بالاست. گوگل به‌طور مداوم به‌روزرسانی‌های امنیتی را منتشر می‌کند و کاربران باید مطمئن شوند که مرورگر خود را به‌روز نگه می‌دارند تا از این به‌روزرسانی‌ها بهره‌مند شوند. در نهایت، امنیت در مرورگر کروم یک موضوع پیچیده و چند لایه است که نیاز به توجه مداوم به جزئیات و همکاری کاربران و توسعه‌دهندگان عزیز دارد.

درباره آسیب پذیری

کتابخانه Libxslt به طور پیش‌فرض به عنوان کتابخانه XSL در مرورگرهای مبتنی بر WebKit مانند کروم، سافاری و غیره استفاده می‌شود. Libxslt اجازه می‌دهد که موجودیت‌های خارجی در اسناد که با روش document() XSL بارگذاری می‌شوند، وجود داشته باشند. این قابلیت می‌تواند به مهاجمان امکان دهد تا محدودیت‌های امنیتی را دور بزنند، از URLهای (http(s// به URLهای file// دسترسی پیدا کنند و به فایل‌ها دسترسی یابند. در شرایط عادی، مرورگرهای مبتنی بر WebKit از یک مکانیزم sandbox برای محدود کردن دسترسی‌ها استفاده می‌کنند. این sandbox به گونه‌ای طراحی شده است که اجرای کدها و دسترسی به منابع سیستم را محدود کند تا مهاجمان نتوانند به راحتی به فایل‌های حساس دسترسی یابند. با این حال، حتی با وجود این مکانیزم‌های امنیتی، امکان سوءاستفاده از Libxslt وجود دارد.

مثال‌هایی از آسیب‌پذیری :

ا.دسترسی به فایل /etc/hosts

- در دستگاه‌های iOS (سافاری/کروم)
- در دستگاه‌های macOS سافاری/کروم)
- در دستگاه‌های اندرویدی (کروم)
- در تلویزیون‌های سامسونگ (مرورگر پیش‌فرض)

مهاجم می‌تواند با بهره‌گیری از Libxslt و روش document()به فایل /etc/hosts دسترسی پیدا کند و آن را بخواند. این فایل حاوی اطلاعات شبکه‌ای و DNS است که می‌تواند برای حملات بیشتر مورد استفاده قرار گیرد.

۲.حمله با استفاده از no-sandbox

- در زمانی که از مرورگرهای Electron یا PhantomJS با گزینه -no-sandbox استفاده می‌شود، مهاجم می‌تواند به هر فایلی در هر سیستم عاملی دسترسی پیدا کند. این وضعیت بسیار خطرناک است زیرا هیچ محدودیتی برای دسترسی به فایل‌ها وجود ندارد و مهاجم می‌تواند به اطلاعات بسیار حساس و محرمانه دسترسی یابد.

۳.دسترسی به فایل /etc/shadow

- این فایل در سیستم‌های یونیکس و لینوکس قرار دارد و اطلاعات مربوط به کاربران سیستم را شامل می‌شود. اگرچه این فایل به طور پیش‌فرض حاوی رمزهای عبور نیست، اما شامل اطلاعاتی مانند نام کاربری، شناسه کاربر (UID)، شناسه گروه (GID)، دایرکتوری و شل پیش‌فرض است و فایل‌های مهم دیگر که در زیر لیستی از آنها وجود دارد.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE test [ <!ENTITY xxe SYSTEM "file:///etc/passwd"> ]>
<root>&xxe;</root>
```

- C:\Windows\System32\config\system
- /Users/[username]/Library/Keychains/login.keychain-db
- /proc/[pid]/cmdline

شرحی برآسیب‌پذیری و بررسی فناوری XSL

فناوری که درمورد آن صحبت می‌کنیم فناوری است XSL مخفف eXtensible Stylesheet Language است و یک زبان تخصصی مبتنی بر XML است که می‌تواند برای اصلاح یا بازیابی داده‌ها، چه در داخل XML و چه خارج از آن، مورد استفاده قرار گیرد. مرورگرکروم، XSL را پشتیبانی می‌کند و کتابخانه که از آن استفاده می‌کند، LibXSLT است. می‌توان این موضوع را با استفاده از تابع «system-property(«xsl:vendor) تأیید کرد، همان‌طور که در مثال زیر نشان داده شده است.

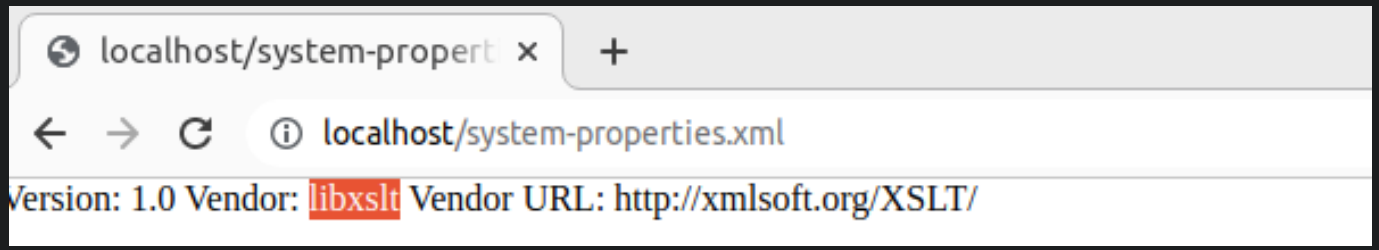
فایل system-properties.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="system-properties.xsl" type="text/xsl"?>
<root/>
```

فایل system-properties.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <p>
      Version: <xsl:value-of select="system-property('xsl:version')" /> <br />
      Vendor: <xsl:value-of select="system-property('xsl:vendor')" /> <br />
      Vendor URL: <xsl:value-of select="system-property('xsl:vendor-url')" />
    </p>
  </xsl:template>
</xsl:stylesheet>
```

خروجی فایل system-properties.xml در کروم



پس از بارگذاری فایل system-properties.xml روی یک وب‌سرور محلی و باز کردن آن در مرورگر کروم، خروجی به صورت بالا نمایش داده می‌شود.

کتابخانه LibXSLT و کاربردهای آن:

کتابخانه LibXSLT که برای اولین بار در ۲۳ سپتامبر ۱۹۹۹ منتشر شد، یکی از کتابخانه‌های قدیمی و پرکاربرد است. این کتابخانه به صورت پیش‌فرض در بسیاری از برنامه‌ها و سیستم‌ها از جمله مرورگرهای کروم و سافاری، Python، Oracle Database، PostgreSQL، PHP و بسیاری از دیگر برنامه‌ها مورد استفاده قرار می‌گیرد.

توضیحات فنی :

تابع unparsed-entity-uri()در XSLT برای بازیابی URI موجودیت‌های تعریف نشده استفاده می‌شود. این تابع مسیر کامل فایل مرتبط با موجودیت تعریف نشده را بازمی‌گرداند. در مثال زیر، ent به مسیر نسبی «?» اشاره می‌کند که وقتی توسط تابع گفته شده پردازش می‌شود، مسیر کامل فایل XML جاری را برمی‌گرداند.

۱. فایل get-location.xml

```
<?xml-stylesheet href="get-location.xsl" type="text/xsl"?>
<!DOCTYPE test [
    <!ENTITY ent SYSTEM "?" NDATA aaa>
]>
<test>
<getLocation test="ent"/>
</test>
```

۲. فایل get-location.xsl

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html"/>
  <xsl:template match="getLocation">
    <input type="text" value="{unparsed-entity-uri(@test)}" />
  </xsl:template>
</xsl:stylesheet>
```

مکانیزم عملکرد:

تعریف موجودیت در DTD

این خط کد موجودیت خارجیent را تعریف می‌کند که به مسیر نسبی «?» اشاره دارد.

```
<!ENTITY ent SYSTEM "?" NDATA aaa>
```

استفاده از تابع unparsed-entity-uri()

این خط کد در فایل XSL، مقدار کامل URI موجودیت «ent» را درون یک فیلد متنی قرار می‌دهد. تابع unparsed-entity-uri(@test) URI کامل موجودیت ent را بازمی‌گرداند که به مسیر فایل XML جاری اشاره دارد.

```
<input type="text" value="file:///path/to/your/xml/file/get-location.xml" />
```

خروجی نهایی

هنگامی که فایل XML و XSL روی یک سرور وب بارگذاری می‌شوند و در مرورگر کروم باز می‌شوند، نتیجه به صورت زیر نمایش داده می‌شود:

```
<input type="text" value="file:///path/to/your/xml/file/get-location.xml" />
```

این خروجی نشان می‌دهد که تابع unparsed-entity-uri() مسیر کامل فایل XML جاری را بازمی‌گرداند.

آسیب‌پذیری‌های مرتبط با LibXSLT

اگرچه مثال بالا خود به تنهایی یک آسیب‌پذیری نیست، اما وجود قابلیت‌های پیشرفته در XSL و کتابخانه LibXSLT می‌تواند به مهاجمان امکان دهد تا از این ویژگی‌ها به شکل سوءاستفاده کنند. به عنوان مثال، قابلیت‌های پردازش XML و XSLT می‌تواند برای حملات XXE (XML External Entity) مورد استفاده قرار گیرد که منجر به دسترسی غیرمجاز به فایل‌های سیستم یا انجام درخواست‌های شبکه غیرمجاز می‌شود.

نتیجه‌گیری:

تابع unparsed-entity-uri() در XSLT می‌تواند برای بازیابی مسیر کامل فایل‌ها مورد استفاده قرار گیرد.

XSL و بارگذاری محتوای خارجی:

تقریباً همه‌ی زبان‌های مبتنی بر XML، امکان بارگذاری یا نمایش فایل‌های خارجی را دارند، مشابه تگ iframe در HTML. از آنجا که XSL مبتنی بر XML است، استفاده از موجودیت‌های خارجی XXE) XML) باید اولین گزینه برای بررسی باشد.

مثالی از XXE

```
<?xml version="1.0"?>
<!DOCTYPE foo [
<!ENTITY xxe SYSTEM "file:///etc/passwd">
]>
<test>&xxe;</test>
```

XInclude

```xml
<?xml version="1.0"?>
<test xmlns:xi="http://www.w3.org/2001/XInclude">
  <xi:include href="file:///etc/passwd"/>
</test>
```

تگ‌های xsl:import و xsl:include در XSL

این تگ‌ها می‌توانند برای بارگذاری فایل‌ها به عنوان استایل‌شیت‌های XSL استفاده شوند،

```xml
<?xml version="1.0" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:include href="file:///etc/passwd"/>
  <xsl:import href="file:///etc/passwd"/>
</xsl:stylesheet>
```

تابع document() در XSL

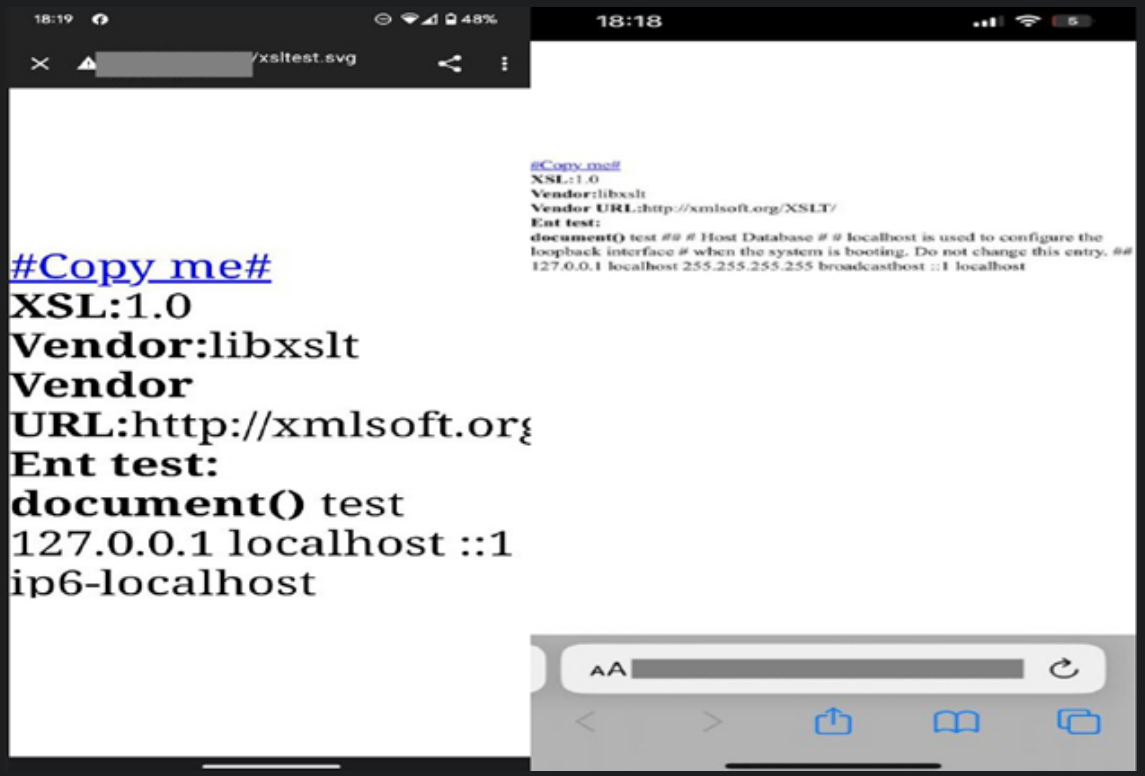
تابع document() در XSL می‌تواند برای بارگذاری فایل‌ها به عنوان اسناد XML استفاده شود.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet  version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <xsl:copy-of select="document('file:///etc/passwd')"/>
  </xsl:template>
</xsl:stylesheet>
```

ترکیب XXE و تابع document ()

یک فایل XSL ایجاد کردیم که تابع document() را با موجودیت‌های خارجی XML در فایل آرگومان ترکیب کرده است، سپس محتوای فایل XSL را به یک فایل XML وارد کردیم.

اگر فایل XML خود را از طریق یک URL HTTP از گوشی همراه خود باز کنید فایل /etc/hosts دستگاه خود را مشاهده میکنید .



این مثال‌ها نشان می‌دهند که چگونه قابلیت‌های پیشرفته XSL و XML می‌توانند برای بارگذاری محتوای راه دور و دسترسی به فایل‌های سیستم سوءاستفاده شوند. این قابلیت‌ها می‌توانند برای حملات XXE (XML External Entity) و دیگر حملات مشابه مورد استفاده قرار گیرند. و بالا بردن سطح امنیتی این بخش با تنظیمات مناسب، پردازش موجودیت‌های خارجی XML را محدود کنید. و درادامه به تمام تست ها را مشاهده می‌کنید.

Test Scenario	Accessible Files
Android + Chrome	/etc/hosts
iOS + Safari	/etc/group, /etc/hosts, /etc/passwd
Windows + Chrome	–
Ubuntu + Chrome	–
PlayStation 4 + Chrome	–
Samsung TV + Chrome	/etc/group, /etc/hosts, /etc/passwd

بهره‌برداری از آسیب‌پذیری

این لینک فایلی است که شامل تست‌هایی هستند که برای بررسی آسیب‌پذیری‌ها و آزمون توابع خواندن فایل در مرورگرها استفاده می‌شوند. اولین بخش یک صفحه HTML ساده است که شامل چند iframe است برای آزمون تمامی قابلیت‌های خواندن فایل و روش‌های آن است.
https:/swarm.ptsecurity.com/wp-content/uploads/2024/05/d34e0a53-libxslt.zip

توصیه های امنیتی و جلوگیری

- بروزرسانی مرورگرها اطمینان حاصل کنید که مرورگرها و سیستم‌های عامل شما همیشه به‌روز باشند. توسعه‌دهندگان مرورگرها به‌طور مداوم در حال اصلاح آسیب‌پذیری‌ها و بهبود مکانیزم‌های امنیتی هستند.
- استفاده از Sandbox: حتی در محیط‌های توسعه و آزمایش نیز از مکانیزم‌های sandbox استفاده کنید تا محدودیت‌های امنیتی به خوبی اعمال شوند.
- محدود کردن دسترسی به XSL وXML: اگر امکان‌پذیر است، دسترسی به قابلیت‌های پیشرفته XSL و XML را محدود کنید و تنها به موارد ضروری اجازه استفاده بدهید.
- بررسی و نظارت: به طور مداوم سیستم‌ها و نرم‌افزارهای خود را برای یافتن هرگونه فعالیت مشکوک و غیرمعمول بررسی کنید و نظارت فعال داشته باشید.

در نهایت، امنیت یک فرآیند پیوسته است و نیاز به توجه مداوم و اقدامات پیشگیرانه دارد. با درک بهتر از آسیب‌پذیری‌های موجود و اتخاذ تدابیر مناسب، می‌توان به میزان قابل توجهی از ریسک‌ها کاست و امنیت سیستم‌ها و داده‌ها را حفظ کرد

منابع

- https://chromereleases.googleblog.com/
- https://issues.chromium.org/
- https://www.cve.org/
- https://ubuntu.com/
- https://swarm.ptsecurity.com
- https://sploitus.com/
- https://nvd.nist.gov/
- https://cve.mitre.org
- https://x.com/
- https://www.tenable.com/


LIANGROUP.NET
021 9100 41 51 (300)