A Dozen Years of Shellphish

From DEFCON to the Cyber Grand Challenge
Zardus
HEX on the beach
Northeastern and Boston University

UC Santa Barbara

antonoob
zardus
cavedon
spermachine
odo
giullo
irish
voltaire
 sicko
virus
weaver
reyammer
nullptr
zanardi
marco
adamd

TU Vienna

balzaroth
pizzaman

NORTHEASTERN AND BOSTON UNIVERSITY

TU VIENNA

engiman
pizzaman

SIMULATION 2006 - 2011
SLA--
BEFORE: practice!
DURING: hack!
AFTER: party!
BEFORE: write tools!
DURING: debug tools!
AFTER: plan new tools!
DEFCON®

x86

SYSCALL IDS

0% SLA!

FAIL

-2 Hackers
DEFCON

2010

REMINISCING

x86

DTRACE

-2 Hackers
0% SLA!
-4 Hackers
DEFCON

0% SLA!
https://github.com/shellphish/puppeteer
https://github.com/reyammer/shellnoob
https://github.com/acama/xrop
https://github.com/zardus/preeny
https://github.com/zardus/memcurses
https://github.com/zardus/idalink

https://github.com/angr/angr
The next generation of binary analysis.
2005 Hex-Rays was founded
2007 Hex-Rays Decompiler 1.0
2009 Hex-Rays IDA 5.5
2011 Hex-Rays IDA 6.1
2013 Hex-Rays IDA 6.4
2015 WHY???????
- Making binary analysis techniques usable!

- Open-source: https://github.com/angr/angr (star it!)

- Written in Python!
  - installable with `pip install angr`
  - interactive shell (using ipython)
  - GUI

- Architecture independent
  - x86, amd64, mips, mips64, arm, aarch64, ppc, ppc64
  - ELF, CGC, PE
angr

- Binary Loader
- Static Analysis Routines
- Symbolic Execution Engine

- Control-Flow Graph
- Data-Flow Analysis
- Value-Set Analysis

WTF
"How do I trigger path X or condition Y?"

1. Interpret the application to identify symbolic variables.
   
   ```c
   read(0, &x, 4);
   ```

2. Track "constraints" on symbolic variables.

   ```c
   if (x < 100 && x >= 10)
   ```

3. When the required condition is triggered, "concretize" to obtain a possible input.

   ```c
   if (x < 100 && x >= 10)
   puts("You win!");
   ```
KEEP CALM ITS TIME FOR A DEMO!
if (prefilter(input))
    error("PLEASE CHECK AGAIN!")
else
    puts("INPUT OK")

do_lot_of_spikey_crypto(input, result)
if (strcmp(result, "growfish") == 0)
    printf("FLAG: %s\n", ...)
else
    puts("INPUT IS NOT GOOD ENOUGH.")
Program

Security policies

Symbolic execution engine

Security policy checker

Exploit
Security policies

- Arbitrary memory accesses
- Controllable stack pointer
- Controllable instruction pointer

CGC Qualifiers Goal: CRASH
CGC Finals Goal: EXPLOIT
~0% SLA!
56% SLA!
Looking toward the future!

- CGC finals!
  - $2,000,000 top prize
  - Shellphish VS Shellphish CRS

- CTFs
  - Can our tools finally help us?

- What do we do with our $750,000?
References:
this presentation: https://goo.gl/xFYItI
hitcon ent presentation: http://goo.gl/3ulxRa
angr: https://github.com/angr/angr
contact: yans@cs.ucsb.edu, @Zardus
<table>
<thead>
<tr>
<th>Year</th>
<th>Artist</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004</td>
<td>zanardi</td>
</tr>
<tr>
<td></td>
<td>VOID</td>
</tr>
<tr>
<td></td>
<td>NULLPTR</td>
</tr>
<tr>
<td></td>
<td>balzaroth</td>
</tr>
<tr>
<td></td>
<td>sicko</td>
</tr>
<tr>
<td></td>
<td>irish</td>
</tr>
<tr>
<td>2005</td>
<td>engiman</td>
</tr>
<tr>
<td></td>
<td>marco</td>
</tr>
<tr>
<td></td>
<td>virus</td>
</tr>
<tr>
<td></td>
<td>beetal</td>
</tr>
<tr>
<td></td>
<td>weaver</td>
</tr>
<tr>
<td>2006</td>
<td>collin</td>
</tr>
<tr>
<td></td>
<td>pizzaman</td>
</tr>
<tr>
<td>2008</td>
<td>odo</td>
</tr>
<tr>
<td></td>
<td>adamd</td>
</tr>
<tr>
<td></td>
<td>giullo</td>
</tr>
<tr>
<td></td>
<td>voltaire</td>
</tr>
<tr>
<td></td>
<td>bboe</td>
</tr>
</tbody>
</table>