PASTIS: A Collaborative Approach to Combine Heterogeneous Software Testing Techniques

SBFT 2023, Melbourne, Australia

Robin David <rdavid@quarkslab.com>
Christian Heitman <cheitman@quarkslab.com>
Richard Abou Chaaya <rabouchaaya@quarkslab.com>

Quarkslab

What testing approach using? *(for security)*

Greybox Fuzzing

Testing approaches relying on executing repetitively pseudo-randomly generated inputs on the program to test. Relies on an instrumentation to obtain feedback on execution and further mutate the input if satisfactory.

+ **Very fast**
+ **Nowadays very optimized** *(constants, dictionary etc..)*

– **brutal approach**
– **No direct link between input and path taken**

What testing approach using? *(for security)*

Greybox Fuzzing

Testing approaches relying on executing repetitively pseudo-randomly generated inputs on the program to test. Relies on an instrumentation to obtain feedback on execution and further mutate the input if satisfactory.

+ **Very fast**
+ **Nowadays very optimized**
  (constants, dictionary etc..)

– **brutal approach**
– **No direct link between input and path taken**

Dynamic Symbolic Execution

(aka Whitebox Fuzzing)

Formal approach representing the path taken in the program as a mathematical formula that can be used to solve constraints in order to cover other paths.

+ **Very precise path modeling**
+ **Can solve hard paths**

– **very slow**
– **Require precise semantic modeling**

Which approach to choose ?

Which approach to choose ?

⇒

Outline

Goal

Combining greybox and whitebox fuzzing to leverage their respective strengths *(on OSS software).*

Takeaway

➢ Performed an **experimental study** of how combining different approaches together in order to **assess the relevance** of the combination.

Subsequent Contributions

TritonDSE

(whitebox fuzzer)

PASTIS

(ensemble fuzzing framework)

(Also a study on how to configure TritonDSE)

Research Questions

RQ1

Can DSE **help** a greybox fuzzing engine in a **collaborative** environment?

RQ2

Can a collaborative approach like ensemble fuzzing reach better **coverage** than the sum of its parts ?

Ensemble Fuzzing Modes

Half Duplex $\overrightarrow{\cup}$

Aggregates all inputs from engines and computes the resulting coverage *(sum)*
(not sharing mode)

Full Duplex $\overset{\leftrightarrows}{\cap}$

Maximal input sharing mode. Computes the resulting coverage *(sum with info sharing)*
(sharing mode)

How ?

Enable exchanging inputs* between engines
via a **broker** which perform configuration dispatching and
inputs sharing *(depending on mode)*. It aggregates all results.

Details:

- All communications performed over the network using a communication library called `libpastis`
- Any number of agents can connect and from anywhere *(comms in TCP)*
- One can **add a new fuzzing** agent by using libpastis and implementing few callbacks

* More in-depth information sharing have been considered but hardly suitable for heterogeneous approaches.

Supported Fuzzers

Honggfuzz

Greybox fuzzer developed by Robert Swiecki.

Modifications:

- **dynamic input injection**
- statistics retrieval

Instrumentation backend:

- source-based *(clang, gcc)*
- **QBDI** *(binary-only targets)*

AFL++

Greybox fuzzer developed as a rewrite of AFL in C++.

Modifications: ∅

Instrumentation backend:

- source-based *(clang, gcc)*
- QEMU *(binary-only targets)*

Supported whitebox fuzzer ⇒ TritonDSE

Low-level framework
Need to provide manually every instructions to execute symbolically.

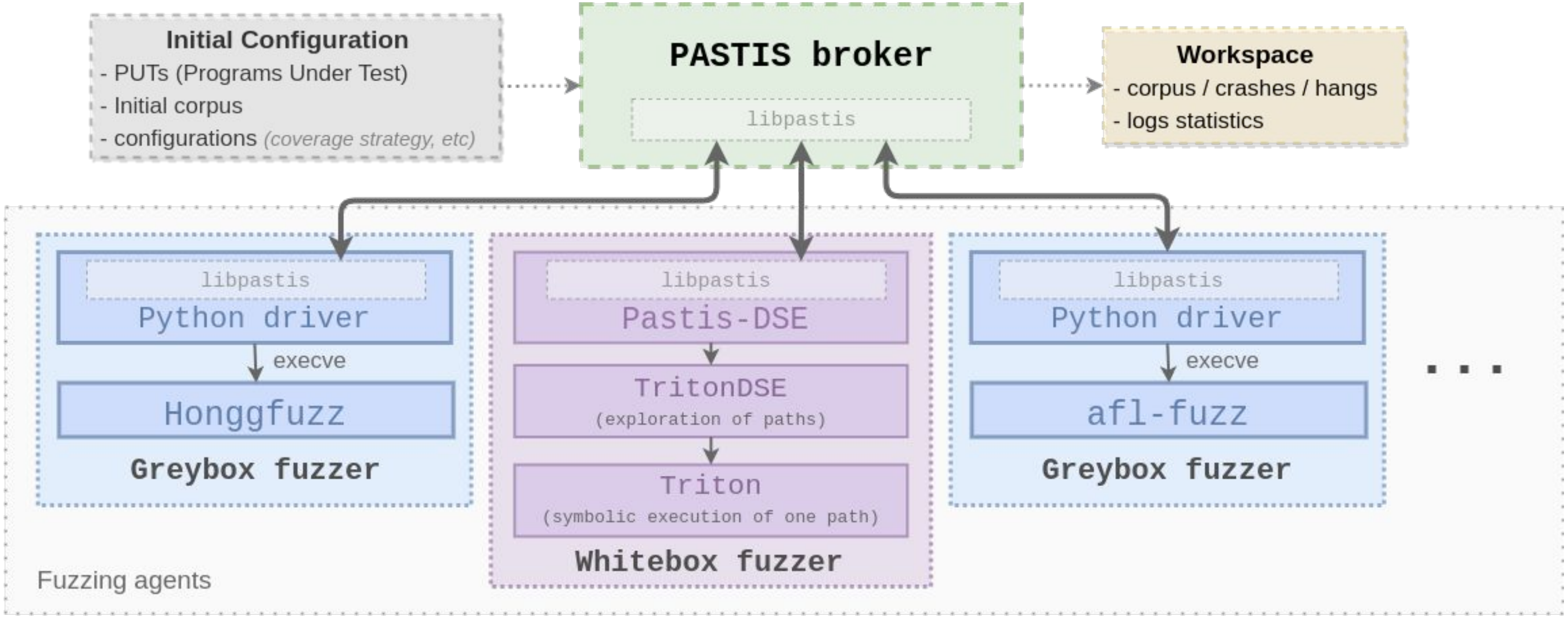
TRITON

⇓

High-level framework.
Provide all primitives to perform exploration and to craft a whitebox fuzzer

TritonDSE

Overview Collaboration



Benchmark Results

Coverage Results

	AFL++ (AFL)	honggfuzz (HF)	TritonDSE (TT)	half-duplex ($\overrightarrow{\cup}$)		full-duplex ($\overleftrightarrow{\cap}$)		
				cov	incr-HF	cov	incr-HF	incr-$\overrightarrow{\cup}$
cyclone	1249	1541	1149	**1546**	+5	1544	+3	-2
freetype	3703	12946	3305	**13046**	+100	12865	-81	-181
harfbuzz	4083	7773	3702	7773	+0	7678	-95	-95
libjpeg	1588	1944	841	1945	+1	**2180**	+236	+237
libpng	797	1005	432	**1016**	+11	978	-27	-38
openthread	1693	2084	1095	**2097**	+13	1963	-121	-134
vorbis	1480	1593	1022	1594	+1	**1596**	+3	+2
zlib	537	541	87	541	+0	534	-7	-7

TABLE IV: Final coverage comparison, fuzzers alone and half/full-duplex

Coverage Results

Very good results

	AFL++ (AFL)	honggfuzz (HF)	TritonDSE (TT)	half-duplex ($\overrightarrow{\cup}$)		full-duplex ($\overset{\rightleftarrows}{\cap}$)		
				cov	incr-HF	cov	incr-HF	incr-$\overrightarrow{\cup}$
cyclone	1249	1541	1149	**1546**	+5	1544	+3	-2
freetype	3703	12946	3305	**13046**	+100	12865	-81	-181
harfbuzz	4083	7773	3702	7773	+0	7678	-95	-95
libjpeg	1588	1944	841	1945	+1	**2180**	+236	+237
libpng	797	1005	432	**1016**	+11	978	-27	-38
openthread	1693	2084	1095	**2097**	+13	1963	-121	-134
vorbis	1480	1593	1022	1594	+1	**1596**	+3	+2
zlib	537	541	87	541	+0	534	-7	-7

TABLE IV: Final coverage comparison, fuzzers alone and half/full-duplex

Coverage Results

Very good results

Slight improvement in half-duplex *(AFL++, TritonDSE find input that HF don't)*

	AFL++ (AFL)	honggfuzz (HF)	TritonDSE (TT)	half-duplex ($\overrightarrow{\cup}$)		full-duplex ($\overleftrightarrow{\cap}$)		
				cov	incr-HF	cov	incr-HF	incr-$\overrightarrow{\cup}$
cyclone	1249	1541	1149	**1546**	+5	1544	+3	-2
freetype	3703	12946	3305	**13046**	+100	12865	-81	-181
harfbuzz	4083	7773	3702	7773	+0	7678	-95	-95
libjpeg	1588	1944	841	1945	+1	**2180**	+236	+237
libpng	797	1005	432	**1016**	+11	978	-27	-38
openthread	1693	2084	1095	**2097**	+13	1963	-121	-134
vorbis	1480	1593	1022	1594	+1	**1596**	+3	+2
zlib	537	541	87	541	+0	534	-7	-7

TABLE IV: Final coverage comparison, fuzzers alone and half/full-duplex

Coverage Results

Very good results

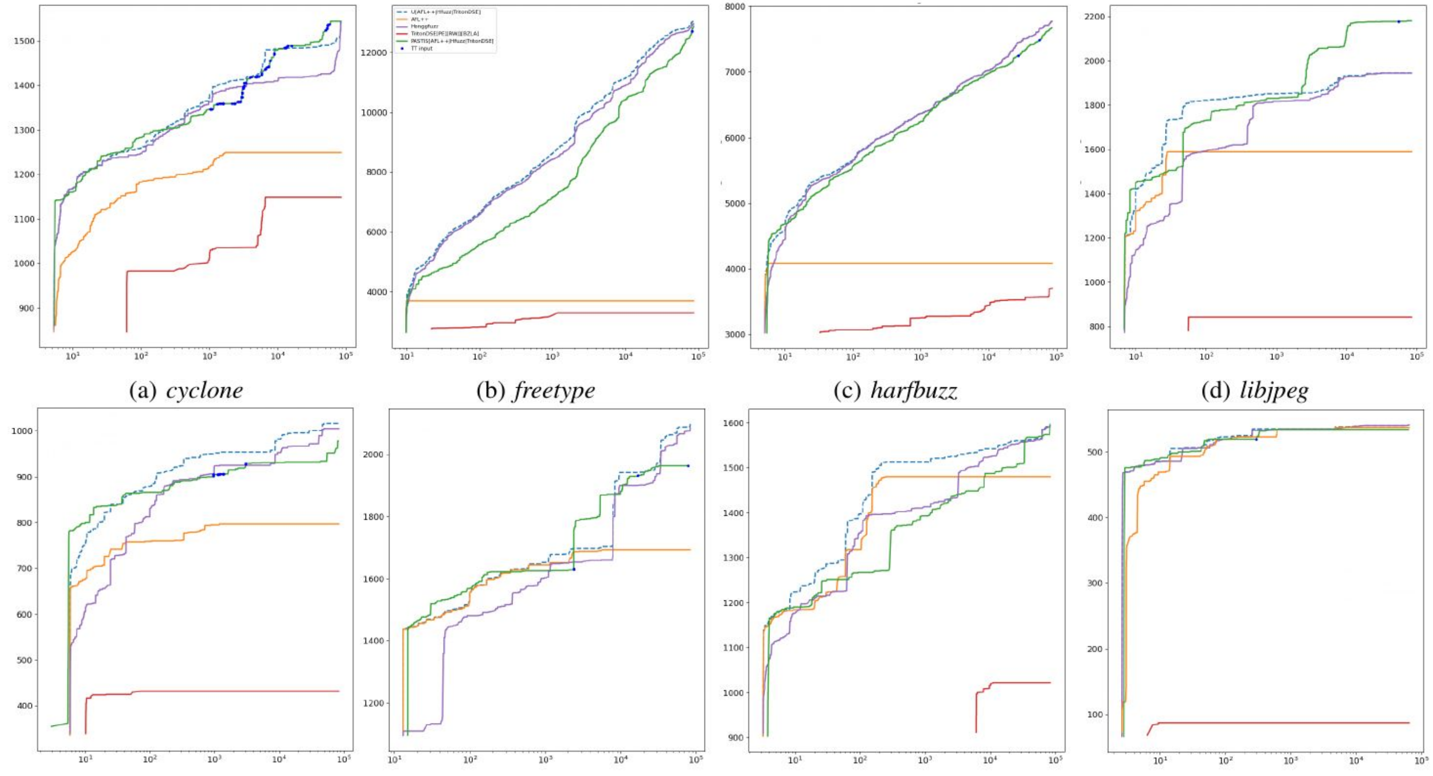
Slight improvement in half-duplex *(AFL++, TritonDSE find input that HF don't)*

Full-duplex outperform on two targets *(solely)*

	AFL++ (AFL)	honggfuzz (HF)	TritonDSE (TT)	half-duplex ($\overrightarrow{\cup}$)		full-duplex ($\overset{\rightleftarrows}{\cap}$)		
				cov	incr-HF	cov	incr-HF	incr-$\overrightarrow{\cup}$
cyclone	1249	1541	1149	**1546**	+5	1544	+3	-2
freetype	3703	12946	3305	**13046**	+100	12865	-81	-181
harfbuzz	4083	7773	3702	7773	+0	7678	-95	-95
libjpeg	1588	1944	841	1945	+1	**2180**	+236	+237
libpng	797	1005	432	**1016**	+11	978	-27	-38
openthread	1693	2084	1095	**2097**	+13	1963	-121	-134
vorbis	1480	1593	1022	1594	+1	**1596**	+3	+2
zlib	537	541	87	541	+0	534	-7	-7

TABLE IV: Final coverage comparison, fuzzers alone and half/full-duplex

Coverage Evolution *(24h)*

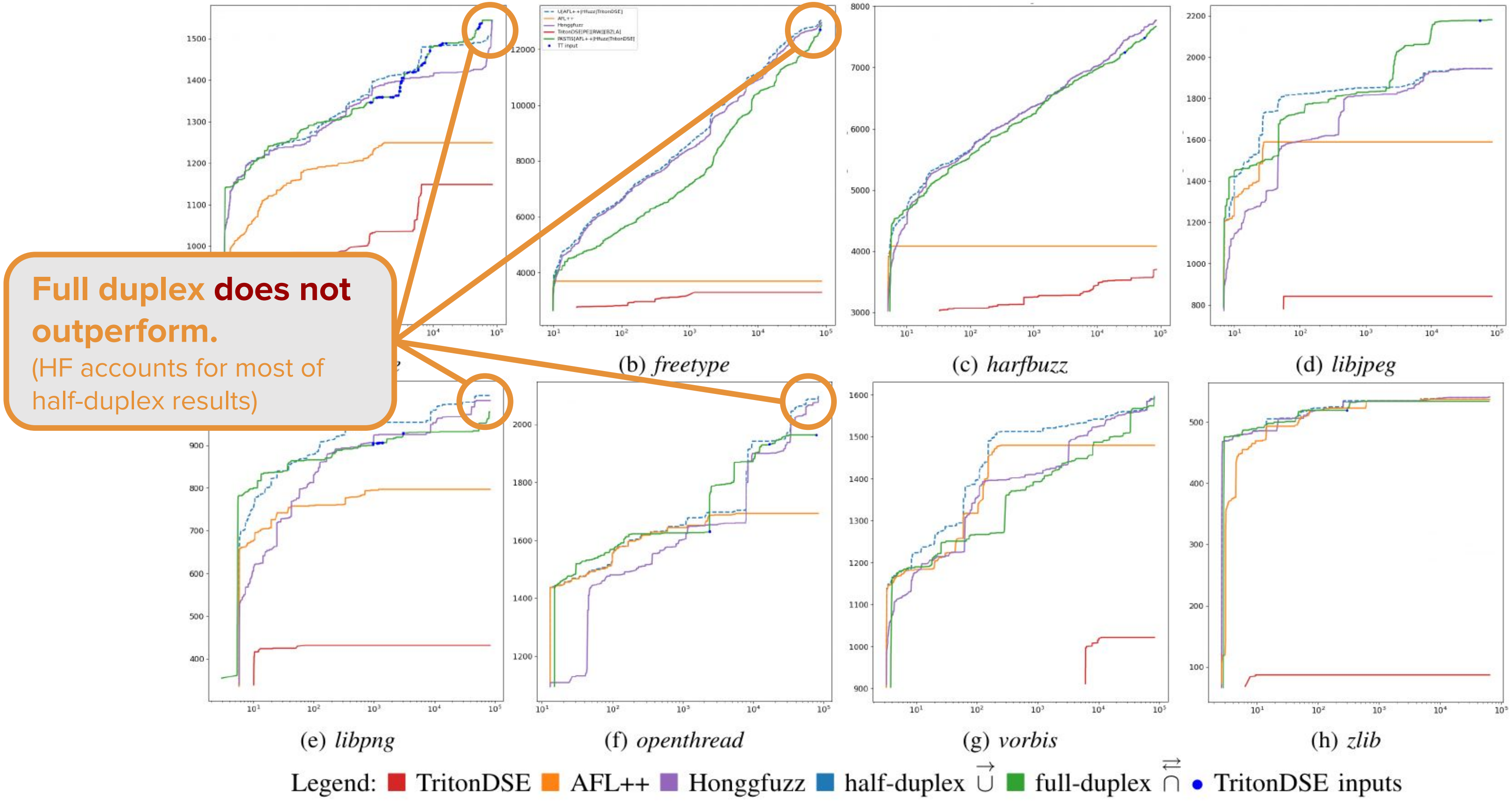


(a) *cyclone* (b) *freetype* (c) *harfbuzz* (d) *libjpeg*

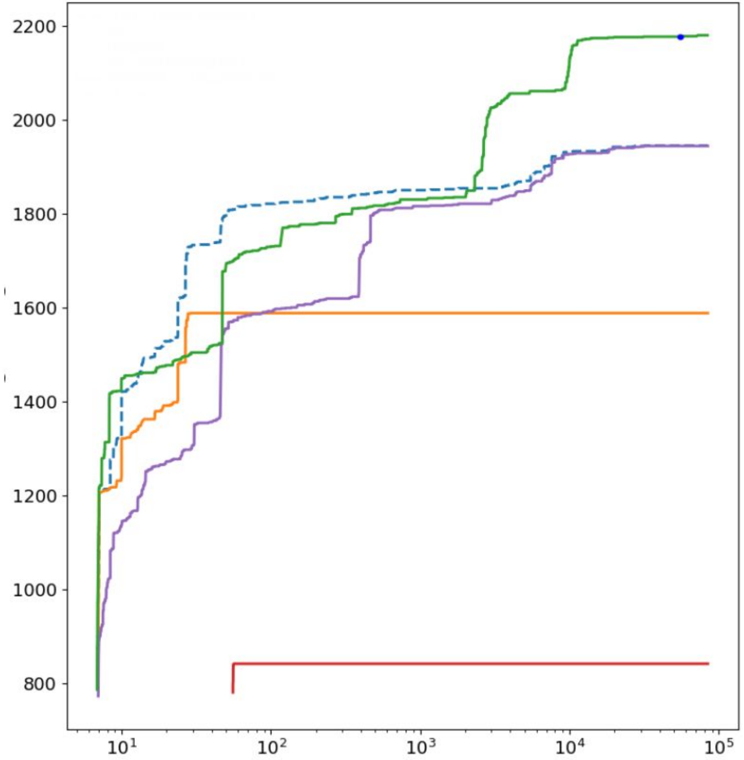
(e) *libpng* (f) *openthread* (g) *vorbis* (h) *zlib*

Legend: TritonDSE AFL++ Honggfuzz half-duplex $\overrightarrow{\cup}$ full-duplex $\overset{\rightleftarrows}{\cap}$ • TritonDSE inputs

Coverage Evolution *(24h)*



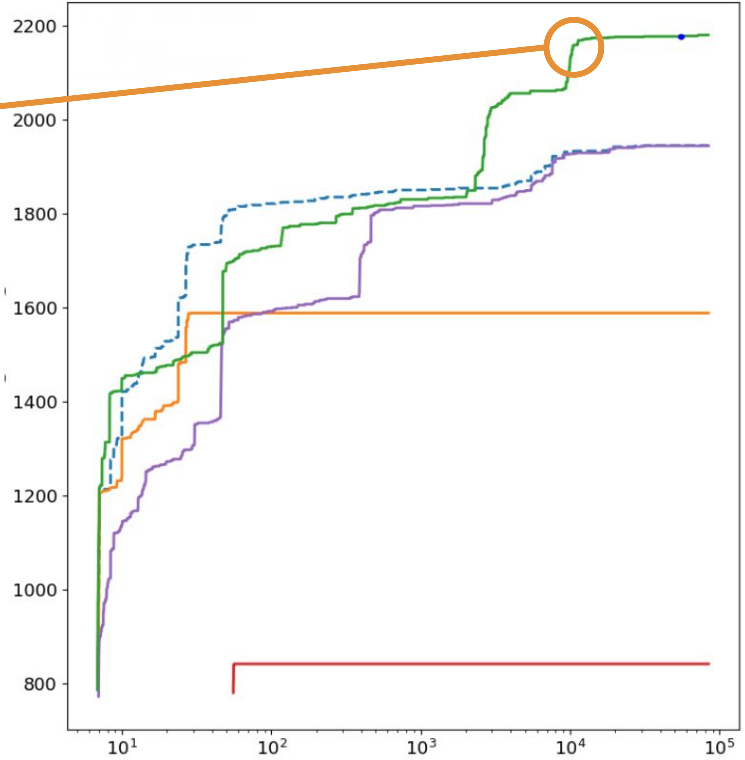
Zoom (1/3): Libjpeg



Legend: TritonDSE AFL++ Honggfuzz half-duplex $\overrightarrow{\cup}$ full-duplex $\overleftrightarrow{\cap}$ • TritonDSE inputs

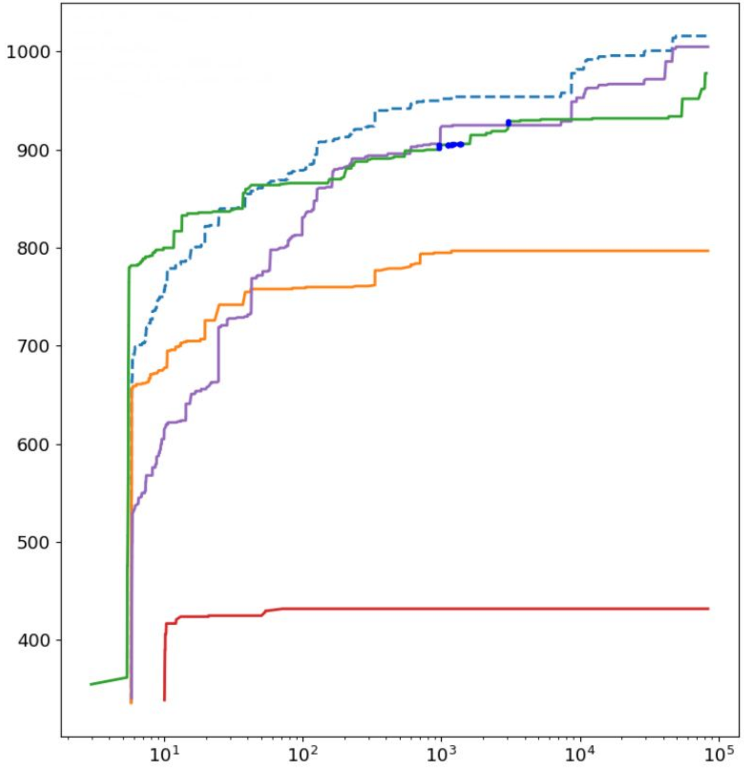
Zoom (1/3): Libjpeg

Full-duplex outperform
(only HF and AFL++)



Legend: TritonDSE AFL++ Honggfuzz half-duplex $\overrightarrow{\cup}$ full-duplex $\overleftrightarrow{\cap}$ • TritonDSE inputs

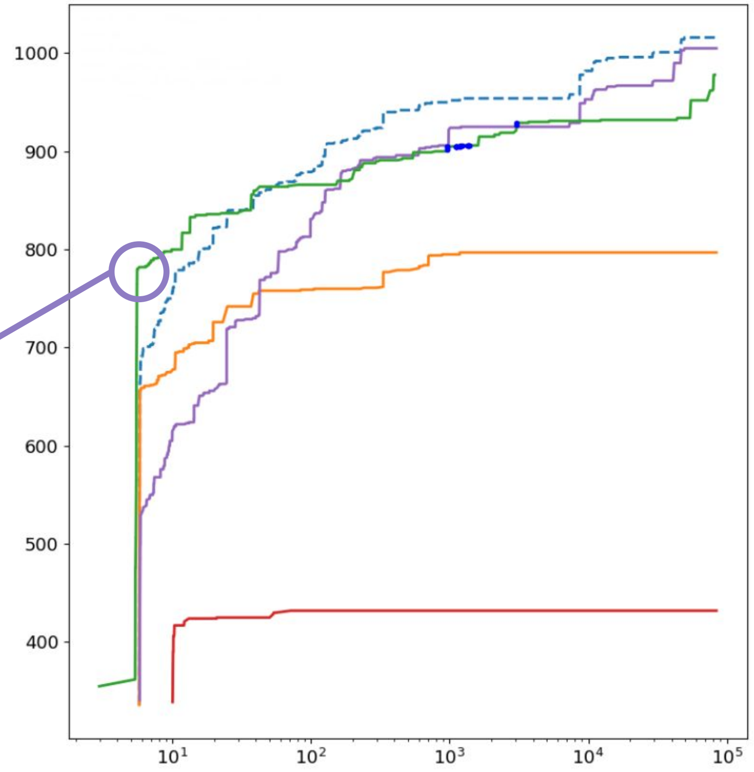
Zoom (2/3): Libpng



Legend: TritonDSE AFL++ Honggfuzz half-duplex $\overrightarrow{\cup}$ full-duplex $\overleftrightarrow{\cap}$ • TritonDSE inputs

Zoom (2/3): Libpng

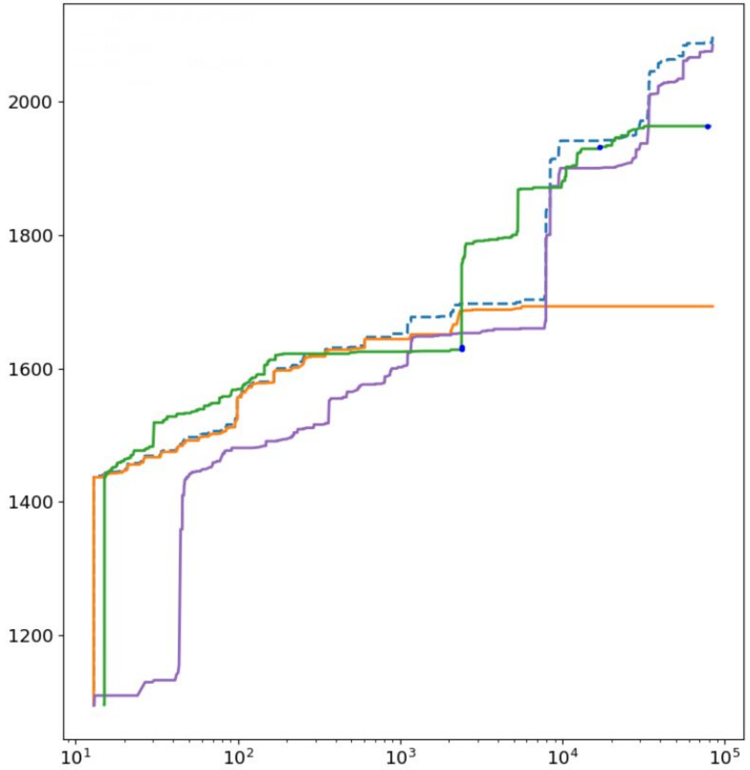
Temporarily outperform *(short-term campaign)*



Legend: TritonDSE AFL++ Honggfuzz half-duplex $\overrightarrow{\cup}$ full-duplex $\overleftrightarrow{\cap}$ • TritonDSE inputs

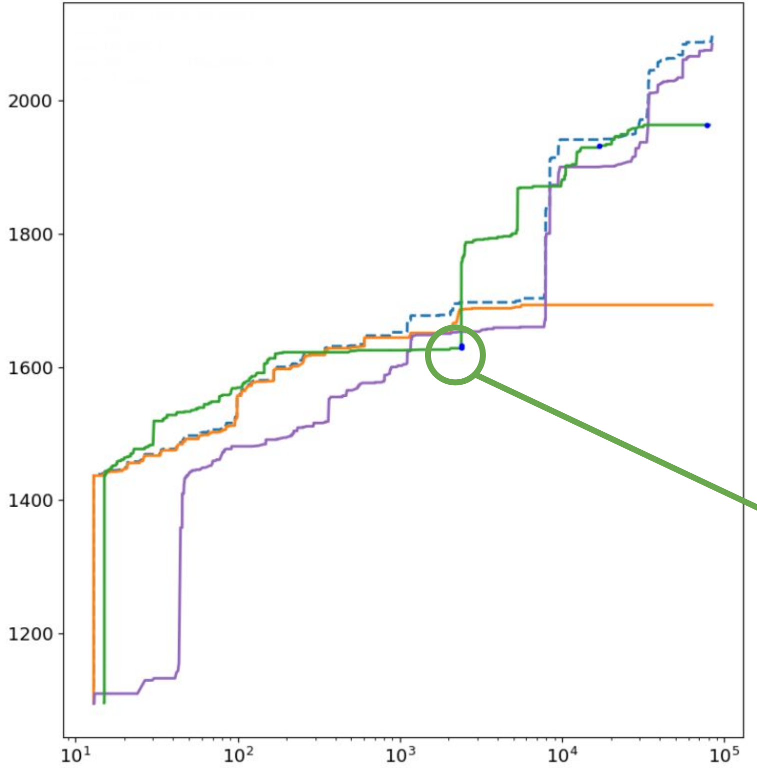
Might explain why PASTIS performs well on competition bug finding

Zoom (3/3): Openthread



Legend: TritonDSE AFL++ Honggfuzz half-duplex $\overrightarrow{\cup}$ full-duplex $\overleftrightarrows{\cap}$ • TritonDSE inputs

Zoom (3/3): Openthread



TritonDSE unlocks the coverage

Legend: ■ TritonDSE ■ AFL++ ■ Honggfuzz ■ half-duplex $\overrightarrow{\cup}$ ■ full-duplex $\overset{\rightleftarrows}{\cap}$ • TritonDSE inputs

Conclusion

&

Future Work

Quarkslab

Conclusions & Future Work

Conclusion:

- Honggfuzz is very effective and produces **numerous** inputs
  (half/full duplex gain is marginal)
- On few targets DSE helps *(RQ#1)* and the collaborative fuzzing **can** provides better results on **some** targets *(RQ#2)*
- Contrasting instrumentation *(HF vs TritonDSE)*

Future Work:

⇒ Pure **binary-only** experiments ! *(already ongoing..)*

⇒ Leveraging the Fuzzbench framework *(for averaged results computation or to cast PASTIS results into fuzzbench format)*

Questions ?

Quarkslab