

What the Fuzz

Cornelius Aschermann

Researcher at Ruhr University Bochum

Verification & Automated Bug Finding

Security Consultant

 @is_eqv

 github.com/eqv

 cornelius.aschermann@rub.de

Sergej Schumilo


Researcher at Ruhr University Bochum

Automated Bug Finding & Everything Low Level

Security Consultant

 @ms_s3c

 github.com/schumilo

 sergej.schumilo@rub.de



Manual Analysis

(doesn't scale that well)

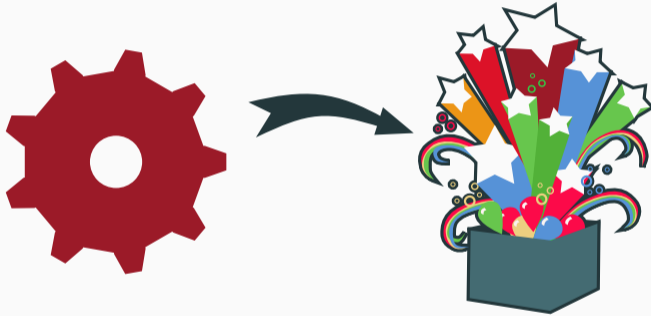


Verification

~~Verification~~

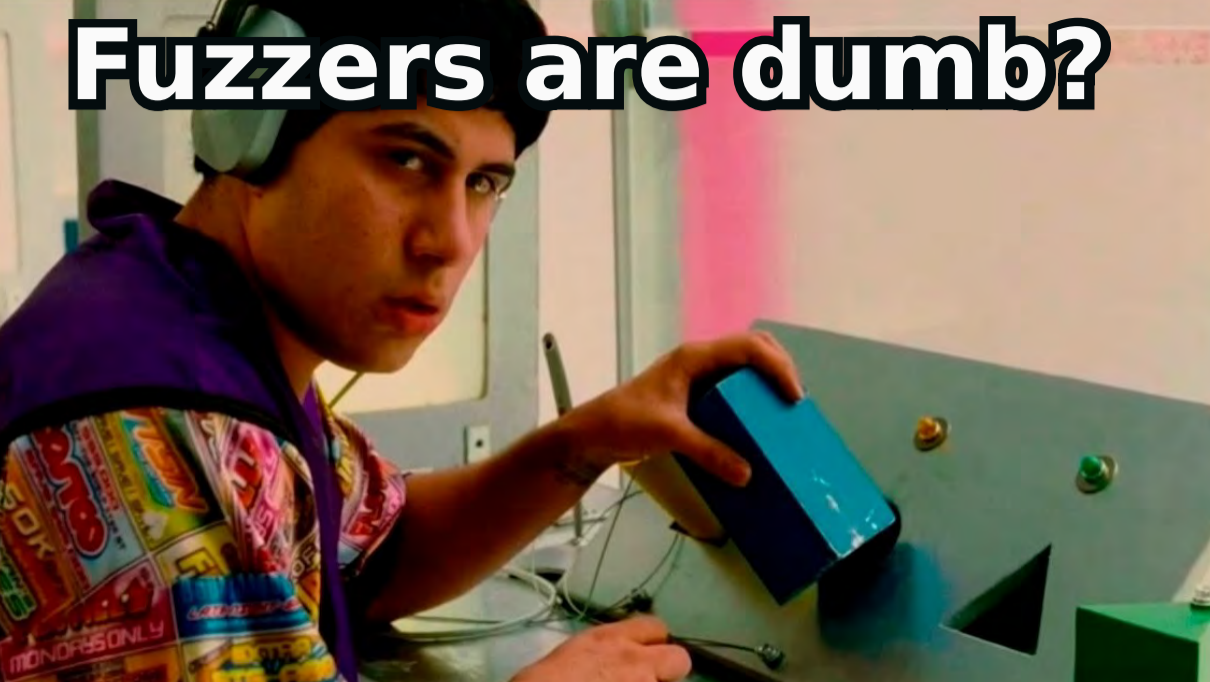
Fuzzers



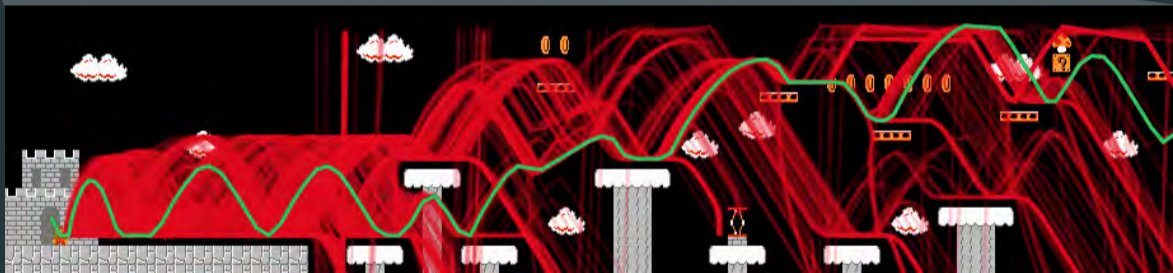


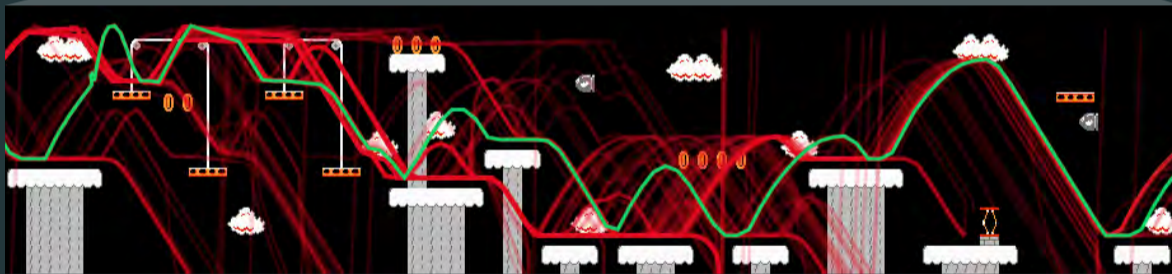


Fuzzers are dumb?



Demo







8 min

How do Fuzzer Work?

How do Fuzzer Work?



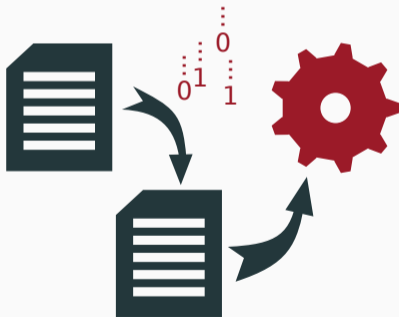
How do Fuzzer Work?



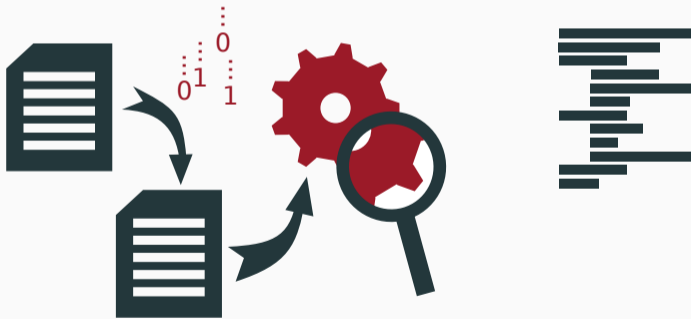
How do Fuzzer Work?



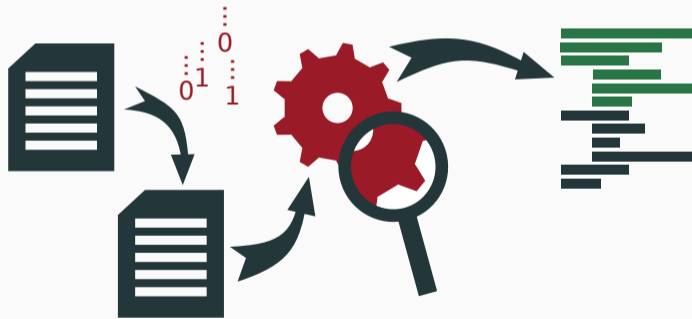
How do Fuzzer Work?



How do Fuzzer Work?



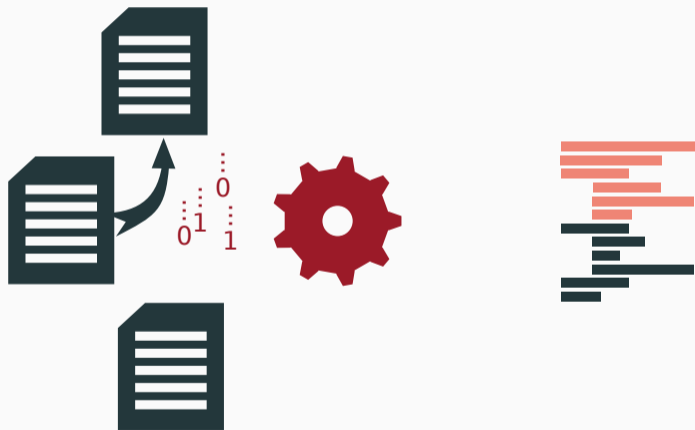
How do Fuzzer Work?



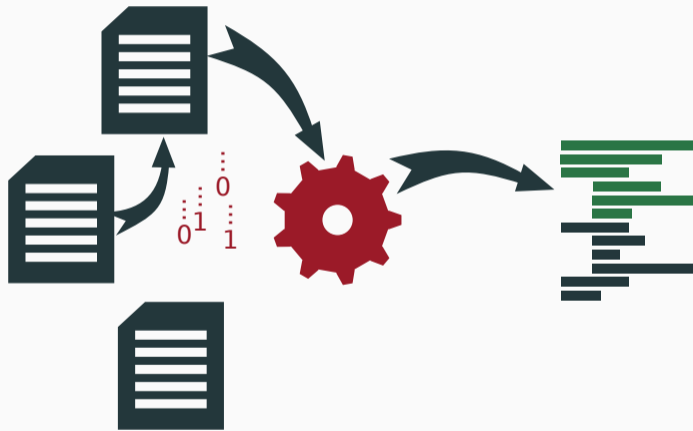
How do Fuzzer Work?



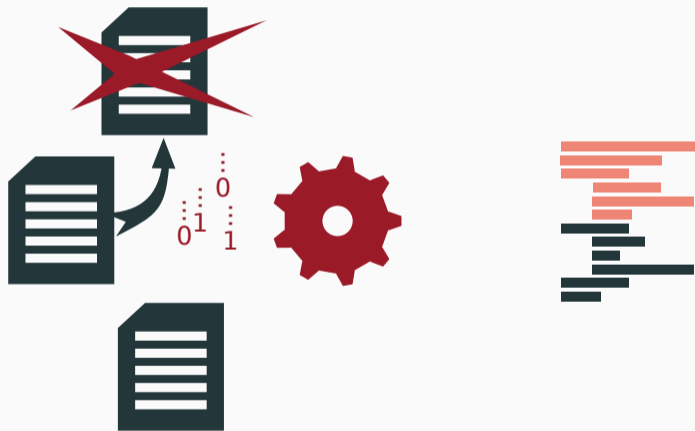
How do Fuzzer Work?



How do Fuzzer Work?



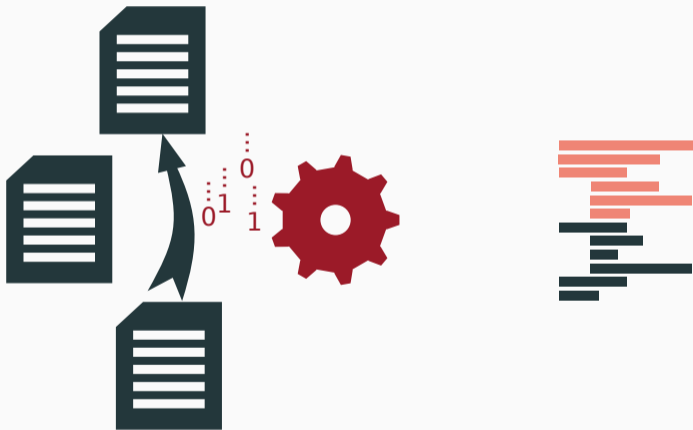
How do Fuzzer Work?



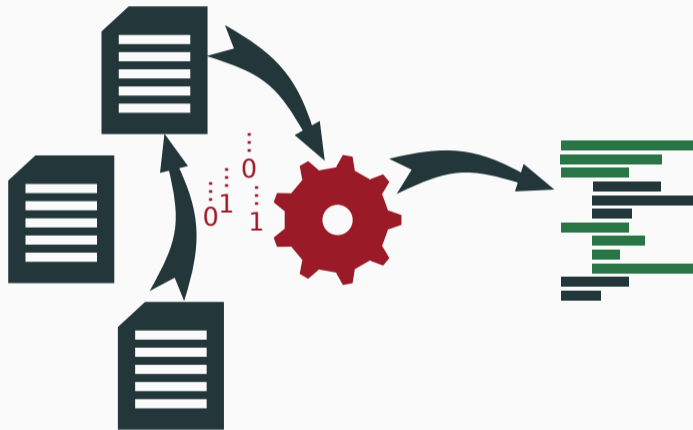
How do Fuzzer Work?



How do Fuzzer Work?



How do Fuzzer Work?



How do Fuzzer Work?



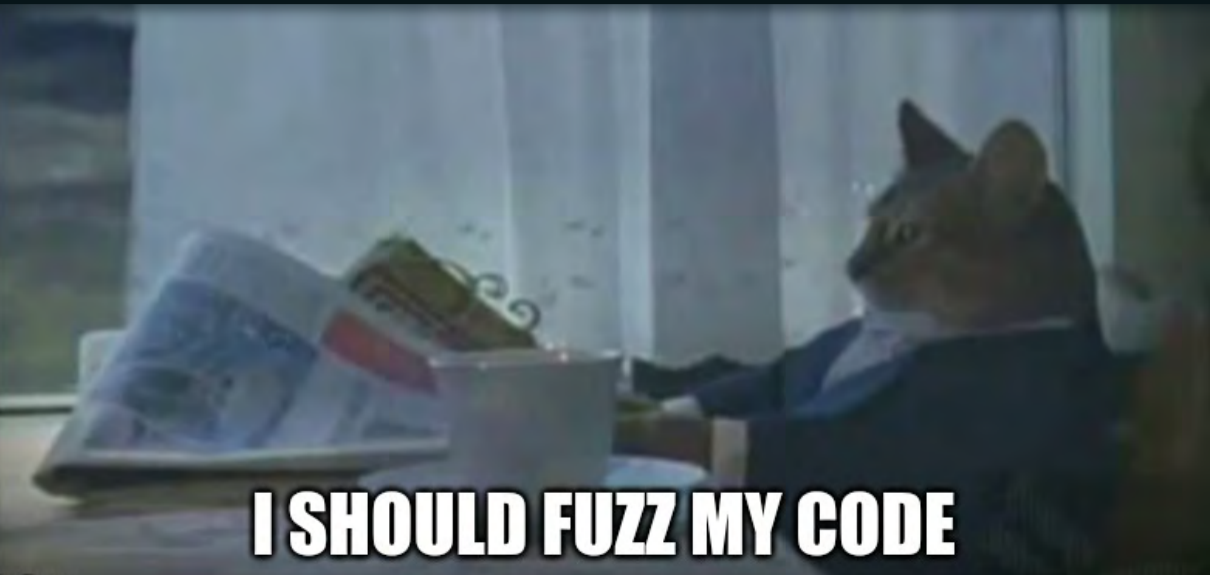
Fuzzers

Artificial Intelligence for Testcase Generation

Key Takeaways:



Key Takeaways:



I SHOULD FUZZ MY CODE

The Rest of this Talk



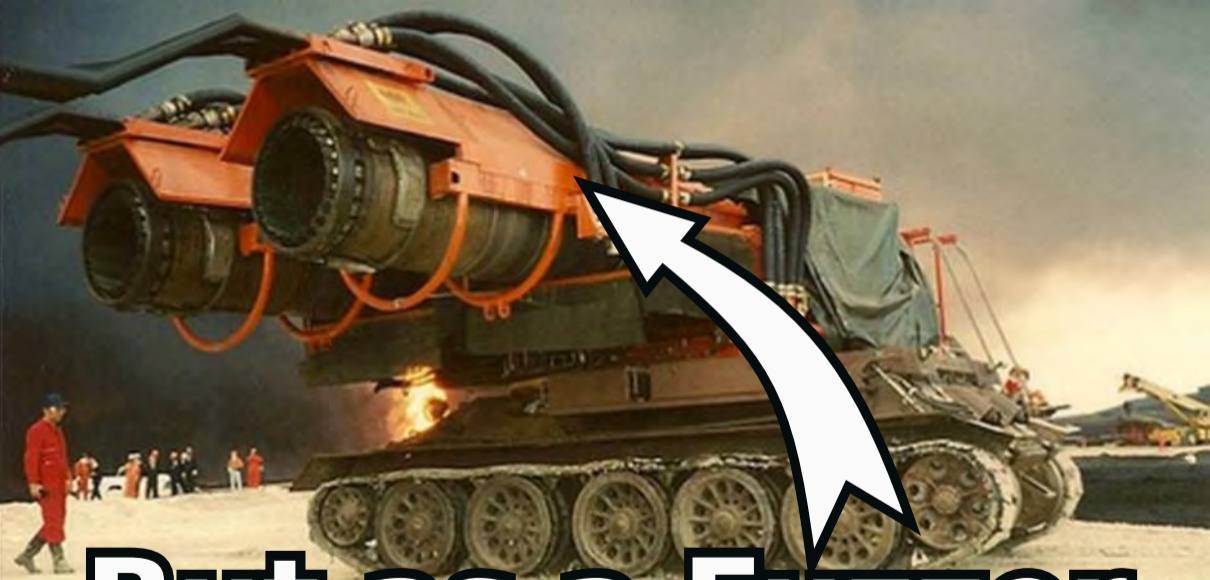
The Rest of this Talk

Past Research



Goal





But as a Fuzzer

Objective C

C

Pascal

Haskell

C++

x86

Ada

Go

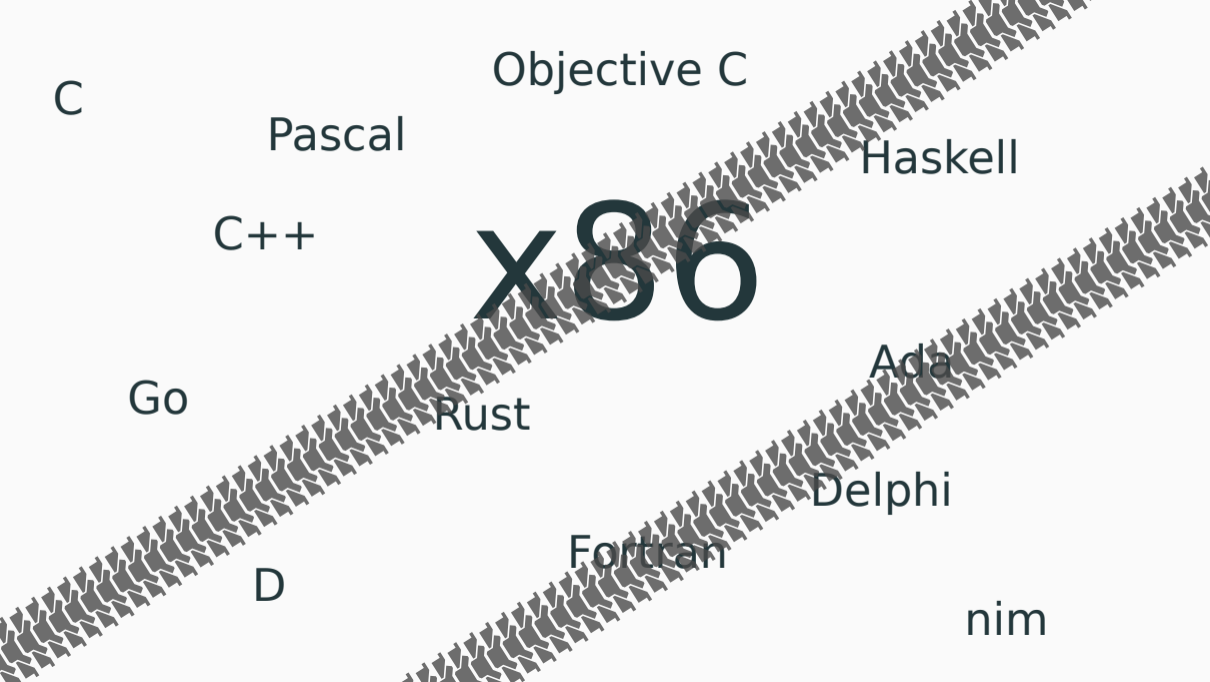
Rust

Delphi

D

Fortran

nim



C

Objective C

Pascal

Haskell

C++

X86

Go

Rust

Ada

Delphi

Fortran

D

nim



Mac

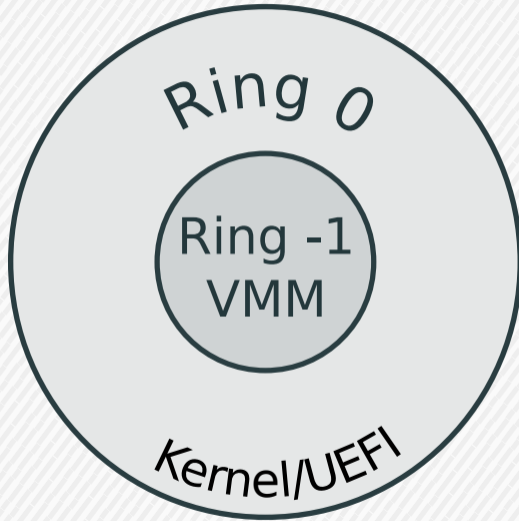




Mac

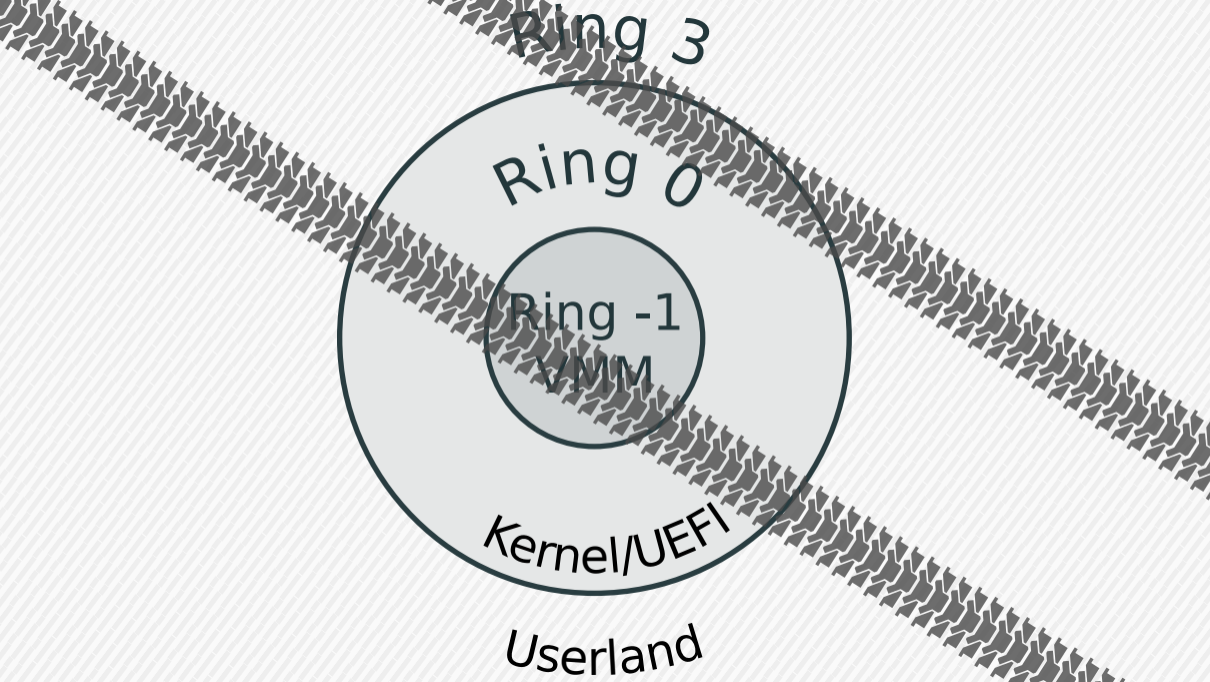


Ring 3

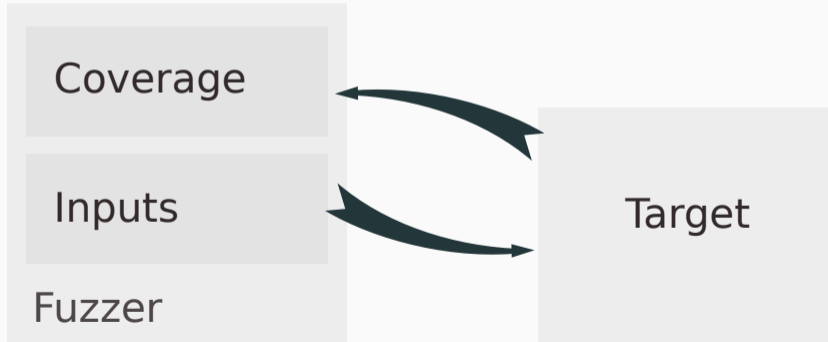


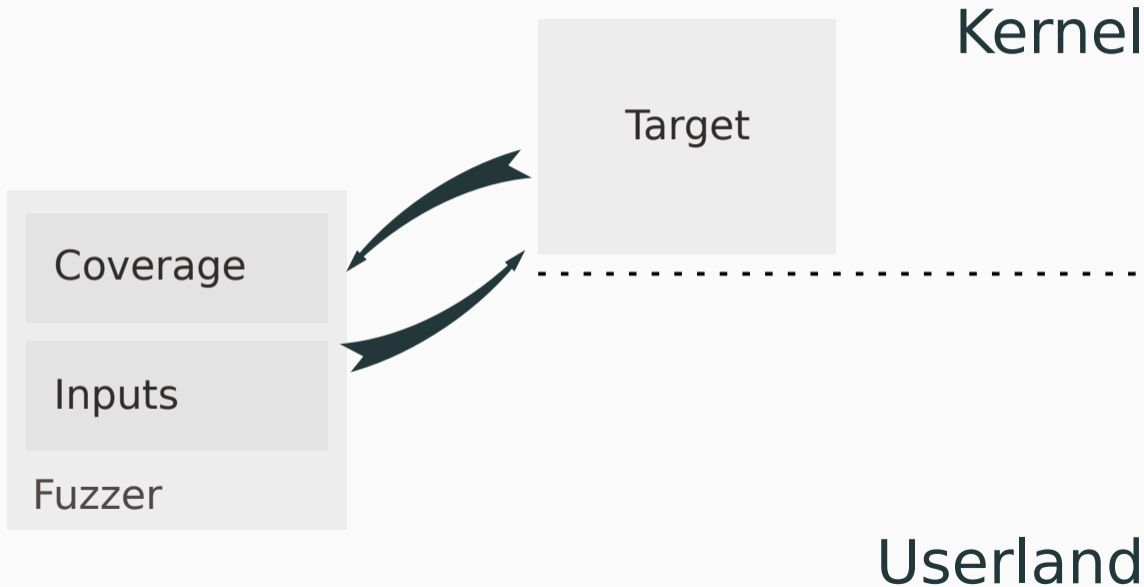
Kernel/UEFI

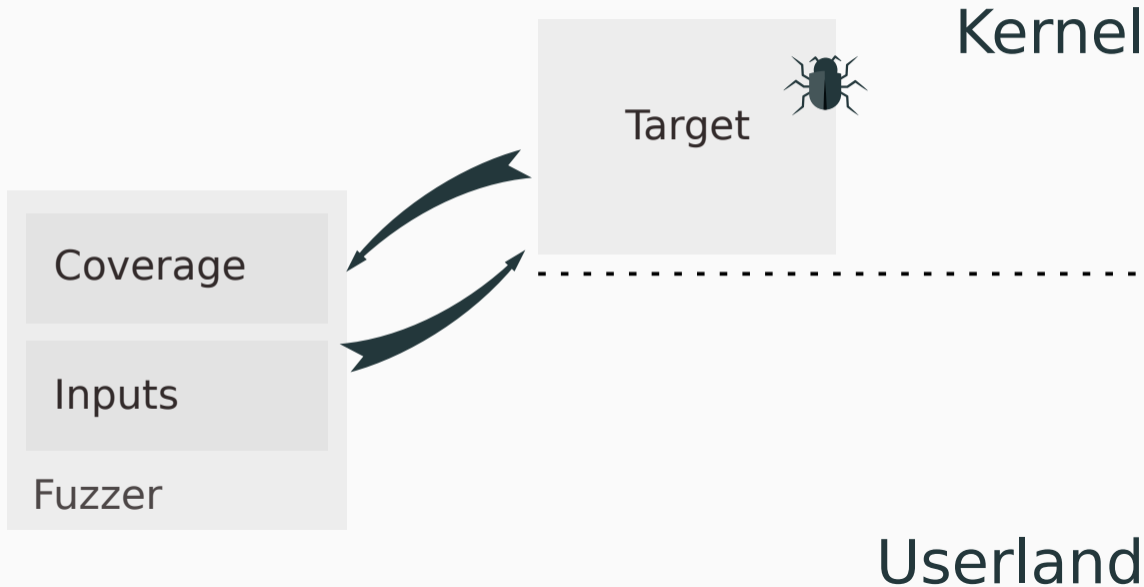
Userland



Nyx







Kernel

Target



Coverage

Inputs

Fuzzer

Userland

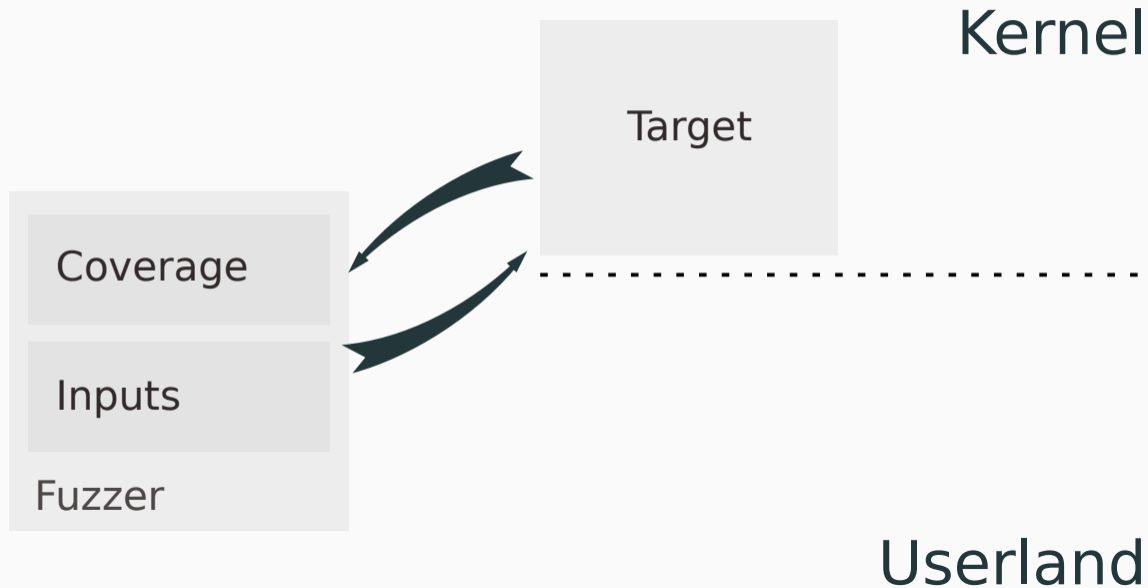
BOOM

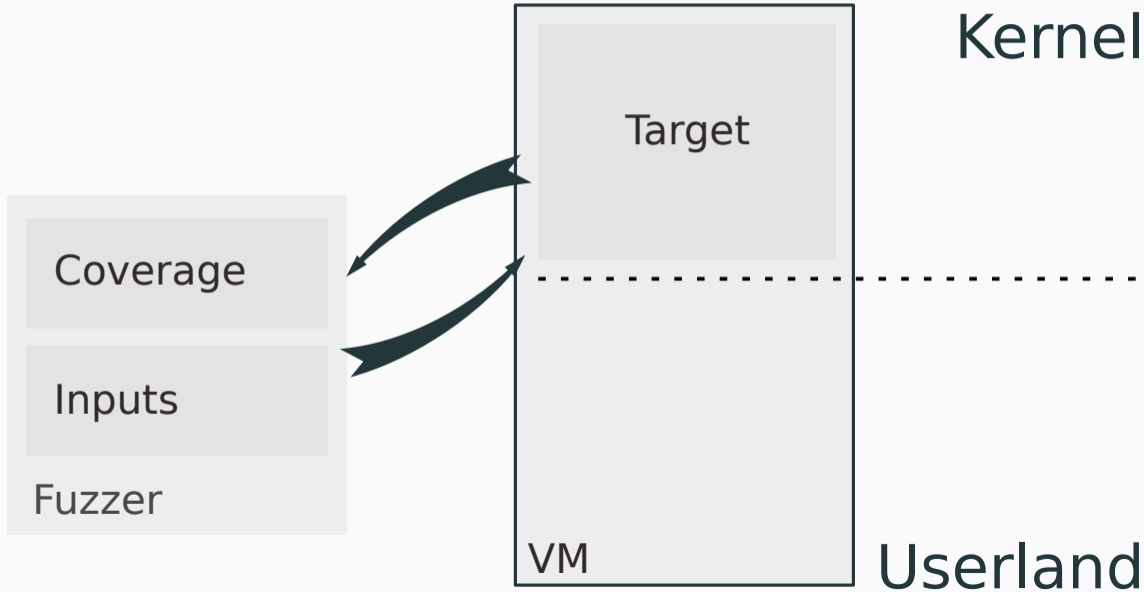


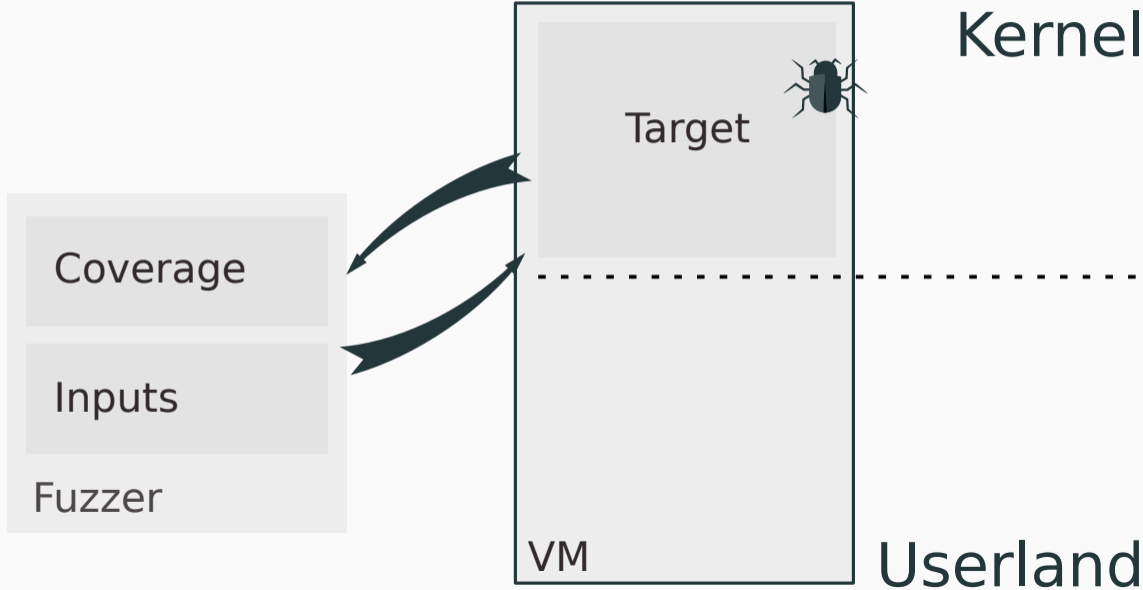
Input

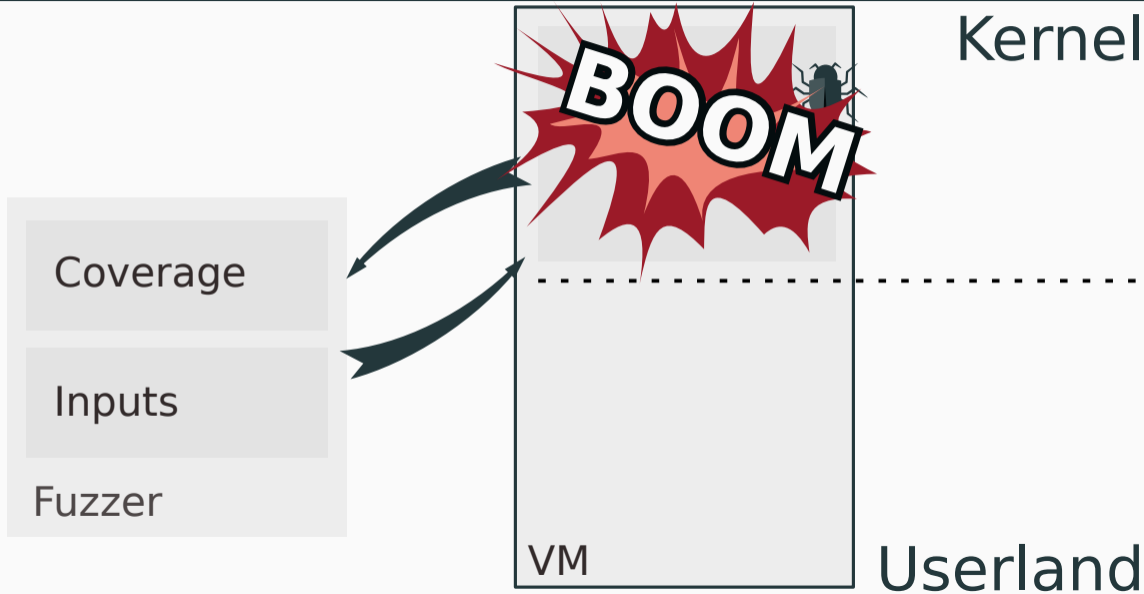
Fuzzer

Userland









Kernel

VM

Userland

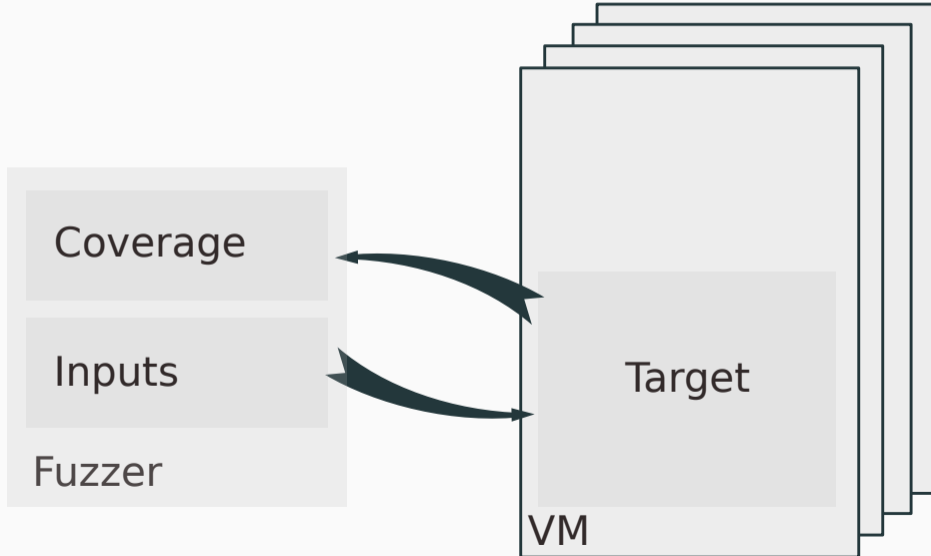
Target in a VM:

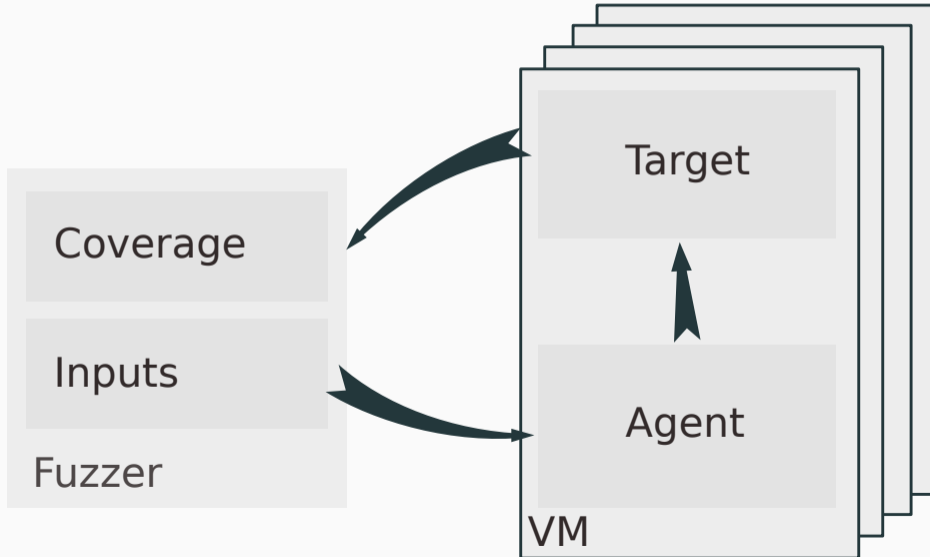
+ Fault Tolerance

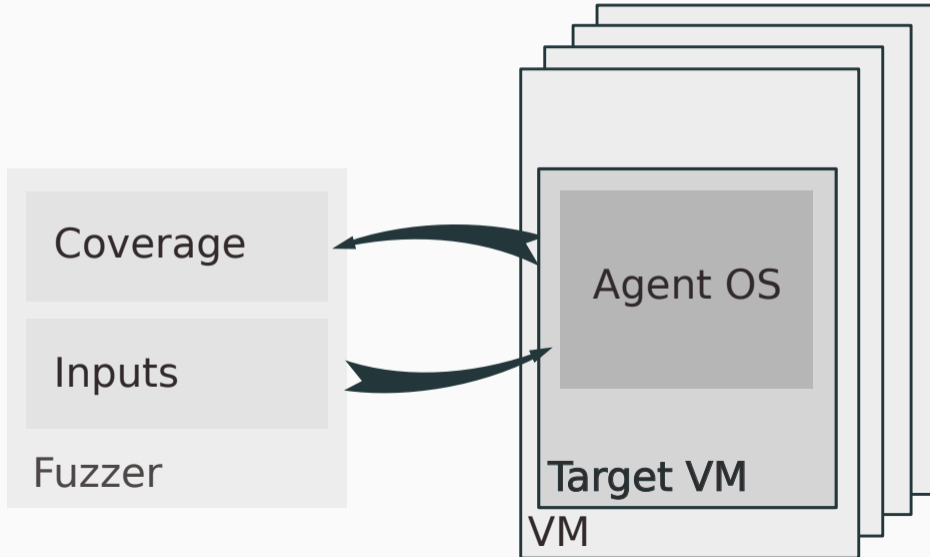
Target in a VM:

+ Fault Tolerance

+ Parallelization







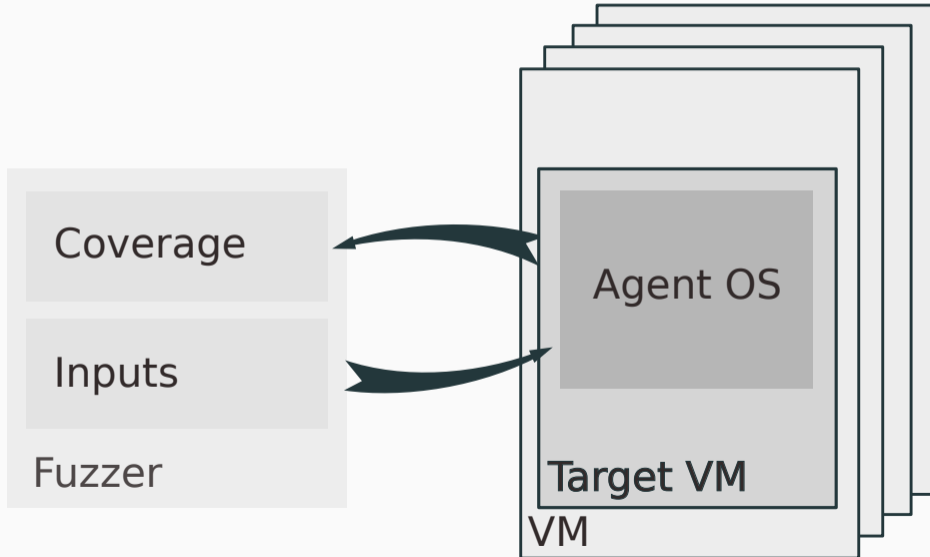
Key Takeaways:

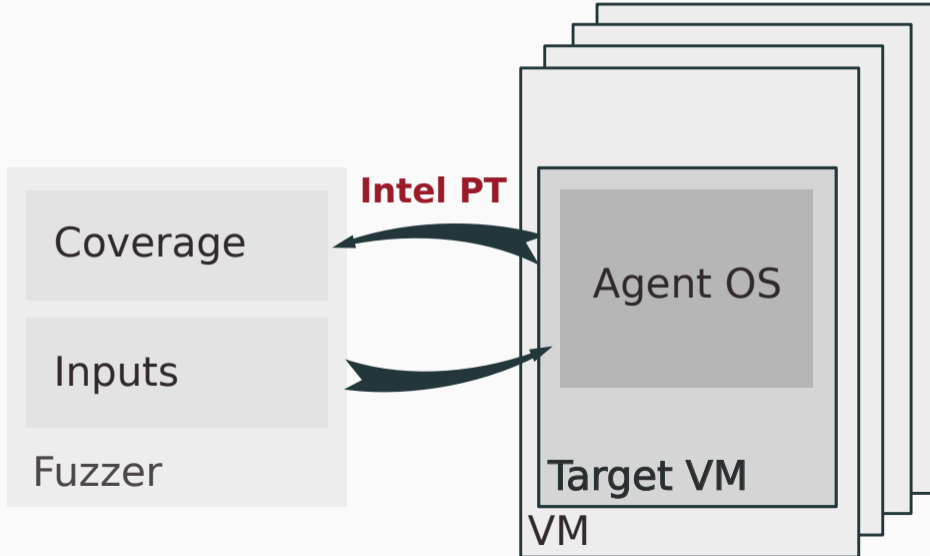


Key Takeaways:

**YO DAWG I HEARD YOU LIKE VIRTUAL
MACHINES**

**SO I PUT A VIRTUAL MACHINE IN YOUR VIRTUAL
MACHINE SO YOU CAN VIRTUAL MACHINE WHILE YOU
VIRTUAL MACHINE**





Intel PT Data

Taken

Not Taken

Target IP (0x1009)

Target IP (0x1055)

Memory Dump

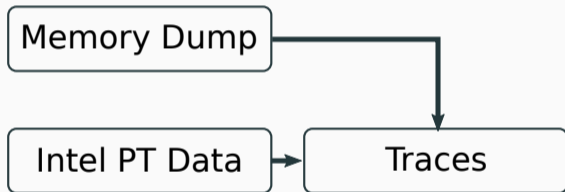
Intel PT Data

Taken

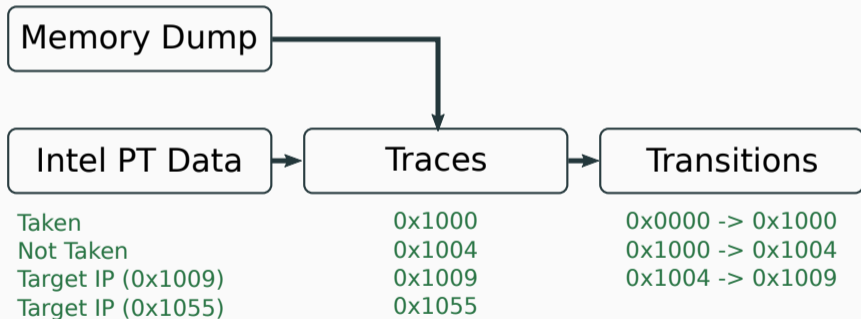
Not Taken

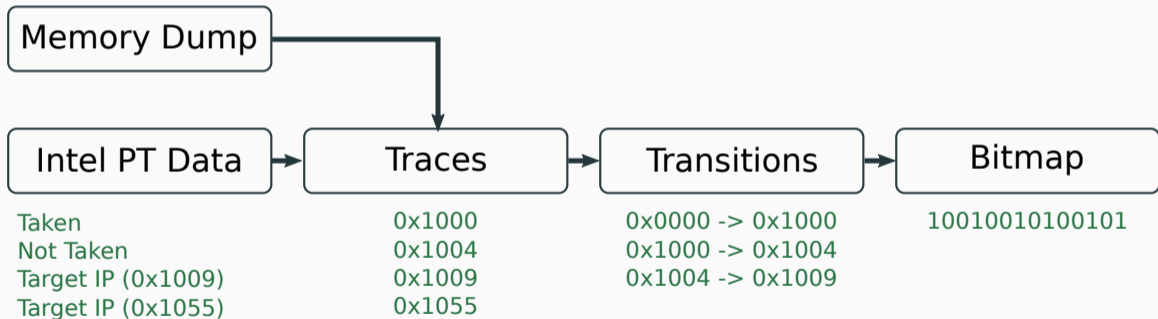
Target IP (0x1009)

Target IP (0x1055)



Taken	0x1000
Not Taken	0x1004
Target IP (0x1009)	0x1009
Target IP (0x1055)	0x1055





Host

VM

ring 0

KVM-PT

Target

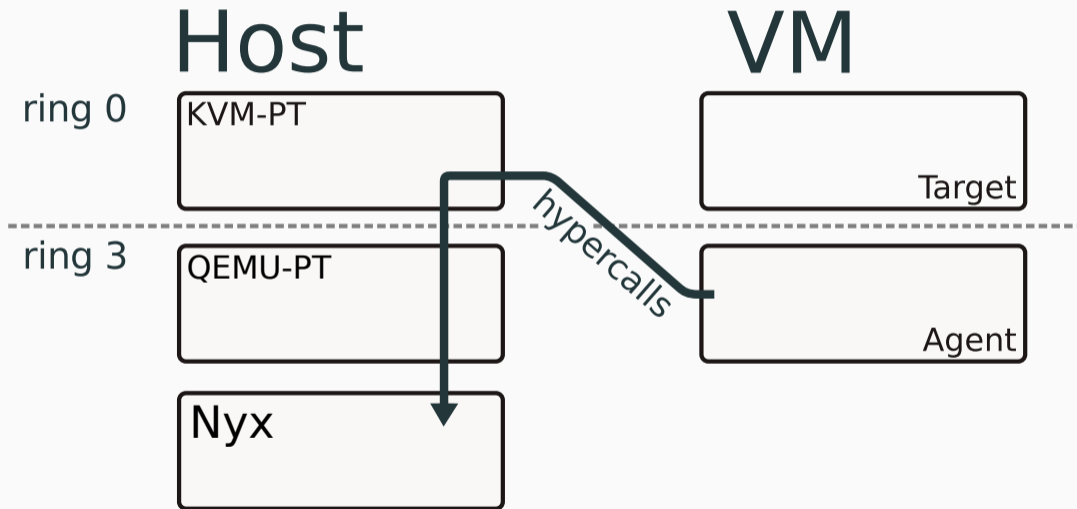
ring 3

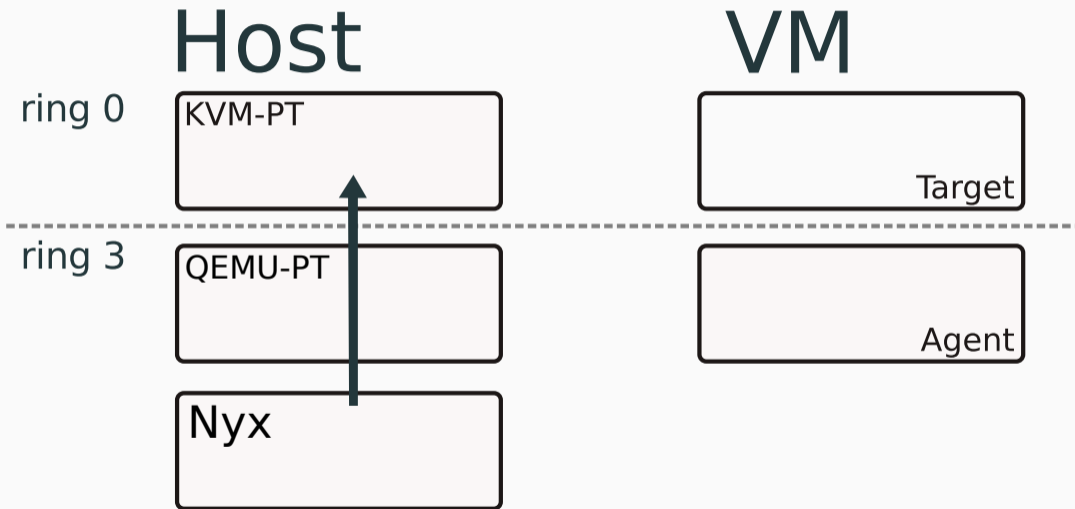
QEMU-PT

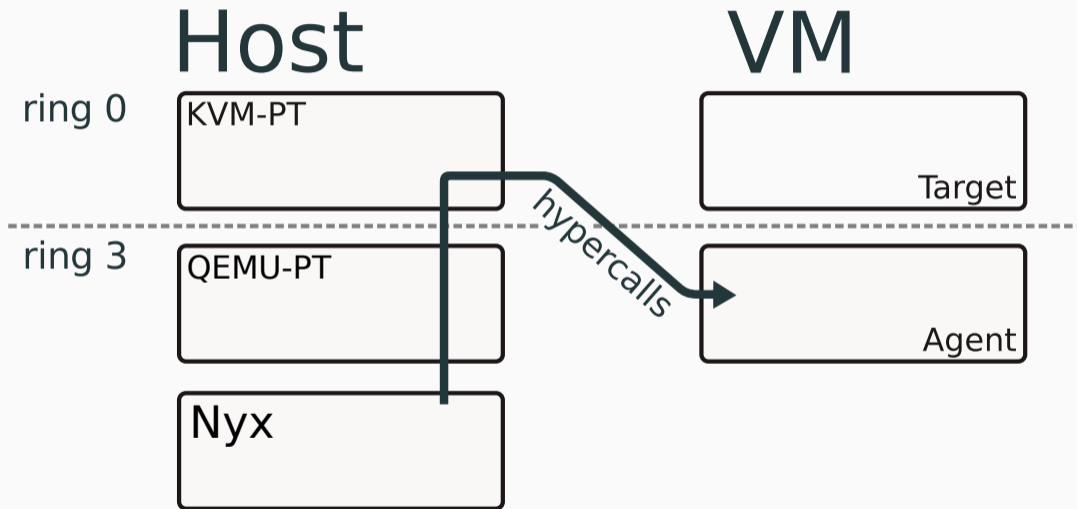
Agent

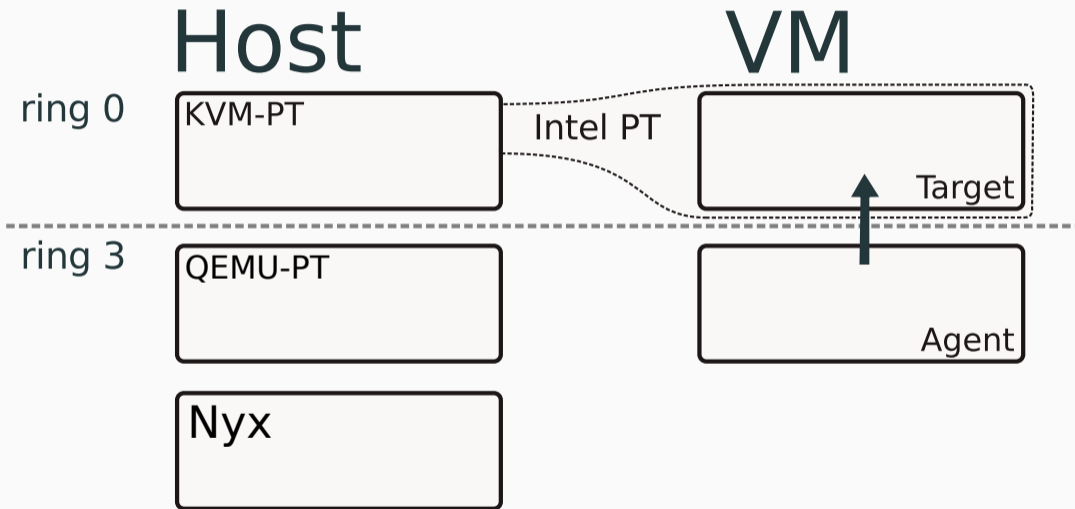
Nyx

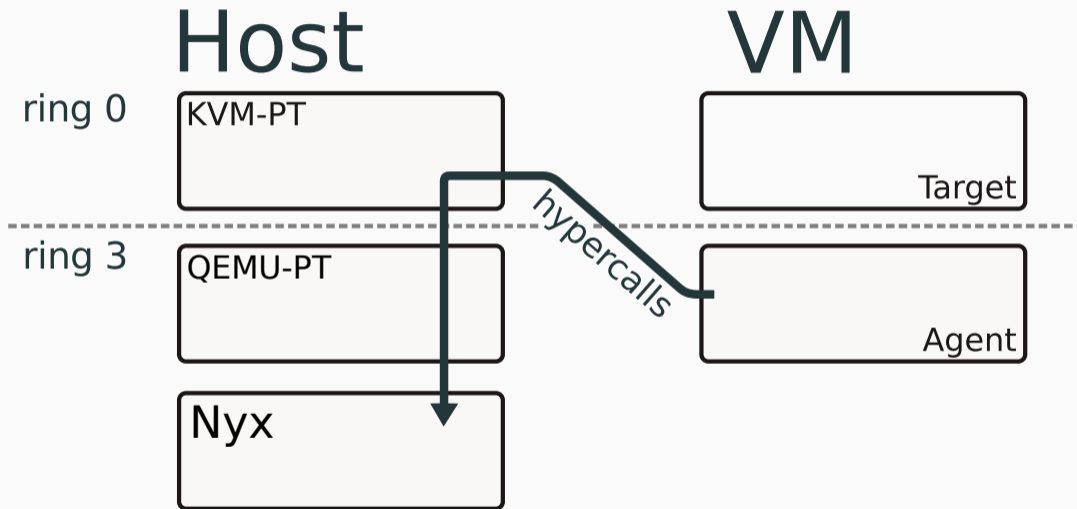


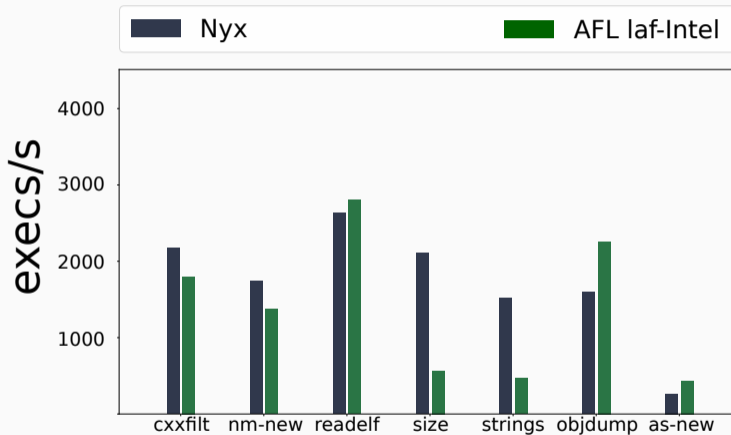












Key Takeaways:



What if I told you



What if I told you



We can be even faster!

Super Fast VM Reloads

~ 6000

times per second

Flatten Qemu VMState Tree

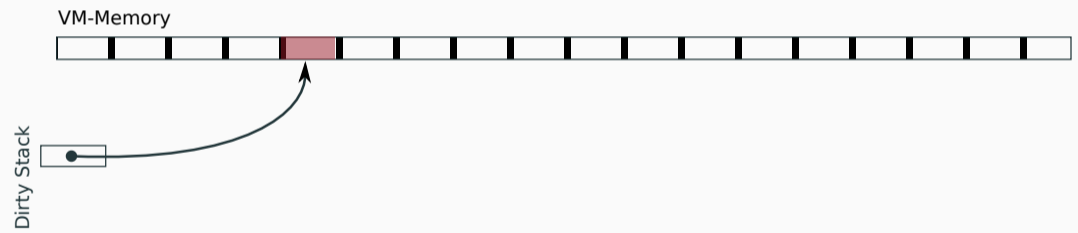


Dirty Page Logging

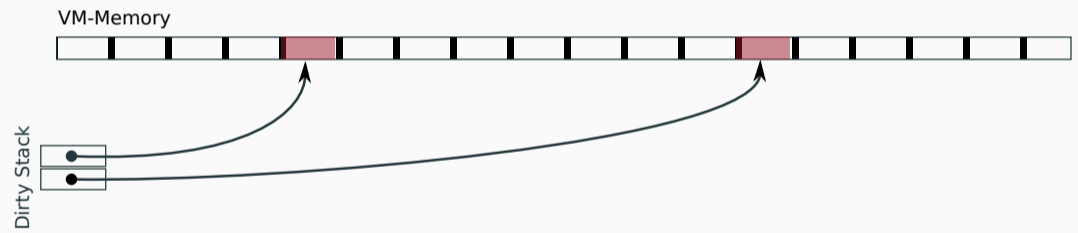
VM-Memory



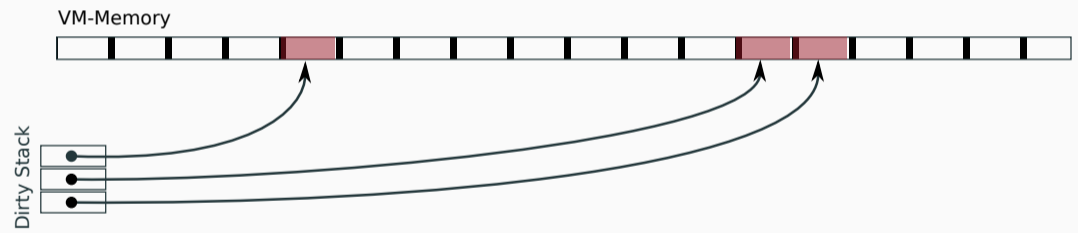
Dirty Page Logging



Dirty Page Logging



Dirty Page Logging



Custom In-Memory
Copy-On-Write
Block Device Layer

$O(1)$ Reset

Test: `gdiplus_test.exe`

Test: `gdiplus_test.exe`

Just Spawing Processes
~80 *execs/sec*

Test: gdiplus_test.exe

Just Spawing Processes

~80 execs/sec

Spawn & File Write

~40 execs/sec

Test: gdiplus_test.exe

Just Spawing Processes

~80 execs/sec

Spawn & File Write

~40 execs/sec

Nyx w. Intel PT & File Writes & Full System Reloads

~145 execs/sec!!!



Faster than Light

Snapshots:

Avoid Startup Time

Snapshots:

Avoid Startup Time

+ Noise free

Snapshots:

Avoid Startup Time

+ Noise free

+ Statefulness

Bugs



macOS High Sierra



TigerVNC



WINE



Parallels Desktop



binutils



Perl



Qtjs



vmware Fusion



ACRN



OpenLDAP



curl://



QEMU



mruby



the
netwide
assembler



Libxml2



Lua



FreeBSD

bhyve



libtiff

TCPDUMP



Chakra
Core



Gnuplot



BASH
THE BOURNE-AGAIN SHELL

Fraunhofer FDK
for Android



Windows

COUNTER STRIKE
SOURCE



macOS High Sierra



TigerVNC



WINE



Parallels Desktop



binutils



Perl



Qtjs



vmware Fusion



ACRN



OpenLDAP



curl://



EMU



ruby



the
netwide
assembler



Libxml2



Gnuplot



Lua



FreeBSD

bhyve



libtiff

TCPDUMP



Chakra
Core



BASH
THE BOURNE AGAIN SHELL

Fraunhofer FDK
for Android



Windows

COUNTER STRIKE
SOURCE



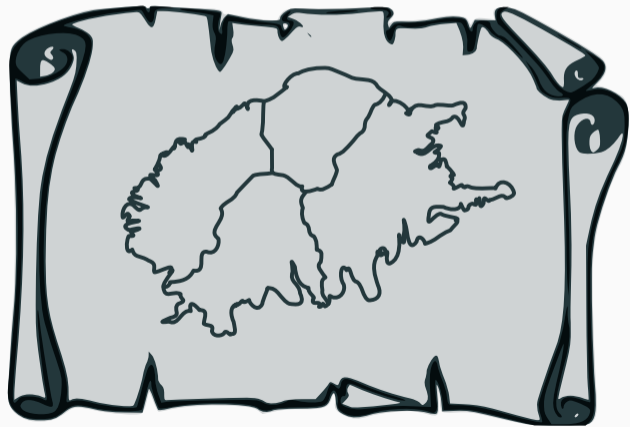
Key Takeaways:



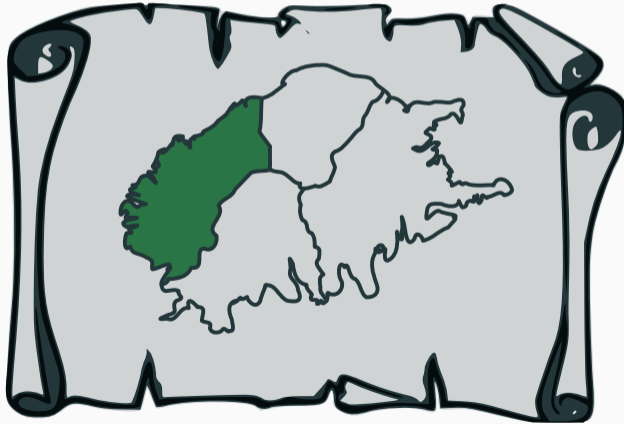
The Rest of this Talk



The Future of
Fuzzing



Harnesses



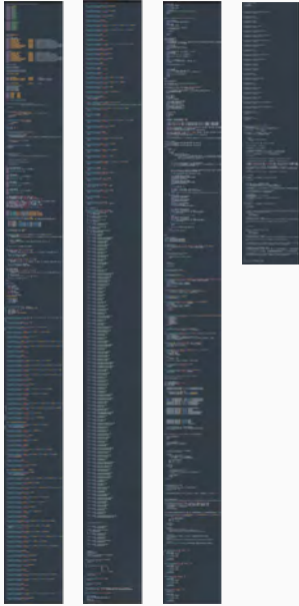




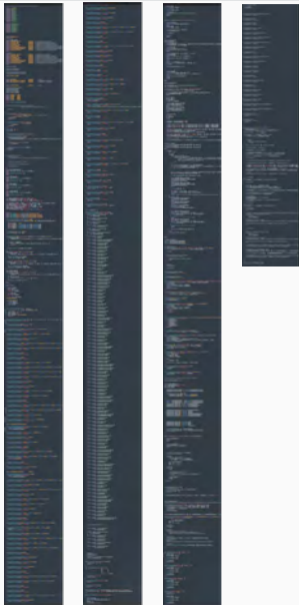
COUNTER STRIKE[™]
SOURCE[™]



Harnesses



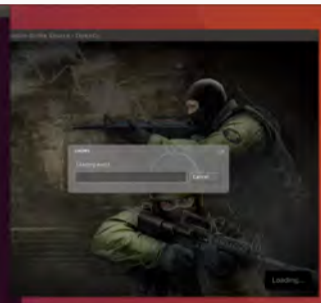
Harnesses



```
serge@ubuntu: ~/steam/steam/steamapps/common/Counter-Strike Source
4
Continuing.
New Thread 0xe1700040 (LWP 112252)
New Thread 0xe1c0f040 (LWP 112253)
New Thread 0xe130e040 (LWP 112254)
Thread 0xe130e040 (LWP 112254) exited
New Thread 0xe1a0d040 (LWP 112255)
New Thread 0xe190c040 (LWP 112256)
New Thread 0xe180b040 (LWP 112257)
Thread 0xe190c040 (LWP 112256) exited
Thread 0xe180b040 (LWP 112257) exited
New Thread 0xe170a040 (LWP 112258)
New Thread 0xe1609040 (LWP 112259)
Thread 0xe170a040 (LWP 112258) exited
New Thread 0xe1508040 (LWP 112260)
New Thread 0xe1407040 (LWP 112261)
New Thread 0xe1306040 (LWP 112262)

Thread 1 "hl2_linux" received signal SIGSEGV, Segmentation fault.
[regs]
000 0x00000000 000 0x00000000 000 0x00000000 000 0x00000000 000 0x00000000 000 0x00000000 000 0x00000000 000 0x00000000
001 0x00000000 000 0x00000000 000 0x00000000 000 0x00000000 000 0x00000000 000 0x00000000 000 0x00000000 000 0x00000000
002 0x00000000 000 0x00000000 000 0x00000000 000 0x00000000 000 0x00000000 000 0x00000000 000 0x00000000 000 0x00000000
[code]
-> 0xf7e57fff <_mempcy_sse2_unaligned+039>: movntdq xmmword ptr [ebx], xmm0
0xf7e58003 <_mempcy_sse2_unaligned+043>: movntdq xmmword ptr [ebx+8x18], xmm0
0xf7e58009 <_mempcy_sse2_unaligned+047>: movntdq xmmword ptr [ebx+8x20], xmm0
0xf7e5800d <_mempcy_sse2_unaligned+051>: movntdq xmmword ptr [ebx+8x28], xmm0
0xf7e58012 <_mempcy_sse2_unaligned+058>: movntdq xmmword ptr [ebx+8x40], xmm0
0xf7e58017 <_mempcy_sse2_unaligned+063>: movntdq xmmword ptr [ebx+8x50], xmm0
0xf7e5801c <_mempcy_sse2_unaligned+068>: movntdq xmmword ptr [ebx+8x60], xmm0
0xf7e58021 <_mempcy_sse2_unaligned+073>: movntdq xmmword ptr [ebx+8x78], xmm0

_mempcy_sse2_unaligned () at ../sysdeps/i386/i686/multiarch/mempcy_sse2_unaligned.S:428
000  ../sysdeps/i386/i686/multiarch/mempcy_sse2_unaligned.S: No such file or directory.
[main]
```





```
$ gdb ./target
GNU gdb (GDB) 7.8
Copyright (C) 2014 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-unknown-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
WARNING: Readline services not available or not loaded.
WARNING: The auto-indent feature requires the readline library
Reading symbols from target...done.
gdb> █
```

```
$ gdb ./target
GNU gdb (GDB) 7.8
Copyright (C) 2014 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-unknown-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
WARNING: Readline services not available or not loaded.
WARNING: The auto-indent feature requires the readline library
Reading symbols from target...done.
gdb> b recv
Breakpoint 1 at 0x406d70
gdb> █
```

```
$ gdb ./target
GNU gdb (GDB) 7.8
Copyright (C) 2014 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-unknown-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
WARNING: Readline services not available or not loaded.
WARNING: The auto-indent feature requires the readline library
Reading symbols from target...done.
gdb> b recv
Breakpoint 1 at 0x406d70
gdb> run
Starting program: /tmp/target

Breakpoint 1, 0x406d70 in main ()
gdb> █
```

```
$ gdb ./target
GNU gdb (GDB) 7.8
Copyright (C) 2014 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-unknown-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
WARNING: Readline services not available or not loaded.
WARNING: The auto-indent feature requires the readline library
Reading symbols from target...done.
gdb> b recv
Breakpoint 1 at 0x406d70
gdb> run
Starting program: /tmp/target

Breakpoint 1, 0x406d70 in main ()
gdb> list
>///...
123 recv_size = recv(fd, buf, max_size, flags);
//...
gdb> █
```

```
$ gdb ./target
GNU gdb (GDB) 7.8
Copyright (C) 2014 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-unknown-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
WARNING: Readline services not available or not loaded.
WARNING: The auto-indent feature requires the readline library
Reading symbols from target...done.
gdb> b recv
Breakpoint 1 at 0x406d70
gdb> run
Starting program: /tmp/target

Breakpoint 1, 0x406d70 in main ()
gdb> list
>///...
123 recv_size = recv(fd, buf, max_size, flags);
//...
gdb> fuzz_set_size &recv_size
gdb> █
```



```
$ gdb ./target
GNU gdb (GDB) 7.8
Copyright (C) 2014 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-unknown-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
WARNING: Readline services not available or not loaded.
WARNING: The auto-indent feature requires the readline library
Reading symbols from target...done.
gdb> b recv
Breakpoint 1 at 0x406d70
gdb> run
Starting program: /tmp/target

Breakpoint 1, 0x406d70 in main ()
gdb> list
>///...
123 recv_size = recv(fd, buf, max_size, flags);
//...
gdb> fuzz_set_size &recv_size
gdb> fuzz_set_max_size max_size
gdb> █
```

```
$ gdb ./target
GNU gdb (GDB) 7.8
Copyright (C) 2014 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-unknown-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
WARNING: Readline services not available or not loaded.
WARNING: The auto-indent feature requires the readline library
Reading symbols from target...done.
gdb> b recv
Breakpoint 1 at 0x406d70
gdb> run
Starting program: /tmp/target

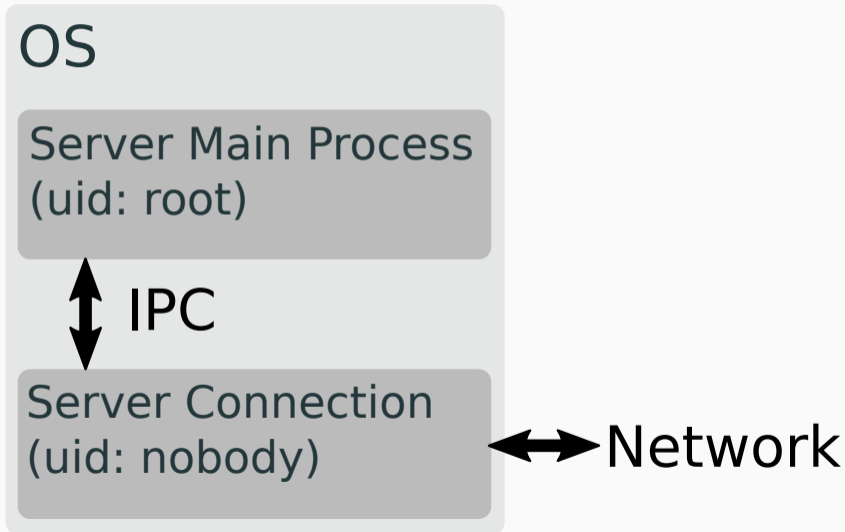
Breakpoint 1, 0x406d70 in main ()
gdb> list
>///...
123 recv_size = recv(fd, buf, max_size, flags);
//...
gdb> fuzz_set_size &recv_size
gdb> fuzz_set_max_size max_size
gdb> fuzz_set_buff buf
gdb> █
```

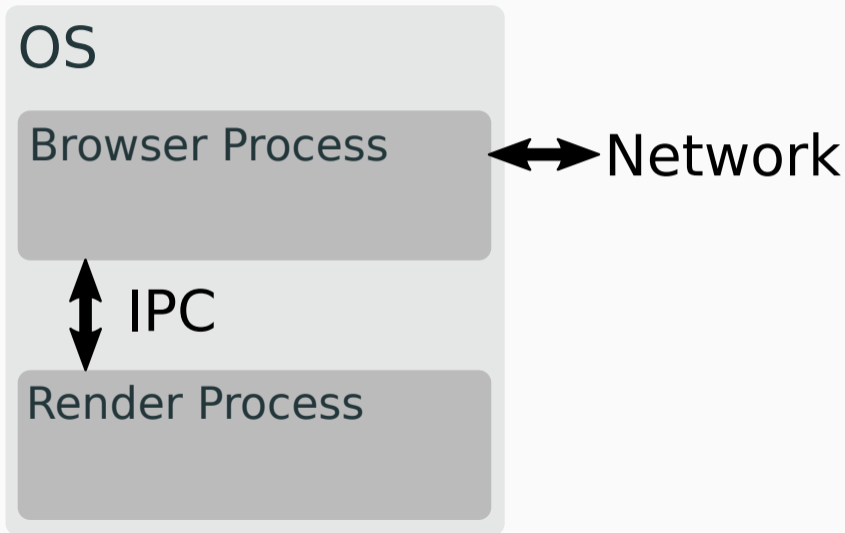
```
$ gdb ./target
GNU gdb (GDB) 7.8
Copyright (C) 2014 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-unknown-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
WARNING: Readline services not available or not loaded.
WARNING: The auto-indent feature requires the readline library
Reading symbols from target...done.
gdb> b recv
Breakpoint 1 at 0x406d70
gdb> run
Starting program: /tmp/target

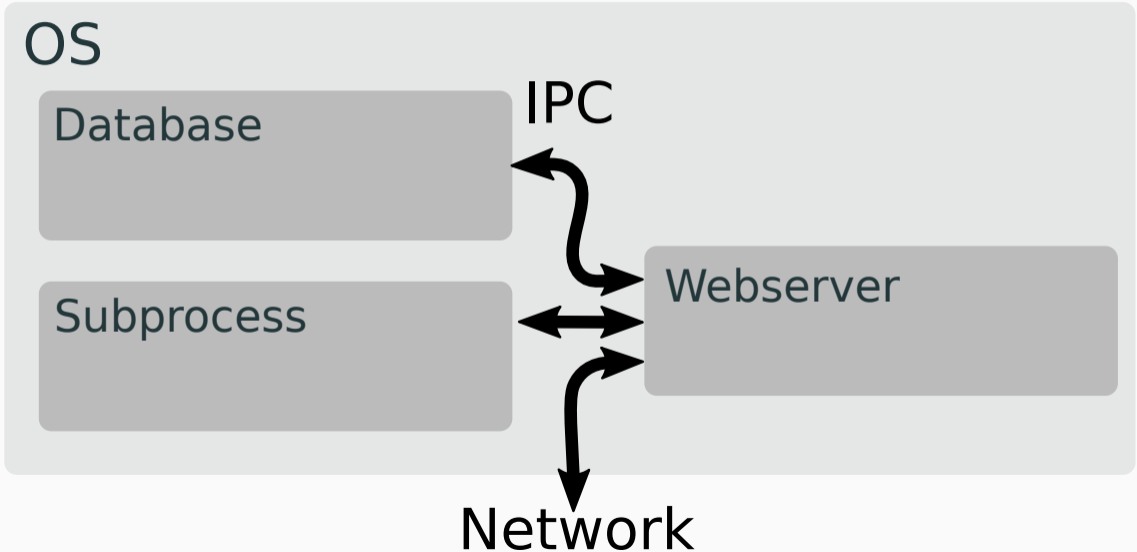
Breakpoint 1, 0x406d70 in main ()
gdb> list
>///...
123 recv_size = recv(fd, buf, max_size, flags);
//...
gdb> fuzz_set_size &recv_size
gdb> fuzz_set_max_size max_size
gdb> fuzz_set_buff buf
gdb> run_fuzzer
```

american fuzzy lop 2.52b (target)

process timing		overall results
run time : 0 days, 0 hrs, 0 min, 23 sec		cycles done : 0
last new path : 0 days, 0 hrs, 0 min, 0 sec		total paths : 142
last uniq crash : none seen yet		uniq crashes : 0
last uniq hang : none seen yet		uniq hangs : 0
cycle progress	map coverage	
now processing : 81 (57.04%)	map density : 0.44% / 2.22%	
paths timed out : 0 (0.00%)	count coverage : 1.73 bits/tuple	
stage progress	findings in depth	
now trying : splice 13	avored paths : 72 (50.70%)	
stage execs : 23/24 (95.83%)	new edges on : 100 (70.42%)	
total execs : 73.9k	total crashes : 0 (0 unique)	
exec speed : 3140/sec	total tmouts : 0 (0 unique)	
fuzzing strategy yields	path geometry	
bit flips : n/a, n/a, n/a	levels : 4	
byte flips : n/a, n/a, n/a	pending : 94	
arithmetics : n/a, n/a, n/a	pend fav : 35	
known ints : n/a, n/a, n/a	own finds : 124	
dictionary : n/a, n/a, n/a	imported : 16	
havoc : 99/47.7k, 25/24.0k	stability : 100.00%	
trim : 60.68%/992, n/a		
[cpu000: 14%]		







Database



Subprocess



Webserver



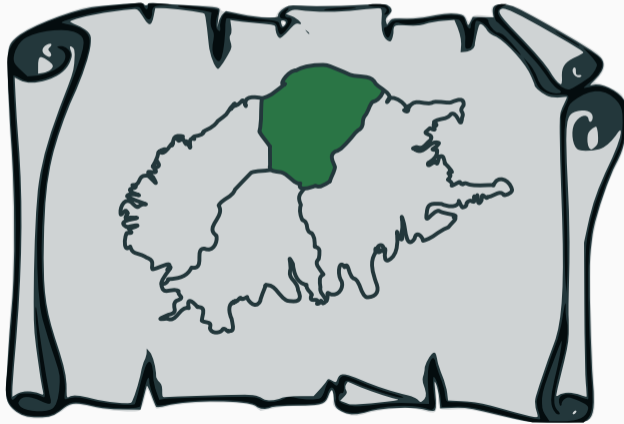
Key Takeaways:



Improve

Usability

Interactive Targets





```
img = Data(0x00, 0x23, 0x54, ... )  
mnt = mount(img);  
dat = "";  
path = "/a"  
mnt.create_file(path, data);  
mnt.cwd(path);  
mnt.umount();
```

```
img = Data(0x00, 0x23, 0x54, ... )  
mnt = mount(img);  
dat = "";  
path = "/a"  
mnt.create_file(path, data);  
mnt.cwd(path);  
mnt.umount();
```

Grammar
Fuzzing?

```
img = Data(0x00, 0x23, 0x54, ... )  
mnt = mount(img);  
dat = "";  
path = "/a"  
mnt.create_file(path, data);  
mnt.cwd(path);  
mnt.umount();
```



Mutated
AFL-Style

```
img = NtfsImg(headers, clusters, ...)  
mnt = mount(img);  
dat = "";  
path = "/a"  
mnt.create_file(path, data);  
mnt.cwd(path);  
mnt.umount();
```

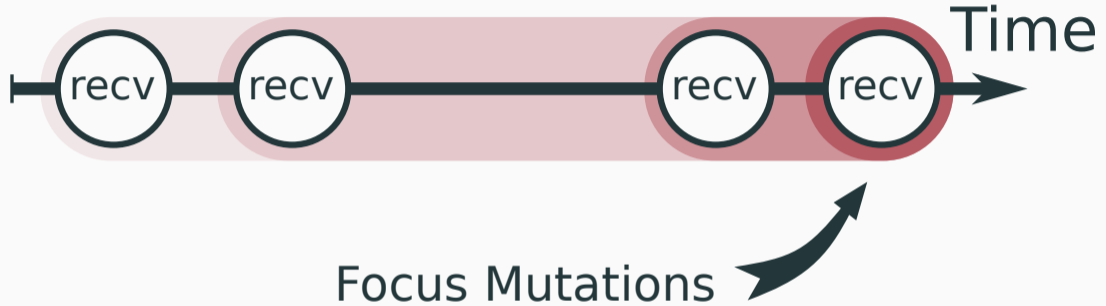


Structural
Mutations

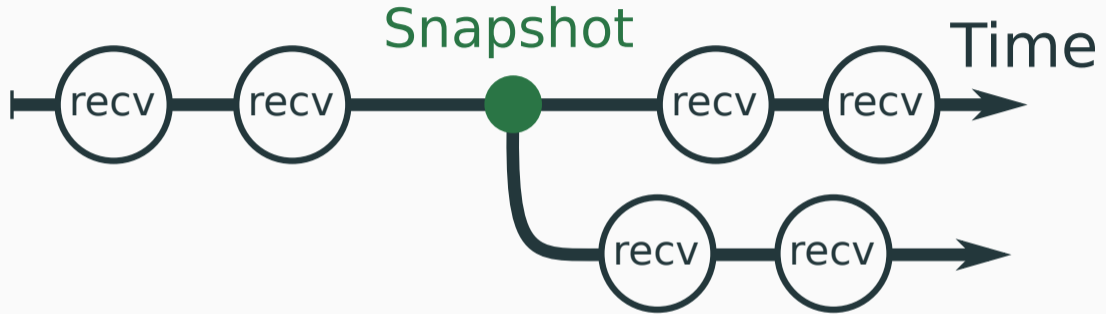
```
img = Data(0x00, 0x23, 0x54, ... )  
mnt = mount(img);  
dat = "";  
path = "/a"  
mnt.create_file(path, data);  
mnt.cwd(path);  
mnt.umount();
```

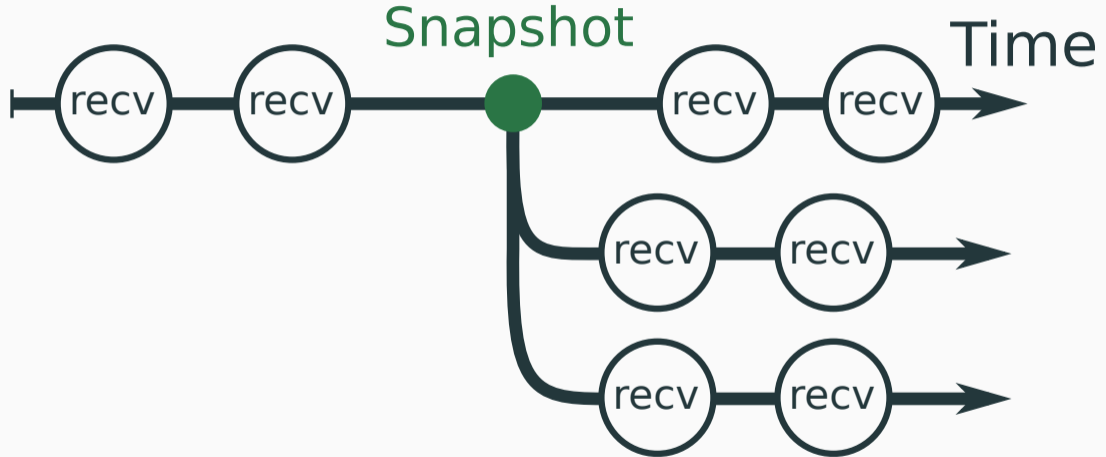


Not reused









Kernel Testing meets Feedback Fuzzing

[1] <https://github.com/google/syzkaller>

Network Protocol meets Feedback Fuzzing

Webcrawler meets Feedback Fuzzing

Key Takeaways:

We need

Bigger Guns



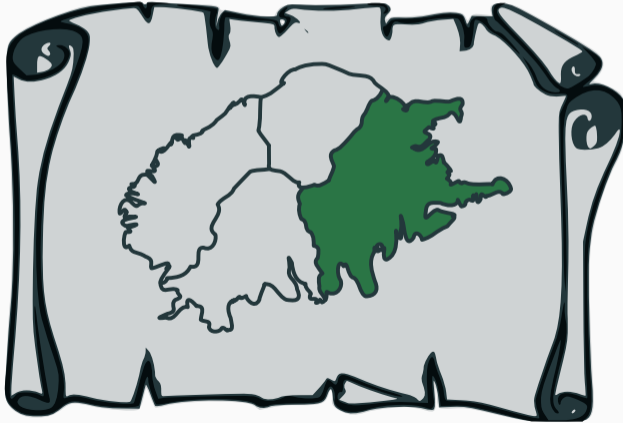
Key Takeaways:

We need

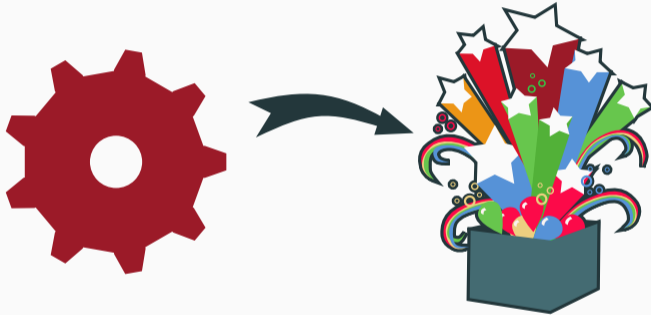
Better Specs



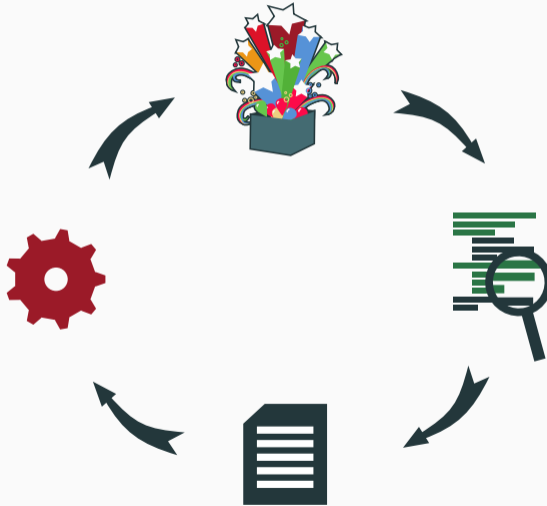
Using Fuzzers

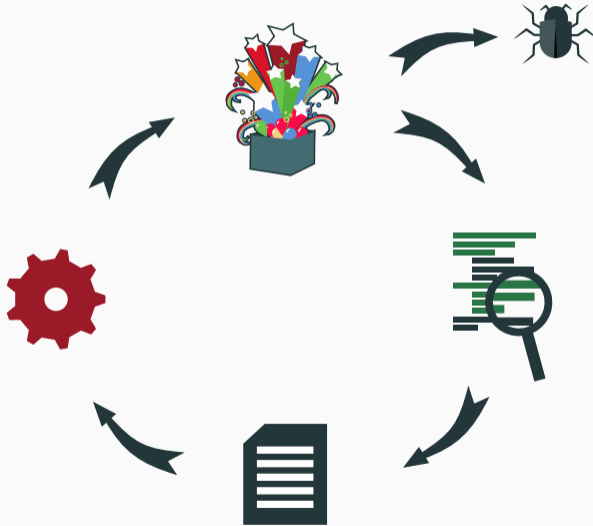












LJON Dashboard Coverage (Lines)

Files Transitions Inputs (1) Search Filter

Ret

0x400220
0x400230

test_data/test.c:22/24

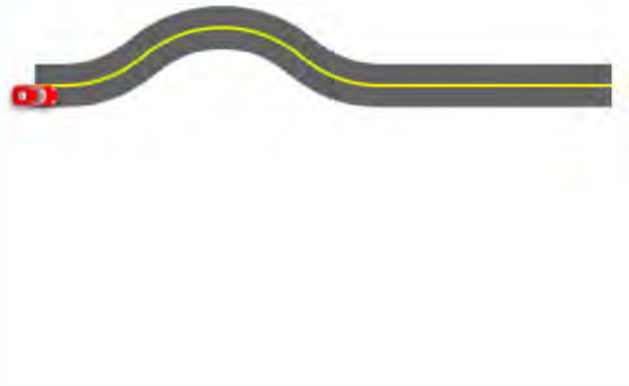
```
1 void foo(int a) {  
2     printf("got %d\n", a);  
3 }  
4  
5 void test(int a, int b) {  
6     while(a) {  
7         if(b) { printf("b=0\n"); return; }  
8         a--;  
9         b--;  
10    }  
11  
12    if(b) { printf("b=1\n"); return; }  
13    printf("b=1\n");  
14    return;  
15 }  
16  
17 int main() {  
18     int a, b;  
19  
20     printf("Enter something\n");  
21     scanf("%d", &a);  
22     printf("Enter something\n");  
23     scanf("%d", &b);  
24     if(a >= 0 && a <= 50) {  
25         if(b >= 0 && b <= 50)  
26             foo(a);  
27         foo(b);  
28     }  
29     test(a, b);  
30 }  
31  
32 return 0;  
33 }
```

Demo

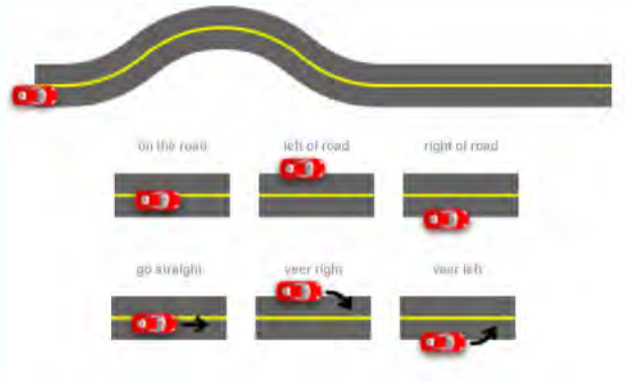
```
00400040 push rbp  
00400047 mov rbp, rsp  
0040004a sub rsp, 0x10  
0040004e mov dword ptr [rbp - 4], esi  
00400051 mov esi, dword ptr [rbp - 4]  
00400054 mov esi, eax  
00400056 mov edi, 0x400074  
00400059 mov eax, 0  
00400060 call 0x400010  
00400065 nop  
00400066 leave  
00400067 ret  
00400068 push rbp  
00400069 mov rbp, rsp  
0040006c sub rsp, 0x10  
00400070 mov dword ptr [rbp - 4], esi  
00400073 mov dword ptr [rbp - 8], esi  
00400076 jmp 0x400067  
00400078 cmp dword ptr [rbp - 4], 0  
0040007c jne 0x40008a  
0040007e mov edi, 0x400074  
00400083 call 0x400070  
00400088 jmp 0x400065  
0040008a sub dword ptr [rbp - 4], 1  
0040008d sub dword ptr [rbp - 8], 1  
00400092 cmp dword ptr [rbp - 4], 0  
00400096 jg 0x400010  
00400098 cmp dword ptr [rbp - 8], 0  
0040009c jne 0x40009a  
0040009e mov esi, 0x400060  
004000a3 call 0x400070  
004000a8 jmp 0x400065  
004000aa mov esi, 0x400060  
004000af call 0x400070  
004000b4 nop  
004000b5 leave  
004000b6 ret  
004000b7 push rbp  
004000b8 mov rbp, rsp  
004000bd sub rsp, 0x10  
004000c7 mov rax, qword ptr [rbp - 8]  
004000cc xor rax, rax  
004000ce mov edi, 0x400060  
004000d3 call 0x400070  
004000d8 lea rax, qword ptr [rbp - 0x10]  
004000db mov rsi, rax  
004000df mov edi, 0x40001a  
004000e4 mov rax, 0  
004000e9 call 0x400010  
004000ee mov edi, 0x40001a  
004000f3 call 0x400070  
004000f8 lea rax, qword ptr [rbp - 0xc]  
004000fc mov rsi, rax  
00400107 mov edi, 0x40001a  
00400114 mov eax, 0
```

Create Analysis Tools that
Abstract away Inputs

Fuzzing



Fuzzing



Fuzzing



Debugger (gdb, olly, ...)

Fuzzing



Debugger (gdb, olly, ...)

Fuzzing



Debugger (gdb, olly, ...)

Fuzzing



Debugger (gdb, olly, ...)

Fuzzing



Debugger (gdb, olly, ...)

Fuzzing



Debugger (gdb, olly, ...)



Abstract away Time
Time traveling Debugger
(rr, REVEN, ...)

Fuzzing



Debugger (gdb, olly, ...)

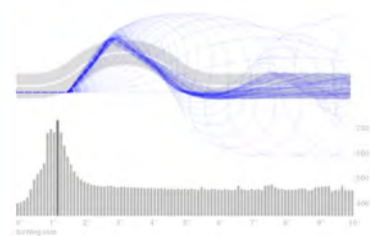


Abstract away Time
Time traveling Debugger
(rr, REVEN, ...)



Abstract away Inputs
Fuzzing Debugger
????

Fuzzing



Debugger (gdb, olly, ...)

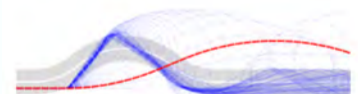


Abstract away Time
Time traveling Debugger
(rr, REVEN, ...)



Abstract away Inputs
Fuzzing Debugger
????

Fuzzing



Debugger (gdb, olly, ...)

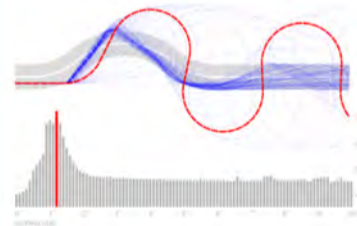


Abstract away Time
Time traveling Debugger
(rr, REVEN, ...)



Abstract away Inputs
Fuzzing Debugger
????

Fuzzing



Debugger (gdb, olly, ...)

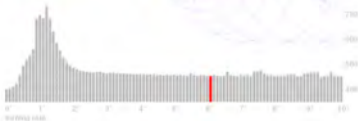


Abstract away Time
Time traveling Debugger
(rr, REVEN, ...)



Abstract away Inputs
Fuzzing Debugger
????

Fuzzing



Debugger (gdb, olly, ...)



Abstract away Time
Time traveling Debugger
(rr, REVEN, ...)



Abstract away Inputs
Fuzzing Debugger
????

A Better Tool...

LJON Dashboard Coverage Queries

Files Transitions Inputs (6) Search Filter

Taken:

0x000000

NotTaken:

0x000001

/home/me/ljon_evil/cb-multios/challenges/Multitpass/src/main.c 76/170

```
transaction = NULL;
if (last_id != NULL && *last_id != pkthdr.transaction_id)
    cgc_send_error(ERRNO_MP_NOT_FOUND, NOT_FOUND_MSG);
goto fail;

transaction_t *last_transaction = lookup_transaction(pkthdr.transaction_id);
ASS if (pkthdr.op_code == ISSUE)
    if (pkthdr.pkt_type == ENIT)
        transaction = cgc_new_transaction();
        if (!transaction)
            cgc_send_error(ERRNO_MP_ALLOC, ALLOC_MSG);
            goto fail;

        if (cgc_read_data(transaction) != 0)
            cgc_send_error(ERRNO_MP_SINK, SINK_ERROR_MSG);
            goto fail;

        /* [source] == [target] ? */
        goto fail;

        last_id = &transaction->id;
    else if (pkthdr.pkt_type == FIN && !last_transaction)
        transaction = last_transaction;

__asm__ volatile (
    "xor esi, esi\n"
    "cmp eax, ebx\n"
    "jz 0x00400021\n"
    "mov ecx, dword ptr [0x00400020]\n"
    "mov ebx, dword ptr gs:[edi]\n"
    "mov esi, 0x10ff\n"
    "xor ebx, esi\n"
    "inc byte ptr [ecx + edi]\n"
    "mov dword ptr gs:[edi], 0xadf\n"
    "mov ecx, dword ptr [0x00400020]\n"
    "lea ecx, dword ptr [ecx + ecx*2]\n"
    "lea ecx, dword ptr [ebx + ecx*8]\n"
    "cpg ebx, ecx\n"
    "mov ecx, 0\n"
    "cmovbe eax, ecx\n"
    "mov ecx, eax\n"
    "mov eax, dword ptr [0x00400020]\n"
    "mov ebx, dword ptr gs:[edi]\n"
    "mov esi, 0x0000\n"
    "xor ebx, esi\n"
    "inc byte ptr [eax + ebx]\n"
    "mov dword ptr gs:[edi], 0x7000\n"
    "movzx ebx, byte ptr [0x00400020]\n"
    "shl ebx, 0x10\n"
    "movzx eax, word ptr [0x00400020]\n"
    "or eax, ebx\n"
    "movzx ebx, sb\n"
    "cpg ebx, 8\n"
    "inc 0x00400020\n"
    "mov ebx, dword ptr [0x00400020]\n"
    "mov esi, dword ptr gs:[edi]\n"
    "mov ebx, 0x1000f\n"
    "xor esi, ebx\n"
    "inc byte ptr [ebx + esi]\n"
    "mov dword ptr gs:[edi], 0x100f\n"
    "test al, al\n"
    "jz 0x00400027\n"
    "mov ebx, dword ptr [0x00400020]"
);
```

id:000008_src:000001_op:havoc_rep:8,+cov

```
00000000: 0000 1337 0000 0000 3730 0000 0000 0000
00000010: 3df8 0600 0000 0000 0204 000f a85d 492b
00000020: 0404 004d c0bb e574 0708 00dc a741 0c05
00000030: aa10 361b 040c 00da bf00 9138 3a50 1ba2
00000040: c381 a37b 0908 00ff 9853 25c5 15cd 5991
00000050: 0e08 005f 30b5 89fd 294f 454f
```

```
...7...70....
=.....]I+
..M...t....A..
..6.....:8:P..
...{.....S%...Y.
..._0...)0E0
```

Watch Points

Address	Value
0x00000000	0x00000000
0x00000010	0x00000000
0x00000020	0x00000000
0x00000030	0x00000000
0x00000040	0x00000000
0x00000050	0x00000000

A Better Tool...

LJON Dashboard Coverage Queries

Files Transitions Inputs (6) Search Filter

Taken: 0x00000000
NotTaken: 0x00000001

/home/me/ljon_evsl/cb-multios/challenges/Multipass/src/main.c 76/170

```
transaction = NULL;
if (last_id != NULL && *last_id != pkthdr.transaction_id)
    cgc_send_error(ERRNO_MP_NOT_FOUND, NOT_FOUND_MSG);
goto fail;

transaction_t *last_transaction = lookup_transaction(pkthdr.transaction_id);
ASS if (pkthdr.op_code == ISSUE)
    if (pkthdr.pkt_type == ENIT)
        transaction = cgc_new_transaction();
        if (!transaction)
            cgc_send_error(ERRNO_MP_ALLOC, ALLOC_MSG);
            goto fail;

        if (cgc_read_data(transaction) != 0)
            cgc_send_error(ERRNO_MP_SINK, SINK_ERROR_MSG);
            goto fail;

        /* [source] == [dest] ? */
        goto fail;

        last_id = &transaction->id;
    else if (pkthdr.pkt_type == FIN && !last_transaction)
        transaction = last_transaction;
```

```
0004c7e9 xor esi, esi
0004c7eb cmp eax, ebx
0004c7ed jk 0x0040001
0004c7ef mov ecx, dword ptr [0x00500020]
0004c7f5 mov ebx, dword ptr gs:[edi]
0004c7f8 mov esi, 0x10ff
0004c7ff xor ebx, esi
0004c7ff inc byte ptr [ecx + ebx]
0004d002 mov dword ptr gs:[edi], 0xadf
0004d009 mov ecx, dword ptr [0x00500020]
0004d00f lea ecx, dword ptr [ecx + ecx*2]
0004d012 lea ecx, dword ptr [ebx + ecx*8]
0004d015 cmp ebx, ecx
0004d017 mov ecx, 0
0004d01c cmovae ebx, ecx
0004d021 mov ecx, eax
0004d023 mov eax, dword ptr [0x00500020]
0004d026 mov ebx, dword ptr gs:[edi]
0004d029 mov esi, 0x0000
0004d02e xor ebx, esi
0004d030 inc byte ptr [eax + ebx]
0004d033 mov dword ptr gs:[edi], 0x7000
0004d036 movzx ebx, byte ptr [0x00500020]
0004d041 shl ebx, 0x10
0004d044 movzx eax, word ptr [0x00500020]
0004d04b or eax, edx
0004d04d movzx ebx, al
0004d050 cmp ebx, 8
0004d053 jne 0x0040001
0004d059 mov ebx, dword ptr [0x00500020]
0004d05f mov esi, dword ptr gs:[edi]
0004d062 mov ebx, 0x1000f
0004d067 xor esi, ebx
0004d069 inc byte ptr [ebx + esi]
0004d06c mov dword ptr gs:[edi], 0x100f
0004d073 test al, al
0004d075 jz 0x0040001f
0004d07b mov ebx, dword ptr [0x00500020]
```

id:000008_src:000001_op:havoc_rep:8,+cov

```
00000000: 0000 1337 0000 0000 3730 0000 0000 0000
00000010: 3df8 0600 0000 0000 0204 000f a85d 492b
00000020: 0404 004d c0bb e574 0708 00dc a741 0c05
00000030: aa10 361b 040c 00da bf00 9138 3a50 1ba2
00000040: c381 a37b 0908 00ff 9853 25c5 15cd 5991
00000050: 0e08 005f 30b5 89fd 294f 454f
```

```
...7...70....
=.....]I+
..M...t....A..
..6.....:8:P..
...{.....S%...Y.
..._0...)0EO
```

Watch Points

Watch Points

A Better Tool...

LJON Dashboard Coverage Queries

Files Transitions Inputs (6) Search Filter

Taken: 0x00000000
NotTaken: 0x00000001

/home/me/ljon_evil/cb-multios/challenges/Multitpass/src/main.c 76/170

```
...
775 transaction = NULL;
776 if (last_id != NULL && *last_id != pkthdr.transaction_id)
777     continue;
778
779     cgc_send_error(ERRNO_MP_NOT_FOUND, NOT_FOUND_MSG);
780     goto FAIL;
781
782     transaction = lookup_transaction(pkthdr.transaction_id);
783
784     if (pkthdr.op_code == ISSUE)
785     {
786         if (pkthdr.pkt_type == ENIT)
787         {
788             transaction = cgc_new_transaction();
789             if (!transaction)
790                 continue;
791
792             cgc_send_error(ERRNO_MP_ALLOC, ALLOC_MSG);
793             goto FAIL;
794         }
795
796         if (cgc_read_data(transaction) != 0)
797         {
798             cgc_send_error(ERRNO_MP_SINK, SINK_ERROR_MSG);
799             goto FAIL;
800         }
801
802         last_id = &transaction->id;
803     }
804     else if (pkthdr.pkt_type == FIN && !last_transaction)
805     {
806         transaction = last_transaction;
807     }
808 }
809
810 return transaction;
811 }
```

```
0004c190 xor esi, esi
0004c1e6 cmp eax, ebx
0004c1ef jb 0x04d021
0004c1ef mov ecx, dword ptr [eax+0x020]
0004c1f5 mov ebx, dword ptr [esi]
0004c1f6 mov esi, 0x10ff
0004c1ff xor ebx, esi
0004c1ff inc byte ptr [ecx + ebx]
0004c202 mov dword ptr [esi], ebx
0004c209 mov ecx, dword ptr [0x04d020]
0004c20f lea ecx, dword ptr [ecx + ecx*2]
0004c212 lea ecx, dword ptr [ebx + ecx*8]
0004c215 cmp ebx, ecx
0004c217 mov ecx, 0
0004c21c cmovae ebx, ecx
0004c21f mov ecx, ebx
0004c221 mov eax, dword ptr [0x04d020]
0004c226 mov ebx, dword ptr [esi]
0004c229 mov esi, 0x0000
0004c22c xor ebx, esi
0004c230 inc byte ptr [eax + ebx]
0004c233 mov dword ptr [esi], ebx
0004c236 movzx ebx, byte ptr [0x04d020]
0004c241 shl ebx, 0x10
0004c244 movzx eax, word ptr [0x04d020]
0004c24b or eax, ebx
0004c24d movzx ebx, al
0004c250 cmp ebx, 8
0004c253 jng 0x04d017
0004c255 mov ebx, dword ptr [esi]
0004c25f mov esi, dword ptr [esi]
0004c262 mov ebx, 0x10ff
0004c267 xor esi, ebx
0004c269 inc byte ptr [ebx + esi]
0004c26c mov dword ptr [esi], ebx
0004c273 test al, al
0004c275 jz 0x04d017
0004c27b mov ebx, dword ptr [0x04d020]
```

Add Watch Point

id:000008_src:000001_op:havoc_rep:8,+cov

```
00000000: 0000 1337 0000 0000 3730 0000 0000 0000
00000010: 3df8 0600 0000 0000 0204 000f a85d 492b
00000020: 0404 004d c0bb e574 0708 00dc a741 0c05
00000030: aa10 361b 040c 00da bf00 9138 3a50 1ba2
00000040: c381 a37b 0908 00ff 9853 25c5 15cd 5991
00000050: 0e08 005f 30b5 89fd 294f 454f
```

```
...7...70....
=.....]I+
..M...t....A..
..6.....:8:P..
...{.....S%...Y.
..._0...)0E0
```

Watch Points

Address	Value
00000000	0000
00000010	3df8
00000020	0404
00000030	aa10
00000040	c381
00000050	0e08

A Better Tool...

LJON

Files Transitions Inputs (6) Search Filter

Taken: 0x00000000
NotTaken: 0x00000001

/home/me/ljon_evsl/cb-multios/challenges/Multipass/src/main.c 76/170

```
transaction = NULL;
if (last_id != NULL && *last_id != pkthdr.transaction_id)
    cgc_send_error(ERRNO_MP_NOT_FOUND, NOT_FOUND_MSG);
goto FAIL;

transaction_t *last_transaction = lookup_transaction(pkthdr.transaction_id);
ASS if (pkthdr.op_code == ISSUE)
    if (pkthdr.pkt_type == ENIT)
        transaction = cgc_new_transaction();
        if (!transaction)
            cgc_send_error(ERRNO_MP_ALLOC, ALLOC_MSG);
            goto FAIL;

        if (cgc_read_data(transaction) != 0)
            cgc_send_error(ERRNO_MP_SINK, SINK_ERROR_MSG);
            goto FAIL;

        if (!lookup_transaction(pkthdr.transaction_id))
            goto FAIL;

        last_id = &transaction->id;
    else if (pkthdr.pkt_type == FIN && !last_transaction)
        transaction = last_transaction;
```

```
0004c190 xor esi, esi
0004c1e6 cmp eax, ebx
0004c1ef jb 0x04d021
0004c1ef mov ecx, dword ptr [0x04d028]
0004c1f5 mov ebx, dword ptr [esi]
0004c1f6 mov esi, 0x10ff
0004c1f6 xor ebx, esi
0004c1ff inc byte ptr [ecx + esi]
0004c202 mov dword ptr [esi], 0x04d021
0004c209 mov ecx, dword ptr [0x04d028]
0004c20f lea ecx, dword ptr [ecx + ecx*2]
0004c212 lea ecx, dword ptr [ebx + ecx*8]
0004c215 cmp ebx, ecx
0004c217 mov ecx, 0
0004c21c cmovae ebx, ecx
0004c21f mov ecx, ebx
0004c221 mov eax, dword ptr [0x04d028]
0004c226 mov ebx, dword ptr [esi]
0004c229 mov esi, 0x0000
0004c22c xor ebx, esi
0004c230 inc byte ptr [eax + esi]
0004c233 mov dword ptr [esi], 0x7000
0004c236 movzx ebx, byte ptr [0x04d028]
0004c241 shl ebx, 0x10
0004c244 movzx eax, word ptr [0x04d02c]
0004c247 or esi, esi
0004c24a movzx ebx, al
0004c24c cmp ebx, 8
0004c24e jne 0x04d074
0004c253 mov ebx, dword ptr [esi]
0004c25f mov esi, dword ptr [esi]
0004c262 mov ebx, 0x04c1f6
0004c267 xor esi, ebx
0004c269 inc byte ptr [ebx + esi]
0004c26c mov dword ptr [esi], 0x10ff
0004c273 test al, al
0004c275 je 0x04d07f
0004c27b mov ebx, dword ptr [0x04d028]
```

Add Watch Point

id:000008_src:000001_op:havoc_rep:8,+cov

```
00000000: 0000 1337 0000 0000 3730 0000 0000 0000
00000010: 3df8 0600 0000 0000 0204 000f a85d 492b
00000020: 0404 004d c0bb e574 0708 00dc a741 0c05
00000030: aa10 361b 040c 00da bf00 9138 3a50 1ba2
00000040: c381 a37b 0908 00ff 9853 25c5 15cd 5991
00000050: 0e08 005f 30b5 89fd 294f 454f
```

```
...7...70....
=.....]I+
..M...t....A..
..6.....:8:P..
...{.....S%...Y.
..._0...)0E0
```

Watch Points

Address	Value
00000000	0000
00000010	3df8
00000020	0404
00000030	aa10
00000040	c381
00000050	0e08

A Better Tool...

LJON Dashboard Coverage Queries

Files Transitions Inputs (6) Search Filter

Taken: 0x00000000
NotTaken: 0x00000001

/home/me/ljon_evil/cb-multios/challenges/Multipass/src/main.c 76/170

```
transaction = NULL;
if (last_id != NULL && *last_id != pkthdr.transaction_id)
    cgc_send_error(ERRNO_MP_NOT_FOUND, NOT_FOUND_MSG);
goto fail;

transaction_t *last_transaction = lookup_transaction(pkthdr.transaction_id);
ASS if (pkthdr.op_code == ISSUE)
    if (pkthdr.pkt_type == ENIT)
        transaction = cgc_new_transaction();
        if (!transaction)
            cgc_send_error(ERRNO_MP_ALLOC, ALLOC_MSG);
            goto fail;

        if (cgc_read_data(transaction) != 0)
            cgc_send_error(ERRNO_MP_SINK, SINK_ERROR_MSG);
            goto fail;

        /* [source] == [target] ? */
        goto fail;

        last_id = &transaction->id;
    else if (pkthdr.pkt_type == FIN && !last_transaction)
        transaction = last_transaction;
```

```
0004cfe9 xor esi, esi
0004cfef cmp eax, ebx
0004cfef jb 0x004d011
0004cfef mov ecx, dword ptr [0x004d028]
0004cf75 mov ebx, dword ptr [esi]
0004cf78 mov esi, 0x10ff
0004cf7e xor ebx, esi
0004cf7f inc byte ptr [ecx + esi]
0004cf82 mov dword ptr [esi + edi], 0xadf
0004cf89 mov ecx, dword ptr [0x004d028]
0004cf8f lea ecx, dword ptr [ecx + ecx*2]
0004cf92 lea ecx, dword ptr [ebx + ecx*8]
0004cf95 cmp ebx, ecx
0004cf97 mov ecx, 0
0004cf9c cmovae ebx, ecx
0004cf9f mov ecx, ebx
0004cf9f mov eax, dword ptr [0x004d028]
0004cf9f mov ebx, dword ptr [esi]
0004cf9f mov esi, 0x0000
0004cf9f xor ebx, esi
0004cf9f inc byte ptr [eax + esi]
0004cf9f mov dword ptr [esi], 0x700
0004cf9f movzx ebx, byte ptr [0x004d028]
0004cf9f shl ebx, 0x10
0004cf9f movzx eax, word ptr [0x004d028]
0004cf9f or esi, esi
0004cf9f movzx ebx, sb
0004cf9f cmp ebx, 0
0004cf9f jne 0x004d017
0004cf9f mov ebx, dword ptr [esi]
0004cf9f mov esi, dword ptr [esi]
0004cf9f mov ebx, 0x10cf
0004cf9f xor esi, ebx
0004cf9f inc byte ptr [ebx + esi]
0004cf9f mov dword ptr [esi], 0x10cf
0004cf9f test al, al
0004cf9f jz 0x004d017
0004cf9f mov ebx, dword ptr [0x004d028]
```

Expr

OK

id:000008,src:000001,op:havoc,rep:8,+cov

```
00000000: 0000 1337 0000 0000 3730 0000 0000 0000
00000010: 3df8 0600 0000 0000 0204 000f a85d 492b
00000020: 0404 004d c0bb e574 0708 00dc a741 0c05
00000030: aa10 361b 040c 00da bf00 9138 3a50 1ba2
00000040: c381 a37b 0908 00ff 9853 25c5 15cd 5991
00000050: 0e08 005f 30b5 89fd 294f 454f
```

```
...7...70....
=.....]I+
..M...t....A..
..6.....:8:P..
...{.....S%...Y.
..._0...)0E0
```

Watch Points

Address	Value
00000000	00000000
00000010	00000000
00000020	00000000
00000030	00000000
00000040	00000000
00000050	00000000

A Better Tool...

LJON Dashboard Coverage Queries

Files Transitions Inputs (6) Search Filter

Taken: 0x00000000
NotTaken: 0x00000001

/home/me/ljon_evil/cb-multios/challenges/MultiPass/src/main.c 76/170

```
transaction = NULL;
if (last_id != NULL && *last_id != pkthdr.transaction_id)
    cgc_send_error(ERRNO_MP_NOT_FOUND, NOT_FOUND_MSG);
goto fail;

transaction_t *last_transaction = lookup_transaction(pkthdr.transaction_id);
ASS if (pkthdr.op_code == ISSUE)
    if (pkthdr.pkt_type == ENIT)
        transaction = cgc_new_transaction();
        if (!transaction)
            cgc_send_error(ERRNO_MP_ALLOC, ALLOC_MSG);
            goto fail;

        if (cgc_read_data(transaction) != 0)
            cgc_send_error(ERRNO_MP_SINK, SINK_ERROR_MSG);
            goto fail;

        /* [source] == [target] ? */
        goto fail;

        last_id = &transaction->id;
    else if (pkthdr.pkt_type == FIN && !last_transaction)
        transaction = last_transaction;
```

```
0004c190 xor esi, esi
0004c1e6 cmp eax, ebx
0004c1ef jb 0x04d021
0004c1ef mov ecx, dword ptr [0x04d028]
0004c1f5 mov ebx, dword ptr [esi]
0004c1f6 mov esi, 0x10ff
0004c1f6 xor ebx, esi
0004c1ff inc byte ptr [ecx + esi]
0004d002 mov dword ptr [esi + edi], 0x04d007
0004d009 mov ecx, dword ptr [0x04d008]
0004d00f lea ecx, dword ptr [ecx + ecx*2]
0004d012 lea ecx, dword ptr [ebx + ecx*8]
0004d015 cmp ebx, ecx
0004d017 mov ecx, 0
0004d01c cmovae ebx, ecx
0004d021 mov ecx, ebx
0004d021 mov eax, dword ptr [0x04d028]
0004d026 mov ebx, dword ptr [esi]
0004d029 mov esi, 0x0000
0004d02e xor ebx, esi
0004d030 inc byte ptr [eax + esi]
0004d033 mov dword ptr [esi], 0x7000
0004d036 movzx ebx, byte ptr [0x04d036]
0004d041 shl ebx, 0x10
0004d044 movzx eax, word ptr [0x04d040]
0004d04d or esi, esi
0004d054 movzx ebx, al
0004d056 cmp ebx, 8
0004d05b jne 0x04d074
0004d05b mov ebx, dword ptr [esi]
0004d05f mov esi, dword ptr [esi]
0004d062 mov ebx, 0x04c1f6
0004d067 xor esi, ebx
0004d069 inc byte ptr [ebx + esi]
0004d06c mov dword ptr [esi], 0x1007
0004d073 test al, al
0004d075 jz 0x04d07f
0004d07b mov ebx, dword ptr [0x04d028]
```

Expr
edx
OK

id:000008_src:000001_op:havoc_rep:8_cov

```
00000000: 0000 1337 0000 0000 3730 0000 0000 0000
00000010: 3df8 0600 0000 0000 0204 000f a85d 492b
00000020: 0404 004d c0bb e574 0708 00dc a741 0c05
00000030: aa10 361b 040c 00da bf00 9138 3a50 1ba2
00000040: c381 a37b 0908 00ff 9853 25c5 15cd 5991
00000050: 0e08 005f 30b5 89fd 294f 454f
```

```
...7...70....
=.....]I+
..M...t....A..
..6.....:8:P..
...{.....S%...Y.
..._0...)0E0
```

Watch Points

Address	Value
00000000	00000000
00000010	00000000
00000020	00000000
00000030	00000000
00000040	00000000
00000050	00000000

A Better Tool...

LJON Dashboard Coverage Queries

Files Transitions Inputs (6) Search Filter

Taken:

loop 000

NotTaken:

loop 001

/home/me/ljon_evil/cb-multios/challenges/Multipass/src/main.c 76/170

```
transaction = NULL;
if (last_id != NULL && *last_id != pkthdr.transaction_id)
    cgc_send_error(ERRNO_MP_NOT_FOUND, NOT_FOUND_MSG);
goto fail;

transaction_t *last_transaction = lookup_transaction(pkthdr.transaction_id);
ABR 1 if (pkthdr.op_code == ISSUE)
    if (pkthdr.pkt_type == ENIT)
        transaction = cgc_new_transaction();
        if (!transaction)
            cgc_send_error(ERRNO_MP_ALLOC, ALLOC_MSG);
            goto fail;

        if (cgc_read_data(transaction) != 0)
            cgc_send_error(ERRNO_MP_SINK, SINK_ERROR_MSG);
            goto fail;

        /* [source] == [target] ? */
        goto fail;

        last_id = &transaction->id;
    else if (pkthdr.pkt_type == FIN && !last_transaction)
        transaction = last_transaction;

xor esi, esi
cmp esi, esi
jz 0x00400021
mov ecx, dword ptr [0x00400020]
mov ebx, dword ptr [esi]
mov esi, 0x10ff
xor ebx, esi
inc byte ptr [ecx + esi]
mov dword ptr [esi], 0xadf
mov ecx, dword ptr [0x00400020]
lea ecx, dword ptr [ecx + ecx*2]
mov ecx, dword ptr [ebx + ecx*8]
cmp ebx, ecx
mov ebx, 0
cmovae ebx, ecx
mov ecx, esi
mov eax, dword ptr [0x00400020]
mov ebx, dword ptr [esi]
mov esi, 0x0000
xor ebx, esi
inc byte ptr [eax + esi]
mov dword ptr [esi], 0x700
movzx ebx, byte ptr [0x00400020]
shl ebx, 0x10
movzx eax, word ptr [0x00400020]
or esi, esi
movzx ebx, sb
cmp ebx, 8
jnc 0x00400027
mov ebx, dword ptr [0x00400020]
mov esi, dword ptr [esi]
mov ebx, 0x1000f
xor esi, ebx
inc byte ptr [ebx + esi]
mov dword ptr [esi], 0x1000f
test al, al
jz 0x00400027
mov ebx, dword ptr [0x00400020]
```

id:000008,src:000001,op:havoc,rep:8,+cov

```
00000000: 0000 1337 0000 0000 3730 0000 0000 0000
00000010: 3df8 0600 0000 0000 0204 000f a85d 492b
00000020: 0404 004d c0bb e574 0708 00dc a741 0c05
00000030: aa10 361b 040c 00da bf00 9138 3a50 1ba2
00000040: c381 a37b 0908 00ff 9853 25c5 15cd 5991
00000050: 0e08 005f 30b5 89fd 294f 454f
```

```
...7...70....
=.....]I+
..M...t....A..
..6.....:8:P..
...{.....S%...Y.
..._0...)0E0
```

Watch Points

0804d050: edx

A Better Tool...

LJON Dashboard Coverage Queries

Files Transitions Inputs (6) Search Filter

Taken: 0x00000000
NotTaken: 0x00000001

/home/me/ljon_evil/cb-multios/challenges/MultiPass/src/main.c 76/170

```
transaction = NULL;
if (last_id != NULL && *last_id != pkthdr.transaction_id)
    cgc_send_error(ERRNO_MP_NOT_FOUND, NOT_FOUND_MSG);
    goto fail;

transaction_t *last_transaction = lookup_transaction(pkthdr.transaction_id);
ABR 1 if (pkthdr.op_code == ISSUE)
    if (pkthdr.pkt_type == ENIT)
        transaction = cgc_new_transaction();
        if (!transaction)
            cgc_send_error(ERRNO_MP_ALLOC, ALLOC_MSG);
            goto fail;

        if (cgc_read_data(transaction) != 0)
            cgc_send_error(ERRNO_MP_SINK, SINK_ERROR_MSG);
            goto fail;

        /* [source] == [dest] ? */
        goto fail;

        last_id = &transaction->id;
    else if (pkthdr.pkt_type == FIN && !last_transaction)
        transaction = last_transaction;

__asm__ volatile (
    "xor    ecx, ecx\n",
    "cmp    eax, ebx\n",
    "jnb    0x040021\n",
    "mov    ecx, dword ptr [0x040020]\n",
    "mov    ebx, dword ptr gs:[edi]\n",
    "mov    esi, 0x10ff\n",
    "xor    ebx, esi\n",
    "inc    byte ptr [ecx + esi]\n",
    "mov    dword ptr gs:[edi], ebx\n",
    "mov    ecx, dword ptr [0x040020]\n",
    "lea    ecx, dword ptr [ecx + ecx*2]\n",
    "mov    ecx, dword ptr [ebx + ecx*4]\n",
    "cmp    ebx, ecx\n",
    "mov    ecx, 0\n",
    "cmovbe ecx, ecx\n",
    "mov    ecx, eax\n",
    "mov    eax, dword ptr [0x040020]\n",
    "mov    ebx, dword ptr gs:[edi]\n",
    "mov    esi, 0x0000\n",
    "xor    ebx, esi\n",
    "inc    byte ptr [eax + ebx]\n",
    "mov    dword ptr gs:[edi], ebx\n",
    "movzx  ebx, byte ptr [0x040020]\n",
    "shl    ebx, 0x10\n",
    "movzx  eax, word ptr [0x040020]\n",
    "or     esi, esi\n",
    "movzx  ebx, byte ptr [esi]\n",
    "cmp    ebx, 8\n",
    "jnb    0x040027\n",
    "mov    ebx, dword ptr [0x040020]\n",
    "mov    esi, dword ptr gs:[edi]\n",
    "mov    ebx, 0x1000f\n",
    "xor    esi, ebx\n",
    "inc    byte ptr [ebx + esi]\n",
    "mov    dword ptr gs:[edi], 0x1000f\n",
    "test   al, al\n",
    "jne    0x040027\n",
    "mov    ebx, dword ptr [0x040020]\n",
    "mov    ebx, dword ptr [0x040020]"
);
```

id:000008,src:000001,op:havoc,rep:8,+cov

```
00000000: 0000 1337 0000 0000 3730 0000 0000 0000
00000010: 3df8 0600 0000 0000 0204 000f a85d 492b
00000020: 0404 004d c0bb e574 0708 00dc a741 0c05
00000030: aa10 361b 040c 00da bf00 9138 3a50 1ba2
00000040: c381 a37b 0908 00ff 9853 25c5 15cd 5991
00000050: 0e08 005f 30b5 89fd 294f 454f
```

```
...7...70....
=.....]I+
..M...t....A..
..6.....:8:P..
...{.....S%...Y.
..._0...)0E0
```

Watch Points

0804d050: ecx

A Better Tool...

LJON Dashboard Coverage Queries

Files Transitions Inputs (6) Search Filter

Taken:

loop 000

NotTaken:

loop 001

/home/me/ljon_evil/cb-multios/challenges/Multitpass/src/main.c 76/170

```
transaction = NULL;
if (last_id != NULL && *last_id != pkthdr.transaction_id)
    cgc_send_error(ERRNO_MP_NOT_FOUND, NOT_FOUND_MSG);
    goto fail;

transaction_t *last_transaction = lookup_transaction(pkthdr.transaction_id);
ABR 1 if (pkthdr.op_code == ISSUE)
    if (pkthdr.pkt_type == ENIT)
        transaction = cgc_new_transaction();
        if (!transaction)
            cgc_send_error(ERRNO_MP_ALLOC, ALLOC_MSG);
            goto fail;

        if (cgc_read_data(transaction) != 0)
            cgc_send_error(ERRNO_MP_SINK, SINK_ERROR_MSG);
            goto fail;

        /* [source] == [dest] ? */
        goto fail;

        last_id = &transaction->id;
    else if (pkthdr.pkt_type == FIN && !last_transaction)
        transaction = last_transaction;

__asm__ volatile ("xor %eax, %eax\n");
__asm__ volatile ("cmp %eax, %ebx\n");
__asm__ volatile ("jlt %ebx,%eax,1\n");
__asm__ volatile ("mov %eax, %edx\n");
__asm__ volatile ("mov %ebx, %ecx\n");
__asm__ volatile ("inc byte ptr [%eax + %edx]\n");
__asm__ volatile ("mov %edx, ptr [%eax + %edx]\n");
__asm__ volatile ("lea %ecx, %edx\n");
__asm__ volatile ("cpl %eax, %ecx\n");
__asm__ volatile ("mov %ecx, 0\n");
__asm__ volatile ("cmovbe %eax, %ecx\n");
__asm__ volatile ("mov %ecx, %eax\n");
__asm__ volatile ("mov %eax, %edx\n");
__asm__ volatile ("mov %edx, %ecx\n");
__asm__ volatile ("xor %edx, %ecx\n");
__asm__ volatile ("inc byte ptr [%eax + %edx]\n");
__asm__ volatile ("mov %edx, ptr [%eax + %edx]\n");
__asm__ volatile ("movzx %eax, byte ptr [%eax + %edx]\n");
__asm__ volatile ("shl %edx, %edx\n");
__asm__ volatile ("movzx %eax, word ptr [%eax + %edx]\n");
__asm__ volatile ("or %eax, %edx\n");
__asm__ volatile ("movzx %edx, %eax\n");
__asm__ volatile ("cpl %edx, %eax\n");
__asm__ volatile ("jng %ebx,%edx,1\n");
__asm__ volatile ("mov %edx, %ecx\n");
__asm__ volatile ("mov %ecx, %edx\n");
__asm__ volatile ("xor %esi, %edx\n");
__asm__ volatile ("inc byte ptr [%edx + %esi]\n");
__asm__ volatile ("mov %edx, ptr [%edx + %esi]\n");
__asm__ volatile ("test %al, %al\n");
__asm__ volatile ("jle %ebx,%edx,1\n");
__asm__ volatile ("mov %edx, %ecx\n");
```

id:000008_src:000001_op:havoc_rep:8_cov

```
00000000: 0000 1337 0000 0000 3730 0000 0000 0000
00000010: 3df8 0600 0000 0000 0204 000f a85d 492b
00000020: 0404 004d c0bb e574 0708 00dc a741 0c05
00000030: aa10 361b 040c 00da bf00 9138 3a50 1ba2
00000040: c381 a37b 0908 00ff 9853 25c5 15cd 5991
00000050: 0e08 005f 30b5 89fd 294f 454f
```

```
...7...70....
=.....]I+
..M...t....A..
..6.....:8:P..
...{.....S%...Y.
..._0...)0E0
```

Watch Points

0804d050: edx Analyze Input

A Better Tool...

LJON Dashboard Coverage Queries

Files Transitions Inputs (6) Search Filter

Taken: 0x00000000
NotTaken: 0x00000001

/home/me/ljon_evil/cb-multios/challenges/Multitpass/src/main.c 76/170

```
...
775 transaction = NULL;
776 if (last_id != NULL && *last_id != pkthdr.transaction_id)
777     cgc_send_error(ERRNO_MP_NOT_FOUND, NOT_FOUND_MSG);
778     goto FAIL;
779
780 transaction_t *last_transaction = lookup_transaction(pkthdr.transaction_id);
781
782 ABR if (pkthdr.op_code == ISSUE)
783     if (pkthdr.pkt_type == ENIT)
784         transaction = cgc_new_transaction();
785         if (!transaction)
786             cgc_send_error(ERRNO_MP_ALLOC, ALLOC_MSG);
787             goto FAIL;
788
789         if (cgc_read_data(transaction) != 0)
790             cgc_send_error(ERRNO_MP_SINK, SINK_ERROR_MSG);
791             goto FAIL;
792
793         /* [source] == [dest] ? */
794         goto FAIL;
795
796         last_id = &transaction->id;
797
798     else if (pkthdr.pkt_type == FIN && !last_transaction)
799         transaction = last_transaction;
800
801 ...
```

```
...
0004c1e9 xor esi, esi
0004c1eb cmp esi, esi
0004c1ed jk 0x0040021
0004c1ef mov ecx, dword ptr [0x0050020]
0004c1f5 mov ebx, dword ptr gs:[edi]
0004c1f8 mov esi, 0x10ff
0004c1ff xor ebx, esi
0004c202 inc byte ptr [ecx + esi]
0004c205 mov dword ptr gs:[edi], 0x00ff
0004c208 mov ecx, dword ptr [ecx + ecx*2]
0004c212 lea ecx, dword ptr [ebx + ecx*4]
0004c215 cmp ebx, ecx
0004c217 mov ecx, 0
0004c21c cmovae ebx, ecx
0004c21f mov ecx, esi
0004c221 mov eax, dword ptr [0x0050020]
0004c224 mov ebx, dword ptr gs:[edi]
0004c227 mov esi, 0x0000
0004c22a xor ebx, esi
0004c22d inc byte ptr [eax + esi]
0004c230 mov dword ptr gs:[edi], 0x7000
0004c233 movzx ebx, byte ptr [0x0050020]
0004c236 shl ebx, 0x10
0004c239 movzx eax, word ptr [0x0050020]
0004c23c or esi, esi
0004c23e movzx ebx, sb
0004c240 cmp ebx, 0
0004c243 jng 0x0040174
0004c246 mov ebx, dword ptr [0x0050020]
0004c249 mov esi, dword ptr gs:[edi]
0004c24c mov ebx, 0x200f
0004c24f xor esi, ebx
0004c251 inc byte ptr [ebx + esi]
0004c254 mov dword ptr gs:[edi], 0x1007
0004c257 test al, al
0004c259 jz 0x004017f
0004c25b mov ebx, dword ptr [0x0050020]
```

id:000008_src:000001_op:havoc_rep:8,+cov

```
00000000: 0000 1337 0000 0000 3730 0000 0000 0000
00000010: 3df8 0600 0000 0000 0204 000f a85d 492b
00000020: 0404 004d c0bb e574 0708 00dc a741 0c05
00000030: aa10 361b 040c 00da bf00 9138 3a50 1ba2
00000040: c381 a37b 0908 00ff 9853 25c5 15cd 5991
00000050: 0e08 005f 30b5 89fd 294f 454f
```

```
...7...70.....
=.....]I+
..M...t....A..
..6.....:8:P..
...{.....S%...Y.
..._0...)0E0
```

Watch Points

0804d050: edx Analyze Input

A Better Tool...

LJON Dashboard Coverage Queries

Files Transitions Inputs (6) Search Filter

Taken: 0x000000
NotTaken: 0x000001

/home/me/ljon_evsl/cb-multios/challenges/Multitpass/src/main.c 76/170

```
transaction = NULL;
if (last_id != NULL && *last_id != pkthdr.transaction_id)
    cgc_send_error(ERRNO_MP_NOT_FOUND, NOT_FOUND_MSG);
goto FAIL;

transaction_t *last_transaction = lookup_transaction(pkthdr.transaction_id);
ABR 1 if (pkthdr.op_code == ISSUE)
    if (pkthdr.pkt_type == ENIT)
        transaction = cgc_new_transaction();
        if (!transaction)
            cgc_send_error(ERRNO_MP_ALLOC, ALLOC_MSG);
            goto FAIL;
        if (cgc_read_data(transaction) != 0)
            cgc_send_error(ERRNO_MP_SINK, SINK_ERROR_MSG);
            goto FAIL;
        /* [source] == [target] ? */
        goto FAIL;

        last_id = &transaction->id;
    else if (pkthdr.pkt_type == FIN && !last_transaction)
        transaction = last_transaction;
```

```
0004c7e0 xor esi, esi
0004c7e1 cmp eax, ebx
0004c7e2 jb 0x004d0021
0004c7e3 mov ecx, dword ptr [0x00000020]
0004c7e4 mov ebx, dword ptr [esi]
0004c7e5 mov esi, 0x10ff
0004c7e6 xor ebx, esi
0004c7e7 inc byte ptr [ecx + ebx]
0004c7e8 mov dword ptr [esi], 0xadf
0004c7e9 mov ecx, dword ptr [0x00000020]
0004c7ea lea ecx, dword ptr [ecx + ecx*2]
0004c7eb lea ecx, dword ptr [ebx + ecx*4]
0004c7ec cmp ebx, ecx
0004c7ed mov ecx, 0
0004c7ee cmovae ebx, ecx
0004c7ef mov ecx, ebx
0004c7f0 mov eax, dword ptr [0x00000020]
0004c7f1 mov ebx, dword ptr [esi]
0004c7f2 mov esi, 0x0000
0004c7f3 xor ebx, esi
0004c7f4 inc byte ptr [eax + ebx]
0004c7f5 mov dword ptr [esi], 0x700
0004c7f6 movzx ebx, byte ptr [0x00000020]
0004c7f7 shl ebx, 0x10
0004c7f8 movzx eax, word ptr [0x00000020]
0004c7f9 or esi, esi
0004c7fa movzx ebx, sb
0004c7fb cmp ebx, 0
0004c7fc jne 0x004d0074
0004c7fd mov ebx, dword ptr [0x00000020]
0004c7fe mov esi, dword ptr [esi]
0004c7ff mov ebx, 0x1000f
0004c800 xor esi, ebx
0004c801 inc byte ptr [ebx + esi]
0004c802 mov dword ptr [esi], 0x100f
0004c803 test al, al
0004c804 jz 0x004d007f
0004c805 mov ebx, dword ptr [0x00000020]
```

id:000008_src:000001_op:havoc_rep:8,+cov

```
00000000: 0000 1337 0000 0000 3730 0000 0000 0000
00000010: 3df8 0600 0000 0000 0204 000f a85d 492b
00000020: 0404 004d c0bb e574 0708 00dc a741 0c05
00000030: aa10 361b 040c 00da bf00 9138 3a50 1ba2
00000040: c381 a37b 0908 00ff 9853 25c5 15cd 5991
00000050: 0e08 005f 30b5 89fd 294f 454f
```

```
...7...70....
=.....]I+
...M...t....A..
..6.....:8:P..
...{.....S%...Y.
..._0...)0E0
```

Watch Points

0804d050: edx

A Better Tool...

LJON Dashboard Coverage Queries

Files Transitions Inputs (6) Search Filter

Taken: 0x000000
NotTaken: 0x000000

/home/me/ljon_evsl/cb-multios/challenges/Multipass/src/main.c 76/170

```
...
775 transaction = NULL;
776 if (last_id != NULL && *last_id != pkthdr.transaction_id)
777     cgc_send_error(ERRNO_MP_NOT_FOUND, NOT_FOUND_MSG);
778     goto FAIL;
779
780 transaction = lookup_transaction(pkthdr.transaction_id);
781
782 ABR if (pkthdr.op_code == ISSUE)
783     if (pkthdr.pkt_type == ENIT)
784         transaction = cgc_new_transaction();
785         if (!transaction)
786             cgc_send_error(ERRNO_MP_ALLOC, ALLOC_MSG);
787             goto FAIL;
788
789         if (cgc_read_data(transaction) != 0)
790             cgc_send_error(ERRNO_MP_SINK, SINK_ERROR_MSG);
791             goto FAIL;
792
793         last_id = &transaction->id;
794     else if (pkthdr.pkt_type == FIN && !last_transaction)
795         transaction = last_transaction;
796
797     ...

```

```
...
0004cfe9 xor esi, esi
0004cfef cmp eax, ebx
0004cfef jk 0x040021
0004cfef mov ecx, dword ptr [0x040020]
0004cf75 mov ebx, dword ptr [esi]
0004cf78 mov esi, 0x10ff
0004cf7f xor ebx, esi
0004cf7f inc byte ptr [ecx + ebx]
0004cf80 mov dword ptr [esi], 0x04f
0004cf89 mov ecx, dword ptr [0x040020]
0004cf8f lea ecx, dword ptr [ecx + ecx*2]
0004cf92 lea ecx, dword ptr [ebx + ecx*4]
0004cf95 cmp ebx, ecx
0004cf97 mov ebx, 0
0004cf9c cmovbe ebx, ecx
0004cf9f mov ecx, ebx
0004cf9f mov eax, dword ptr [0x040020]
0004cf9f mov ebx, dword ptr [esi]
0004cf9f mov esi, 0x0000
0004cf9f xor ebx, esi
0004cf9f inc byte ptr [eax + ebx]
0004cf9f mov dword ptr [esi], 0x700
0004cf9f movzx ebx, byte ptr [0x040020]
0004cf9f shl ebx, 0x10
0004cf9f movzx eax, word ptr [0x040020]
0004cf9f or ebx, edx
0004cf9f movzx ebx, sb
0004cf9f cmp ebx, 0
0004cf9f jne 0x040021
0004cf9f mov ebx, dword ptr [0x040020]
0004cf9f mov esi, dword ptr [esi]
0004cf9f mov ebx, dword ptr [0x040020]

```



if changed, we don't reach the breakpoint

```
id:000008_src:000001_op:havoc_rep:8,+
00000000: 0000 1337 0000 0000 3730 0000 0000 0000
00000010: 3df8 0600 0000 0000 0204 000f a85d 492b
00000020: 0404 004d c0bb e574 0708 00dc a741 0c05
00000030: aa10 361b 040c 00da bf00 9138 3a50 1ba2
00000040: c381 a37b 0908 00ff 9853 25c5 15cd 5991
00000050: 0e08 005f 30b5 89fd 294f 454f
```

Watch Points	
0804d050:	edx

A Better Tool...

LJON Dashboard Coverage Queries

Files Transitions Inputs (6) Search Filter

Taken: 0x000000
NotTaken: 0x000000

/home/me/ljon_evil/cb-multios/challenges/Multipass/src/main.c 76/170

```
transaction = NULL;
if (last_id != NULL && *last_id != pkthdr.transaction_id)
    cgc_send_error(ERRNO_MP_NOT_FOUND, NOT_FOUND_MSG);
goto FAIL;

transaction = lookup_transaction(pkthdr.transaction_id);
ABR  if (pkthdr.op_code == ISSUE)
    if (pkthdr.pkt_type == ENIT)
        transaction = cgc_new_transaction();
        if (!transaction)
            cgc_send_error(ERRNO_MP_ALLOC, ALLOC_MSG);
            goto FAIL;

        if (cgc_read_data(transaction) != 0)
            cgc_send_error(ERRNO_MP_SINK, SINK_ERROR_MSG);
            goto FAIL;

        /* [source] == [target] */
        goto FAIL;

        last_id = &transaction->id;

    else if (pkthdr.pkt_type == FIN && !transaction)
        transaction = last_transaction;
```

```
0004cfe9 xor esi, esi
0004cfef cmp esi, ebx
0004cfef jk 0x040021
0004cfef mov ecx, dword ptr [0x040020]
0004cf75 mov ebx, dword ptr [esi]
0004cf78 mov esi, 0x10ff
0004cf7f xor ebx, esi
0004cfff inc byte ptr [ecx + ebx]
0004d002 mov dword ptr [esi], ebx
0004d009 mov ecx, dword ptr [0x040020]
0004d00f lea ecx, dword ptr [ecx + ecx*2]
0004d012 lea ecx, dword ptr [ebx + ecx*4]
0004d015 cmp ebx, ecx
0004d017 mov ecx, 0
0004d01c cmovbe ebx, ecx
0004d021 mov ecx, ebx
0004d021 mov eax, dword ptr [0x040020]
0004d026 mov ebx, dword ptr [esi]
0004d029 mov esi, 0x0000
0004d02c xor ebx, esi
0004d030 inc byte ptr [eax + ebx]
0004d033 mov dword ptr [esi], ebx
0004d036 movzx ebx, byte ptr [0x040020]
0004d041 shl ebx, 0x10
0004d044 movzx eax, word ptr [0x040020]
0004d04b or esi, eax
0004d04d movzx ebx, al
0004d050 cmp ebx, 0
0004d053 jne 0x040074
0004d059 mov ebx, dword ptr [0x040020]
0004d05f mov esi, dword ptr [esi]
0004d062 mov ebx, dword ptr [esi]
0004d067 xor esi, ebx
```

directly affects the value

id:000008_src:000001_op:havoc_rep:8_cov

```
00000000: 0000 1337 0000 0000 3...000 0000 0000
00000010: 3df8 0600 0000 0000 0204 000f a85d 492b
00000020: 0404 004d c0bb e574 0708 00dc a741 0c05
00000030: aa10 361b 040c 00da bf00 9138 3a50 1ba2
00000040: c381 a37b 0908 00ff 9853 25c5 15cd 5991
00000050: 0e08 005f 30b5 89fd 294f 454f
```

```
...7...70....
=.....]I+
..M...t....A..
..6.....8:P..
...{....S%...Y.
..._0...)0E0
```

Watch Points

0804d050: edx

A Better Tool...

LJON Dashboard Coverage Queries

Files Transitions Inputs (6) Search Filter

Taken: 0x000000
NotTaken: 0x000000

/home/me/ljon_evil/cb-multios/challenges/Multitpass/src/main.c 76/170

```
transaction = NULL;
if (last_id != NULL && *last_id != pkthdr.transaction_id)
    cgc_send_error(ERRNO_MP_NOT_FOUND, NOT_FOUND_MSG);
goto FAIL;

transaction = lookup_transaction(pkthdr.transaction_id);
ABR  if (pkthdr.op_code == ISSUE)
    if (pkthdr.pkt_type == ENIT)
        transaction = cgc_new_transaction();
        if (!transaction)
            cgc_send_error(ERRNO_MP_ALLOC, ALLOC_MSG);
            goto FAIL;

        if (cgc_read_data(transaction) != 0)
            cgc_send_error(ERRNO_MP_SINK, SINK_ERROR_MSG);
            goto FAIL;

        if (!strcmp((const char *)data, "I"))
            goto FAIL;

        last_id = &transaction->id;
    else if (pkthdr.pkt_type == FIN && !last_transaction)
        transaction = last_transaction;
```

```
0004cfe0 xor esi, esi
0004cfe1 cmp esi, ebx
0004cfe2 jk 0x0040021
0004cfe7 mov ecx, dword ptr [0x0050020]
0004cfe7 mov ebx, dword ptr [esi]
0004cfe8 mov esi, 0x10ff
0004cfe9 xor ebx, esi
0004cfe9 inc byte ptr [ecx + ebx]
0004cfe9 mov dword ptr [esi], 0x00ff
0004cfe9 mov ecx, dword ptr [0x0050020]
0004cfe9 lea ecx, dword ptr [ecx + ecx*2]
0004cfe9 lea ecx, dword ptr [ebx + ecx*8]
0004cfe9 cmp ebx, ecx
0004cfe9 mov ecx, 0
0004cfe9 cmovbe esi, ecx
0004cfe9 mov ecx, esi
0004cfe9 mov esi, dword ptr [0x0050020]
0004cfe9 mov ebx, dword ptr [esi]
0004cfe9 mov esi, 0x0000
0004cfe9 xor ebx, esi
0004cfe9 inc byte ptr [eax + ebx]
0004cfe9 mov dword ptr [esi], 0x7000
0004cfe9 movzx ebx, byte ptr [0x0050020]
0004cfe9 shl ebx, 0x10
0004cfe9 movzx eax, word ptr [0x0050020]
0004cfe9 or esi, eax
0004cfe9 movzx ebx, sb
0004cfe9 cmp ebx, 8
0004cfe9 jnc 0x0040174
0004cfe9 mov ebx, dword ptr [0x0050020]
0004cfe9 mov esi, dword ptr [esi]
0004cfe9 mov ebx, 0x0000
0004cfe9 xor esi, ebx
0004cfe9 inc byte ptr [ebx + esi]
ptr [esi], 0x10ff
si
DIY
dword ptr [0x0050020]
```



doesn't matter

```
id:000008,src:000001,op:havoc,ret:0x00000000
00000000: 0000 1337 0000 0000 0000 3730 0000 0000 0000
00000010: 3df8 0600 0000 0000 0204 000f a85d 492b
00000020: 0404 004d c0bb e574 0708 00dc a741 0c05
00000030: aa10 361b 040c 00da bf00 9138 3a50 1ba2
00000040: c381 a37b 0908 00ff 9853 25c5 15cd 5991
00000050: 0e08 005f 30b5 89fd 294f 454f
```

```
...7...70....
=.....]I+
..M...t....A..
..6.....:8:P..
...{....S%...Y.
..._0...)0E0
```

Watch Points	
0804d050:	edx

A Better Tool...

LJON Dashboard Coverage Queries

Files Transitions Inputs (6) Search Filter

Taken:

NotTaken:

/home/me/ljon_evil/cb-multios/challenges/Multypass/src/main.c 76/170

```
transaction = NULL;
if (last_id != NULL && *last_id != pkthdr.transaction_id)
    cgc_send_error(ERRNO_MP_NOT_FOUND, NOT_FOUND_MSG);
goto fail;

transaction_t *last_transaction = lookup_transaction(pkthdr.transaction_id);
ABR  if (pkthdr.op_code == ISSUE)
    if (pkthdr.pkt_type == ENIT)
        transaction = cgc_new_transaction();
        if (!transaction)
            cgc_send_error(ERRNO_MP_ALLOC, ALLOC_MSG);
            goto fail;

        if (cgc_read_data(transaction) != 0)
            cgc_send_error(ERRNO_MP_SINK, SINK_ERROR_MSG);
            goto fail;

        /* [source] == [dest] ? */
        goto fail;

        last_id = &transaction->id;
    else if (pkthdr.pkt_type == FIN && !last_transaction)
        transaction = last_transaction;
```

```
0004cfe9 xor esi, esi
0004cfef cmp eax, ebx
0004cfef jk 0x004001
0004cfef mov ecx, dword ptr [0x0050028]
0004cf75 mov ebx, dword ptr qi:[edi]
0004cf78 mov esi, 0x10ff
0004cf7e xor ebx, esi
0004cf7f inc byte ptr [ecx + ebx]
0004cf82 mov dword ptr qi:[edi], 0xadf
0004cf89 mov ecx, dword ptr [0x0050028]
0004cf8f lea ecx, dword ptr [ecx + ecx*2]
0004cf92 lea ecx, dword ptr [ebx + ecx*4]
0004cf95 cmp ebx, ecx
0004cf97 mov ecx, 0
0004cf9c cmovae ebx, ecx
0004cf9f mov ecx, ebx
0004cf9f mov eax, dword ptr [0x0050028]
0004cf9f mov ebx, dword ptr qi:[edi]
0004cf9f mov esi, 0x0000
0004cf9f xor ebx, esi
0004cf9f inc byte ptr [eax + ebx]
0004cf9f mov dword ptr qi:[edi], 0x700
0004cf9f movzx ebx, byte ptr [0x0050028]
0004cf9f shl ebx, 0x10
0004cf9f movzx eax, word ptr [0x0050028]
0004cf9f or esi, esi
0004cf9f movzx ebx, sb
0004cf9f cmp ebx, 0
0004cf9f jnc 0x0040174
0004cf9f mov ebx, dword ptr [0x0050028]
0004cf9f mov esi, dword ptr qi:[edi]
0004cf9f mov ebx, 0x20cf
0004cf9f xor esi, ebx
0004cf9f inc byte ptr [ebx + esi]
0004cf9f mov dword ptr qi:[edi], 0x10ef
0004cf9f test al, al
0004cf9f je 0x004037f
0004cf9f mov ebx, dword ptr [0x0050028]
```

id:000008_src:000001_op:havoc_rep:8_cov

```
00000000: 0000 1337 0000 0000
00000010: 3df8 0600 0000 0000 04 00ff a741 0c05
00000020: 0404 004d c0bb 0708 00dc a741 0c05
00000030: aa10 361b 040c 00da bf00 9138 3a50 1ba2
00000040: c381 a37b 0908 00ff 9853 25c5 15cd 5991
00000050: 0e08 005f 30b5 89fd 294f 454f
```

affects the number how often the BP was hit

A Better Tool...

LJON Dashboard Coverage Queries

Files Transitions Inputs (6) Search Filter

Taken: 0x000000
NotTaken: 0x000001

/home/me/ljon_evil/cb-multios/challenges/Multipass/src/main.c 76/170

```
transaction = NULL;
if (last_id != NULL && *last_id != pkthdr.transaction_id)
    cgc_send_error(ERRNO_MP_NOT_FOUND, NOT_FOUND_MSG);
goto FAIL;

transaction_t *last_transaction = lookup_transaction(pkthdr.transaction_id);
ABR 1 if (pkthdr.op_code == ISSUE)
    if (pkthdr.pkt_type == ENIT)
        transaction = cgc_new_transaction();
        if (!transaction)
            cgc_send_error(ERRNO_MP_ALLOC, ALLOC_MSG);
            goto FAIL;
        if (cgc_read_data(transaction) != 0)
            cgc_send_error(ERRNO_MP_SINK, SINK_ERROR_MSG);
            goto FAIL;
        if (!memcmp((void *)0, (void *)0, 1))
            goto FAIL;
        last_id = &transaction->id;
    else if (pkthdr.pkt_type == FIN && !last_transaction)
        transaction = last_transaction;
```

```
0004c7e0 xor esi, esi
0004c7e1 cmp eax, ebx
0004c7e2 jb 0x04c801
0004c7e3 mov ecx, dword ptr [0x04c808]
0004c7e4 mov ebx, dword ptr [esi]
0004c7e5 mov esi, 0x10ff
0004c7e6 xor ebx, esi
0004c7e7 inc byte ptr [ecx + ebx]
0004c7e8 mov dword ptr [esi], 0x04f
0004c7e9 mov ecx, dword ptr [0x04c808]
0004c7ea lea ecx, dword ptr [ecx + ecx*2]
0004c7eb lea ecx, dword ptr [ebx + ecx*8]
0004c7ec cmp ebx, ecx
0004c7ed mov ebx, 0
0004c7ee cmovae ebx, ecx
0004c7ef mov ecx, ebx
0004c7f0 mov eax, dword ptr [0x04c808]
0004c7f1 mov ebx, dword ptr [esi]
0004c7f2 mov esi, 0x0000
0004c7f3 xor ebx, esi
0004c7f4 inc byte ptr [eax + ebx]
0004c7f5 mov dword ptr [esi], 0x700
0004c7f6 movzx ebx, byte ptr [0x04c808]
0004c7f7 shl ebx, 0x10
0004c7f8 movzx eax, word ptr [0x04c808]
0004c7f9 or esi, eax
0004c7fa movzx ebx, sb
0004c7fb cmp ebx, 8
0004c7fc jnc 0x04c817
0004c7fd mov ebx, dword ptr [0x04c808]
0004c7fe mov esi, dword ptr [esi]
0004c7ff mov ebx, 0x04c7f
0004c800 xor esi, ebx
0004c801 inc byte ptr [ebx + esi]
0004c802 mov dword ptr [esi], 0x10ff
0004c803 test al, al
0004c804 jz 0x04c817
0004c805 mov ebx, dword ptr [0x04c808]
```

id:000008,src:000001,op:havoc,rep:8,+cov

```
00000000: 0000 1337 0000 0000 3730 0000 0000 0000
00000010: 3df8 0600 0000 0000 0204 000f a85d 492b
00000020: 0404 004d c0bb e574 0708 00dc a741 0c05
00000030: aa10 361b 040c 00da bf00 9138 3a50 1ba2
00000040: c381 a37b 0908 00ff 9853 25c5 15cd 5991
00000050: 0e08 005f 30b5 89fd 294f 454f
```

```
...7...70....
=.....]I+
...M...t....A..
..6.....:8:P..
...{.....S%...Y.
..._0...)0E0
```

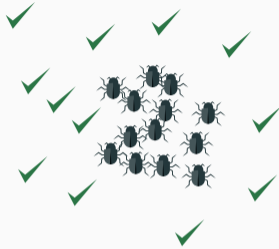
Watch Points

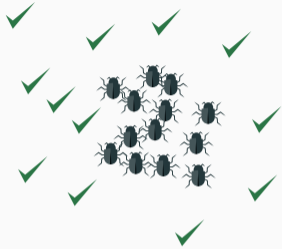
0804d050: edx

Root Cause Analysis

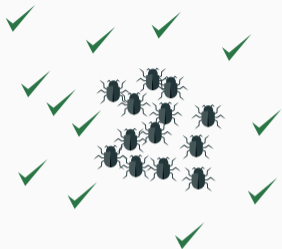


Root Cause Analysis



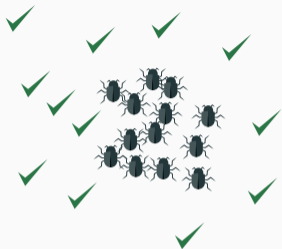


```
val.type != MRB_TT_EXCEPTION
```



```
val.type != MRB_TT_EXCEPTION
```

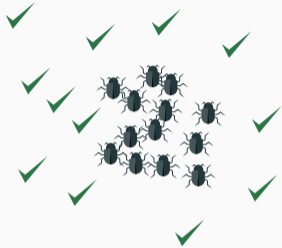
```
val.type < 123
```



```
val.type != MRB_TT_EXCEPTION
```

```
val.type < 123
```

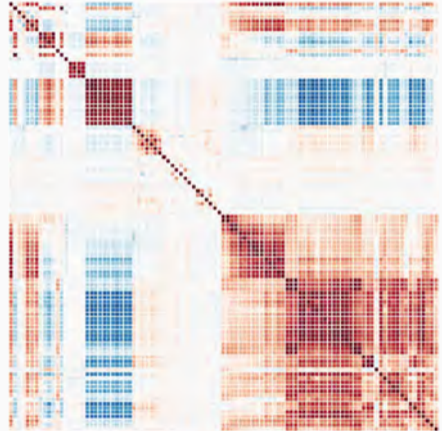
```
▪  
▪  
▪
```

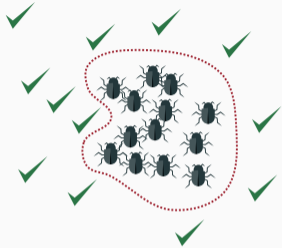


```
val.type != MRB_TT_EXCEPTION
```

```
val.type < 123
```

-
-
-

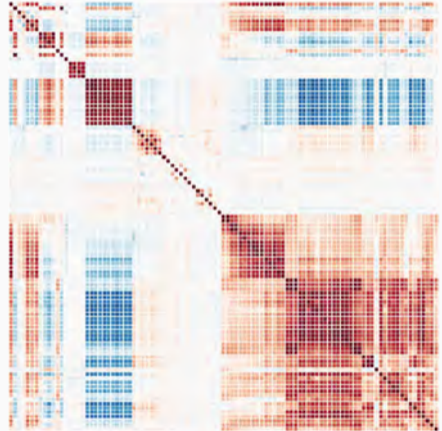




```
val.type != MRB_TT_EXCEPTION
```

```
val.type < 123
```

-
-
-





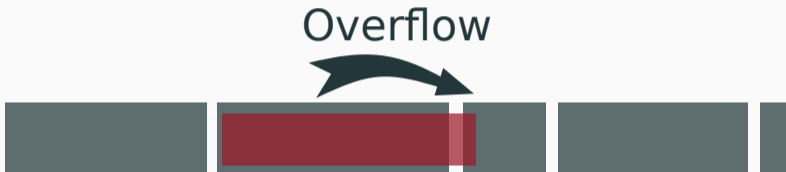
<https://sean.heelan.io/heaplayout/>



<https://sean.heelan.io/heaplayout/>



<https://sean.heelan.io/heaplayout/>



<https://sean.heelan.io/heaplayout/>

Key Takeaways:

printf

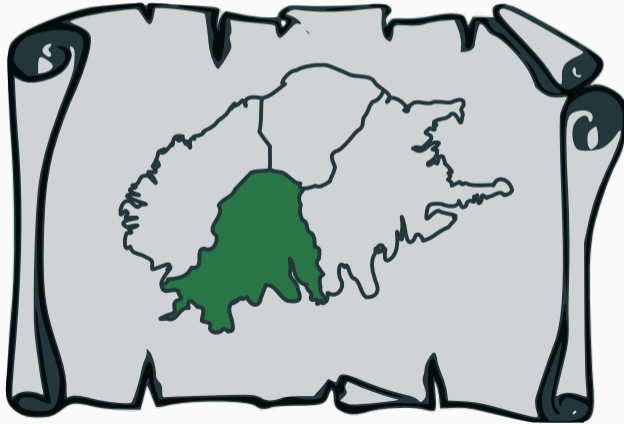
Debugger

**Time Traveling
Debugger**

Fuzzer + Debugger



Unfuzzable Code



Code that doesn't run?

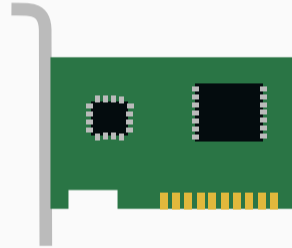
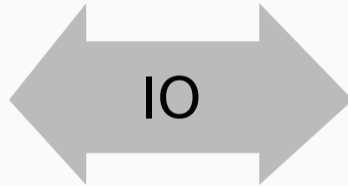
Code that doesn't run?

Firmware

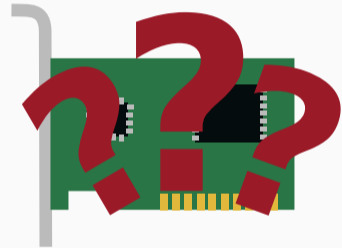
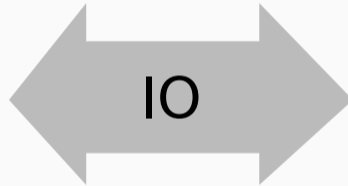


```
████████████████████  
██████████████████  
████████████████  
  ████████████████  
  ████████████████  
  ██████████  
████████████████  
  ██████████  
  ██████████  
  ██████████  
████████████████████  
██████████████████  
██████████████████  
██████████████████  
██████████████████  
██████████████████  
  ██████████  
  ██████████  
  ██████████  
██████████████████  
██████████████████  
██████████████████
```

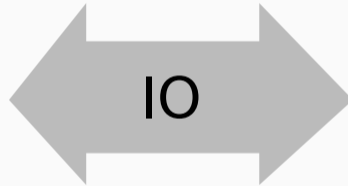
Code that doesn't run?



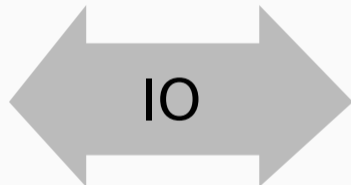
Code that doesn't run?



Code that doesn't run?



Code that doesn't run?



P²IM [1]
HALucinator [2]

[1] <https://arxiv.org/pdf/1909.06472.pdf>

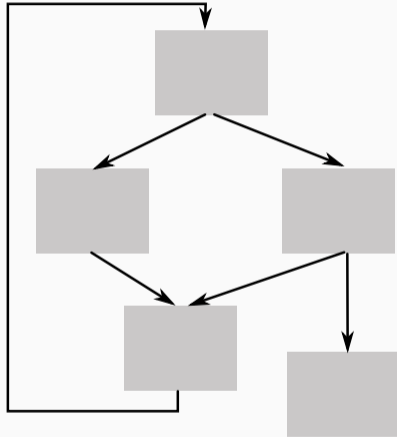
[2] <https://nebelwelt.net/publications/files/20SEC2.pdf>

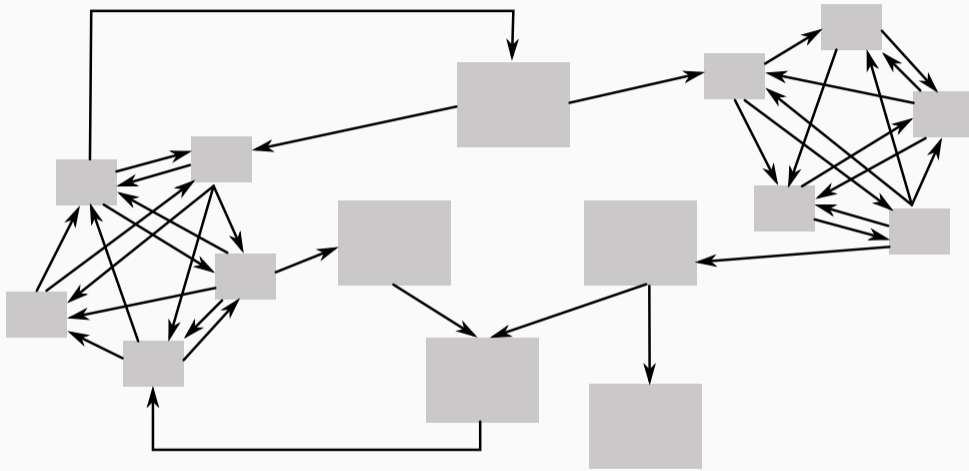
Key Takeaways:

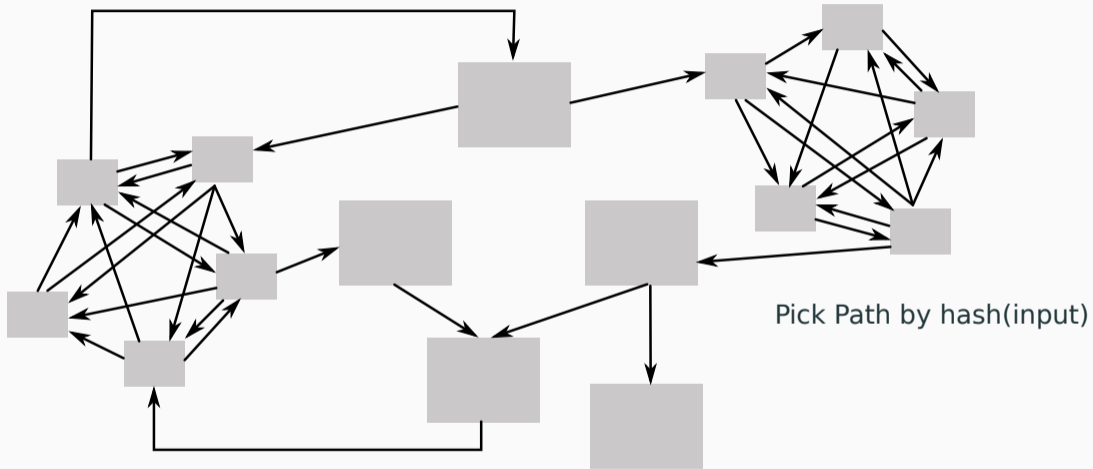


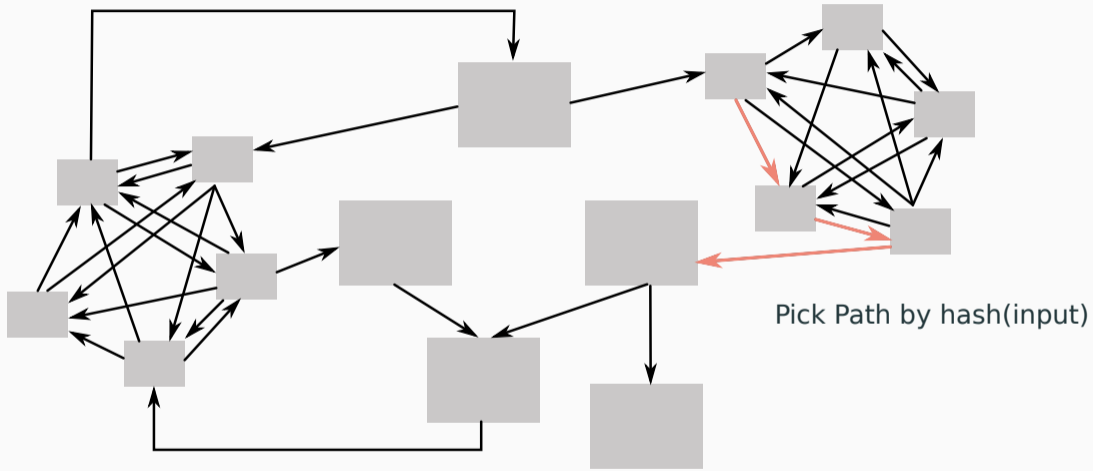
**You underestimate
the fuzzer**

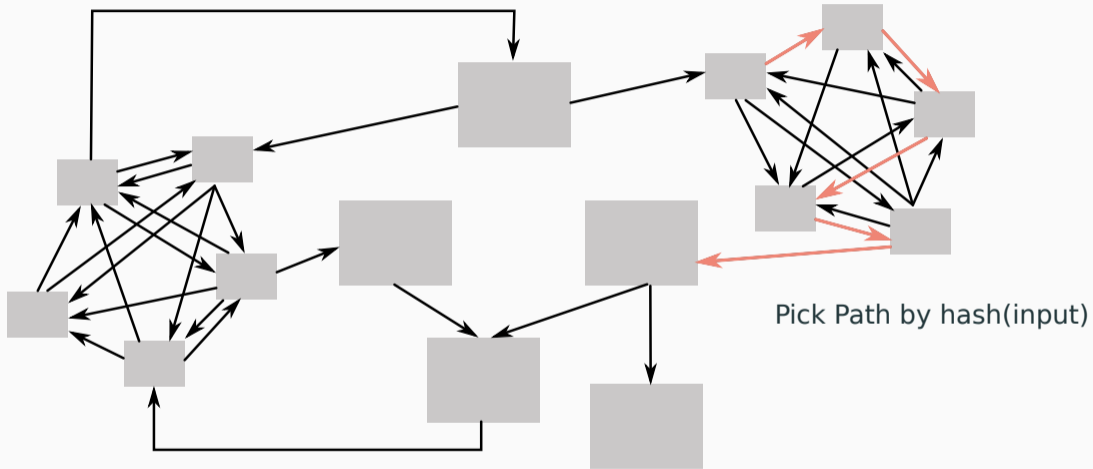
AntiFuzz















```
if(parse_error){  
    printf("couldn't parse\n");  
    exit(1);  
}
```



```
if(parse_error){  
    printf("couldn't parse\n");  
    delay(1); //expensive calc  
    exit(1);  
}
```

**SLOW
DOWN!**

```
if(parse_error){  
    printf("couldn't parse\n");  
    delay(1); //expensive calc  
    exit(1);  
}
```



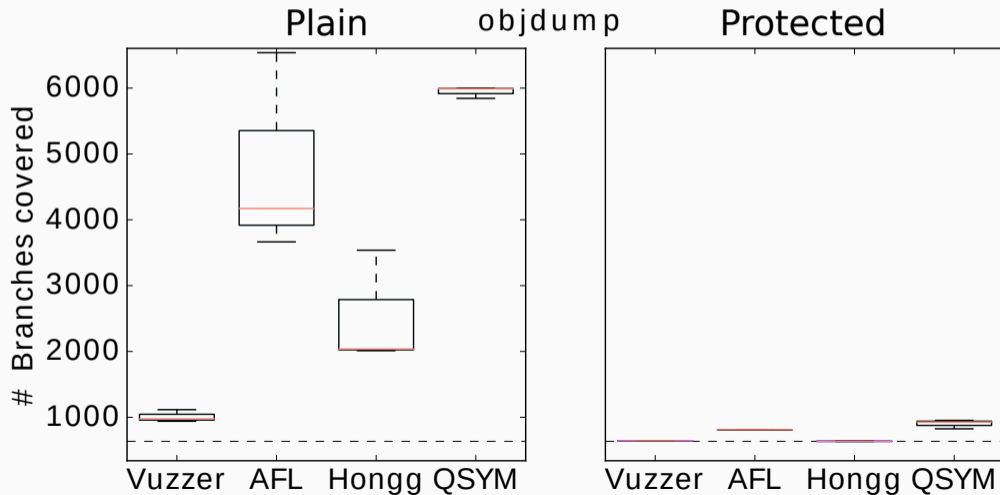
Signal Handler!

Symbolic Execution? Taint Tracking?

Symbolic Execution? Taint Tracking?

```
input = enc(dec(input));
```

objdump



Key Takeaways:



Katelyn Gadd @antumbral · 29.05

В отговор на [@johnregehr](#)

this sure seems like hostile research to me

"how can I ensure that software is full of vulnerabilities only known to me"



Katelyn Gadd @antumbral · 29.05

we need a term for hostile researchers like this, sort of the "you are undermining the future of the human race" equivalent of "class traitor"

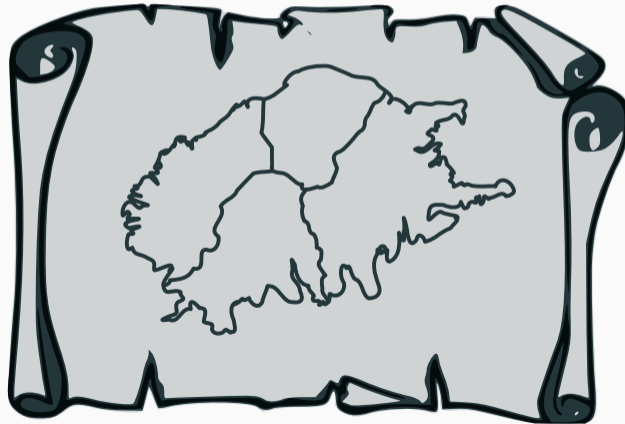


Key Takeaways:



**PROTECT SOFTWARE
AGAINST FUZZER**

DONT GET ANY BUG REPORTS



Fuzzers

As Building Blocks for a
New Generation of Tools



Let's Build Better Tools

github.com/RUB-SysSec/kAFL


github.com/RUB-SysSec/nautilus


github.com/RUB-SysSec/grimoire

github.com/RUB-SysSec/antifuzz


github.com/eqv/fuzz_ui





 @is_eqv

 github.com/eqv

 cornelius.aschermann@rub.de

 @ms_s3c

 github.com/schumilo

 sergej.schumilo@rub.de

Special Thanks to:

Ali Abbasi, Tim Blazytko, Robert Gawlik,
Emre Güler, Thorsten Holz, Moritz Schlögel,
Daniel Teuchert, Simone Wörner, and all the
others that made this research possible.

