



DockerCleaner: Automatic Repair of Security Smells in Dockerfiles

Quang-Cuong Bui, Malte Laukötter, Riccardo Scandariato

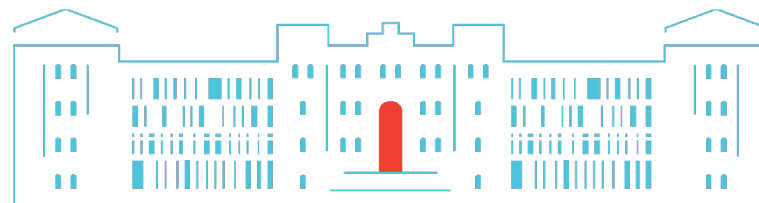
Institute of Software Security

Hamburg University of Technology, Germany



ICSME'23 - 5th Oct, 2023

Bogotá, Colombia





Introduction



- Security smells and vulnerabilities not only exist in program source code
 - But also in **Infrastructure configuration code**^{1 2}

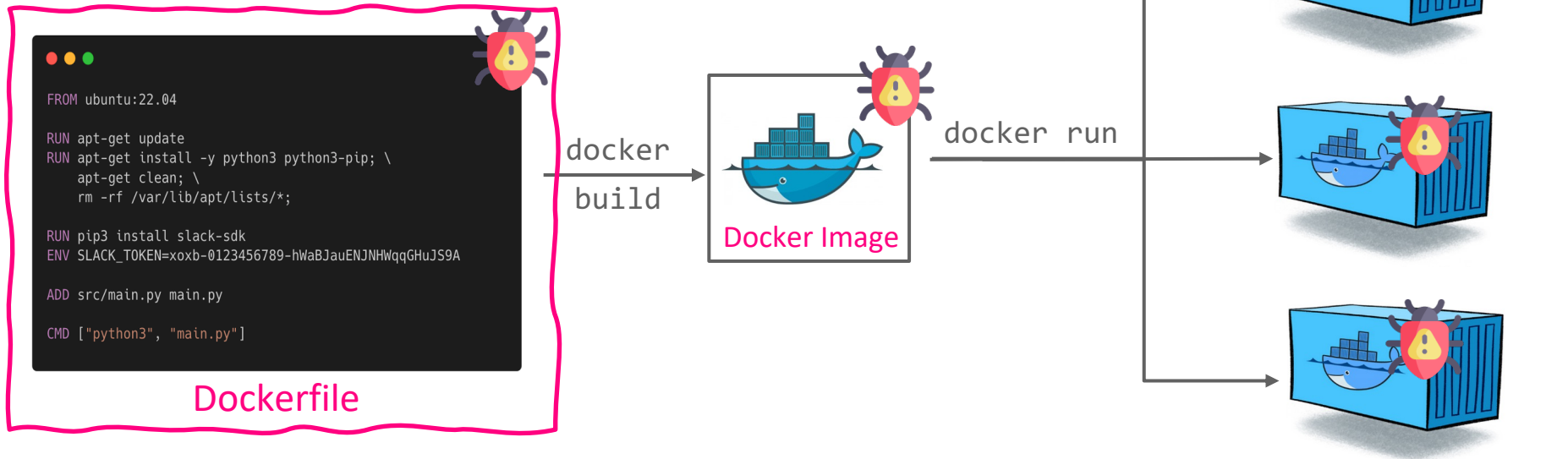


Docker is the **most loved and wanted** tool by professional developers!

¹ Rahman et al. *The Seven Sins: Security Smells in Infrastructure as Code Scripts*. ICSE'19.

² Shu et al. *A Study of Security Vulnerabilities on Docker Hub*. CODASPY'17.

How Docker works



--> We focus on **Security Smells** in Dockerfiles!

Docker Containers

Security tools for Docker

- Many techniques and tools are devoted to detecting and localizing security smells/vulnerabilities, yet few are designed for repairing them



Hadolint



Docker Bench for Security

Binnacle¹aqua
trivy

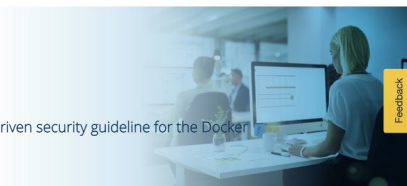
¹ Henkel et al. Learning from, Understanding, and Supporting DevOps Artifacts for Docker. ICSE'20.

Best security practices for Docker

CIS Benchmarks™

Securing Docker

An objective, consensus-driven security guideline for the Docker Server Software.



CIS Docker Benchmark

117 guidelines, latest version: v1.6.0



OWASP Docker Security Cheat Sheet

12 rules to avoid common security mistakes

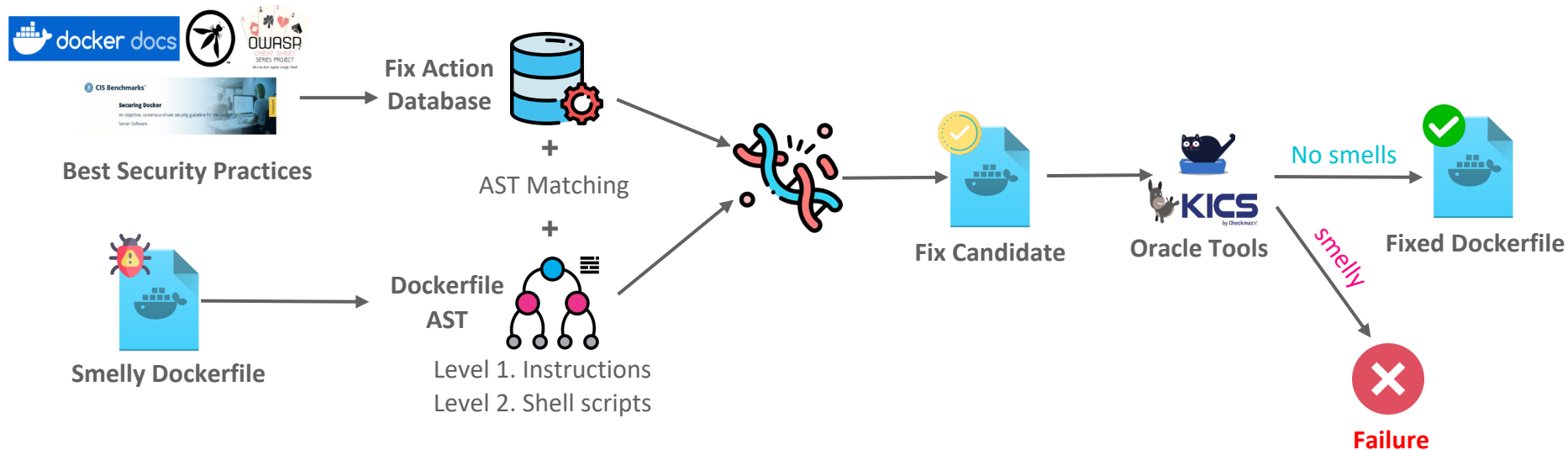


Best practices for writing Dockerfiles

Several practices are security-related



Our idea pipeline






Selecting security smell types for repair



Covered in best security practices



Reported as prevalent in recent studies



Can be detected by stoa scanning tools

No	Chosen Smell Types
1-5	No Version Pinning (apt, apk, npm, pip, gem)
6	Do not use --no-install-recommends
7	Do not use apt-get update alone
8	Use ADD instead of COPY wget
9	Have secrets
10	Do not define HEALTHCHECK
11	Do not run with a <i>non-root</i> USER



Fix Actions

FA = Fully Automated

No	Smell types	Fix strategies	FA
1-5	No Version Pinning (apt, apk, npm, pip, gem)	Query the versions in <i>version sources</i> and pin them to the packages	✓
6	Do not use --no-install-recommends	Add --no-install-recommends flag	✓ ●
7	Do not use apt-get update alone	Add to the starting of every RUN instruction with apt-get install command inside	✓
8	Use ADD instead of COPY wget	- Replace ADD with COPY for <i>a file</i> - Replace ADD with wget for <i>an URL</i>	✓
9	Have secrets	Replace ENV with ARG instruction	●
10	Do not define HEALTHCHECK	Define a HEALTHCHECK for web-based apps	✓ ●
11	Do not run with a <i>non-root</i> USER	Define and run with a <i>non-root</i> USER	✓



Fix Action: Version Pinning

```

RUN pip install urllib3
RUN npm install pm2
RUN gem install draper
    
```

pkg_names =
[urllib3, pm2, draper]



latest versions

urllib3	2.0.6
pm2	5.3.0
draper	2.1.0

```

FROM test:v1.0
RUN apt-get install -y python3
    
```

pkg_name = python3

1.2M+

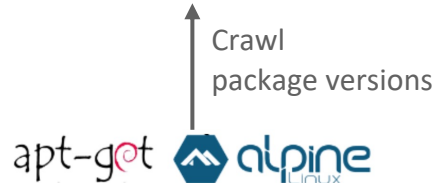


Local Database

python3	3.10.6-1~22.04
---------	----------------

os = ubuntu:22.04

base_img = test:v1.0






SOURCES OF PACKAGE VERSIONS FOR PACKAGE MANAGERS.


Package Manager	Version Sources
apt	http://archive.ubuntu.com, https://archive.debian.org
apk	https://dl-cdn.alpinelinux.org
pip	https://pypi.org
npm	https://registry.npmjs.org
gem	https://rubygems.org


An example of Dockerfile with smells



```
FROM ubuntu:22.04
```

```
RUN apt-get update 
```

```
RUN apt-get install -y python3 python3-pip; \    
apt-get clean; \  
rm -rf /var/lib/apt/lists/*;
```

```
RUN pip3 install slack-sdk 
```

```
ENV SLACK_TOKEN=xoxb-0123-QcXp9MpFwFL9FYJChR8cxeQg 
```

```
   
ADD  src/main.py main.py
```

```

```

```

```

```
CMD ["python3", "main.py"]
```

```
FROM ubuntu:22.04
```

```
RUN apt-get update; \  
apt-get install -y --no-install-recommends \  
python3=3.10.6-1~22.04 python3-pip=22.0.2; \  
apt-get clean; \  
rm -rf /var/lib/apt/lists/*;
```

```
RUN pip3 install slack-sdk=3.23.0
```

```
ARG SLACK_TOKEN # pass during Dockerfile build
```








```
COPY src/main.py main.py
```

```
USER uibahy2i
```

```
# write your HEALTHCHECK instruction here
```

```
CMD ["python3", "main.py"]
```

Detecting smells with oracle tools

No	Smell Types	Tools
1-5	No Version Pinning (apt, apk, npm, pip, gem)	
6	Do not use --no-install-recommends	
7	Do not use apt-get update alone	
8	Use ADD instead of COPY wget	
9	Have secrets	
10	Do not define HEALTHCHECK	
11	Do not run with a <i>non-root</i> USER	

Seven issues were reported to Kics team and resolved during this study!

--> Kics' latest version (incl. fixes) is used to report detection results in the paper!

```
ENV PACKAGES_COMMON="python=2.7"
RUN apt-get -y update && \
    apt-get install -yq --no-install-recommends \
        $PACKAGES_COMMON && \
# Hadolint does not support variable interpolation
```

hadolint/hadolint#329: **False Positive**

```
FROM debian:stretch
CMD ["whoami"]
```

hadolint/hadolint#328: **False Negative**

Research questions & Evaluation metric

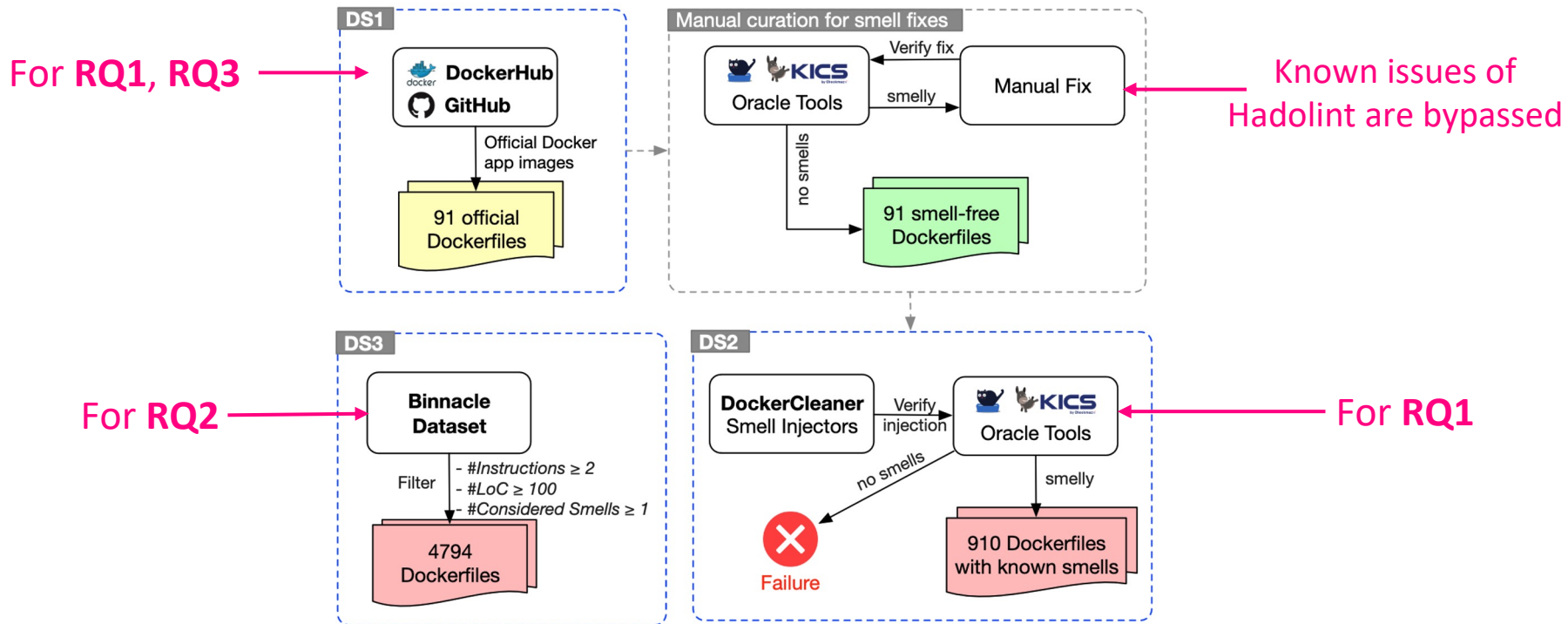
- **RQ1.** How effective is DockerCleaner in repairing known security smells?
- **RQ2.** How effective is DockerCleaner, compared to the state-of-the-art, in repairing security smells in real-world, large-scale Dockerfiles?
- **RQ3.** Do developers of the official Docker images acknowledge security smells and accept repairs suggested by DockerCleaner?

Evaluation Metric:

$$\text{RepairEffectiveness}(S) = \frac{D_{before}^S - D_{after}^S}{D_{before}^S}$$

D_{before}^S	#Dockerfiles containing smell S before repair
D_{after}^S	#Dockerfiles containing smell S after repair

Datasets



RQ1. Performance in repairing injected smells

Private version source repository:
e.g., <https://repo.mongodb.org/apt/>

THE NUMBER OF SMELLY DOCKERFILES DETECTED BEFORE AND AFTER THE REPAIR OF
THE ORIGINAL (DS1) AND INJECTED (DS2) DOCKERFILE DATASETS.

“–” denotes that the smell type does not exist in the mentioned dataset. RE = RepairEffectiveness.

Smell Type	#Smelly Dockerfiles									
	Injected dataset (DS2)			Original dataset (DS1)						
	<i>Before repair</i>	<i>After repair</i>	<i>RE</i>	<i>Before repair</i>	<i>After repair</i>	<i>RE</i>				
noVersionPinningAptGet	233	100.00%	15	6.44%	93.56%	48	52.75%	12	13.19%	75.00%
noVersionPinningApk	151	100.00%	7	4.64%	95.36%	32	35.16%	13	14.29%	59.38%
noVersionPinningPip	13	100.00%	0	0.00%	100.00%	1	1.10%	0	0.00%	100.00%
noVersionPinningNpm	11	100.00%	0	0.00%	100.00%	1	1.10%	0	0.00%	100.00%
noVersionPinningGem	5	100.00%	0	0.00%	100.00%	–	–	–	–	–
noAptGetInstallRec	224	100.00%	0	0.00%	100.00%	17	18.68%	0	0.00%	100.00%
useAptGetUpdateAlone	248	100.00%	0	0.00%	100.00%	–	–	–	–	–
addInsteadOf{Wget, Copy}	357	100.00%	0	0.00%	100.00%	–	–	–	–	–
lastUserIsRoot	280	100.00%	0	0.00%	100.00%	–	–	–	–	–
haveSecrets	450	100.00%	0	0.00%	100.00%	–	–	–	–	–
haveNoHealthcheck	441	100.00%	339	76.87%	23.13%	89	97.80%	69	75.82%	22.47%

Average RepairEffectiveness: 92.67%

76.14%

RQ1. Performance in repairing injected smells

THE BUILD ERROR RATE AFTER APPLYING SMELLS INJECTION AND REPAIR ACTIONS IN DS1, DS2, AND THE SMELL-FREE DATASET.

*The size of the smell-free dataset was multiplied up to 910 Dockerfiles before we injected the smells.

Action: Input dataset → Output dataset	#Buildable		Build
	<i>Before</i>	<i>After</i>	degradation
Smell Injection: Smell-free* → DS2	910	890	2.20%
Smell Repair: DS2 → Fixed	890	884	0.67%
Smell Repair: DS1 → Fixed	91	86	5.49%

RQ2. Performance in repairing real-world smells

THE NUMBER OF SMELLY DOCKERFILES DETECTED BEFORE AND AFTER THE REPAIR OF THE EXTENDED DATASET OF 4794 DOCKERFILES (DS3).

“—” denotes that the mentioned tool does not support the repair of the smell type. RE = RepairEffectiveness.

Smell Type	#Smelly Dockerfiles							
	Before repair		After repair					
			DOCKERCLEANER	RE	PARFUM	RE		
noVersionPinningAptGet	3259	67.98%	2702	56.36%	17.09%	—	—	—
noVersionPinningApk	873	18.21%	696	14.52%	20.27%	—	—	—
noVersionPinningPip	1112	23.20%	435	9.07%	60.88%	—	—	—
noVersionPinningNpm	258	5.38%	19	0.40%	92.64%	—	—	—
noVersionPinningGem	231	4.82%	1	0.02%	99.57%	—	—	—
noAptGetInstallRec	2127	44.37%	193	4.03%	90.93%	73	1.52%	96.57%
useAptGetUpdateAlone	855	17.83%	470	9.80%	45.03%	822	17.15%	3.86%
addInsteadOf{Wget, Copy}	808	16.85%	0	0.00%	100.00%	122	2.54%	84.90%
lastUserIsRoot	275	5.74%	13	0.27%	95.27%	271	5.65%	1.45%
haveSecrets	128	2.67%	32	0.67%	75.00%	—	—	—
haveNoHealthcheck	4746	99.00%	4378	91.32%	7.75%	—	—	—

(All smells) Avg RepairEffectiveness: 64.04% --

(Common smells) Avg RepairEffectiveness: 82.81% 46.70%

--> DockerCleaner outperforms Parfum by **36.11%** in terms of RepairEffectiveness

RQ2. Repair failure causes

- Version Pinning:
 - Base OS image not found or not supported
 - Private version repositories
 - Use multiple version repositories at the same time
 - Use alias name for installing package, e.g., **man** instead of **man-db**
 - Use shell variable for storing package name list
- Too complex shell scripts
- HEALTHCHECK creation requires domain knowledge

RQ3. Developer attitudes toward suggested repairs

THE PULL REQUESTS SUBMITTED TO THE PROJECTS OF DOCKER OFFICIAL IMAGES. BASED ON THE PROPOSED FIXES BY DOCKERCLEANER IN DS1.

PC = Pull Count, #FS = #Fixed Smells, #FD = #Fixed Dockerfiles, Human interv. = Human intervention was needed.

Official image	PC	#FS	#FD	Human interv.	Status
backdrop	6.8M+	1	1		Merged
couchbase	83M+	1	1		Accepted
couchdb	179M+	2	1		Open
hitch	363K+	2	1		Merged
kong	308M+	2	1	Add ca-certificates	Open
mysql	3.6B+	1	1		Open
php-zendserver	4.1M+	2	1	Add ca-certificates,patch	Merged
rethinkdb	73M+	1	1		Merged
silverpeas	1.7M+	2	1		Merged
solr	139M+	2	2		Merged
tomee	21M+	43	43	Add dirmngr	Merged
varnish	13M+	6	3		Merged
Total (12 pull requests)		65	57	8 Merged, 1 Accepted, 3 Open	

- *“That looks great, thanks for the improvement!”*
- *“I was expecting that one :-D Thanks for your work.”*
- ***“Interesting... I hadn’t heard about this ... Thanks for the contribution”***
- *“Thank you for the PR ... I am fine with the --no-install-recommends as it makes sense to me.”*

Merged by yesterday!

<https://github.com/Kong/docker-kong/pull/644>



Questions ?

Thank you for listening!

