



# 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 3<sup>rd</sup> Kind”

```
1 <HTML>
2 <TITLE>Welcome!</TITLE>
3 Hi
4 <SCRIPT>
5 var pos=document.URL.indexOf("name")+5;
6 document.write(document.URL.substring(pos,document.URL.length));
7 </SCRIPT>
8 <BR>
9 Welcome to our system
10 </HTML>
```

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

See “DOM Based Cross Site Scripting or XSS of the Third Kind”. Amit Klein. 2005. <http://www.webappsec.org/projects/articles/071105.shtml>

<http://www.vulnerable.site/welcome.html#foo<script>



# Static Analysis





ory /zz/combo?yui:3.9.1/build/yui/yui-min.js&os/mit/media/p/common/rmp-min-1675959.js&os/mit/m

v-source:<https://www.google.com>



# If that's not enough

## • Anonymous functions

```
> (function(){  
  window.onload = function(){alert(document.location)}  
})()  
undefined  
window.onload  
function (){alert(document.location)}  
>
```

## • Dynamic loading

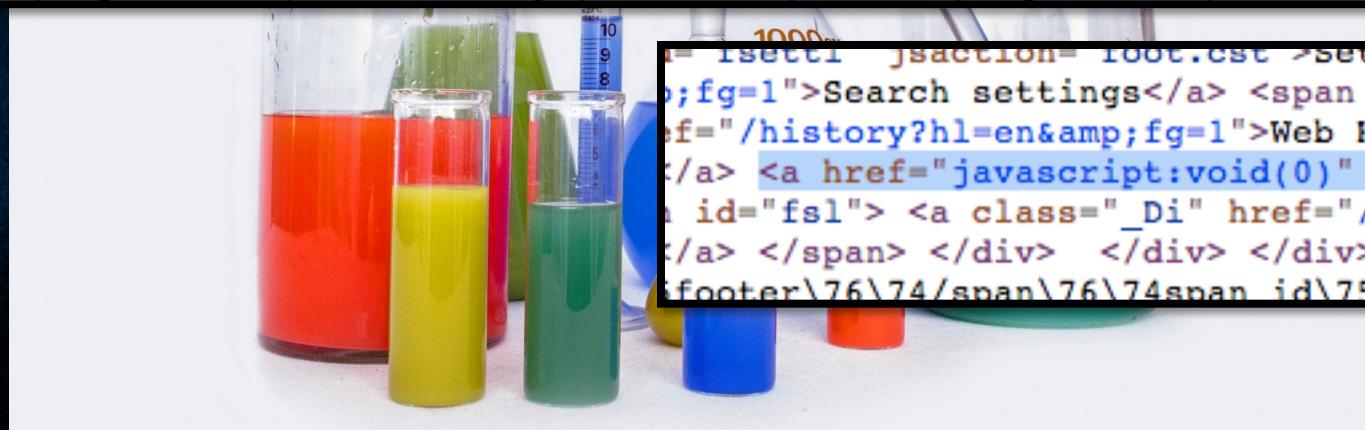
```
<script async src="https://s.yimg.com/lq/lib/3pm/cs_0.2.js"></script>  
► <script type="text/javascript">...</script>  
► <script type="text/javascript">...</script>  
► <script>...</script>  
► <script>...</script>  
► <script type="text/javascript">...</script>  
<style type="text/css">  
  </style>  
<style id="tmp-css"></style>  
<style type="text/css">.jsenabled .lzbgl{background:none!important}</style>  
► <script>...</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.images)new Image().src='/images/nav_logo170.png'" alink="#"><div id="nocs" style="display:none;position:absolute"><div><script>'n':'pc','i':'cst','h':document.getElementById('cst').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(){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(google.Date().getTime());}google.dljp('/xjs/_/js/k\x3dxjs.s.en_US.Qb9R7Hu1644.0/m\x3dc,sb,cr,jp,jp,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}}
```

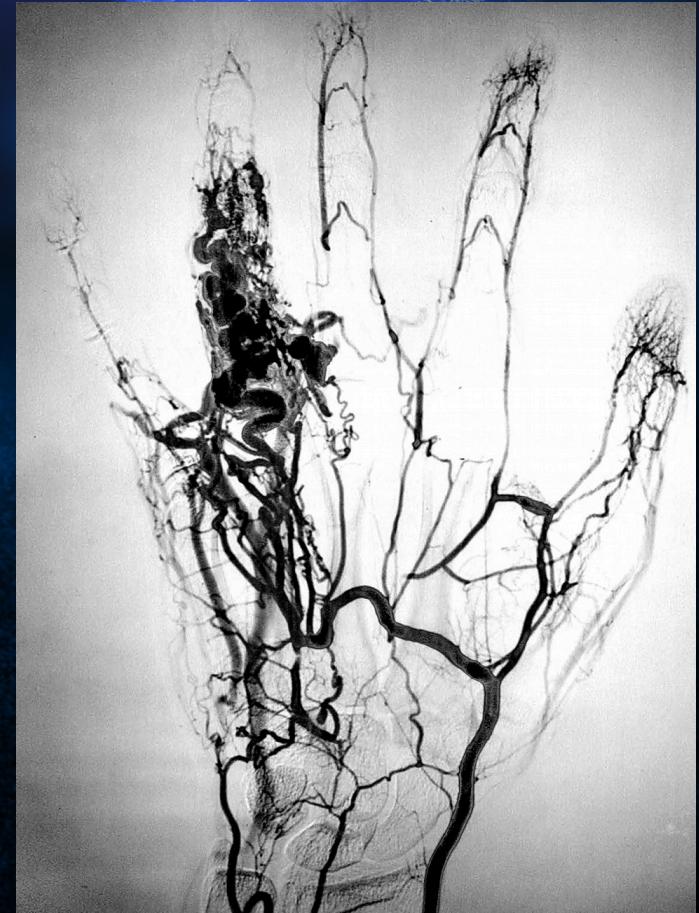




# DOM XSS DETECTION

# What we want to do

- Analysis how “antigen” (untrusted data) get into our “body” (DOM)



# char\*

```
1 apple = "yummy";
2 apple.tainted = True;
3 apple_pie = apple + "pie";
4 console.log(apple_pie.tainted);
```

- 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

```
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->href);
    #ifdef JSC_TAINTED
        TaintedCounter* counter = TaintedCounter::create();
        unsigned int tainted = counter->getTaintedCount();
        result.setTainted(tainted);
    #endif
}
```

## Propagation – String.concat

```
EncodedJSValue JSC_HOST_CALL stringProtoFuncConcat(ExecState* exec)
{
    JSValue thisValue = exec->hostThisValue();

    #ifdef JSC_TAINTED
        unsigned int tainted = 0;
        // https://git.corp.yahoo.com/heraliu/phantomjs-tainted/issues/35
        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();
        }
    #endif
}

EncodedJSValue JSC_HOST_CALL jsHTMLDocumentPrototypeFunctionWriteLn(ExecState* exec)
{
    JSValue thisValue = exec->hostThisValue();
    if (!thisValue.inherits(&JSHTMLDocument::s_info))
        return throwVMTypeError(exec);
    JSHTMLDocument* castedThis = static_cast<JSHTMLDocument*>(asObject(thisValue));
    #ifdef JSC_TAINTED
        // https://git.corp.yahoo.com/heraliu/phantomjs-tainted/issues/35
        JSValue s = exec->argument(0);
        if (s.isString() && s.isTainted() > 0) {
            HTMLDocument* d1 = static_cast<HTMLDocument*>(castedThis->impl());
            d1->setTainted(s.isTainted());
        }

        TaintedStructure trace_struct;
        trace_struct.taintedno = s.isTainted();
        trace_struct.internalfunc = "jsHTMLDocumentPrototypeFunctionWriteLn";
        trace_struct.jsfunc = "document.writeln";
        trace_struct.action = "sink";
    #endif
}
```

# Flow Analysis



# Flow Analysis

```
5 <div id="o"></div>
6 <script>
7 var x = location.search;
8 var o = document.getElementById("o");
9 o.innerHTML = unescape(x.substring(7));
10 </script>
11
```

```
[Fri, 07 Mar 2014 08:55:56 GMT] [TPJS] [CONSOLE] [TRACE] [object HTMLDivElement] id:o tainted:2
[Fri, 07 Mar 2014 08:55:56 GMT] [TPJS] [CONSOLE] [TRACE] =====
[Fri, 07 Mar 2014 08:55:56 GMT] [TPJS] [CONSOLE] [TRACE] source,1,jsLocationHref,location.href,about:blank
[Fri, 07 Mar 2014 08:55:56 GMT] [TPJS] [CONSOLE] [TRACE] source,2,jsLocationSearch,location.search,?1394182556085&
[Fri, 07 Mar 2014 08:55:56 GMT] [TPJS] [CONSOLE] [TRACE] source,3,jsLocationHref,location.href,http://tinyknight.c
[Fri, 07 Mar 2014 08:55:56 GMT] [TPJS] [CONSOLE] [TRACE] =====
[Fri, 07 Mar 2014 08:55:56 GMT] [TPJS] [CONSOLE] [TRACE] sink,2,setJSHTMLElementInnerHTML,HTMLElement.innerHTML,2556085&
[Fri, 07 Mar 2014 08:55:56 GMT] [TPJS] [CONSOLE] [TRACE] =====
[Fri, 07 Mar 2014 08:55:56 GMT] [TPJS] [CONSOLE] [TRACE] propagate,2,stringProtoFuncSubstring,String.substring,?1394182556085&
[Fri, 07 Mar 2014 08:55:56 GMT] [TPJS] [CONSOLE] [TRACE] propagate,2,globalFuncUnescape,unescape,2556085&
[Fri, 07 Mar 2014 08:55:56 GMT] [TPJS] [CONSOLE] [TRACE] propagate,3,stringProtoFuncToLowerCase,String.LowerCase,http://tinyknight.c
[Fri, 07 Mar 2014 08:55:56 GMT] [TPJS] [CONSOLE] [TRACE] propagate,3,stringProtoFuncReplace::exec,RegExp.exec,http://tinyknight.c
[Fri, 07 Mar 2014 08:55:56 GMT] [TPJS] [CONSOLE] [TRACE] propagate,3,re
[Fri, 07 Mar 2014 08:55:56 GMT] [TPJS] [CONSOLE] [TRACE] propagate,3,JS
```

Sources	Propagate	Sinks
<ul style="list-style-type: none"><li>source id: 1</li><li>source internal func: jsLocationHref</li><li>source js func: location.href</li></ul>		
<ul style="list-style-type: none"><li>source id: 2</li><li>source internal func: jsLocationSearch</li><li>source js func: location.search</li></ul>	<ul style="list-style-type: none"><li>propagate id: 2</li><li>propagate internal func: stringProtoFuncSubstring</li><li>propagate js func: String.substring</li></ul> <ul style="list-style-type: none"><li>propagate id: 2</li><li>propagate internal func: globalFuncUnescape</li><li>propagate js func: unescape</li></ul>	<ul style="list-style-type: none"><li>sink id: 2</li><li>sink internal func: setJSHTMLElementInnerHTML</li><li>sink js func: HTMLElement.innerHTML</li></ul>
<ul style="list-style-type: none"><li>source id: 3</li><li>source internal func: jsLocationHref</li><li>source js func: location.href</li></ul>	<ul style="list-style-type: none"><li>propagate id: 3</li><li>propagate internal func: stringProtoFuncToLowerCase</li><li>propagate js func: String.lowercase</li></ul> <ul style="list-style-type: none"><li>propagate id: 3</li><li>propagate internal func: stringProtoFuncReplace::exec</li><li>propagate js func: RegExp.exec</li></ul> <ul style="list-style-type: none"><li>propagate id: 3</li><li>propagate internal func: regExpProtoFuncExec</li><li>propagate js func: RegExp.exec</li></ul> <ul style="list-style-type: none"><li>propagate id: 3</li><li>propagate internal func: JString</li><li>propagate js func: String._manipulation</li></ul>	

# Usable Security

False alarm rate

= non-issues / issues reported

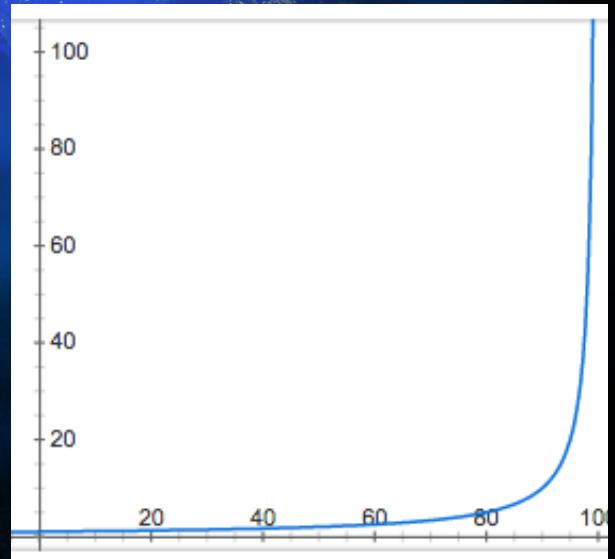
More you fix, the higher the  
false alarm rate

Our ultimate goal:

0 false alarm = 0% rate!

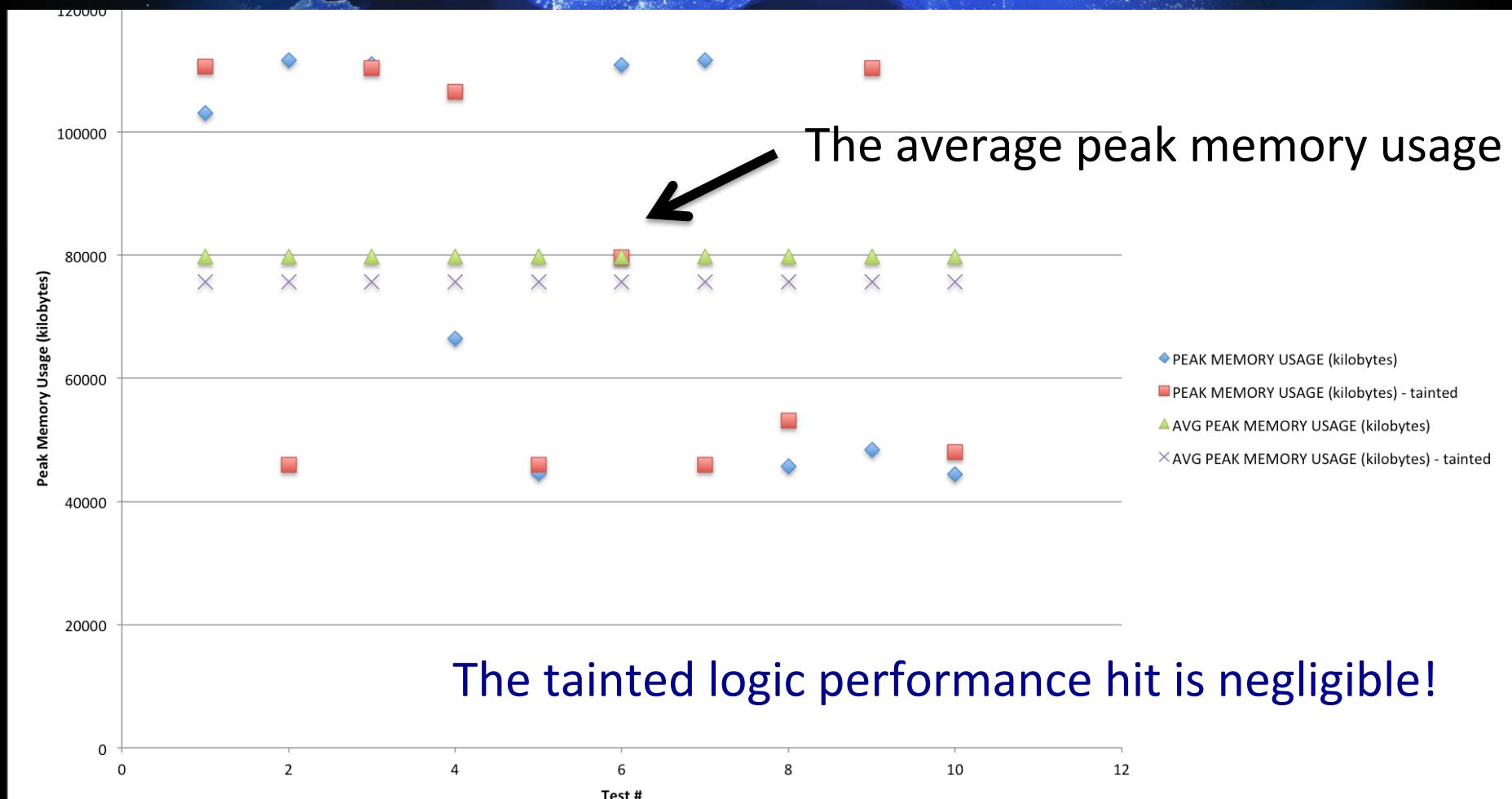
```
[TPJS] [RESULT] document.tainted? true
[TPJS] [RESULT] document.onAlert? false
[TPJS] [RESULT] document.domxss.vulnerable?
```

```
[TPJS] [RESULT] document.tainted? true
[TPJS] [RESULT] document.onAlert? true
[TPJS] [RESULT] document.domxss.vulnerable? true
```



# Benchmark and Comparisons

## 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://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://en.wikipedia.org/wiki/File:Angiome\\_annulaire.JPG](http://en.wikipedia.org/wiki/File:Angiome_annulaire.JPG)
- <http://www.flickr.com/photos/free-stock/4817475664/>
- <http://www.flickr.com/photos/78428166@N00/9604922912/>



THANK YOU!

