



# A Bug Hunter's Reflections on Fuzzing

Alexander Popov

Positive Technologies

25.05.2024



# About Me

- Alexander Popov
- Linux kernel developer since 2012
- Open-source maintainer
- Principal security researcher at

## ■ **positive technologies**

- Speaker at conferences including:

OffensiveCon, Nullcon Goa, Linux Security Summit, Still Hacking Anyway, Zer0Con, Positive Hack Days, ZeroNights, HighLoad++, Open Source Summit, OS Day and Linux Plumbers  
[a13xp0p0v.github.io/conference\\_talks](http://a13xp0p0v.github.io/conference_talks)



## Intro

- Been thinking about this topic for several years
- Have wanted to structure these thoughts
- Giving a talk and creating a discussion
  - is a great way to do that
- Haven't found any conference talks about that 🤔
- People tend to keep their know-how to themselves 😊
- So, let's do this!





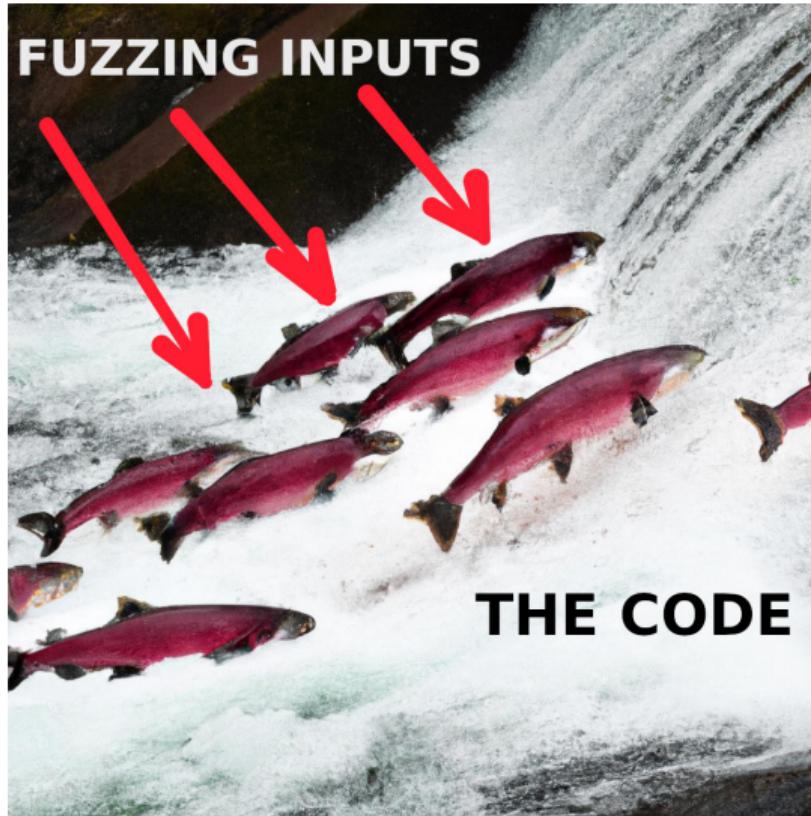
What is fuzzing?

Fuzzing is...



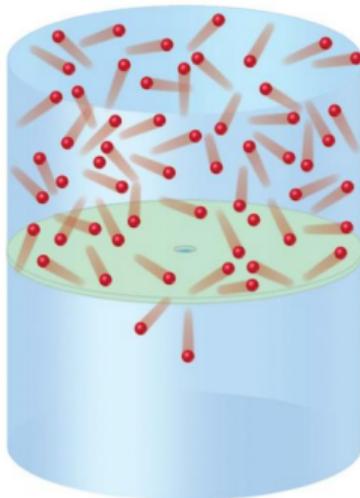
## What is fuzzing?

Ha-ha! I'm joking,  
everybody here already knows  
what fuzzing is.



generated with DALL-E 2

## Effusion

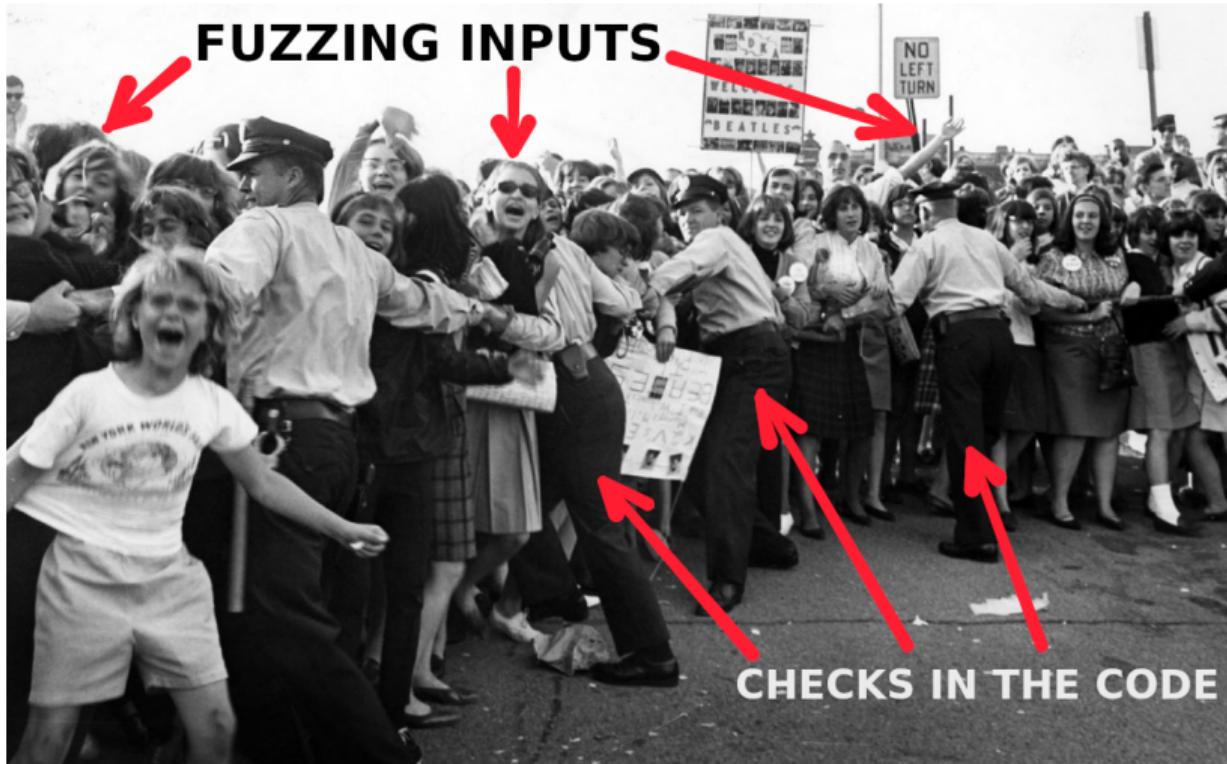


Effusion is the escape of gas molecules through a tiny hole into an evacuated space.

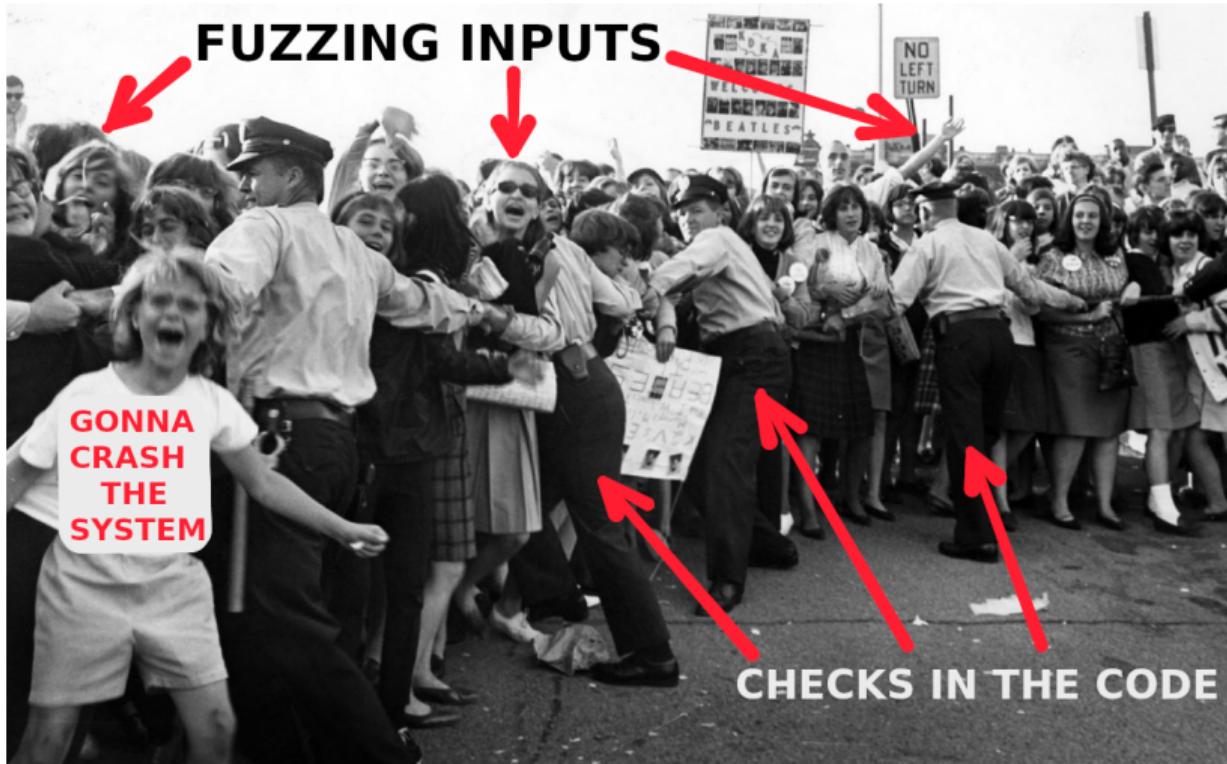


© 2009, Prentice-Hall, Inc.

[Chemistry, The Central Science, 11th edition](#)



Police link arms in an attempt to hold back Beatles fans (Dale Gleason/The Pittsburgh Press)



Police link arms in an attempt to hold back Beatles fans (Dale Gleason/The Pittsburgh Press)

# I Think Fuzzing Is...



Fuzzing is...

a great way to delegate  
boring software testing to computers  
(but you need to have control over it)

## Software developer



- ① Uses fuzzing to search for bugs
- ② Usually interested in all bugs
- ③ Have access to the source code
- ④ Enables all available debug features

## Security researcher



- ① Uses fuzzing to discover vulnerabilities
- ② Not interested in all bugs
- ③ Interested in vulnerabilities (bugs reachable via attack surface)
- ④ May not have access to the source code
- ⑤ More interested in bugs with stable reproducers
- ⑥ More interested in unique bugs

# The Main Question



## Question

What is special about fuzzing  
for vulnerability discovery?

- Security researchers usually don't discuss this...
- But today we will!
- As an example, I'll use my favorite kernel fuzzer, **syzkaller**

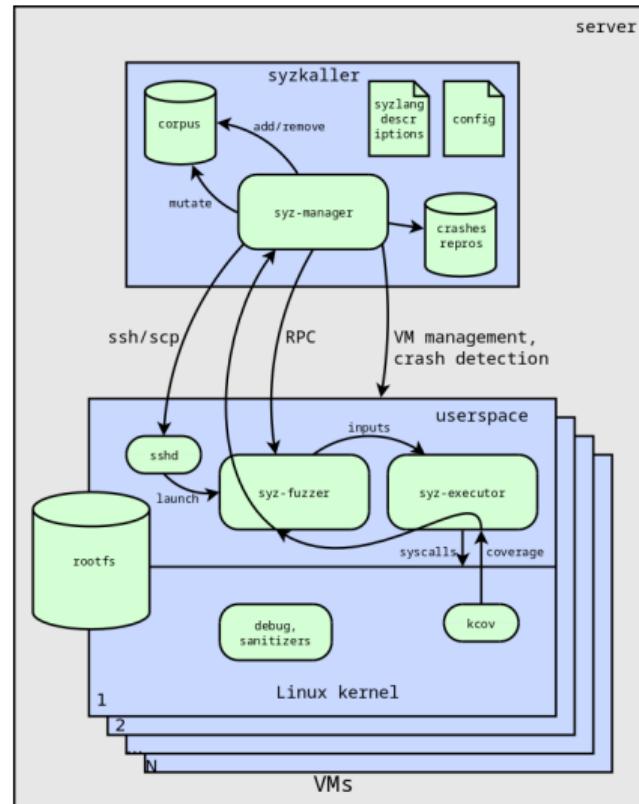
**syzkaller - kernel fuzzer**

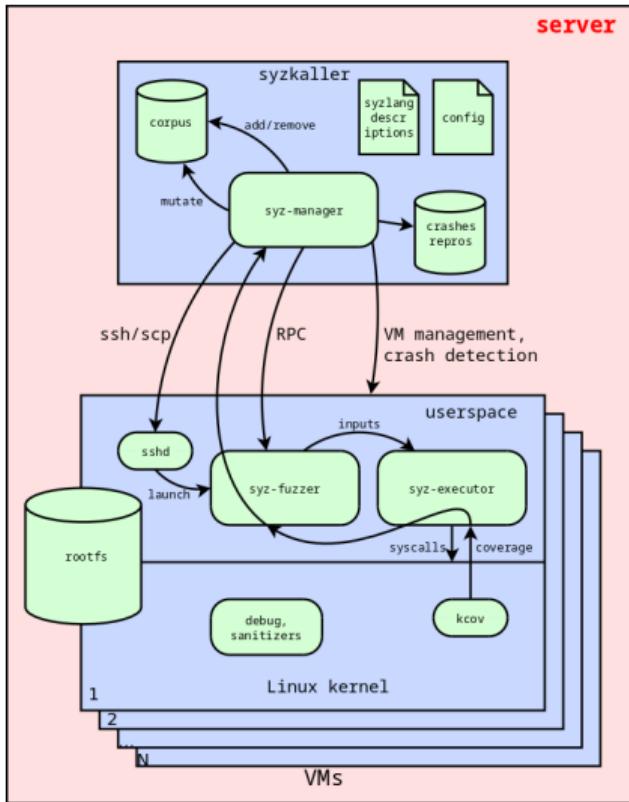
ci passing oss-fuzz fuzzing go report A+ codecov 61% reference License Apache 2.0

syzkaller ([si:z'ko:lə]) is an unsupervised coverage-guided kernel fuzzer.  
Supported OSes: FreeBSD, Fuchsia, gVisor, Linux, NetBSD, OpenBSD, Windows.

# Syzkaller Architecture



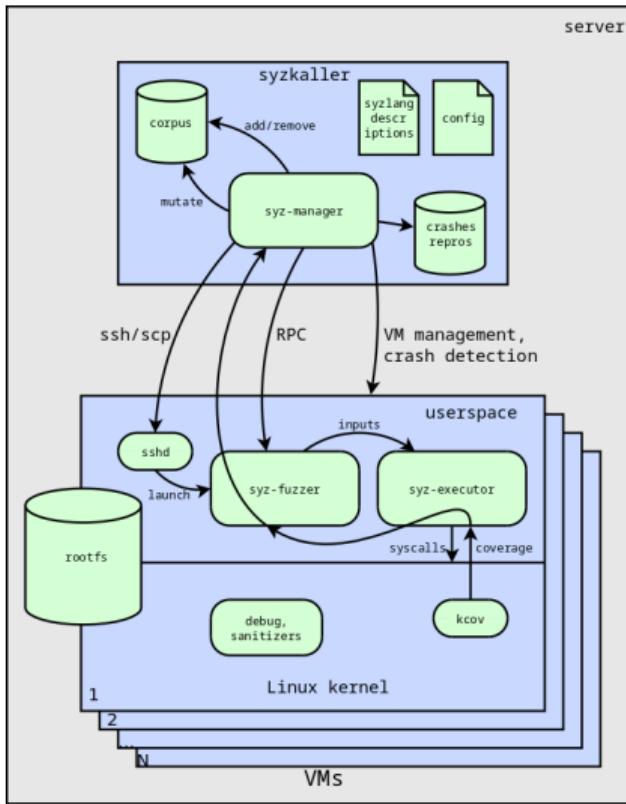
# A Couple of Words About Hardware



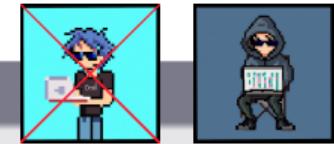
## What about a server for syzkaller?

- ➊ It needs hardware virtualization
- ➋ Unusual characteristics for a server:
  - ➌ Number of CPU cores is crucial
  - ➌ RAM  $\approx 4\text{GB} * (\text{CPU\_N} / 2)$
  - ➌ No huge hard drive needed unless tracing or snapshots are used
- ➌ It can run on:
  - ➌ Dedicated server (it needs to be customized; otherwise, you'll overpay)
  - ➌ VPS with nested virtualization (not many options)

# Back to the Research Goals Again



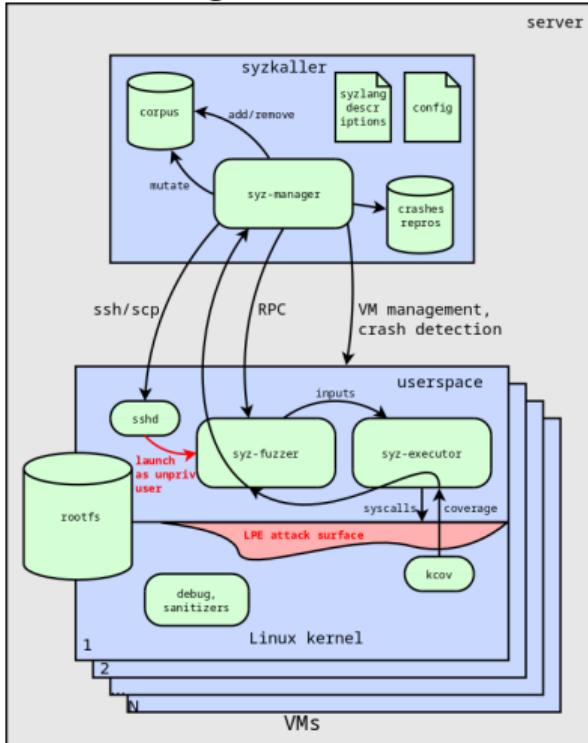
## Adapt to vuln discovery



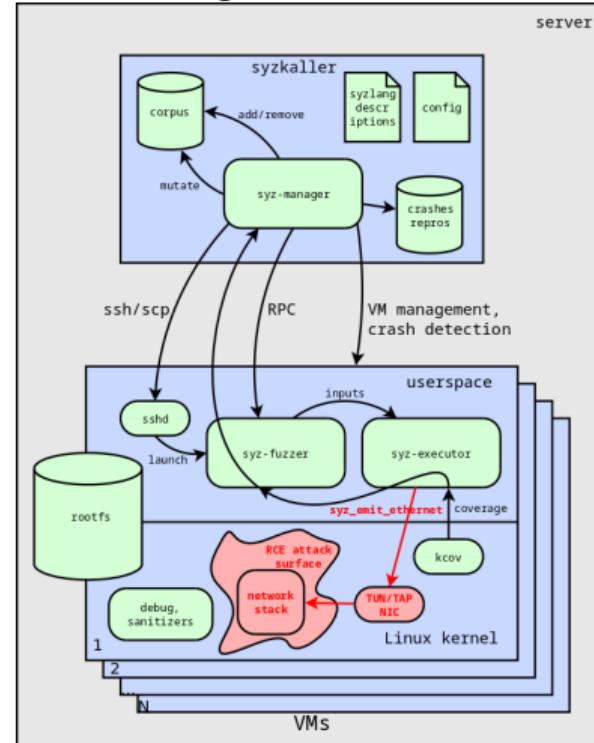
- ① Not interested in all bugs, interested in vulnerabilities (bugs reachable via attack surface)
- ② More interested in bugs with stable reproducers
- ③ More interested in unique bugs

# 1) Not Interested in All Bugs, Interested in Vulnerabilities

## Fuzzing for potential LPE

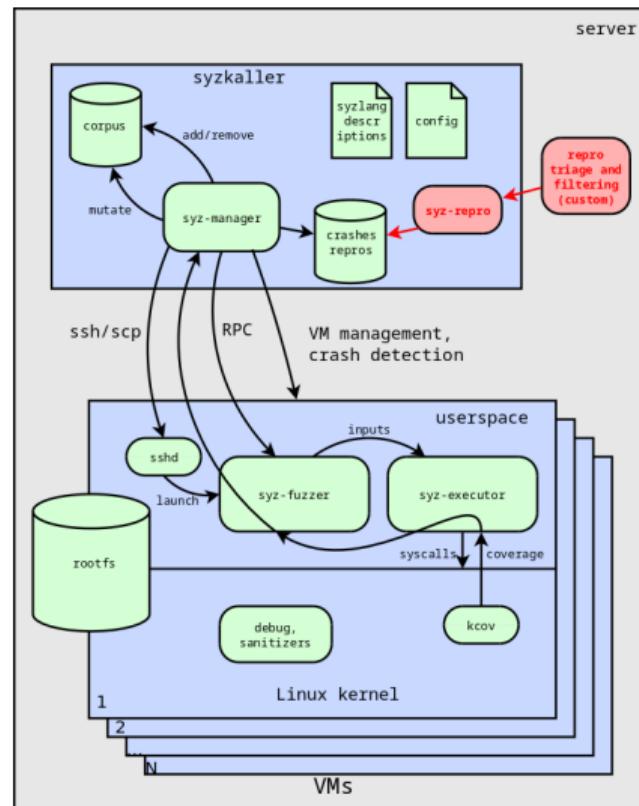


## Fuzzing for potential RCE

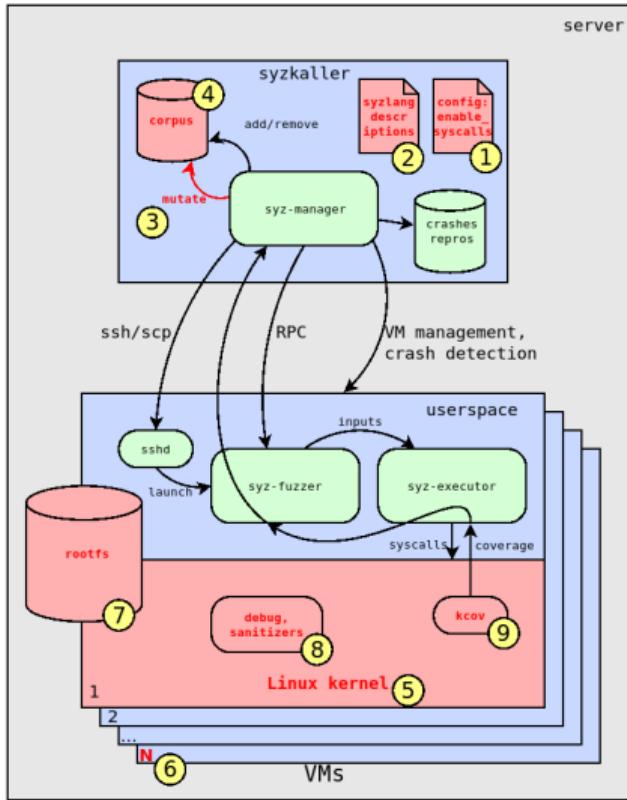


See: <https://xairy.io/articles/syzkaller-external-network>

## 2) More Interested in Bugs with Stable Reproducers



### 3) More Interested in Unique Bugs



#### Ideas on how to make a fuzzer find unique bugs

- ① Limit enabled syscalls to make fuzzing go deeper
- ② Write new syzlang descriptions
- ③ Change the mutation of fuzzing inputs  
(i.e. integrate symbolic execution)
- ④ Start fuzzing from the crafted corpus
- ⑤ Modify the Linux kernel  
(for example, my [CVE-2021-26708](#))
- ⑥ Use more computing power than competitors
- ⑦ Modify the rootfs of fuzzing VMs  
(for example, my [CVE-2017-2636](#))
- ⑧ Improve the Linux kernel bug detectors
- ⑨ Customize kcov or use cover\_filter for directed fuzzing

- It's a **wonderful** research instrument
- It can be used **not only** for vuln discovery  
(for example, it's how I discovered [msg\\_msg heap spraying](#))
- It's an **everyday practice**
- For **unique** findings, your fuzzing setup should be **unique** as well
- You need to be **brave**: you are risking your efforts and computing power \*
- And that's why it's so **exciting** when you eventually find success!



Thank you!  
Questions?



alex.popov@linux.com



a13xp0p0v

positive technologies