Ultimate DOM Based XSS Detection Scanner On Cloud

Nera W. C. Liu & Albert Yu
Paranoids
Yahoo
Agenda

• DOM XSS
• Our solution
• DEMO!
Who are we

What people think we are?

What we think we are?

What we actually are?
DO[r]M XSS
“XSS of the 3rd Kind”

- Does not rely on flaws in application containers
- Easier target for attacker
- Harder for defender to detect

http://www.vulnerable.site/welcome.html#foo<script
Static Analysis
If that’s not enough

• Anonymous functions

```javascript
(function()
  window.onload = function()
    alert(document.location)
  )()
```

• Dynamic loading

```html
<script async src="https://s.yimg.com/lq/lib/3pm/cs_0.2.js"></script>
```
The Chemistry of DOM
what is executable?
The Chemistry of DOM
what is executable?

<body class="hp vasq" onload="try{if(!google.j.b){document.getElementsByTagName('script').innerHTML,'is':''",'r':true,'sc':true}}();}</script><script data-url="/extern_chrome/ce4147df7c7403f4.js?bav=or.r_qf" id="ecs"><script if(google.y)google.y.first=[];}function b(a){window.setTimeout(function(c=document.createElement("script"));c.src=a;document.getElementById(xjsd).appendChild(c)}if(!google.xjs){window._=window._||{};}window._._DumpException=function(e){throw e};if(good Date().getTime());google.dljp('/xjs/_/js/k\x3dxjs.s.en_US.Qb9R7Hu1644.0/m\x3dc,sb,cr,jp,j OrT79TCTH70y_FIAAKw');google.xjs=1;}google.pmc="c:{"mcr":5},"sb":{"agen":false,"cgen":true,"client":"hp","dh":true,"ds":"","eqch":true,"fl":true,"fpol":true}"
DOM XSS DETECTION
What we want to do

• Analysis how “antigen” (untrusted data) get into our “body” (DOM)
• All arithmetic operations need to be overridden
• Enable to propagate through different context (HTML/CSS/JS)
Tainted Phantomjs

• Hacking the JavaScriptCore and WebKit engine by propagating the tainted signal during the javascript execution.
Source code of Tainted PhantomJS

Source – location.href

```c
JSValue jsLocationHref(ExecState* exec, JSValue slotBase, const Identifier&)
{
    JSLocation* castedThis = static_cast<JSLocation*>(asObject(slotBase));
    UNUSED_PARAM(exec);
    Location* imp = static_cast<Location*>(castedThis);
    JSValue result = jsString(exec, imp
    #ifdef JSC_TAINTED
    TaintedCounter* counter = TaintedC
    unsigned int tainted = counter->get
    result.setTainted(tainted);
    #endif
```

Sink – document.writeln

```c
EncodedJSValue JSC_HOST_CALL jsHTMLDocumentPrototypeFunctionWriteLn(ExecState* exec)
{
    JSValue thisValue = exec->hostThisValue();
    if (!thisValue.inherits(&JSHTMLDocument::s_info))
        return throwError(exec);
    JSHTMLDocument* castedThis = static_cast<JSHTMLDocument*>(asObject(thisValue));
    #ifdef JSC_TAINTED
    JSValue s = exec->argument(0);
    if (s.isString() && s.isTainted() > 0) {
        HTMLDocument* d1 = static_cast<HTMLDocument*>(castedThis->impl());
        d1->setTainted(s.isTainted());
```

Propagation – String.concat

```c
EncodedJSValue JSC_HOST_CALL stringProtoFuncConcat(E):
{
    JSValue thisValue = exec->hostThisValue();

    #ifdef JSC_TAINTED
    unsigned int tainted = 0;
    if (thisValue.isString() && thisValue.isTainted()) tainted = thisValue.isTainted();
    if (thisValue.inherits(&StringObject::s_info) && asStringObject(thisValue)->isTainted()) tainted = asStringObject(thisValue)->isTainted();
    if (thisValue.isObject()) {
        UString s = thisValue.toString(exec);
        if (s.isTainted()) tainted = s.isTainted();
    }
```
Flow Analysis
False alarm rate
= non-issues / issues reported
More you fix, the higher the false alarm rate
Our ultimate goal:

0 false alarm = 0% rate!
Benchmark and Comparisons

peak memory usage

The tainted logic performance hit is negligible!

The average peak memory usage
DEMO

http://www.youtube.com/watch?v=VU3YnAwc2Ag
Creative Commons

http://www.flickr.com/photos/58053205@N06/6999839463/
http://www.flickr.com/photos/67272961@N03/6123892769/
http://upload.wikimedia.org/wikipedia/commons/7/75/UCLA_dorm_room.JPG
http://www.flickr.com/photos/44124348109@N01/4682168995/
http://www.flickr.com/photos/15923063@N00/3150765076/
http://www.flickr.com/photos/88063120@N00/3529818070/
http://www.flickr.com/photos/free-stock/4817475664/
http://www.flickr.com/photos/78428166@N00/9604922912/
THANK YOU!