ISCAS 中国科学院软件研究所学术年会'2023
暨计算机科学国家重点实验室开放周

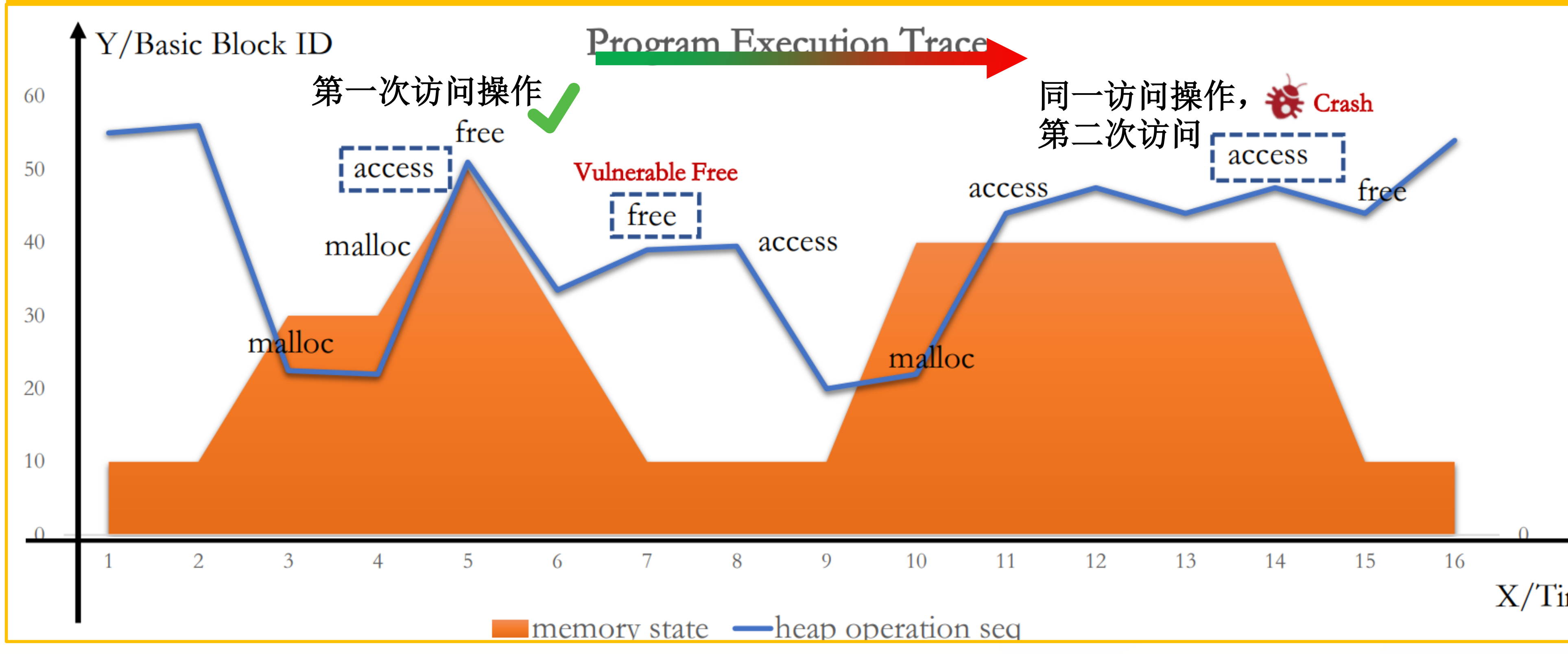
学术论文

HTFuzz: Heap Operation Sequence Sensitive Fuzzing
堆操作序列敏感的模糊测试，ASE 2022（CCF-A）

Yuanping Yu*, Xiangkun Jia*, Yuwei Liu, Yanhao Wang, Qian Sang,
Chao Zhang and Purui Su (联系人：贾相堃，xiangkun@iscas.ac.cn)

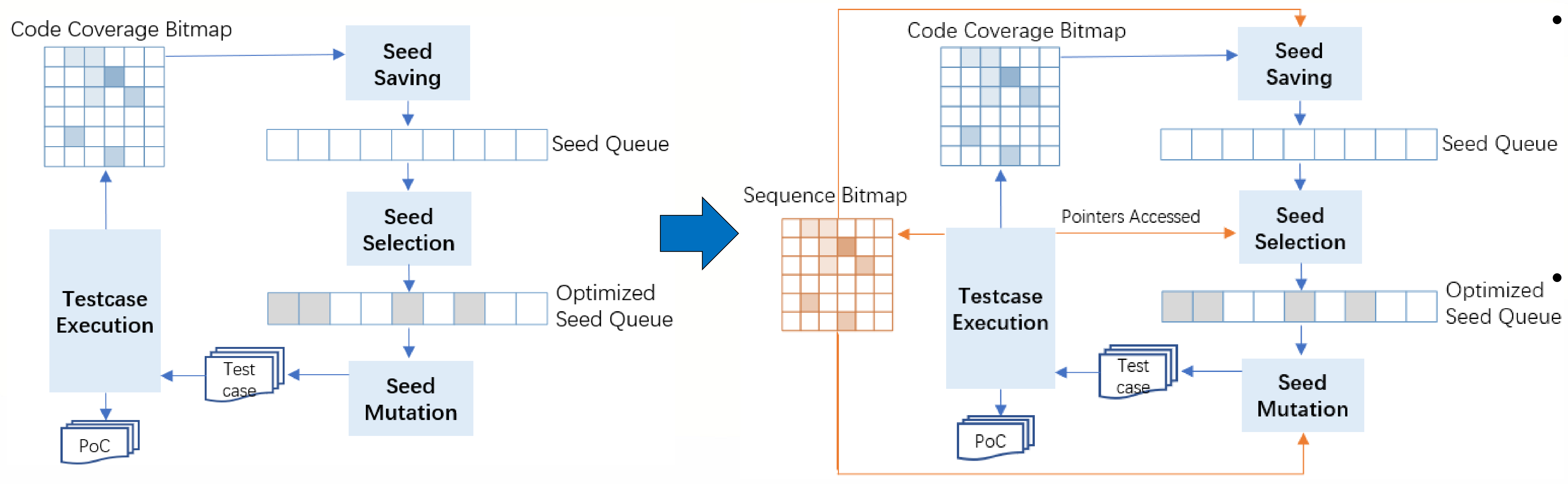
https://github.com/TCA-ISCAS/HTFuzz.git

问题描述



- **时序类漏洞**（Use-After-free, Double-Free 和 Null Pointer Dereference）的发现不仅需要触发相应的内存操作（分配/释放/访问），还需要满足特定的操作序列。
- 以**代码覆盖率反馈**（Coverage Feedback）为导向的模糊测试方法（如AFL）在触发相应内存操作之后，无法感知操作序列，失去导向效果，对时序类漏洞挖掘效果不佳。

解决思路



- 把**堆操作序列**信息加入模糊测试反馈中，保留能够触发新操作序列的种子
- 把**内存访问频度**加入反馈中，提高在不同操作序列下内存访问操作的几率

技术挑战 1

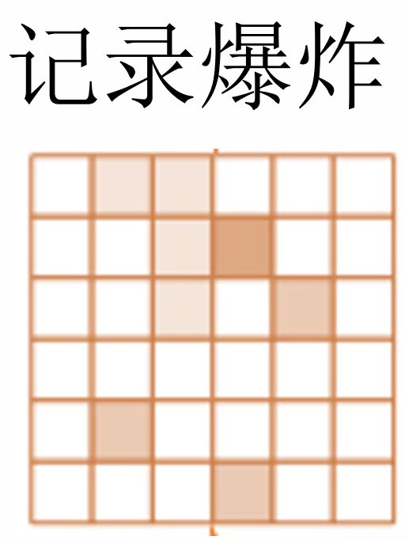
…-> malloc (Size) -> malloc (Size)-> access (Ptr, Size) -> free (Ptr) -> free (Ptr) -> access (Ptr, Size) -> malloc (Size) -> access (Ptr, Size) -> access (Ptr, Size) -> free (Ptr) ->…

1, 1, *access*, 0, 0, *access*, 1, *access*, *access*, 0

[0, 1, 1] [1, 0, 0] [0, 0, 1] [0, 0, 1]

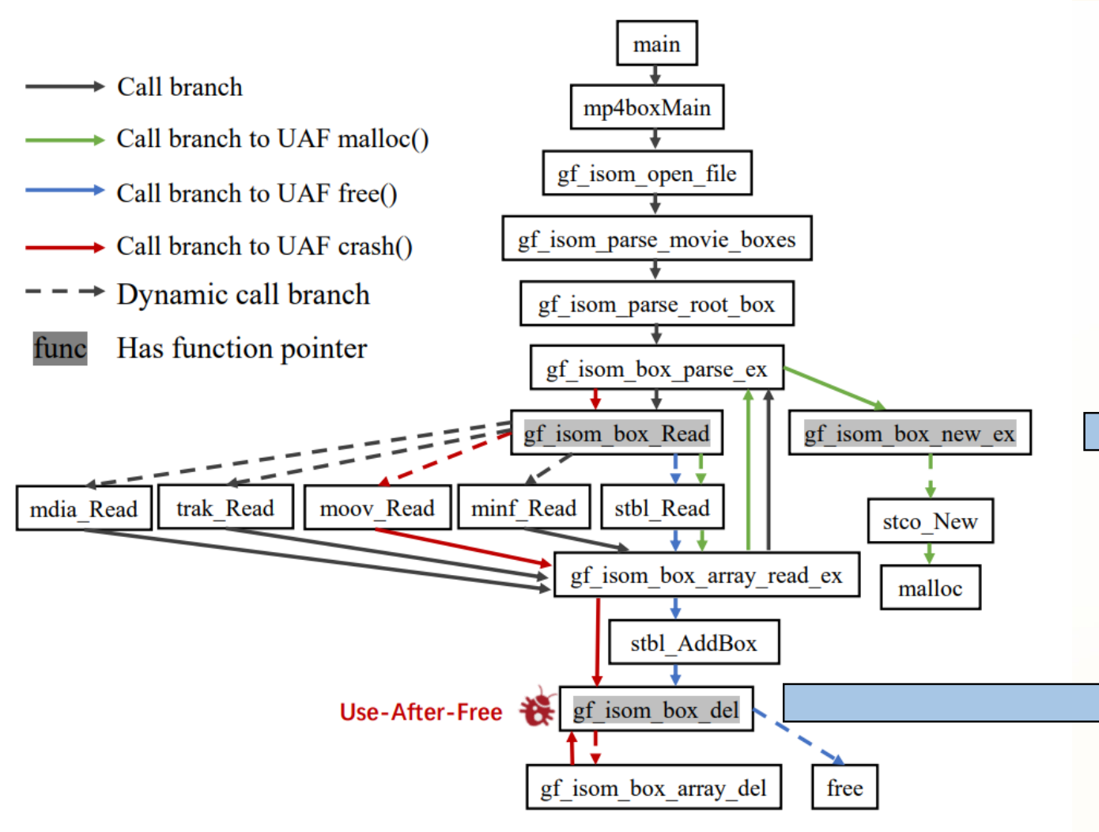
[3] [4] [1] [1]

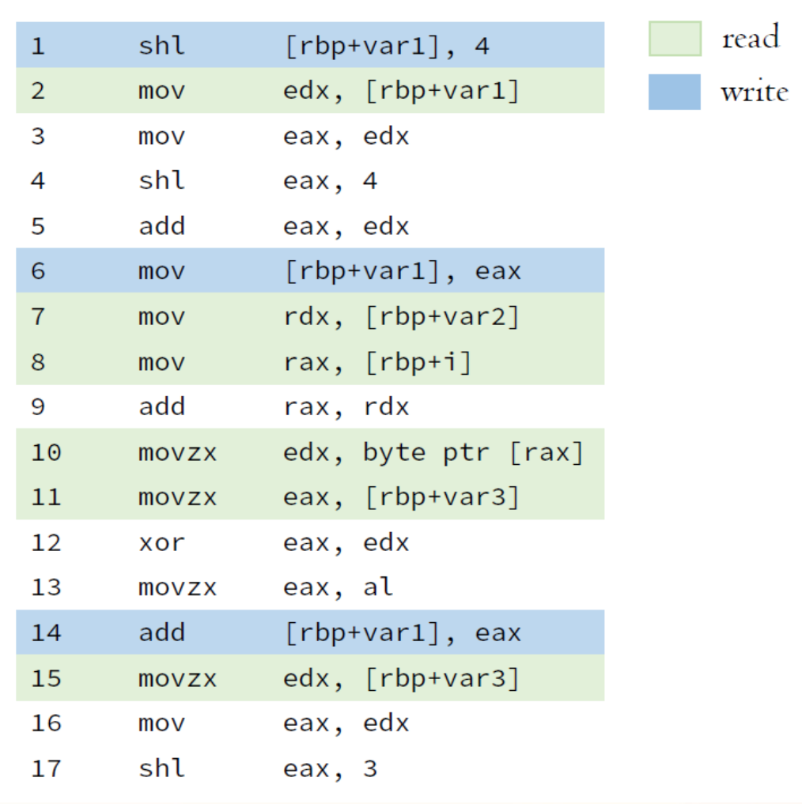
记录爆炸



Seq Bitmap

技术挑战 2





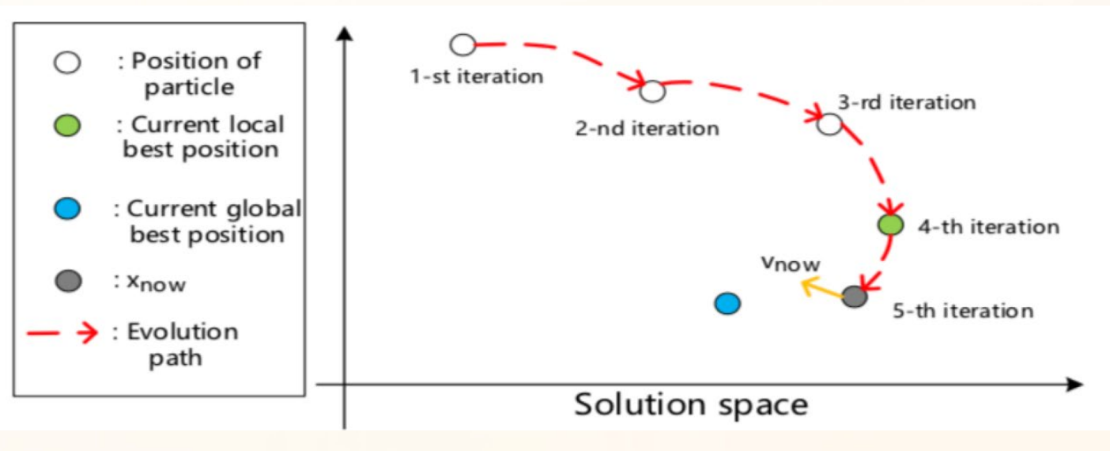
```
shl      [rbp+var1], 4
mov      edx, [rbp+var1]
mov      eax, edx
shl      eax, 4
add      eax, edx
mov      [rbp+var1], eax
mov      rdx, [rbp+var2]
mov      rax, [rbp+i]
add      rax, rdx
movzx    edx, byte ptr [rax]
movzx    eax, [rbp+var3]
xor      eax, edx
movzx    eax, al
add      [rbp+var1], eax
movzx    edx, [rbp+var3]
mov      eax, edx
shl      eax, 3
```

关注包含内存访问的基本块 **插桩内存访问操作**

技术挑战 3

Effector map干扰

变异策略倾向



基于**MOPT**: Optimize Mutation Scheduling for Fuzzers的修改

HTFuzz的效果

漏洞挖掘效果

Bug ID	Version	Type	Status	Vulnerable Function
CVE-2021-33453	LRZIP 0.641	UAF	accepted	ucompthread()
CVE-2019-20169	GPAC 0.8.0	UAF	accepted & fixed	trak_Read()
CVE-2019-20164	GPAC 0.8.0	UAF	accepted & fixed	gf_isom_box_del()
CVE-2019-20168	GPAC 0.8.0	UAF	accepted & fixed	gf_isom_box_dump_ex()
CVE-2020-35980	GPAC 0.8.0	UAF	accepted & fixed	gf_isom_box_del()
CVE-2021-33461	YASM 1.3.0	UAF	accepted	yasm_intnum_destroy()
CVE-2021-33462	YASM 1.3.0	UAF	accepted	expr_traverse_nodes_post()
CVE-2021-33467	YASM 1.3.0	UAF	accepted	pp_getline()
CVE-2021-33468	YASM 1.3.0	UAF	accepted	error()
CVE-2021-33439	MJS version 6	NPD	accepted	gc_compact_strings()
CVE-2021-33440	MJS version 6	NPD	accepted	mjs_bcode_commit()
CVE-2021-33441	MJS version 6	NPD	accepted	exec_expr()
CVE-2021-33442	MJS version 6	NPD	accepted	json_printf()
CVE-2021-33444	MJS version 6	NPD	accepted	getprop_builtin_foreign()
CVE-2021-33445	MJS version 6	NPD	accepted	mjs_string_char_code_at()
CVE-2021-33446	MJS version 6	NPD	accepted	mjs_next()
CVE-2021-33449	MJS version 6	NPD	accepted	mjs_bcode_part_get_by_offset()
CVE-2021-33437	MJS version 6	NPD	accepted	frozen_cb()
CVE-2021-33455	YASM 1.3.0	NPD	accepted	do_directive()
CVE-2021-33456	YASM 1.3.0	NPD	accepted	hash()
CVE-2021-33457	YASM 1.3.0	NPD	accepted	expand_mmac_params()
CVE-2021-33458	YASM 1.3.0	NPD	accepted	find_cc()
CVE-2021-33460	YASM 1.3.0	NPD	accepted	if_condition()
CVE-2021-33463	YASM 1.3.0	NPD	accepted	yasm_expr__copy_except()
CVE-2021-33465	YASM 1.3.0	NPD	accepted	expand_mmacro()
CVE-2021-33466	YASM 1.3.0	NPD	accepted	expand_smacro()
CVE-2019-20163	GPAC 0.8.0	NPD	accepted & fixed	gf_odf_avc_cfg_write_bs()
CVE-2020-35981	GPAC 0.8.0	NPD	accepted & fixed	SetupWriters()
CVE-2020-35982	GPAC 0.8.0	NPD	accepted & fixed	gf_hinter_track_finalize()
CVE-2021-33450	NASM 2.14rc0	NPD	accepted	nasm_calloc()
CVE-2021-33452	NASM 2.14rc0	NPD	accepted	nasm_malloc()
CVE-2021-33451	LRZIP 0.641	NPD	accepted	fill_buffer()
CVE-2021-33438	MJS version 6	BO	accepted	json_parse_array()
CVE-2021-33448	MJS version 6	BO	accepted	unknown-module>
CVE-2021-33443	MJS version 6	BO	accepted	mjs_execute()
CVE-2020-35979	GPAC 0.8.0	BO	accepted & fixed	gp_rtp_builder_do_avc()
CVE-2021-33464	YASM 1.3.0	BO	accepted	inc_fopen()

37个0day（其中32个时序类漏洞）

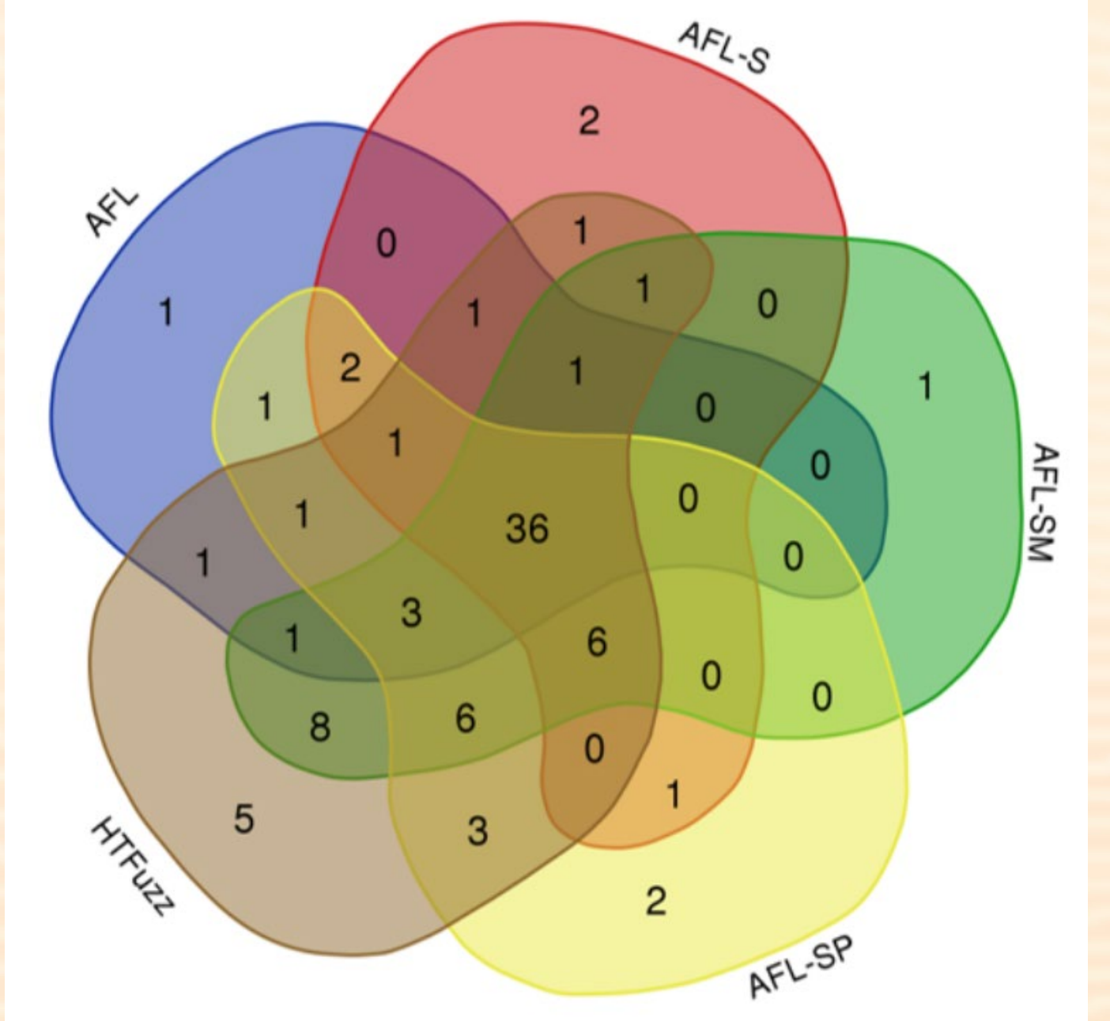
工具对比结果

	HTFuzz	AFL	Memlock	AFL-sen-ma	AFL-sen-mw
0day	37-**32**-0	24-22-0	9-7-0	9-9-0	9-9-0
0day-non-CVE	-	3-0-0	3-0-0	-	-
1day	55-**42**-0	37-26-4	29-20-2	16-12-0	11-9-0
Sum	92-**74**-0	64-48-4	41-27-2	25-21-0	20-18-0

	PathAFL	Tofuzz	MOPT	Angora	Ankou
0day	21-18-0	20-18-0	21-19-0	8-8-0	26-22-0
0day-non-CVE	2-0-0	1-0-0	-	3-2-2	6-1-1
1day	28-23-4	30-20-2	46-32-1	27-25-10	49-36-2
Sum	51-41-4	51-38-2	67-51-1	42-37-14	81-59-3

每项数据X-Y-Z, X是发现的所有类型漏洞, Y是时序类漏洞,Z是HTFuzz相比于其他工具漏掉的漏洞

策略的消融实验



AFL是baseline工具，AFL-S是增加了Seq Bitmap反馈，AFL-SP是AFL-S基础上增加了内存访问频度，AFL-SM是AFL-S基础上增加了MOPT调度策略，HTFuzz是最终的完整方案