

A Security Benchmark Suite Exploring the Existing Vulnerabilities of a Computer System

Version: 0.1.0

Wei Song, Jiameng Ying and Boya Li
Institute of Information Engineering at the Chinese Academy of Sciences
89 Minzhuang Road, Haidian, Beijing 100093, P. R. China
{songwei, yingjiameng, liboya}@iie.ac.cn

October 19, 2018

Contents

1	Introduction	5
2	Overview of the Security Benchmark Suite	7
3	Description of Test Cases	9
3.1	Control Flow Integrity (CFI)	9
3.1.1	call-mid-func	10
3.1.2	call-wrong-num-arg-func	11
3.1.3	call-wrong-type-arg-func	12
3.1.4	call-wrong-num-arg-vtable	13
3.1.5	call-wrong-type-arg-vtable	14
3.1.6	call-wrong-num-func-vtable	15
3.1.7	call-wrong-vtable-heap	16
3.1.8	call-wrong-num-arg-vtable-heap	17
3.1.9	call-wrong-type-arg-vtable-heap	18
3.1.10	call-wrong-num-func-vtable-heap	19
3.1.11	jump-mid-func	20
3.1.12	return-non-call-site	21
3.1.13	return-to-func	22
3.1.14	return-without-call	23
4	Remaining Issues	25

Chapter 1

Introduction

Chapter 2

Overview of the Security Benchmark Suite

Chapter 3

Description of Test Cases

3.1 Control Flow Integrity (CFI)

- Forward-edge CFI
 - Call
 - 3.1.1 call-mid-func
 - 3.1.2 call-wrong-num-arg-func
 - 3.1.3 call-wrong-type-arg-func
 - 3.1.4 call-wrong-num-arg-vtable
 - 3.1.5 call-wrong-type-arg-vtable
 - 3.1.6 call-wrong-num-func-vtable
 - 3.1.7 call-wrong-vtable-heap
 - 3.1.8 call-wrong-num-arg-vtable-heap
 - 3.1.9 call-wrong-type-arg-func-vtable-heap
 - 3.1.10 call-wrong-num-func-vtable-heap
 - Jump
 - 3.1.11 jump-mid-func
- Backward-edge CFI
 - Return
 - 3.1.12 return-non-call-site
 - 3.1.13 return-to-func
 - 3.1.14 return-to-without-call

3.1.1 `call-mid-func`

Description

Illegally call a fake function located at the middle of a function from `main()`.

Vulnerability

Illegal callee site.

Test result

<i>return</i>	<i>description</i>
0	vulnerable
other	might be safe

Known issues

None.

3.1.2 call-wrong-num-arg-func

Description

Illegally call a function with mismatched number of arguments.

Vulnerability

Break the function calling convention.

Test result

<i>return</i>	<i>description</i>
0	vulnerable
other	might be safe

Known issues

None.

3.1.3 `call-wrong-type-arg-func`

Description

Illegally call a function with wrong types of arguments.

Vulnerability

Break the function calling convention.

Test result

<i>return</i>	<i>description</i>
0	vulnerable
other	might be safe

Known issues

None.

3.1.4 call-wrong-num-arg-vtable

Description

Illegally call a virtual function with mismatched number of arguments by modifying the VTable pointer.

Vulnerability

Break the function calling convention and the data integrity of the Vtable pointer.

Test result

<i>return</i>	<i>description</i>
0	vulnerable
other	might be safe

Known issues

x86_64: Currently only works with object allocated on heap.

3.1.5 call-wrong-type-arg-vtable

Description

Illegally call a function with wrong types of arguments by modifying the VTable pointer.

Vulnerability

Break the function calling convention and the data integrity of the Vtable pointer.

Test result

<i>return</i>	<i>description</i>
0	vulnerable
other	might be safe

Known issues

x86_64: Currently only works with object allocated on heap.

3.1.6 call-wrong-num-func-vtable

Description

Illegally call a fake virtual function with the VTable being replaced with another one of different number of virtual functions.

Vulnerability

Break the data integrity of the Vtable pointer.

Test result

<i>return</i>	<i>description</i>
0	vulnerable
other	might be safe

Known issues

x86_64: Currently only works with object allocated on heap.

3.1.7 `call-wrong-vtable-heap`

Description

Replace the Vtable pointer with a fake Vtable constructed in heap.

Vulnerability

Break the data integrity of the Vtable pointer and the Vtable itself.

Test result

<i>return</i>	<i>description</i>
0	vulnerable
other	might be safe

Known issues

x86_64: Currently only works with object allocated on heap.

3.1.8 call-wrong-num-arg-vtable-heap

Description

Replace the Vtable pointer with a fake Vtable constructed in heap and illegally call a virtual function with mismatched number of arguments.

Vulnerability

Break the function calling convention, the data integrity of the Vtable pointer and the Vtable itself.

Test result

<i>return</i>	<i>description</i>
0	vulnerable
other	might be safe

Known issues

x86_64: Currently only works with object allocated on heap.

3.1.9 call-wrong-type-arg-vtable-heap

Description

Replace the Vtable pointer with a fake Vtable constructed in heap and illegally call a virtual function with wrong types of arguments.

Vulnerability

Break the function calling convention, the data integrity of the Vtable pointer and the Vtable itself.

Test result

<i>return</i>	<i>description</i>
0	vulnerable
other	might be safe

Known issues

x86_64: Currently only works with object allocated on heap.

3.1.10 call-wrong-num-func-vtable-heap

Description

Replace the Vtable pointer with a fake Vtable constructed in heap with different number of virtual functions and illegally call a fake virtual function.

Vulnerability

Break the data integrity of the Vtable pointer and the Vtable itself.

Test result

<i>return</i>	<i>description</i>
0	vulnerable
other	might be safe

Known issues

x86_64: Currently only works with object allocated on heap.

3.1.11 `jump-mid-func`

Description

Illegally jump from the `main()` function to the middle of another function.

Vulnerability

Break the execution compartment complied by most C/C++ programs.

Test result

<i>return</i>	<i>description</i>
0	vulnerable
other	might be safe

Known issues

None.

3.1.12 return-non-call-site

Description

Illegally modify the return address stored on the stack and then return to a non-call-site position.

Vulnerability

Break the backward CFI and the integrity of the return address.

Test result

<i>return</i>	<i>description</i>
0	vulnerable
other	might be safe

Known issues

x86_64: The `rbp` register might be (with `-g`) or not be (with `-O2`) pushed to the stack. The return address is modified by embedded assembly using `rsp` as the base register. See `STACK_STRUCT` in the make file.

3.1.13 return-to-func

Description

Illegally modify the return address stored on the stack and directly return to another function.

Vulnerability

Break the backward CFI and the integrity of the return address.

Test result

<i>return</i>	<i>description</i>
0	vulnerable
other	might be safe

Known issues

3.1.14 `return-without-call`

Description

Illegally add a fake function call onto the stack and return to it..

Vulnerability

Break the backward CFI and the integrity of the return address.

Test result

<i>return</i>	<i>description</i>
0	vulnerable
other	might be safe

Known issues

x86_64: The `rbp` register might be (with `-g`) or not be (with `-O2`) pushed to the stack. Currently the test works only with `-O2`.

Chapter 4

Remaining Issues

- `call-wrong-num-arg-func` 3.1.2: test for arguments passed on stack.
- `call-wrong-type-arg-func` 3.1.3: more importantly, test (data/code) pointer to integer.
- `call-wrong-num-arg-vtable` 3.1.4: known issues.
- `call-wrong-num-func-vtable` 3.1.6: known issues.
- `return-without-call` 3.1.14: known issues.
- call a invisible function (call a local function from outside).
- differentiate between global data, heap and stack.