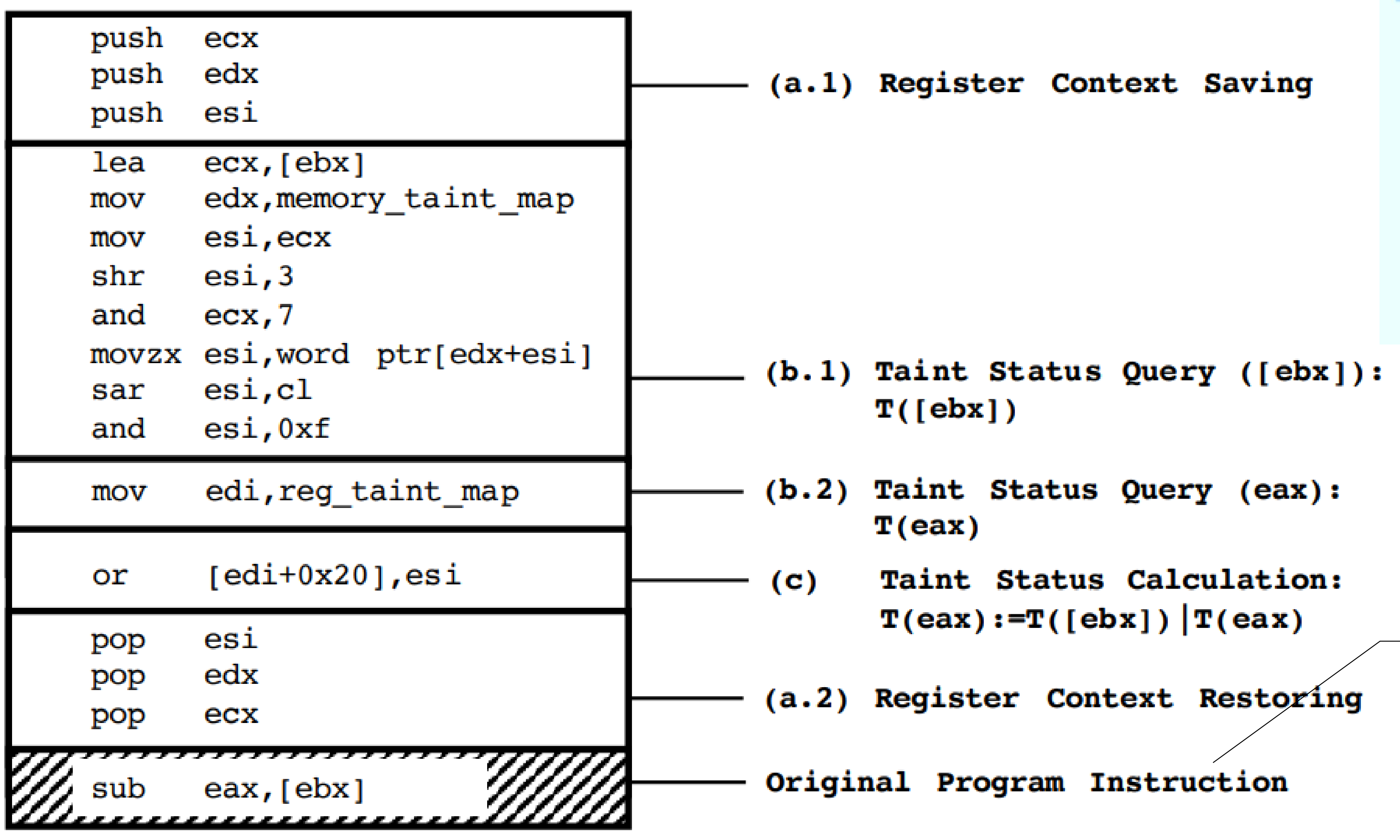
ISCAS 中国科学院软件研究所学术年会’2024
暨 重 点 实 验 室 科 技 活 动 周

学术论文

AirTaint: Making Dynamic Taint Analysis Faster and Easier
更快更易用的动态污点分析

桑倩*，王衍豪*，刘昱玮，贾相堃✉，Tiffany Bao，苏璞睿✉
IEEE S&P 2024（CCF-A），xiangkun@iscas.ac.cn，15652143223

Motivation



```
push   ecx
push   edx
push   esi
lea    ecx,[ebx]
mov    edx,memory_taint_map
mov    esi,ecx
shr    esi,3
and    ecx,7
movzx  esi,word ptr[edx+esi]
sar    esi,cl
and    esi,0xf
mov    edi,reg_taint_map
or     [edi+0x20],esi
pop    esi
pop    edx
pop    ecx
sub    eax,[ebx]
```

(a.1) Register Context Saving

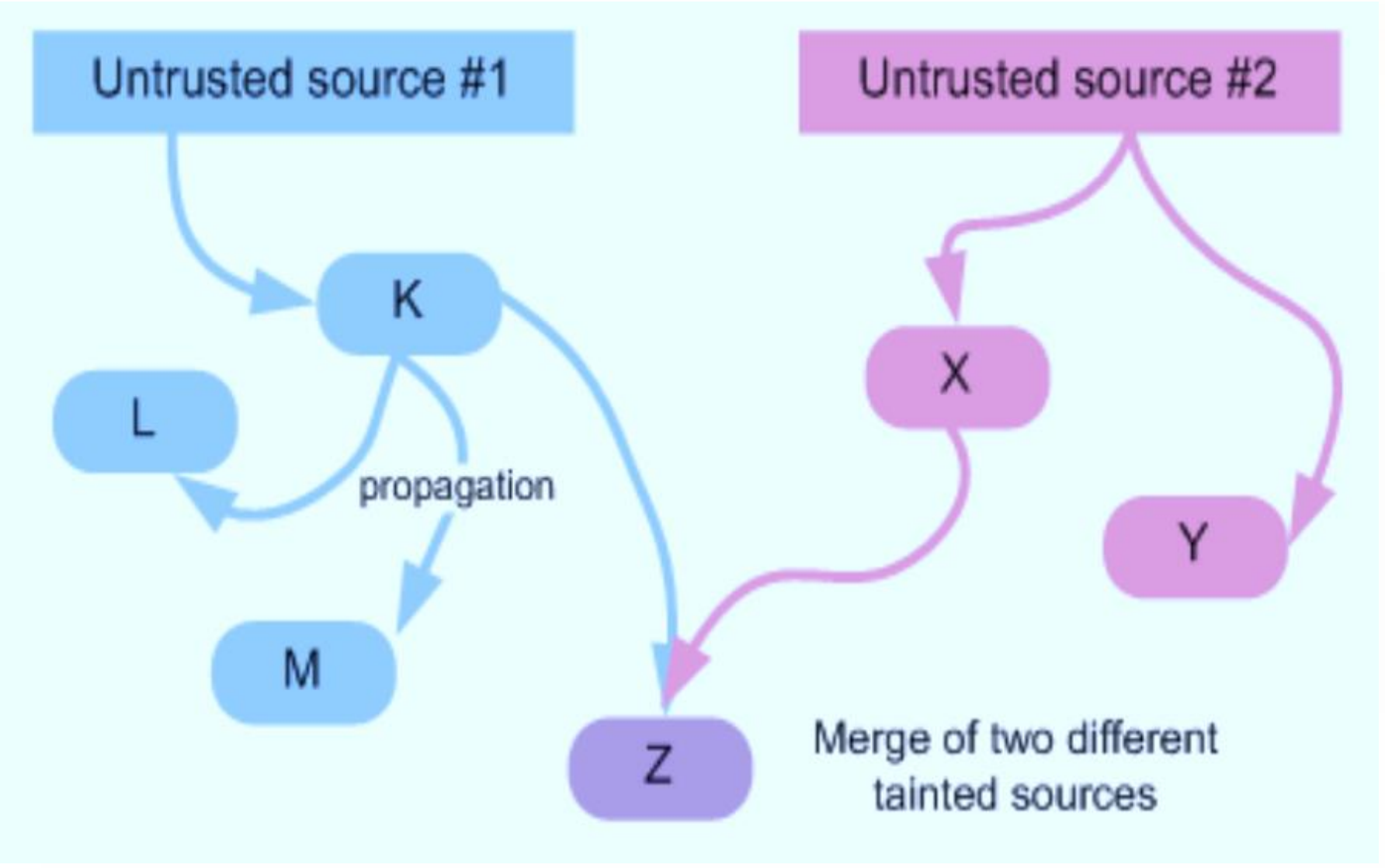
(b.1) Taint Status Query ([ebx]): T([ebx])

(b.2) Taint Status Query (eax): T(eax)

(c) Taint Status Calculation: T(eax):=T([ebx])|T(eax)

(a.2) Register Context Restoring

Original Program Instruction



动态污点分析原理：跟踪每条指令，并按照指令语义进行污点传播计算

为了一条sub指令，需要插桩16条指令（libdft）

Idea

```
1:div dword ptr [ebx+0x8]
2:add ecx, edx
3:sub [ebx+0x8], ecx
4:add ebx, 0x10
5:xor edx, edx
6:div ecx
7:mov ecx, [ebx+0x10]
8:lea ebx, [ecx+esi]
9:shl esi, 3
10:or [edx+esi], ecx
11:lea eax, [ebx+esi]
```

(a) basic block

```
1:T(eax):=T(eax)|T(edx)|T([ebx+0x8])
  T(edx):=T(eax)|T(edx)|T([ebx+0x8])
2:T(ecx):=T(ecx)|T(edx)
3:T([ebx+0x8]):=T([ebx+0x8])|T(ecx)
4:T(ebx):=T(ebx)
5:T(edx):=0x0
6:T(eax):=T(eax)|T(ecx)
  T(edx):=T(eax)|T(ecx)
7:T(ecx):=T([ebx+0x10])
8:T(ebx):=T(ecx)|T(esi)
9:T(esi):=T(esi)
10:T([edx+esi]):=T([edx+esi])|T(ecx)
11:T(eax):=T(ebx)|T(esi)
```

(b) Tradition DTA

1:tmp_1:=T(eax)
  tmp_2:=T(edx)
  tmp_3:=T(ecx)
  tmp_4:=T([ebx+0x8])
3:T([ebx+0x8]):=tmp_1|tmp_2|tmp_3|tmp_4
7:T(edx):=tmp_1|tmp_2|tmp_3|tmp_4
  T(ecx):=T([ebx+0x10])
11:T(ebx):=T(ecx)|T(esi)
  T([edx+esi]):=T([edx+esi])|T(ecx)
  T(eax):=T(ecx)|T(esi)

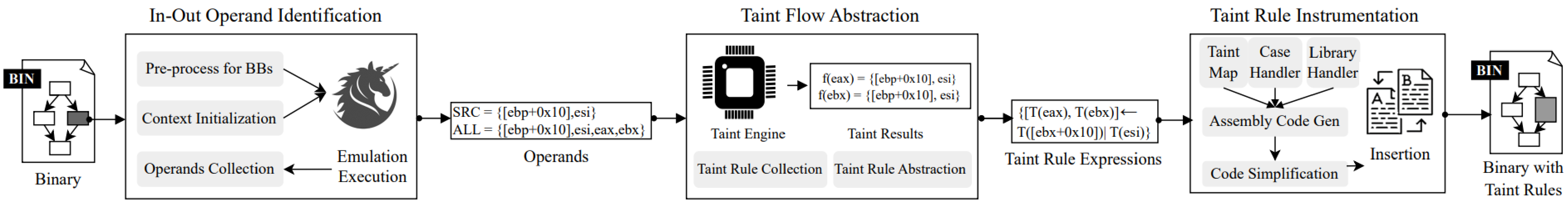
(c) Instrumentations by Jee's work

```
11:T(eax):=T([ebx+0x10])|T(esi)
   T(ebx):=T([ebx+0x10])|T(esi)
   T(ecx):=T([ebx+0x10])
   T(edx):=T(eax)|T(edx)|T([ebx+0x8])|T(ecx)
   T([ebx+0x8]):=T(eax)|T(edx)|T([ebx+0x8])|T(ecx)
   T([edx+esi]):=T([edx+esi])|T([ebx+0x10])
```

(d) Instrumentations by AirTaint

指令级污点计算
-> 基本块级计算

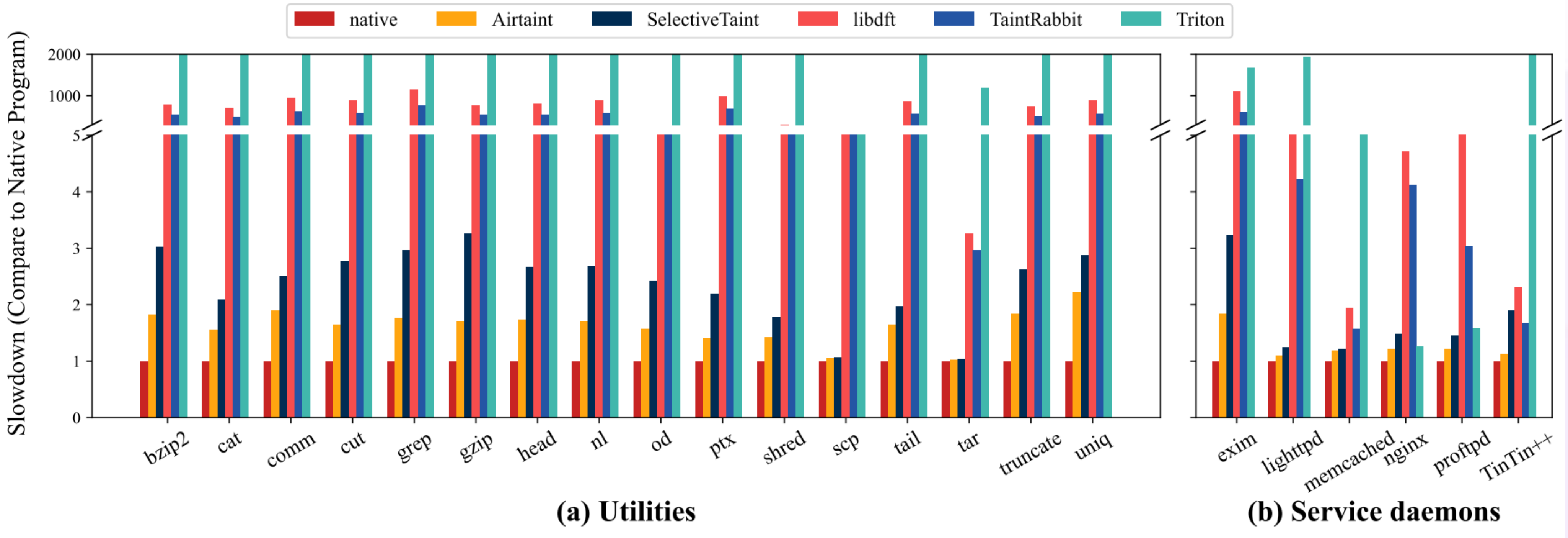
AirTaint



挑战：
□ 如何准确识别基本块的输入输出
□ 如何自动化地提取基本块的污点计算规则
□ 如何实现运行时高效的污点计算

方法：
✓ 基本块的**模拟执行**（基于Unicorn）
✓ 借助**现有污点传播引擎**（如Triton）提取
✓ **静态插桩**（如RetroWrite）插入汇编代码

Evaluation



(a) Utilities

(b) Service daemons

平均比SelectiveTaint[USENIX Security 21]效率提升1.53倍；
比libdft[VEE 12]提升509.0倍；
比TaintRabbit[AisaCCS 20]提升216.3倍

CVE-ID	Program	Vulnerability Type	Source Point	Sink Point	x86	x86-64
CVE-2019-8343	NASM 2.14.02	use-after-free	preproc.c:5165	preproc.c:900	✓	✓
CVE-2019-20352	NASM 2.15rc1	heap-buffer-overflow	preproc.c:1335	preproc.c:387	✓	✓
CVE-2018-11575	ngiflib 0.4	stack-buffer-overflow	git2tga.c:95	ngiflib.c:524	✓	✓
CVE-2019-16346	ngiflib 0.4	heap-buffer-overflow	git2tga.c:95	ngiflib.c:123	✓	✓
CVE-2018-19655	dcraw 9.28	stack-buffer-overflow	dcraw.c:885	dcraw.c:8342	✓	✓
CVE-2021-3624	dcraw 9.28	integer-overflow	dcraw.c:3197, 3198	dcraw.c:3221	✓	✓
CVE-2018-6612	Jhead 3.00	integer-underflow	jpgfile.c:159, 160	exif.c:1034	✓	✓
CVE-2020-26208	Jhead 3.00	heap-buffer-overflow	jpgfile.c:159, 160	jpgfile.c:286	✓	✓
CVE-2017-1000074	Gravity 0.2.6	stack-buffer-overflow	gravity.c:187	gravity_core.c:1595	✓	✓
CVE-2017-14408	MP3Gain 1.5.2	stack-buffer-overflow	mp3gain.c:1778	layer3.c:1255	✓	✓
CVE-2019-7629	TinTin++ 2.01.6	stack-buffer-overflow	update.c:228	parse.c:771	✓	✓
CVE-2018-6789	exim 4.89	heap-buffer-overflow	get_data.c:34	b64decode.c:156	✓	✓
CVE-2020-19143	LibTIFF 4.0.10	global-buffer-overflow	tif_unix.c:169	tif_dir.c:1116	✓	✓
CVE-2018-18557	LibTIFF 4.0.8	heap-buffer-overflow	tif_jbig.c:63	tif_jbig.c:101	✓	✓

支持x86，x86-64平台的漏洞检测能力，且由于方法设计，AirTaint具有很好的扩展性，可更新规则、支持新架构

https://github.com/TCA-ISCAS/AirTaint