Fuzzing BusyBox : Leveraging LLM and Crash Reuse For Embedded Bug Unearthing

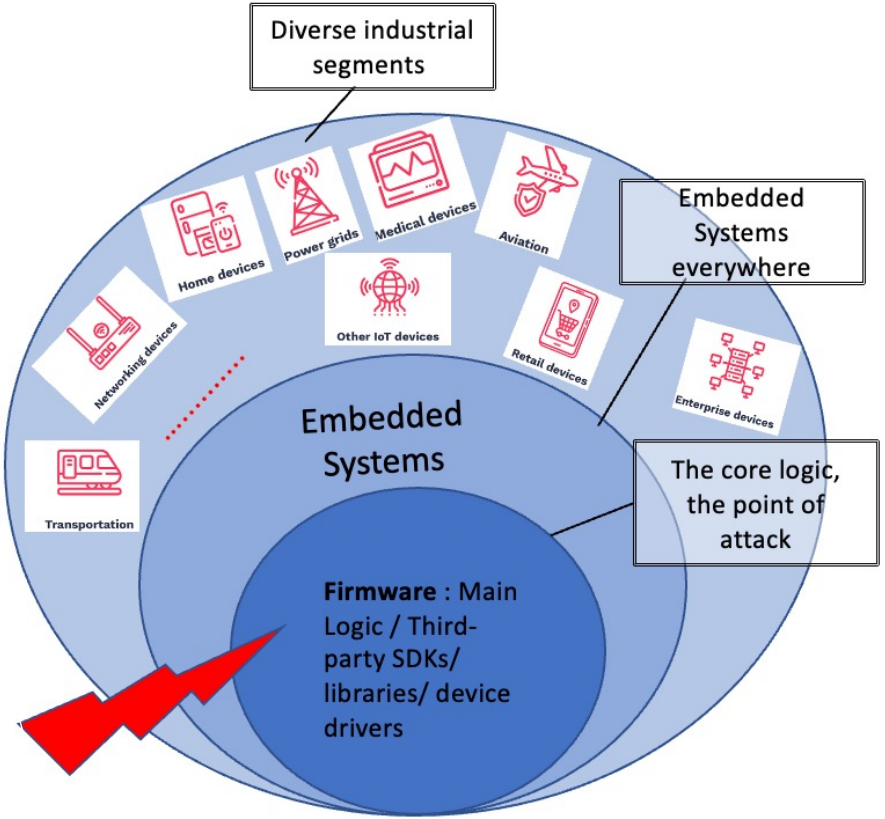
Asmita[1], Yaroslav Oliinyk[2], Michael Scott[2], Ryan Tsang[1], Chongzhou Fang[1], Houman Homayoun[1]

[1]University of California Davis, [2]NetRise

Usenix Security 2024

Introduction

- Surge in the number of IoT devices
- Increase in the number of attacks
- Embedded devices , a central part of IoT ecosystem
- Firmware, the core logic of the system



Icon Image Source – JSOF Tech

Introduction

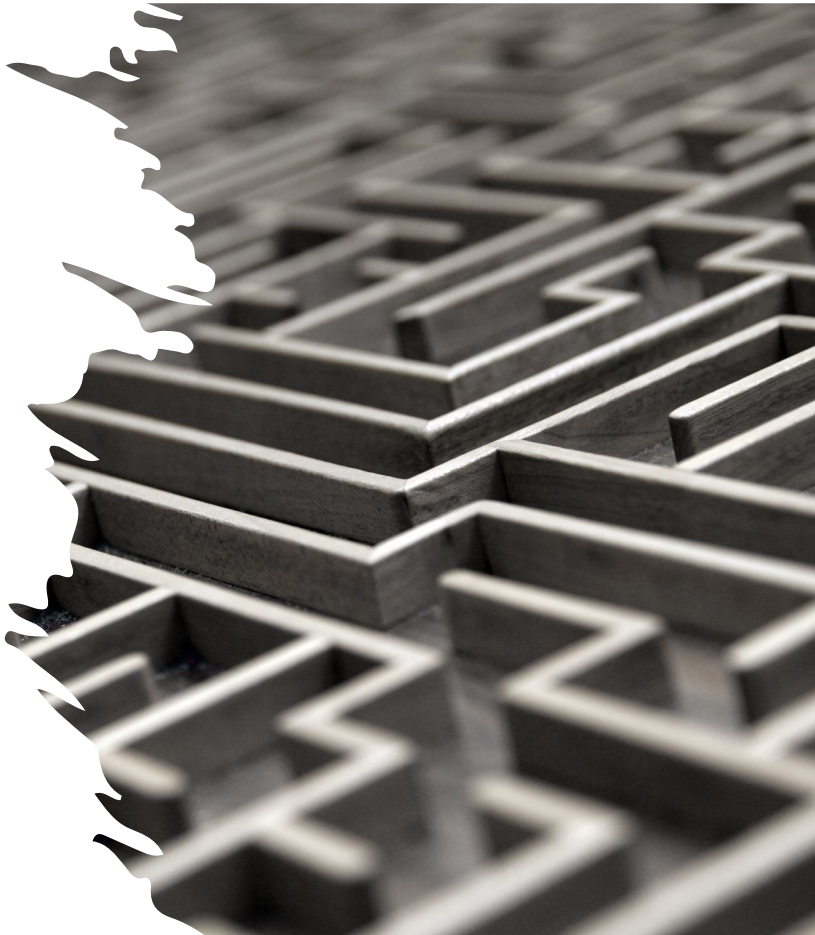
- BusyBox
- Fuzzing
- LLM *(Large language models)*
- Crash Reuse *(Replay crash on variants of a target)*

Existing research leveraging LLM for fuzzing

Research	Target	LLM use case
ChatAFL	Protocols	Extract a machine-readable grammar for a protocol, generate diverse messages for initial seeds
ChatFuzz	Format conforming targets	Used at the mutation stage to generated format conforming mutated inputs
Fuzz4All	Targets that need different programming languages as input	Generate code snippets for different programming languages
WhiteFox	Compiler	Optimization source code analyzer, test input generation
Proposed Work	Embedded applications like BusyBox	Generate diverse and target-specific initial seeds

Research Questions

- How widespread are the variants of BusyBox ?
- Can LLM be leveraged for fuzzing BusyBox ?
- Is there a better way of identifying similar vulnerabilities across variants?

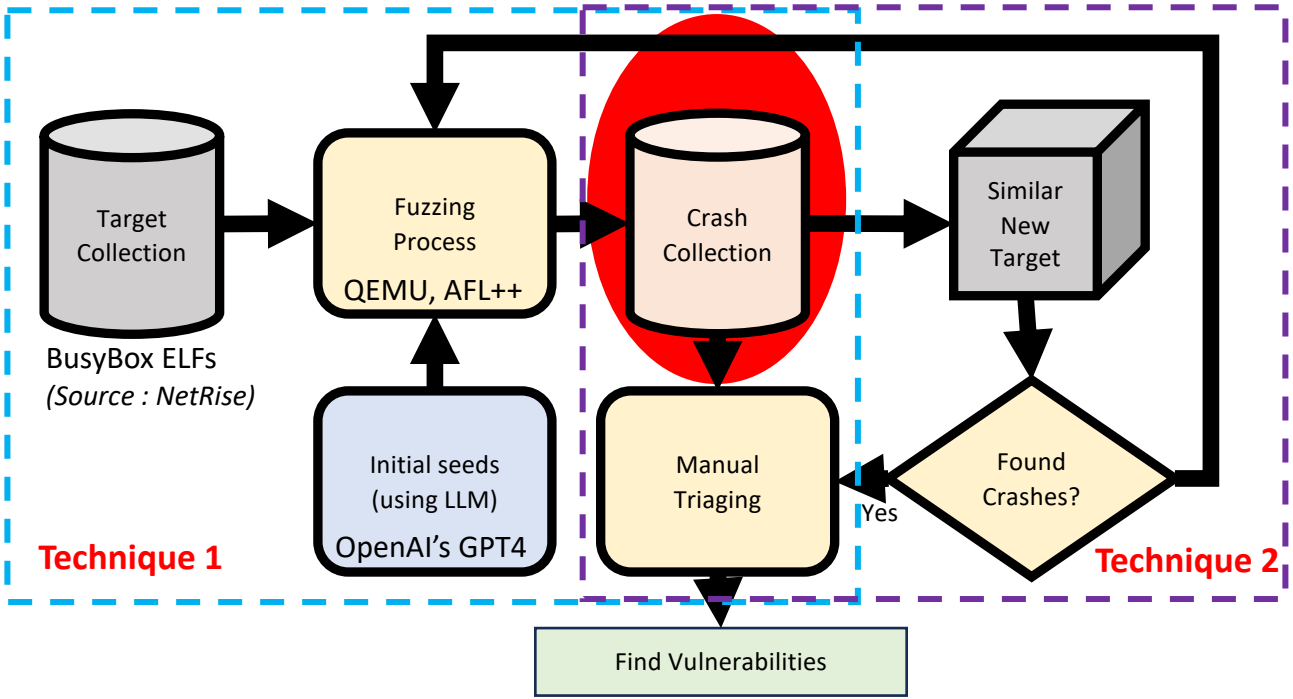


Variants of BusyBox

BusyBox Version	No. of Occurrence	Product Types	BusyBox Version	No. of Occurrence	Product Types	BusyBox Version	No. of Occurrence	Product Types
v1.7.2	2	wireless access point,	v1.19.4	4	network management tool, wireless access point, network hardware, security camera	v1.27.2	1	power distribution unit
v1.10.2	1	telecom device	v1.20.2	2	security camera,	v1.28.3	1	network management tool
v1.11.1	1	building automation	v1.21.1	5	drone, ip phone, medical device, bmc	v1.28.4	1	operating system
v1.13.2	1	building automation	v1.22.1	7	wireless aceess point, network switch, operating system	v1.29.3	1	operating system
v1.15.2	2	wireless aceess point	v1.23.0	3	wireless access point,	v1.30.1	6	wireless access point, building automation, power managemnet system
v1.17.2	1	router	v1.23.1	7	bmc, router,	v1.33.0	1	power management system
v1.17.3	1	telecom device	v1.24.1	7	wireless access point, network switch,	v1.34.0	1	network controller card
v1.17.4	1	printer	v1.25.0	3	drone, network attached storage	v1.34.1	4	building automation, wireless access point
v1.18.2	1	wireless aceess point	v1.25.1	3	wireless access point,	v1.35.0	1	router
v1.19.2	1	wireless aceess point	v1.26.2	6	power management system, building automation,	v1.36.0	1	router

Analyzed ~80 embedded products. Source : NetRise

Research Pipeline

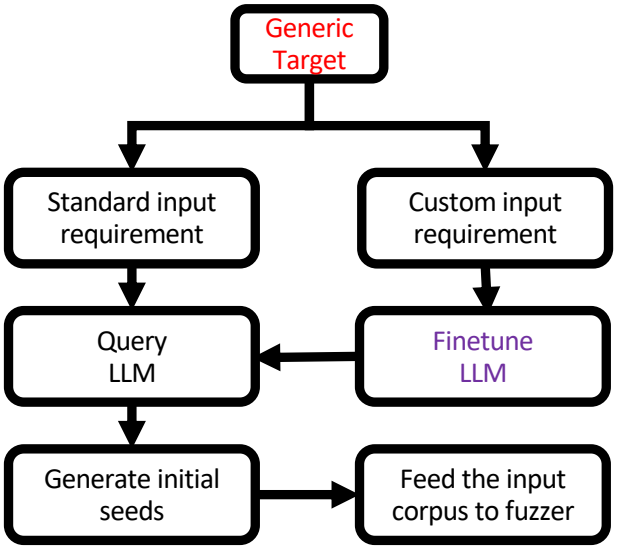


Technique 1 :
Leveraging LLM for Initial Seed Generation

- Case 1 : Input format is known by the LLM knowledge base
- Case 2 : Custom Input format (fine-tuning)

Prompt for BusyBox "awk" applet :

"role": "system", "content": "You are initial seed generator for a fuzzer that has to fuzz BusyBox awk applet. In response only provide the list of awk scripts"
"role": "user", "content": f'Generate initial seed to fuzz Busy-Box awk applet"



- Followed by corpus minimization using afl-cmin

Results :

Leveraging LLM for Initial Seed Generation

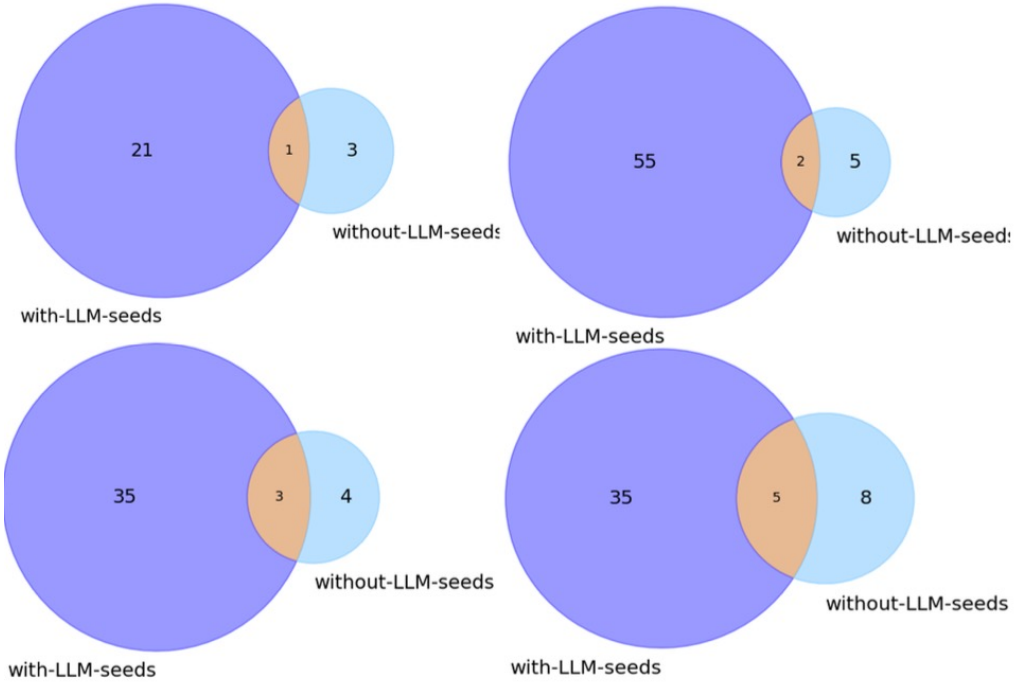
(Technique 1)

Target Version (ARM)	Product Type	No. of crashes w/o LLM	No. of crashes with LLM	Target Version (x86_64)	Product Type	No. of crashes w/o LLM	No. of crashes with LLM
v1.34.1	embedded wireless controller	3	188	v1.23.1	network controller	64	140
v1.29.3	medical device	54	82	v1.22.1	network management tool	43	165
v1.34.1	embedded PLC	3	177	v1.30.1	network management tool	147	229
v1.15.3	building automation	220	404	v1.27.2	operating system	0	114
v1.23.2	camera	50	165	v1.23.1	storage array controller	38	178
v1.18.4	plc	137	224	v1.21.1	firewall	44	99
v1.30.1	operating system	49	106	v1.19.4	network switch	49	172
v1.26.2	security camera	55	70	v1.15.1	operating system	166	357
v1.32.0	power control system	0	70	v1.23.1	network controller	41	98
v1.27.2	drone	34	147	v1.35.0	network management tool	2	193

Source of firmware : NetRise

No. of unique crashes with vs without LLM based initial input seeds.

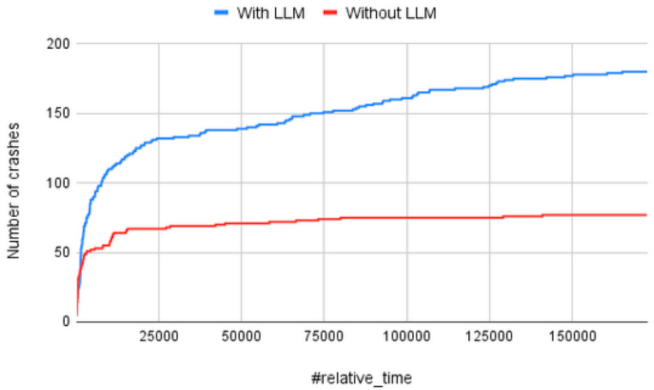
Target : Network controller, Network switch, storage array controller, firewall

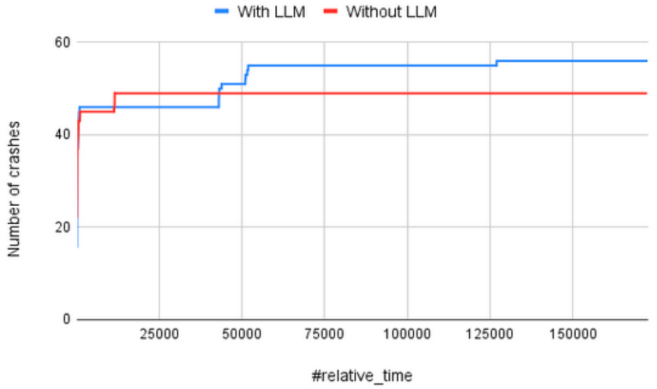


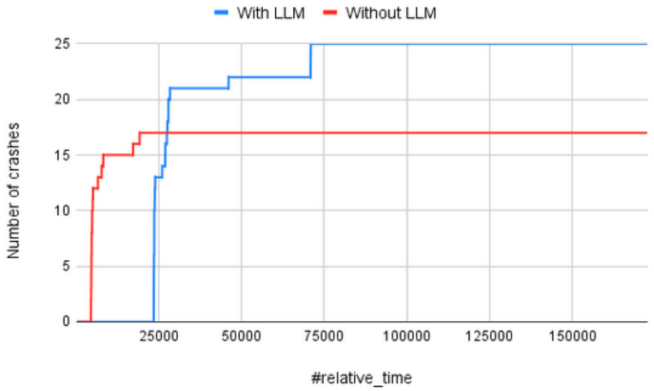
Used AFL-Triage
to identify unique crashes

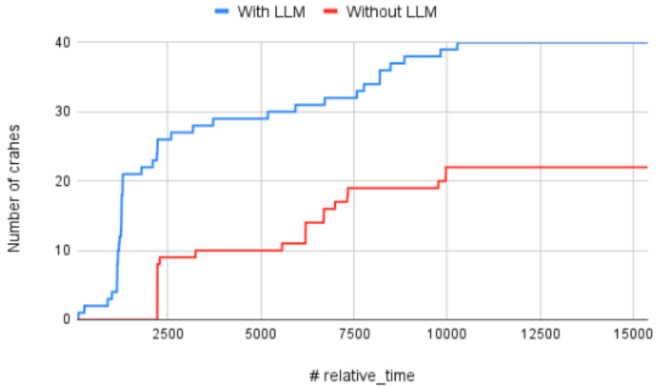
No. of crashes with vs without LLM based initial input seeds (48 hours fuzzing)

Target applets : awk, dc, man, ash clockwise



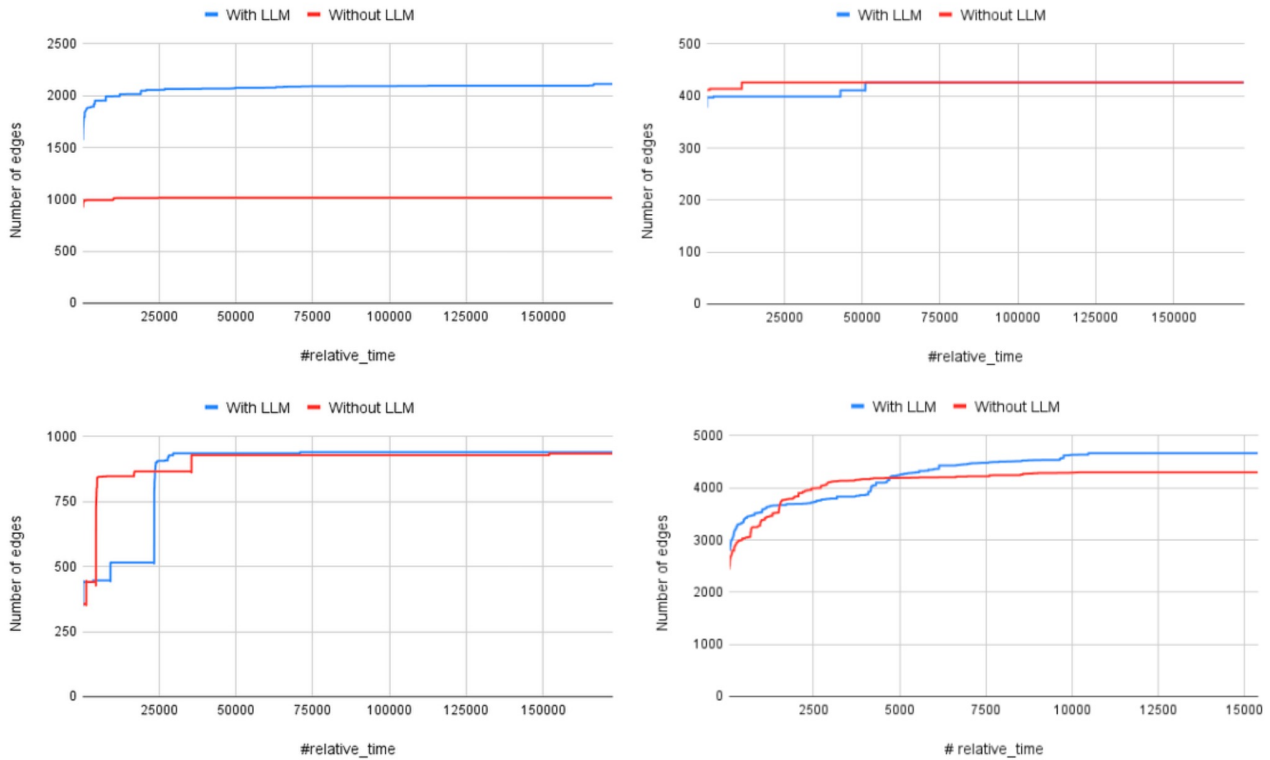






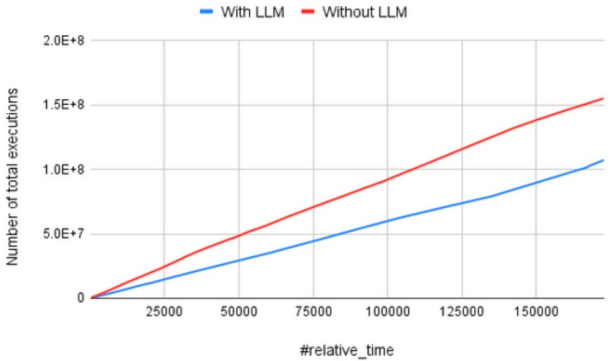
No. of edges with vs without LLM based initial input seeds

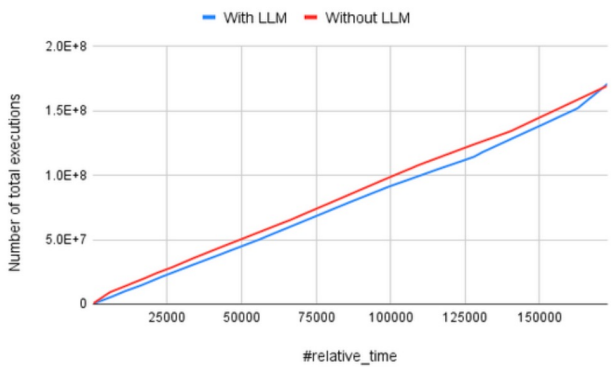
Target applets : awk, dc, man, ash clockwise

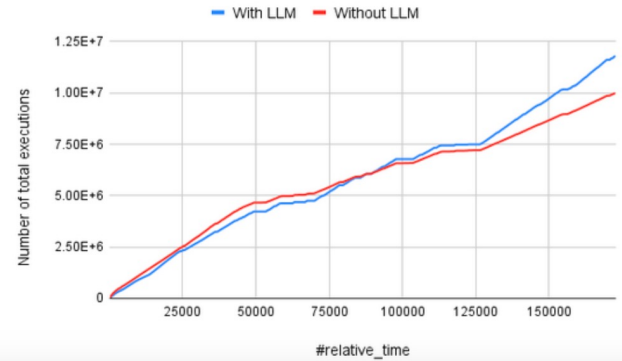


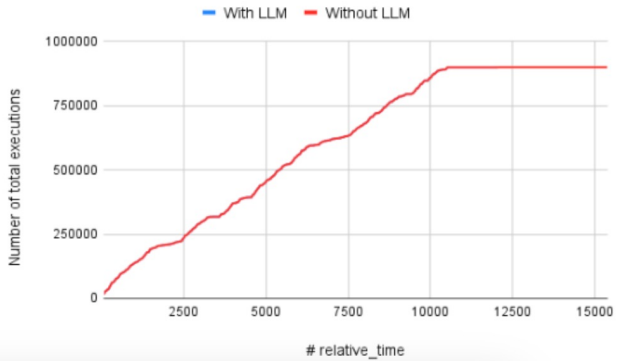
No. of executions with vs without LLM based initial input seeds

Target applets : awk, dc, man, ash clockwise









Result summary with LLM-based initial seeds *(Technique 1)*

- Increase in the number of crashes
- Edge coverage performance is better in some cases
- No significant difference in the number of executions/sec, hence no overhead issues

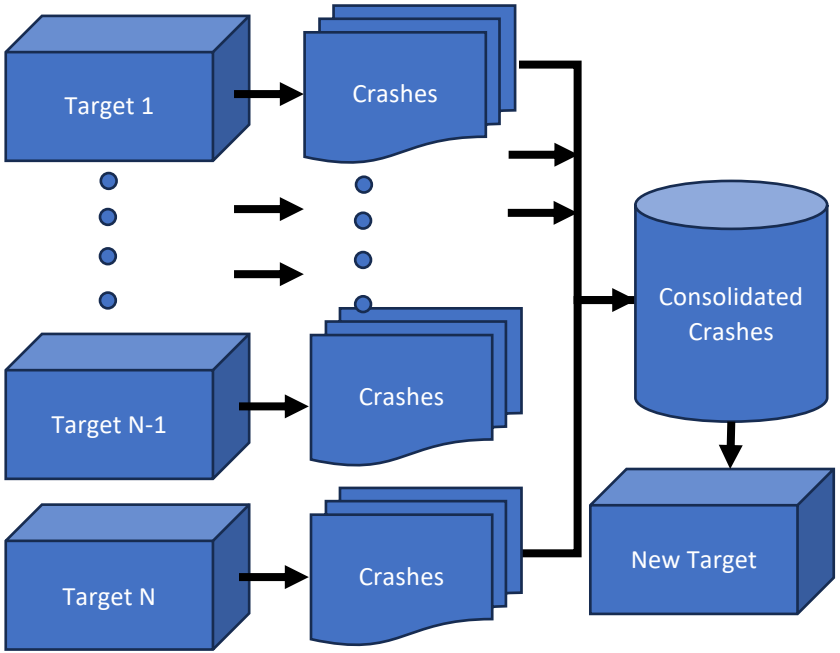
Technique 2 : Crash Reuse

Why ?

Efficiency

Black-box testing

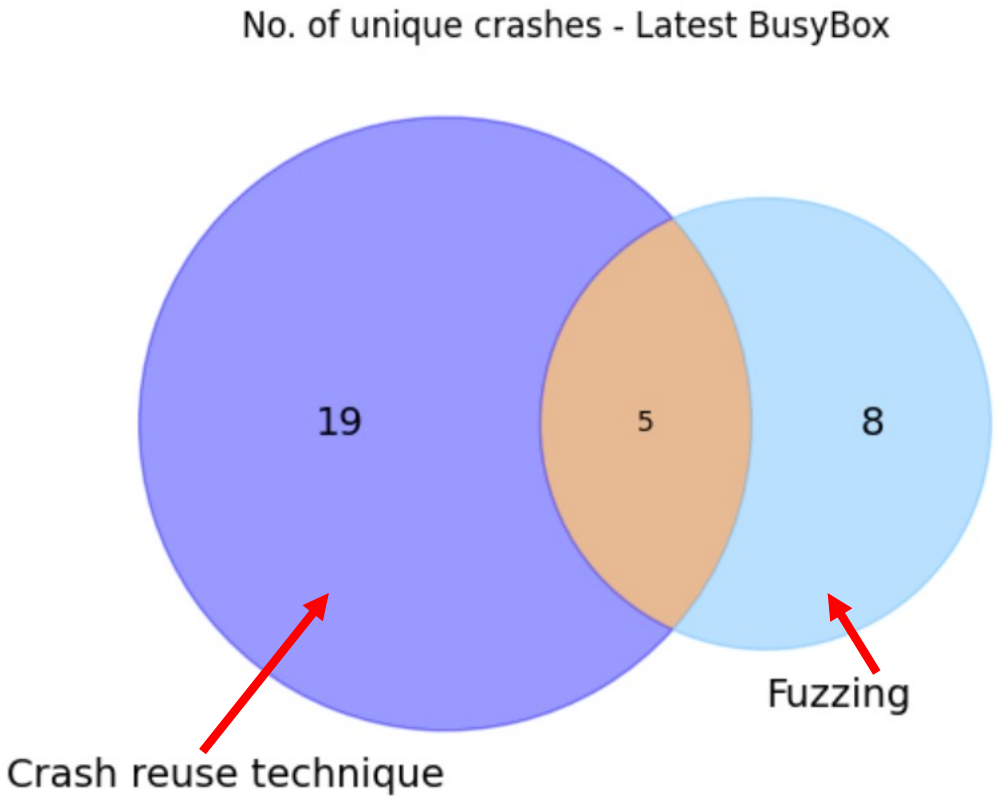
Technique 2 : Crash Reuse



Results : Crash Reuse

(Technique 2)

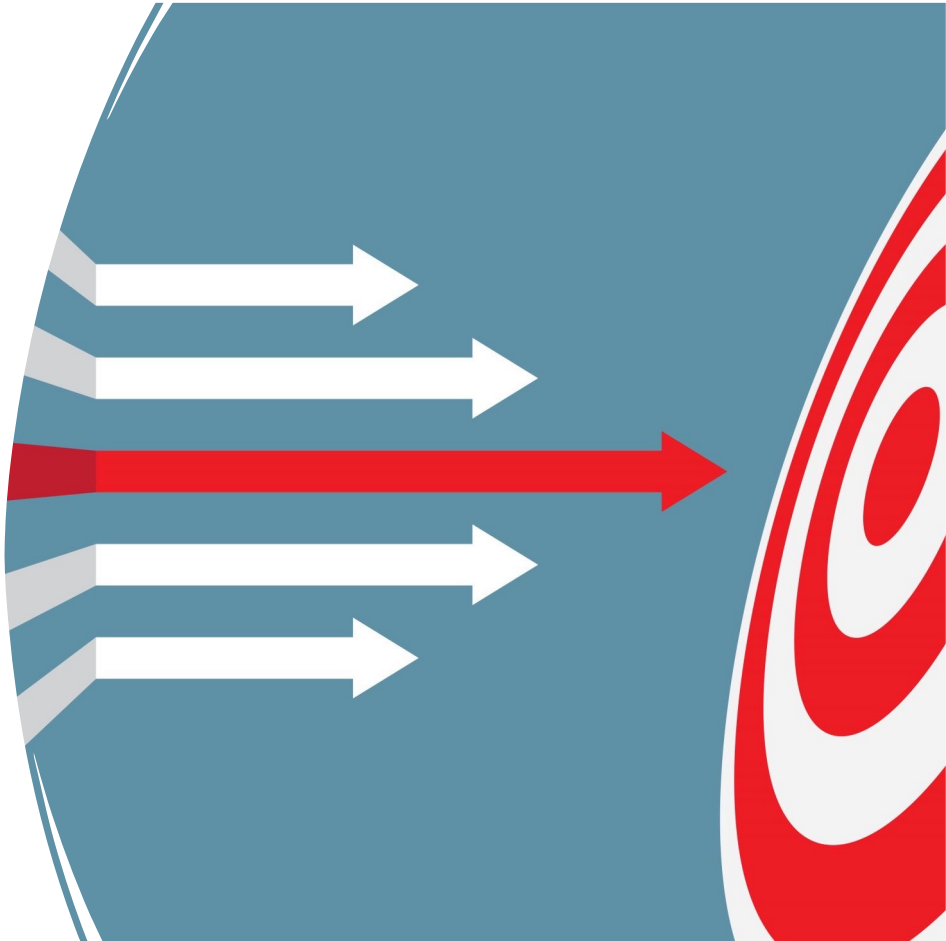
- No. of previously collected crashes from **Technique 1** : ~4500
- Leveraged theses crashes to test new target variant : Latest BusyBox version (v1.36.1 at the time of experiment)
- Identified the presence of previous CVE-2010-4051, CVE-2015-8776 in latest version



Overview : Crash Reuse

(Technique 2)

- Detects potential crashes similar to previously discovered ones
- Applies to similar target variants
- Similar targets : Same program but with variants w.r.t architecture, version, compiler
- Good for initial first phase, but needs fuzzing to identify zero-day vulnerabilities



Limitations

Initial tuning effort when dealing with new targets

Manual verification of generated seeds

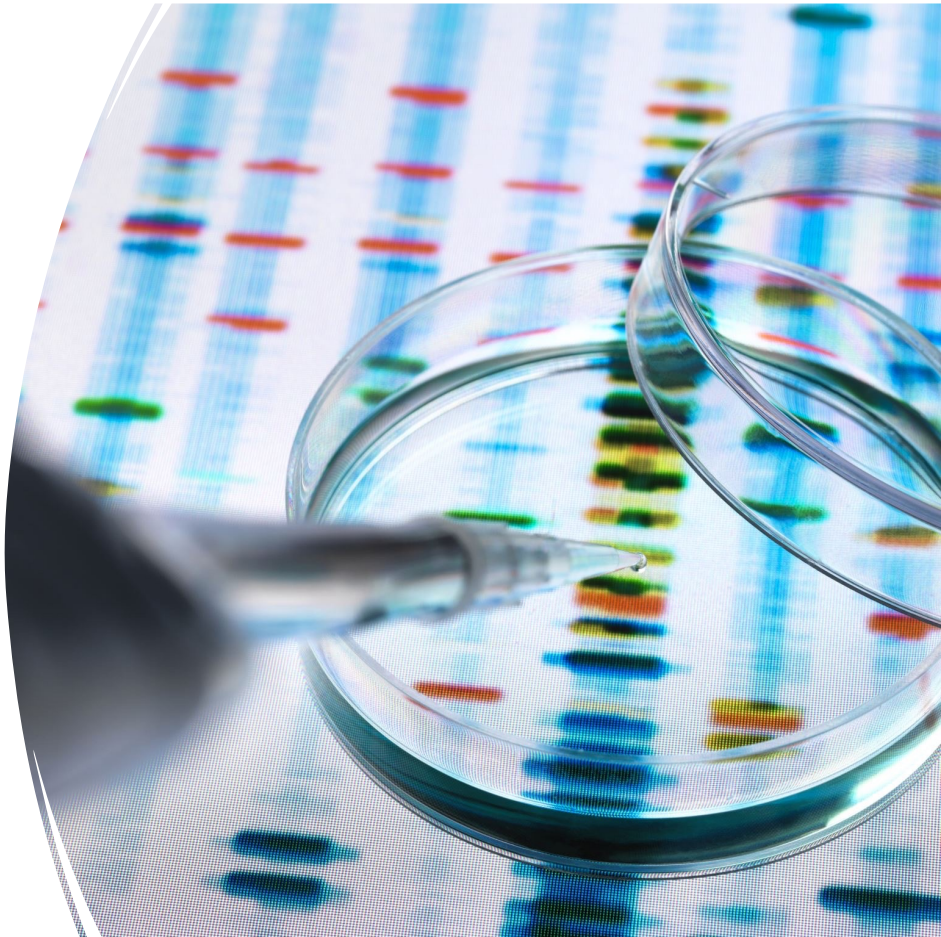
Technique1 : LLM-based seed generation

Does not guarantee identification of all bugs, especially zero-day vulnerabilities.

Technique2 : Crash Reuse

Future Work

- Extend the proposed techniques to other embedded targets other than BusyBox
- Experiment with the inclusion of LLM at different stages of fuzzing apart from the seed generation



Thank you!

Questions?

Contact : aasmita@ucdavis.edu