



Разработка без опасности или сертификация безопасного



Как изменилась команда лаборатории при работе в парадигме РБПО

- Команда статического анализа (5 человек)
- Команда фаззинг тестирования (2 человека)
- Команда архитектурного анализа (2 человека)





Как изменилась команда лаборатории при работе в парадигме РБПО

3

Размечено работок 1600

Более 40 принятых исправлений



<https://github.com/python/cpython/issues/106831>

https://github.com/HdrHistogram/HdrHistogram_c/issues/118

<https://github.com/nodejs/node/issues/50896>

<https://github.com/php/php-src/issues/12791>

<https://github.com/php/php-src/issues/12962>

<https://listengine.tuxfamily.org/chrony.tuxfamily.org/chrony-dev/2023/10/msg00000.html>

<https://listengine.tuxfamily.org/chrony.tuxfamily.org/chrony-dev/2023/10/msg00001.html>

<https://github.com/OpenPrinting/libcupsfilters/issues/42>

<https://github.com/OpenPrinting/cups-filters/issues/552>

<https://github.com/OpenPrinting/cups-filters/issues/551>

<https://github.com/OpenPrinting/cups-browsed/issues/?>



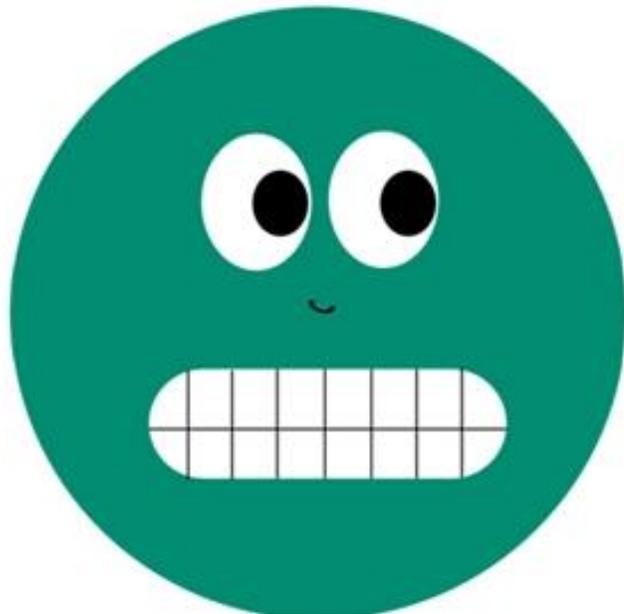


Центр исследования безопасности системного программного обеспечения

- Входящие
- Отправленные
- Черновики

а.kuznetcov@fobos-nt.ru

Входящие 25



Все Непрочитанные

! 📧 📧 📧	ОТ	ТЕМА
	Alexey Khoroshilov Добавлен новый комментарий.	[lvc-svace] REDUNDANT_COMPARISON.ALWAYS_FALSE: src/storage/storage_util.c:3389
	Alexey Khoroshilov Добавлен новый комментарий.	[lvc-svace] REDUNDANT_COMPARISON.ALWAYS_FALSE: src/vbox/vbox_common.c:4817
	Alexey Khoroshilov Добавлен новый комментарий.	[lvc-svace] PASSED_TO_PROC_AFTER_RELEASE: src/qemu/qemu_command.c:9786
	Alexey Khoroshilov Добавлен новый комментарий.	[lvc-svace] PASSED_TO_PROC_AFTER_RELEASE: src/qemu/qemu_command.c:1561
	Alexey Khoroshilov Добавлен новый комментарий.	[lvc-svace] PASSED_TO_PROC_AFTER_RELEASE: src/qemu/qemu_command.c:9787
	Alexey Khoroshilov Добавлен новый комментарий.	[lvc-svace] PASSED_TO_PROC_AFTER_RELEASE: src/qemu/qemu_command.c:8911
	Alexey Khoroshilov Добавлен новый комментарий.	[lvc-svace] PASSED_TO_PROC_AFTER_RELEASE: src/qemu/qemu_command.c:1517
	Alexey Khoroshilov Добавлен новый комментарий.	[lvc-svace] MEMORY_LEAK.EXCEPTION: tools/virt-host-validate-common.c:199
	Alexey Khoroshilov Добавлен новый комментарий.	[lvc-svace] MEMORY_LEAK.EX.EXCEPTION: tools/virsh-completer-domain.c:695
	Alexey Khoroshilov Добавлен новый комментарий.	[lvc-svace] MEMORY_LEAK.EX.EXCEPTION: src/vbox/vbox_snapshot_conf.c:579





Центр исследования безопасности системного программного обеспечения

The image displays multiple overlapping screenshots of the Checkers tool interface. Each screenshot shows a different system component being analyzed, with a list of findings categorized by severity. The findings are color-coded: CRITICAL (red), MAJOR (orange), NORMAL (yellow), and MINOR (green). Each finding includes the number of checkers and markers associated with it.

Visible components and findings include:

- aspnetcore**: # 149 (Custom)
- openssl**: # 25 (Custom)
- podman**: # 600 (Custom)
- libvirt**: # 197 (Custom)
- Chronty**: # 197 (Custom)
- gemu**: # 737 (Custom)
- nginx**: # 737 (Custom)
- clickhouse**: # 293 (Custom)
- Python3**: # 293 (Custom)
- m08-v8.1.0-C6-2024-01**: # 737 (Custom)

Findings shown in the screenshots:

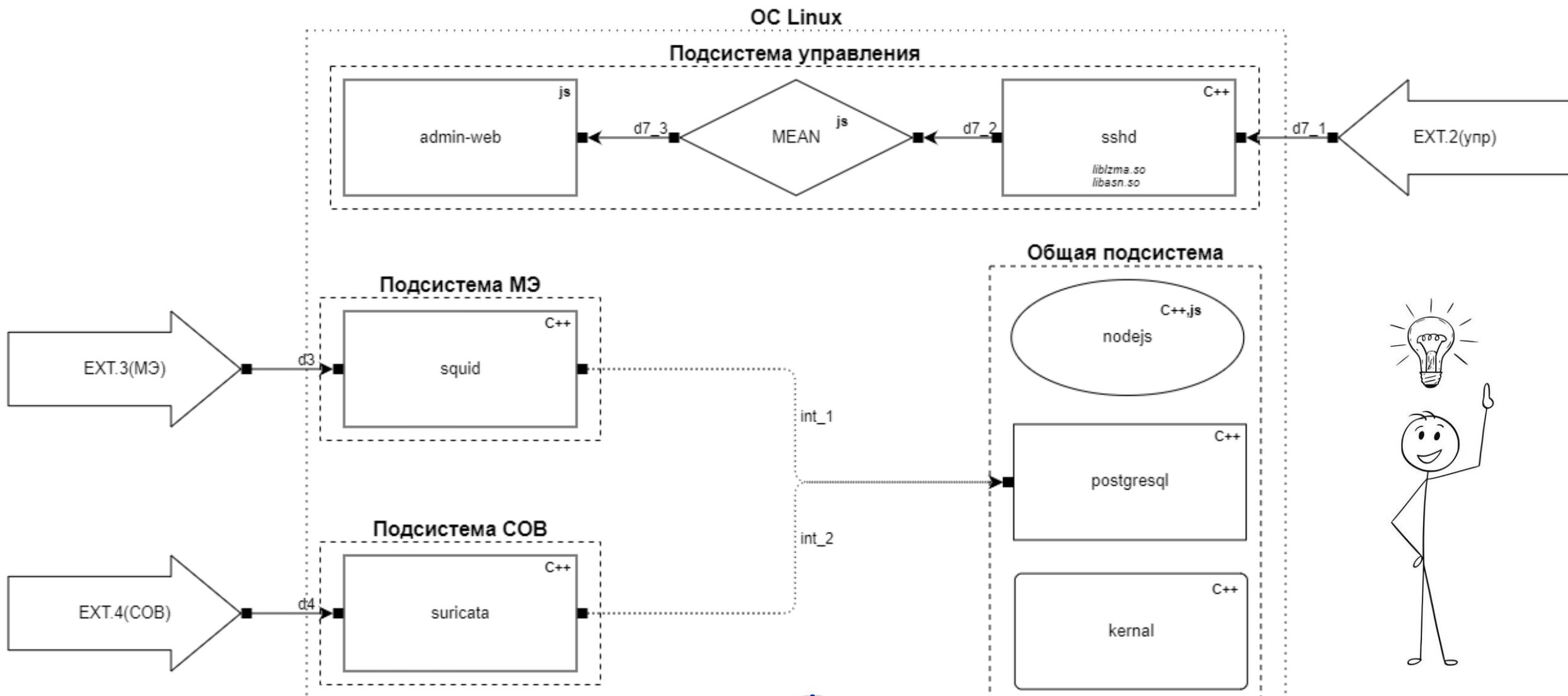
- aspnetcore**: CRITICAL (checkers: 16, markers: 106), MAJOR (checkers: 22, markers: 420), NORMAL (checkers: 17, markers: 301)
- openssl**: CRITICAL (checkers: 2, markers: 7)
- podman**: CRITICAL (checkers: 5, markers: 65), MAJOR (checkers: 5, markers: 65)
- libvirt**: CRITICAL (checkers: 6, markers: 30), MAJOR (checkers: 19, markers: 128), NORMAL (checkers: 6, markers: 67), MINOR (checkers: 7, markers: 68), UNDEFINED (checkers: 0, markers: 0)
- Chronty**: CRITICAL (checkers: 24, markers: 213), MAJOR (checkers: 24, markers: 261), NORMAL (checkers: 9, markers: 238), MINOR (checkers: 2, markers: 25)
- gemu**: CRITICAL (checkers: 8, markers: 35), CRITICAL (checkers: 5, markers: 5)
- nginx**: CRITICAL (checkers: 5, markers: 5)
- clickhouse**: CRITICAL (checkers: 5, markers: 5)
- Python3**: CRITICAL (checkers: 6, markers: 30), MAJOR (checkers: 19, markers: 128), NORMAL (checkers: 6, markers: 67), MINOR (checkers: 7, markers: 68), UNDEFINED (checkers: 0, markers: 0)
- m08-v8.1.0-C6-2024-01**: CRITICAL (checkers: 24, markers: 213), MAJOR (checkers: 24, markers: 261), NORMAL (checkers: 9, markers: 238), MINOR (checkers: 2, markers: 25)

Checker	Severity	Checkers	Markers
1	CRITICAL	2	7
2	CRITICAL	5	65
3	CRITICAL	5	65
4	CRITICAL	5	65
5	CRITICAL	5	65
1	CRITICAL	6	30
2	MAJOR	19	128
3	NORMAL	6	67
4	MINOR	7	68
5	UNDEFINED	0	0
6	UNDEFINED	0	0
7	UNDEFINED	0	0
8	UNDEFINED	0	0



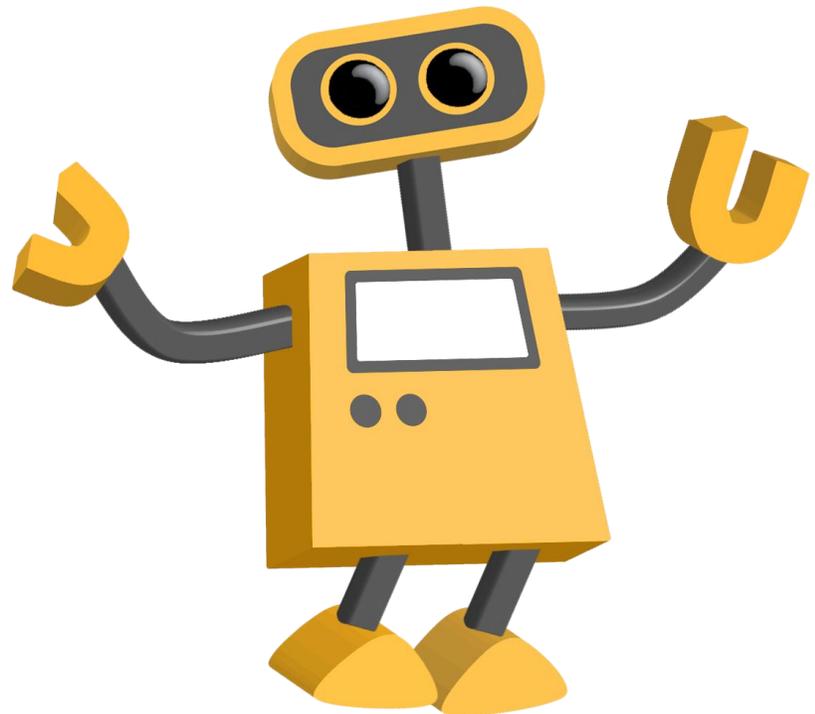
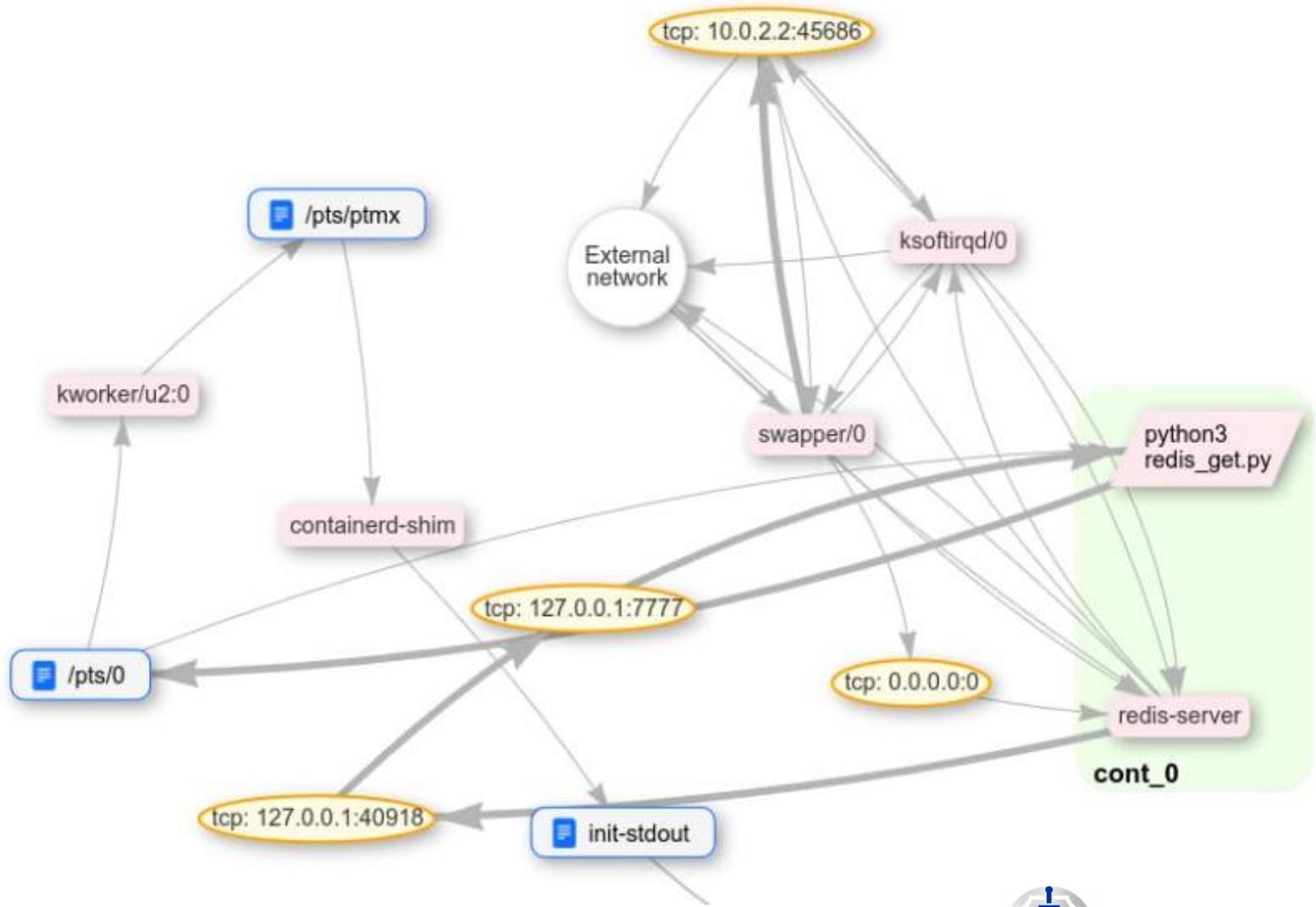


Поверхность атаки





Автоматизация определения поверхности атаки





Применение философии фаззинга для тестирования изоляции Rodman контейнеров (и не только)

Леонид

инженер ООО НТЦ «Фобос-НТ»

студент МГТУ Баумана





Стратегия фаззинг-тестирования Podman контейнеров

The screenshot shows the GitHub repository page for 'trinity-docker' by user 'ewindisch'. The repository is public and has 2 watchers, 4 forks, and 8 stars. The main branch is 'master'. The repository contains a Dockerfile, LICENSE, and README.md. The README.md file is selected and shows the following content:

trinity-docker

Dockerfile for Trinity kernel/syscall fuzzer

This image when run (run as non-root using '-u'), will fuzz the kernel and, should there be a container-breakout vulnerability, may possibly exploit it.

Using this tool, I've personally discovered at least one breakout. One user even told me they managed to crash the Qemu process that their Docker host was running inside.

Authors

Eric Windisch ewindisch@docker.com

The right sidebar shows repository statistics: About (Dockerfile for Trinity kernel/syscall fuzzer), Readme, MIT license, Activity, 8 stars, 2 watching, 4 forks, Report repository, Releases (No releases published), Packages (No packages published), and Languages (Shell 100.0%).

<https://github.com/ewindisch/trinity-docker>





Критические состояния

Нарушение ограничений по потреблению ресурсов (cgroups)

- Потребление памяти больше лимита (> 110%)
- Использование ЦП больше лимита (> 110%)
- Количество дочерних процессов превосходит установленное значение

Побег из пространства имён

- Запущенный контейнер не должен порождать процесс в родительском пространстве имён

Чтение или запись в запрещённые места на хосте

- Запущенный контейнер не должен читать/записывать в недоступные места хостовой системы

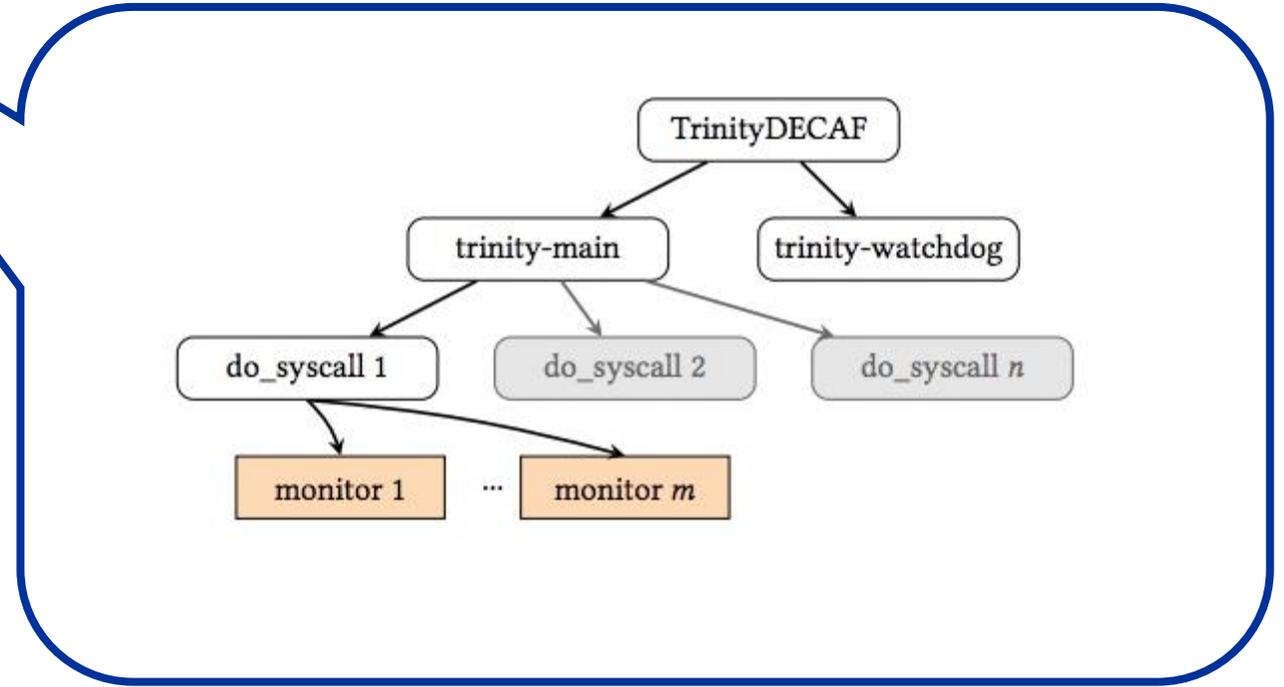
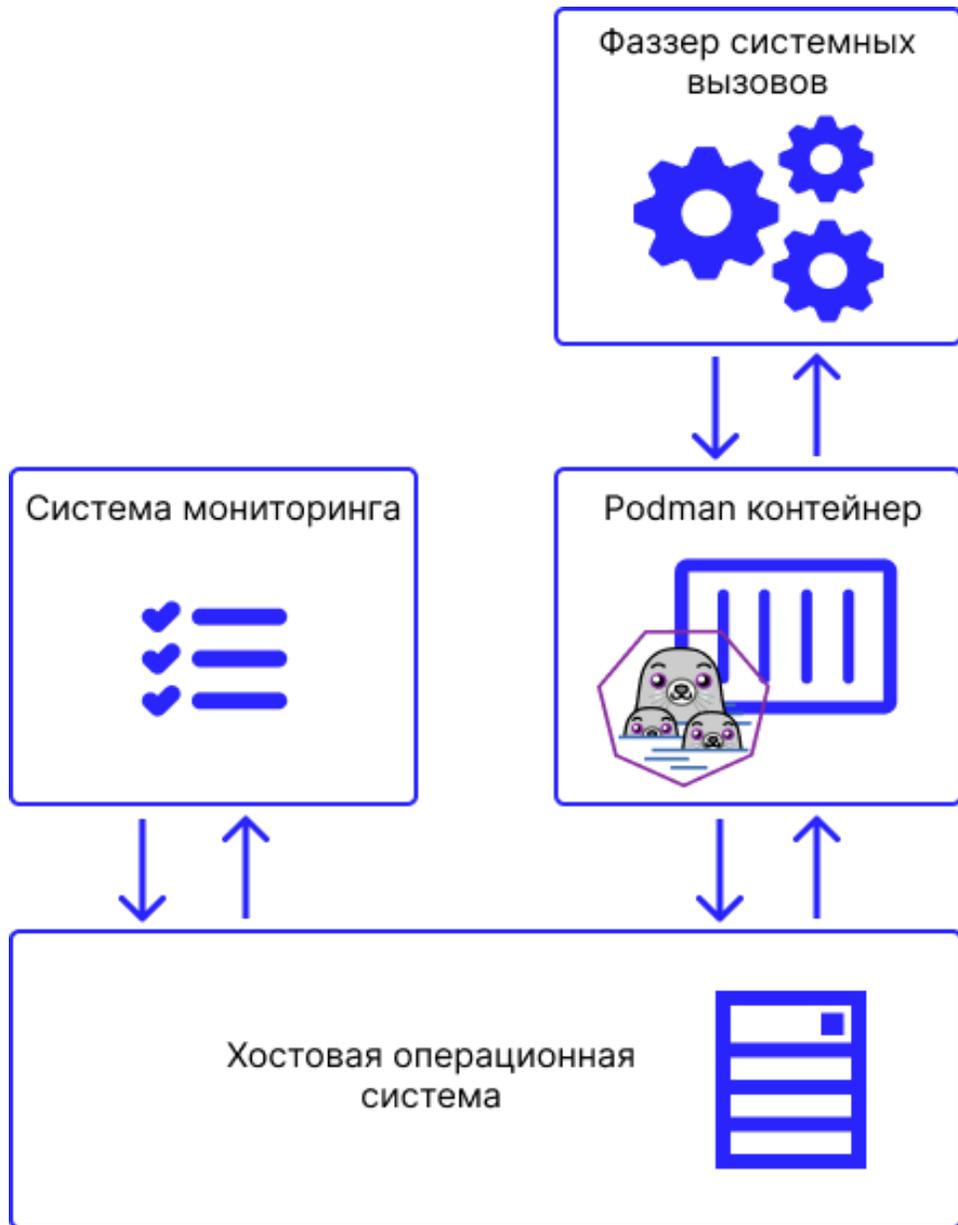
Приобретение новых прав

- Контейнер не должен повышать свои привилегии (capabilities)





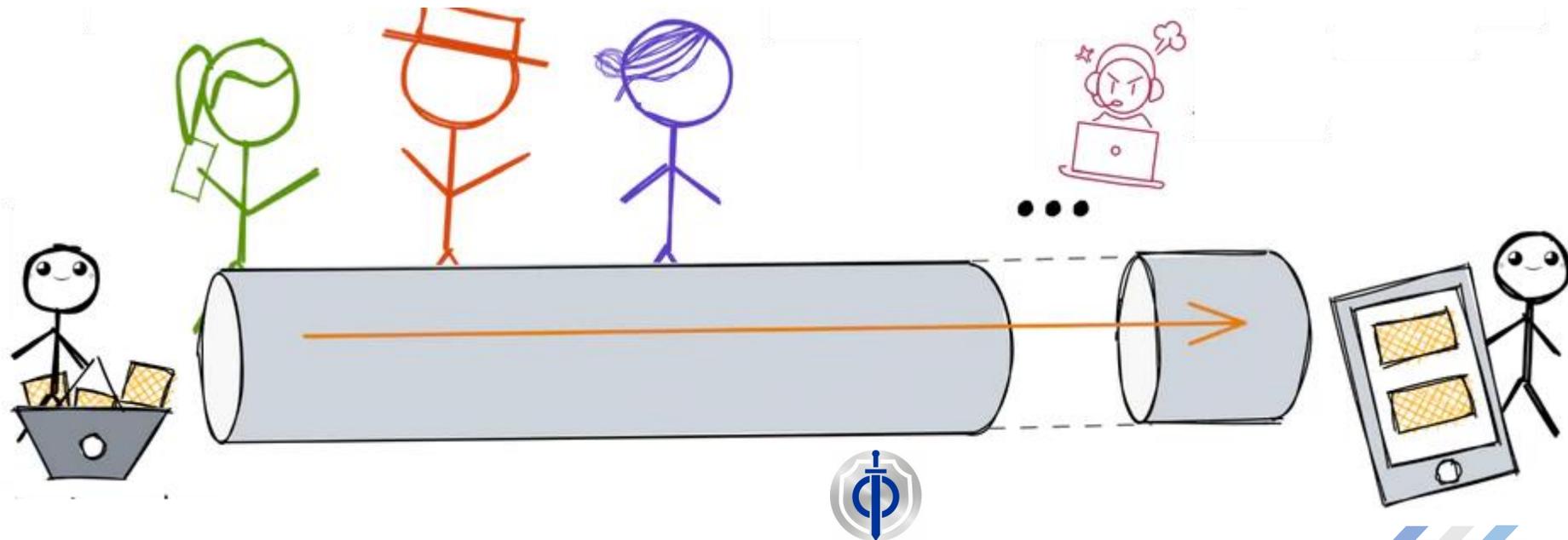
Стратегия фаззинг-тестирования Podman контейнеров





Аудит - сторонний взгляд на процессы разработки

- Подготовка к аудиту
- Сбор данных и информации
- Анализ данных и информации
- Формулирование выводов и рекомендаций





Аудит - двигатель безопасной разработки

Елена

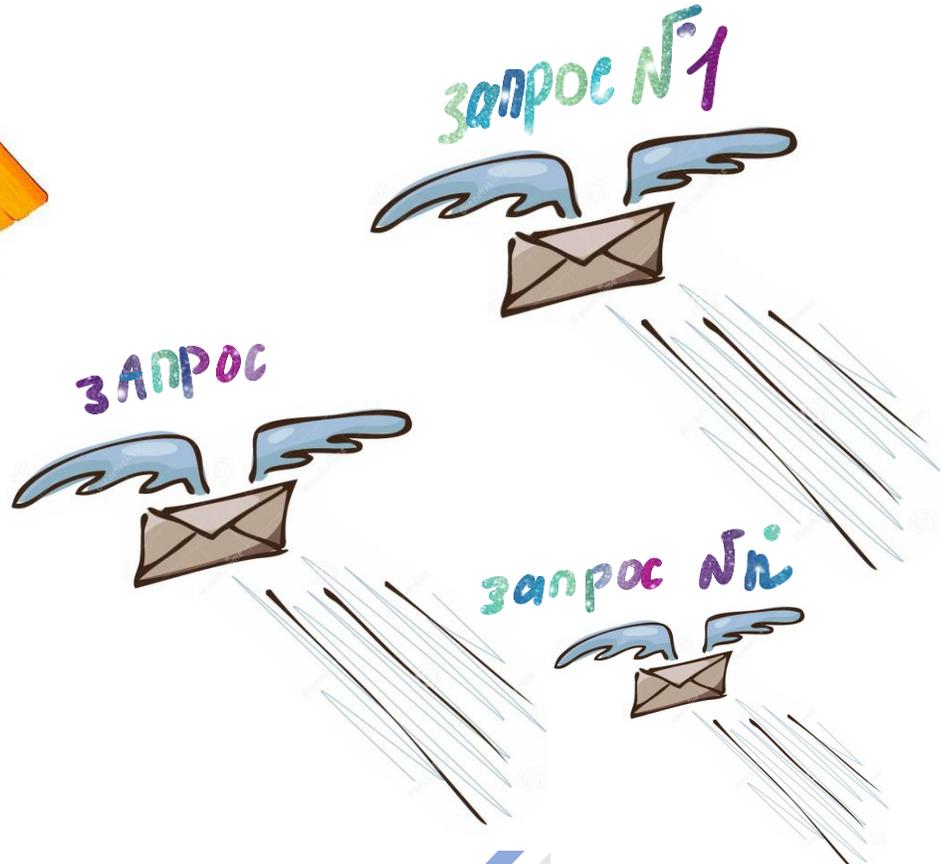
инженер ООО НТЦ «Фобос-НТ»

студент МГТУ Баумана



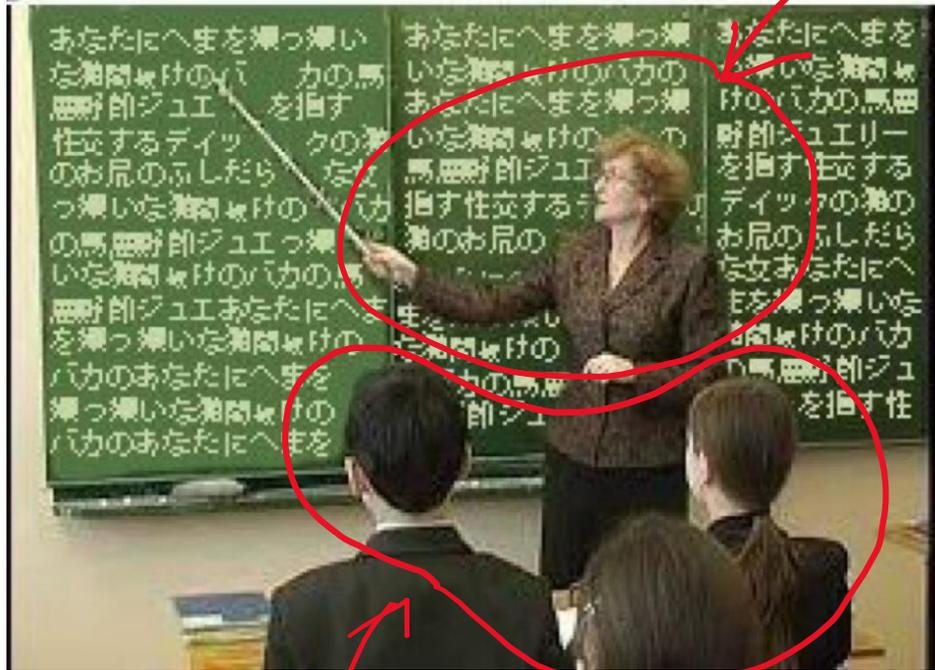


Елена





ЭТО МЫ ВАМ
ПОМОГАЕМ



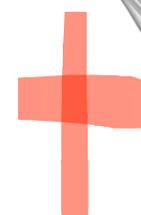
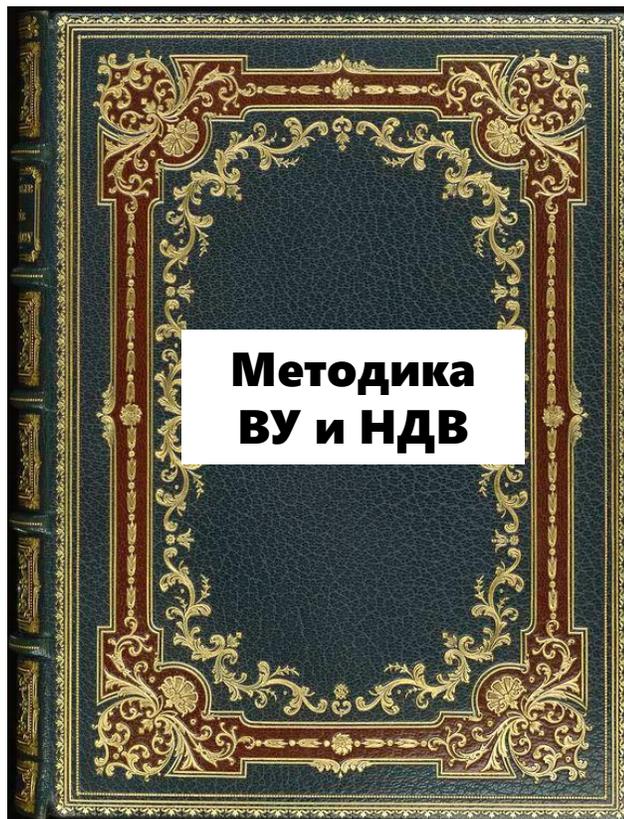
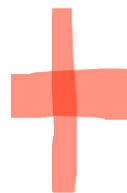
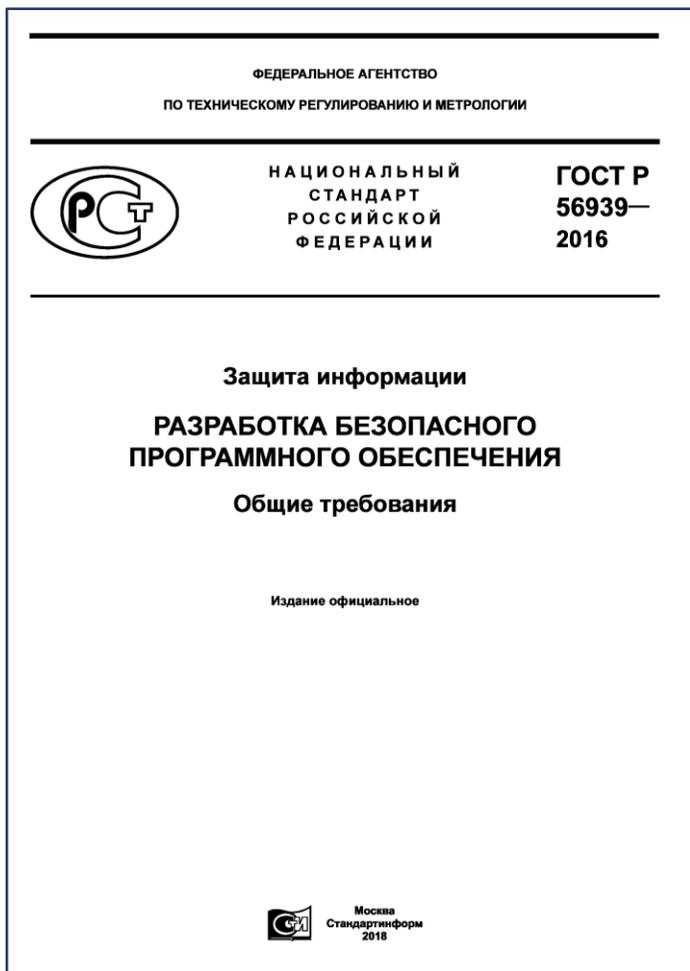
ЭТО ВЫ НАМ
С ИНТЕРЕСОМ
ВНИМАЕТЕ





История из жизни...







Дмитрий Владимирович
Пономарев ♡



kaspersky





Дмитрий Владимирович Пономарев

старший
аудитор



старший
техпис



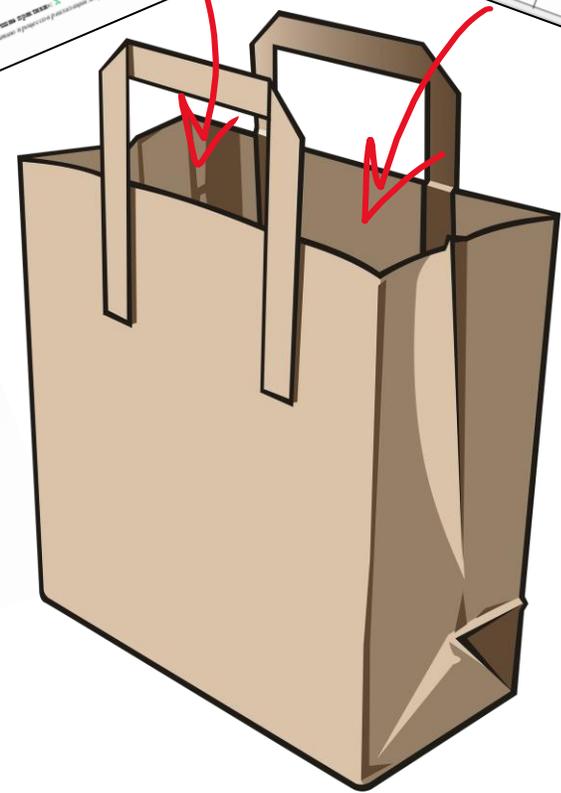
младший
техпис

младший
аудитор





Фобос-НТ
 НАУЧНО-ТЕХНИЧЕСКИЙ ЦЕНТР







Тут нужен
аудит РБПО





“До скорой встречи!” :)

