

7 июня 2023 📍 Москва, МЦК ЗИЛ

# БЕКОН<sup>23</sup>

Первая в России конференция  
по БЕзопасности КОНтейнеров и контейнерных сред

## ОРА с shared Docker executor

Харденим разделяемый docker executor  
в multi-team/multi-tenancy CI/CD

Павел Сорокин





Павел Сорокин

- Telegram: @sorokinpf
- Пентестер → AppSec
- Мои интересы: безопасность CI/CD, K8s, платформы и их мисконфиги



Github

<https://github.com/sorokinpf>



Telegram-канал «Нарыл»

[https://t.me/naryl\\_sec](https://t.me/naryl_sec)

- **Shared Executor в CI/CD: типы и риски**
- **Почему Docker Shared Executor — это плохо**
- **Применение OPA authz plugin для защиты Docker Shared Executor**
- **Разработка правил на rego**

# 1

## Shared executor в CI/CD: типы и риски



- В CI/CD встречаются Shared Executor (runner)
- Типы Executor

Тип	Как работает
Shell	Все джобы запускаются в рамках ОС от имени одного пользователя
Docker	Каждая джоба запускается в контейнере. Чтобы работать с докером в джобу пробрасывается docker API socket
Kubernetes	Каждая джоба запускается в своем поде

# Shared Executor — угрозы

Как одна команда может повлиять на джобы других команд?

Доступ к коду других команд

Доступ к артефактам  
других команд, возможность  
их перезаписать

Доступ к секретам,  
используемым в джобах  
других команд

Повышение привилегий  
на Executor



Тип	Чтение кода	Доступ к артефактам	Доступ к секретам	Повышение привилегий на Executor
Shell	Возможно напрямую	Возможен напрямую	Возможен	В общем-то не требуется

Тип	Чтение кода	Доступ к артефактам	Доступ к секретам	Повышение привилегий на Executor
Shell	Возможно напрямую	Возможен напрямую	Возможен	В общем-то не требуется
Docker	Возможно через docker API socket	Возможен через docker API socket	Возможен через docker API socket	Возможен docker run --privileged ....



Тип	Чтение кода	Доступ к артефактам	Доступ к секретам	Повышение привилегий на Executor
Shell	Возможно напрямую	Возможен напрямую	Возможен	В общем-то не требуется
Docker	Возможно через docker API socket	Возможен через docker API socket	Возможен через docker API socket	Возможен docker run --privileged ....
Kubernetes (*)	-	-	-	-

# Threat Matrix for CI/CD Pipeline



Initial Access	Execution	Persistence	Privilege Escalation	Defence Evasion	Credential Access	Lateral Movement	Exfiltration	Impact
Supply Chain Compromise on CI/CD	Modify CI/CD Configuration	Compromise CI/CD Server	Get Credential for Deployment (CD) on CI Stage	Add Approver using Admin permission	Dumping Env Variables in CI/CD	Exploitation of Remote Services	Exfiltrate data in Production Environment	Deniel of Services
Valid Account of Git Repository (Personal Token, SSH Key, Login Password, Browser Cookie)	Inject Code to iaC Configuration	Implant CI/CD Runner Images	Privileged Escalation and Compromise other CI/CD Pipeline	Bypass Review	Access to Cloud Metadata	(Monorepo) Get Credential of different folder`s context	Clone Git Repositories	
Valid Account of CI/CD Service (Personal Token, Login Password, Browser Cookie)	Inject Code to Source Code	Modify CI/CD Configuration		Access to Secret Manager from CI/CD kicked by different repository	Read Credentials File	Privileged Escalation and Compromise other CI/CD Pipeline		
Valid Admin account of Server hosting Git Repository	Supply Chain Compromise on CI/CD	Inject Code to iaC Configuration		Modify Caches of CI/CD	Get Credential from CI/CD Admin Console			
	Inject bad Dependency	Inject Code to Source Code		Implant CI/CD Runner Images				
	SSH to CI/CD Pipelines	Inject bad Dependency						
	Modify the Configuration of Production Environment							
	Deploy modified applications or Server Images to production Environment							



## Docker — это:

- инструмент для сборки
- runtime

Вместо docker для сборки OCI-образов лучше использовать [kaniko](#)/[buildah](#)

Но если почему-то все-таки  
нужен *именно docker runtime...*





# Docker Shared Executor

ozon{ech

Но если почему-то все-таки нужен именно `docker runtime`...



Но если почему-то все-таки нужен именно docker runtime...



# 2

OPA Docker Authz plugin

**Github:** <https://github.com/open-policy-agent/opa-docker-authz>

**Плагин позволяет создавать  
разрешающие и запрещающие правила  
на основании запроса, отправленного в docker API socket**



## OPA Docker Authz plugin:

<https://github.com/open-policy-agent/opa-docker-authz>

**Плагин позволяет создавать  
разрешающие и запрещающие правила  
на основании запроса, отправленного в docker API socket**

# OPA Docker Authz plugin — OPA

**OPA Docker Authz plugin:**

<https://github.com/open-policy-agent/opa-docker-authz>

**Готовый набор правил...**

# OPA Docker Authz plugin — OPA

**OPA Docker Authz plugin:**

<https://github.com/open-policy-agent/opa-docker-authz>

**Готовый набор правил...**



3

Создаем правила



# ОРА. Что хотим запретить?

- Возможность влиять на сам докер-демон
- На всякий случай на swarm
- Возможность создания привилегированных (\*) контейнеров и, соответственно, выхода из докера

- Возможность монтирования файлов/директорий с хостовой ОС
- Возможность читать данные в другом контейнере
- Возможность внедряться в чужой контейнер

## Плагин получает:

- Method
- Path
- Query
- Headers
- Body

Ну то есть распаршенный  
HTTP-запрос

```
"Body": {
  "HostConfig": {
    ...
    "Privileged": true,
    ...
  },
  "Hostname": "",
  "Image": "hello-world",
  ...
},
"Headers": {
  "Content-Length": "1552",
  "Content-Type": "application/json",
  "User-Agent": "Docker-Client/20.10.12 (linux)"
},
"Method": "POST",
"Path": "/v1.41/containers/create",
```



# ОРА — читаем документацию docker

ozon.tech





- **docker plugin** (можно отключить плагин)
- **docker swarm** (работа в режиме swarm)
- **docker volumes** (возможность создания volumes, в том числе локальных директорий)



- **--privileged** (привилегированный запуск)
- **--cap-add** (запуск с капабилити)
- **--ipc** (IPC namespace)
- **--pid** (PID namespace)
- **--network** (Network namespace)
- **-v** (монтирование volume)
- **--cgroup-parent**
- **--device** (подключение device)
- **--security-opt apparmor/seccomp** (отключение apparmor и seccomp)

# ОРА — ограничения на работу с другими контейнерами

- **docker exec**

- **docker update**

- **docker cp**

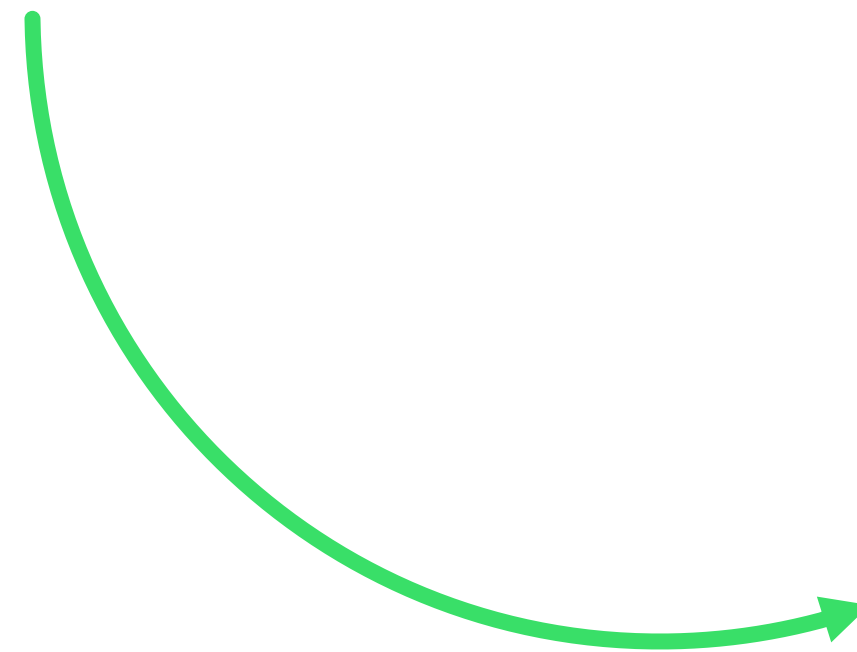
- **docker attach/logs**

- **docker stop/kill/  
pause/restart**

- **docker commit/  
checkpoint**

**Создан набор правил**

[https://github.com/sorokinpf/docker\\_opa](https://github.com/sorokinpf/docker_opa)



# Пример Rego-правила

```
1  package docker.authz
2
3  allow {
4      not devices
5  }
6
7  devices { #only null and empty arrays are allowed
8      input.Body.HostConfig.Devices != null
9      not devices_array
10 }
11
12 devices_array {
13     is_array(input.Body.HostConfig.Devices)
14     count(input.Body.HostConfig.Devices) == 0
15 }
```



- **-p — порты на хосте**

- **docker stop**

- **docker attach/logs**



**Можно повлиять на доступность,  
но не на конфиденциальность**

# OPA — админский пароль для байпаса

```
} {  
  input.Headers["Opa-Bypass"] == "your_secret_password_here"  
}
```

~/.docker/config.json:



```
{ ... ,  
  "HttpHeaders": {  
    "Opa-Bypass": "your_secret_password_here"  
  },  
  ...  
}
```

- Docker executor — это плохо, но можно захарденить
- Ставим docker authz
- И используем готовый список правил по QR-коду





***Ваши  
вопросы***





7 июня 2023 📍 Москва, МЦК ЗИЛ

Первая в России конференция  
по БЕзопасности КОНтейнеров и контейнерных сред



Contacts:

Email: [pavsorokin@ozon.ru](mailto:pavsorokin@ozon.ru)

Tg: [@sorokinpf](https://t.me/sorokinpf)

Site: <https://github.com/sorokinpf>