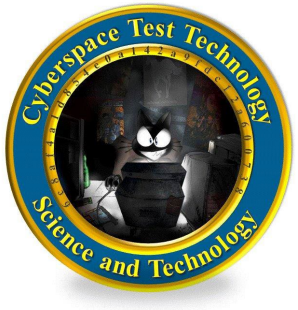


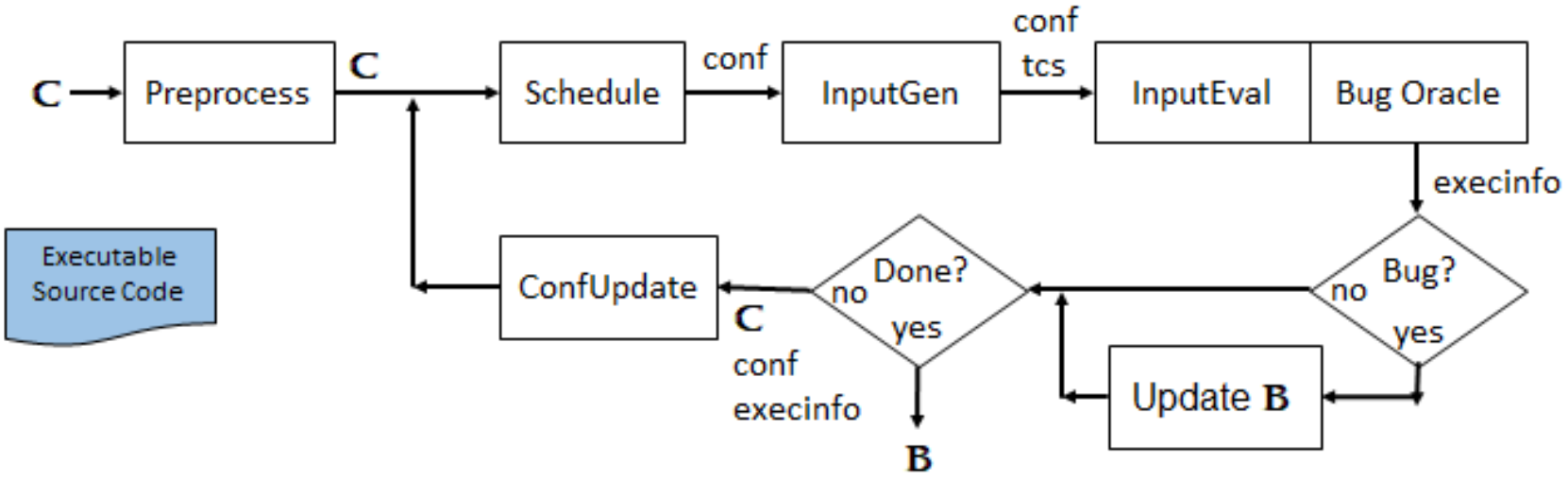
Test Resource Management Center (TRMC)

U.S. Government Fuzzing Initiative

BLUF

- **TRMC is providing fuzzing tools and knowledge to the US Government (USG) to meet USG software security needs**
  - TRMC is developing a modular fuzzer (named the Vader Modular Fuzzer) to address the broad range of USG fuzzing needs
  - TRMC is setting up a USG Fuzzing Working Group to expand fuzzing knowledge and expertise across the USG

The Basic Fuzzing Algorithm



Key
$\mathbb{C}$ – set of configurations
conf – fuzz configuration for an iteration
tcs – the input for an execution
execinfo – information about an execution
$\mathbb{B}$ – set of bugs found

Reference Manès 2019

ALGORITHM 1: Fuzz Testing

```
Input : ℂ, t_limit
Output : 𝔹 // a finite set of bugs
1 𝔹 ← ∅
2 ℂ ← PREPROCESS(ℂ)
3 while t_elapsed < t_limit ∧ CONTINUE(ℂ) do
4     conf ← SCHEDULE(ℂ, t_elapsed, t_limit)
5     tcs ← INPUTGEN(conf)
      // O_bug is embedded in a fuzzer
6     𝔹′, execinfos ← INPUTEVAL(conf, tcs, O_bug)
7     ℂ ← CONFUPDATE(ℂ, conf, execinfos)
8     𝔹 ← 𝔹 ∪ 𝔹′
9 return 𝔹
```

Why Fuzzing - Industry

- **Some interesting quotes from Microsoft**
  - ***"Fuzzing is an effective way to find security bugs in software, so much so that the Microsoft Security Development Lifecycle requires fuzzing at every untrusted interface of every product."***
    - Microsoft, "A brief introduction to fuzzing and why it's and important tool for developers", March 4, 2020
  - *"**'Fuzz testing is a highly effective method for increasing the security and reliability of native code – it is the gold standard for finding and removing costly, exploitable security flaws.'** said Microsoft Security's Justin Campbell, a principal security software engineering lead, and Mike Walker, a senior director, special projects management."*
    - *ZDNet, "Microsoft: Windows 10 is hardened with these fuzzing security tools – now they're open source", September 15, 2020*

Industry is fuzzing extensively.

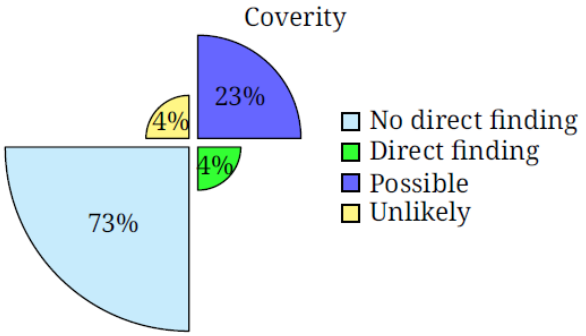
Why Fuzzing – DoD Observations

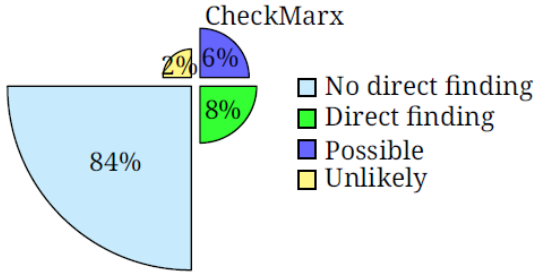
DoD is recognizing value of fuzzing.

- **Quotes from AvMC Report**
  - "Of the 45 failure cases found through fuzzing, neither static code analysis tool found more than ten percent of the failures directly."
  - "The results demonstrate that there are flaws in the software that are not discovered by the [Static Code Analysis] tools employed."
  - "The results suggest that a two-prong approach of using both methods provides a more comprehensive software assurance analysis."

An AvMC report(1) found that fuzzers and static analysis tools found complementary sets of bugs.

(1) Alexander, Hood and Jones, "A Comparison of Fuzzing Dynamic Analysis and Static Code Analysis", DEVCOM-AvMC, Nov 16, 2020





The Challenges of Fuzzing – USG Specific

- **Affordability**
  - Commercial license costs across USG
- **USG Niche Platforms**
  - Real Time OSes (e.g. VxWorks)
  - Cyber Physical Systems (CPS)
  - PowerPCs and other less common processors
- **Selection of Fuzzer Tools (See next three slides)**
  - No single best fuzzer
  - There is an explosion in number of Open Source fuzzers from which to draw
    - More that 300 academic papers on fuzzing are published worldwide each year and there are over 4800 GitHub public repositories related to fuzzing
    - Little concern for extensibility of any given tool
  - Costs associated with having teams be experts in a variety of useful fuzzing tools

Challenges in Fuzzing – No "Best" Fuzzer – View #1

There is no "One Best fuzzer".
Different fuzzers perform differently on different targets(1,2)

Li et al, *UniFuzz: A Holistic and Pragmatic Metrics-Driven Platform for Evaluating Fuzzers*, 2020-10-05

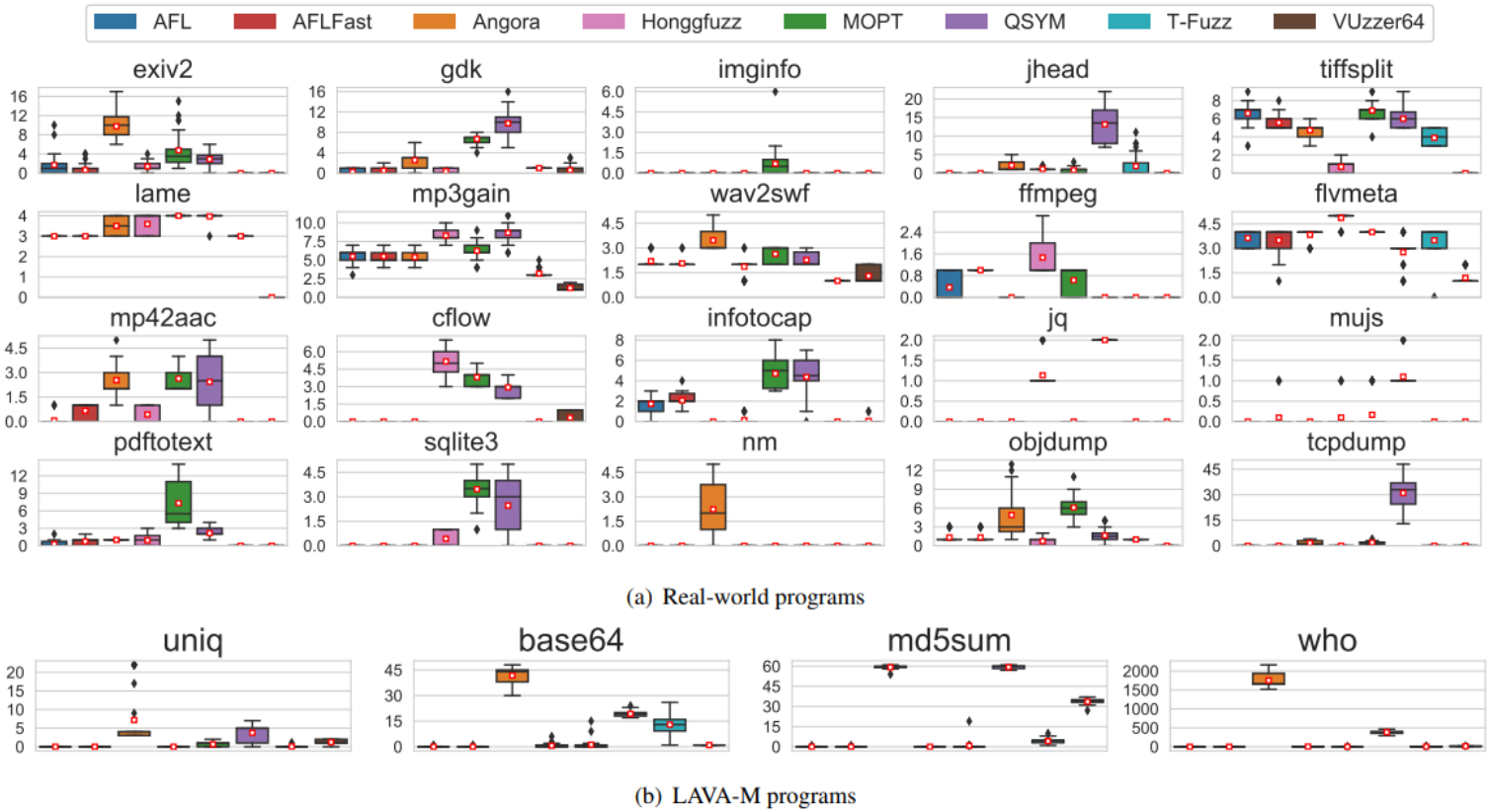


Figure 2: The number of unique bugs detected by fuzzers.

(1) Li et al, *UniFuzz: A Holistic and Pragmatic Metrics-Driven Platform for Evaluating Fuzzers*, 2020-10-05
(2) Hazimeh et al, *Magma: A Ground-Truth Fuzzing Benchmark*, 2020-10-23

TRMC Fuzzing Vision

Get broad USG adoption of fuzzing to better secure USG systems from cyber attack

- Awareness
- Adoption
- Training
- Share Knowledge & Lessons Learned
- Tool Availability

TRMC Actions

- Government Fuzzing Working Group
- Develop New Fuzzing Tools for Government Needs

Fuzzing Tool Vision

USG Fuzzing Tool "Requirements"

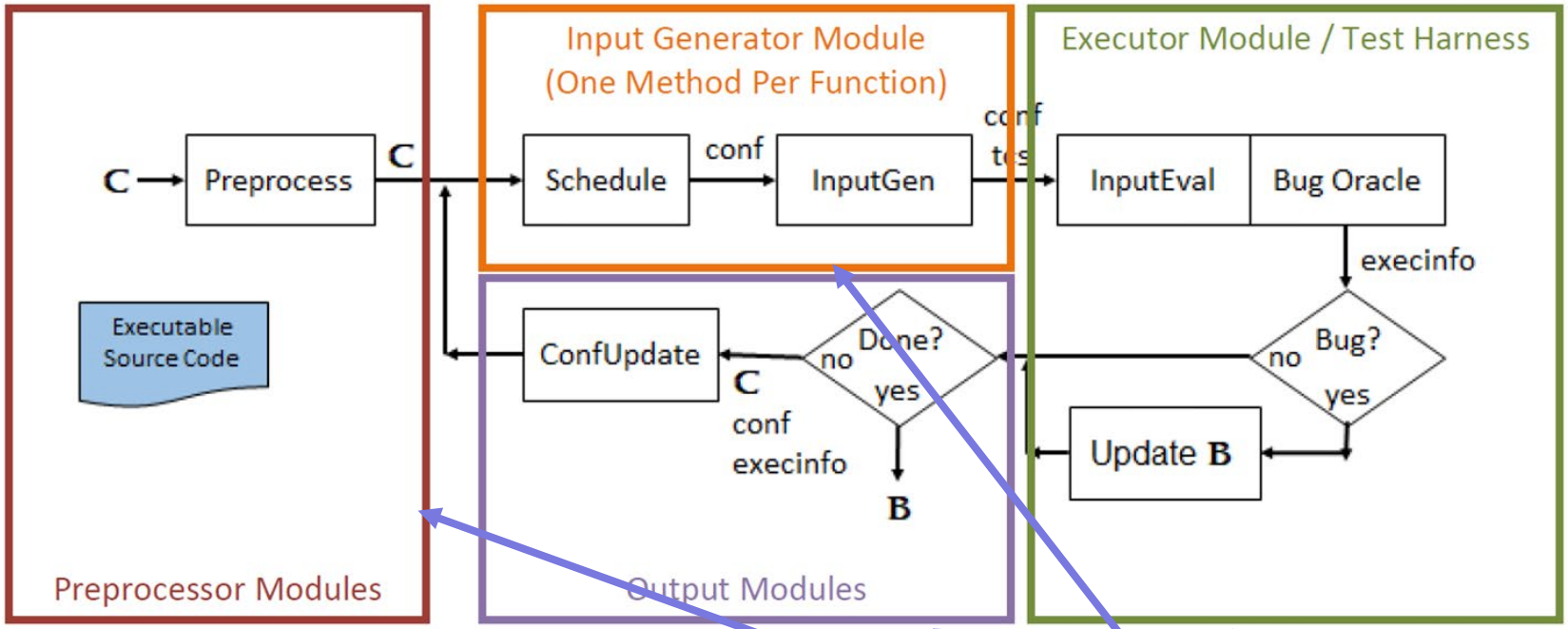
- **Affordable**
  - **Eliminate License Fees**
  - **Leverage unused USG server resources (e.g. National Cyber Range)**
  - **Leverage "Other People's Money" for fuzzer tool development and maintenance**
- **Available to All USG**
- **State of the Art**
- **Tailorable to SUT**
- **Can run in closed spaces**
- **Reduce required expertise**
  (e.g. don't have to know 10 different fuzzers)

Means to that End

- **Build tools leveraging existing open source fuzzers where suitable**
- **Build true MOSA fuzzer architecture to**
  - Allow users to tailor fuzzers to specific USG Program and Tester needs
  - Recreate common fuzzers, but within one user interface instead of dispirit interfaces
  - Create focal point for open source fuzzing development

Vader Modular Fuzzer

What is VMF – *A Modular Fuzzer*



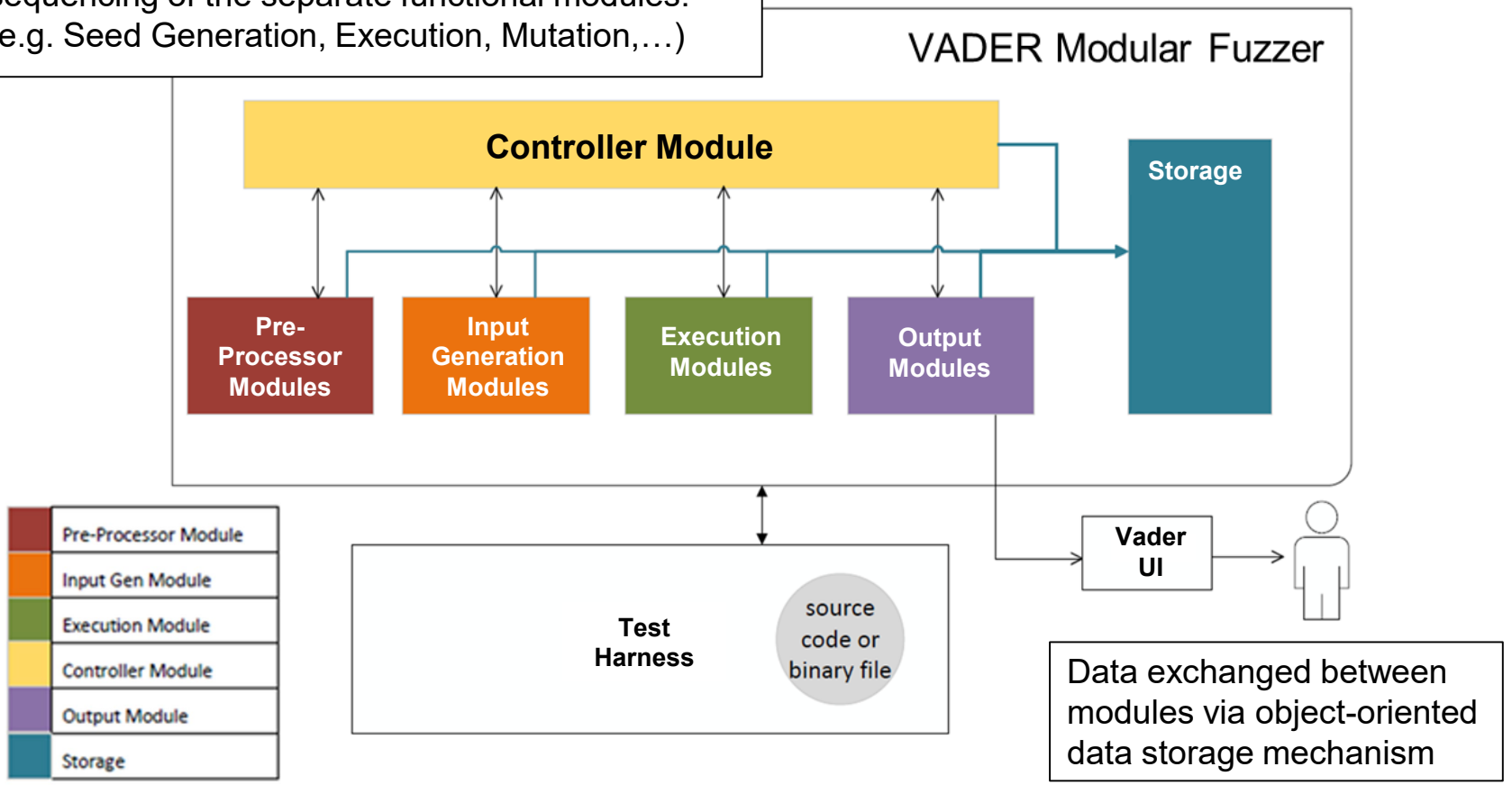
- The Vader Modular Fuzzer breaks the fuzzer functional steps into separate modules instead one monolithic fuzzer.
- Swap/combine modules to compose new fuzzers and fuzzing campaigns

Interchangeable modules

What is VMF – *A Modular Fuzzer*

The Vader Modular Fuzzer is built on an underlying Vader infrastructure that orchestrates the integration and execution of the separate modules/

Controller module calls and coordinates the sequencing of the separate functional modules: (e.g. Seed Generation, Execution, Mutation,…)



Data exchanged between modules via object-oriented data storage mechanism

Vader Modular Fuzzer Status

- **Currently in Phase 1 of Phase 4**
- **Code base and availability**
  - *No Cost, source available to the Government*
  - *Vader 2.0 MVP release – June 2022*
  - *Hosted on JMETC (OSD TRMC site)*
  - *Fuzzing WG provides status and availability*
- **TRL**
  - *TRL 5-6*

US Government Fuzzing Working Group Charter

- Hosted and Led by TRMC (Test Resource Management Center) T&E/S&T (Test & Evaluation/Science & Technology) CTT (Cyberspace Test Technology)
- Purposes:
  - Develop U. S. Government modular fuzzer and promote fuzzing technologies to U. S Government agencies for testing and DevSecOps
  - Provide a venue for information exchange
  - Solve challenges in fuzzing on critical U. S. Government systems
- Membership:
  - DoD and other U.S. Government organizations, support contractors (.mil email addresses), FFRDC, UARC, and academia
- Meeting Frequency
  - Every 3-6 months
- Funding:
  - Travel and labor costs will be borne by the member's own command

WG Meetings

- 1st Meeting
  - 15 June 2022, 1200-1500 ET, Virtual
  - Fuzzing 101
  - Fuzzing Framework overview
  - Fuzzing Research
- 2nd Meeting
  - 24 August 2022, 1200-1500 ET, Virtual
  - Fuzzing Frame Release (initial version) and Training
  - Current U.S. Government fuzzing landscape (A survey form will be provided)
- 3rd Meeting
  - 13-14 September 2022, in-person, Draper, Boston
  - Hands on training of the fuzzing framework; Bring SUT of your interests to the meeting

If interested in participating or getting notified of upcoming meetings, email Travis Watson (travis.watson.5.ctr@us.af.mil) AND Min Kim (jeong.kim@us.af.mil) to be added to the distro