This issue has been migrated to GitHub:
https://github.com/python/cpython/issues/83784

classification

Title: [security][CVE-2020-26116] http.client: HTTP Header Injection in the HTTP method	
Type: security	**Stage**: resolved
Components: Library (Lib), SSL	**Versions**: Python 3.10, Python 3.9, Python 3.8, Python 3.7, Python 3.6, Python 3.5

process

Status: closed	**Resolution**: fixed
Dependencies:	**Superseder**:
Assigned To: christian.heimes	**Nosy List**: Amir, M W2, christian.heimes, gvanrossum, kmaork, larry, lukasz.langa, maxpl0it, mcascella, miss-islington, ned.deily, orsenthil, vstinner, xtreak
Priority: normal	**Keywords**: patch

Created on **2020-02-10 19:29** by **maxpl0it**, last changed **2022-04-11 14:59** by **admin**. This issue is now **closed**.

Pull Requests

URL	Status	Linked	Edit
PR 18480	closed	Amir, 2020-02-12 11:05	
PR 18485	merged	Amir, 2020-02-12 13:54	
PR 21536	merged	miss-islington, 2020-07-18 20:16	
PR 21537	merged	miss-islington, 2020-07-18 20:16	
PR 21538	merged	miss-islington, 2020-07-18 20:16	
PR 21539	merged	miss-islington, 2020-07-18 20:17	
PR 21946	merged	vstinner, 2020-08-24 15:52	

Messages (21)

msg361710 - (view) **Author: Max (maxpl0it)** **Date: 2020-02-10 19:29**

```
I recently came across a bug during a pentest that's allowed me to perform some really interesting attacks on a target. While
originally discovered in requests, I had been forwarded to one of the urllib3 developers after agreeing that fixing it at it's lowest
level would be preferable. I was informed that the vulnerability is also present in http.client and that I should report it here as
well.

The 'method' parameter is not filtered to prevent the injection from altering the entire request.

For example:
>>> conn = http.client.HTTPConnection("localhost", 80)
>>> conn.request(method="GET / HTTP/1.1\r\nHost: abc\r\nRemainder:", url="/index.html")

This will result in the following request being generated:
GET / HTTP/1.1
Host: abc
Remainder: /index.html HTTP/1.1
Host: localhost
Accept-Encoding: identity

This was originally found in an HTTP proxy that was utilising Requests. It allowed me to manipulate the original path to access
different files from an internal server since the developers had assumed that the method would filter out non-standard HTTP methods.

The recommended solution is to only allow the standard HTTP methods of GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, TRACE, and
PATCH.

An alternate solution that would allow programmers to use non-standard methods would be to only support characters [a-z] and stop
reading at any special characters (especially newlines and spaces).
```

msg361808 - (view) **Author: STINNER Victor (vstinner) *** **Date: 2020-02-11 12:29**

```
> The recommended solution is to only allow the standard HTTP methods of GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, TRACE, and
PATCH.

I don't think that we have to be so strict. We can maybe restrict the HTTP method to ASCII letters, or just reject control characters
(U+0000-U+001f).

Similar issues (fixed):

* https://python-security.readthedocs.io/vuln/http-header-injection2.html
* https://python-security.readthedocs.io/vuln/http-header-injection.html
```

msg361818 - (view) **Author: Max (maxpl0it)** **Date: 2020-02-11 14:13**

```
I agree that the solution is quite restrictive.
Restricting to ASCII characters alone would certainly work.
```

msg361828 - (view) Author: Amir Mohamadi (Amir) * Date: 2020-02-11 18:34

can I work on it?!

msg361865 - (view) Author: Amir Mohamadi (Amir) * Date: 2020-02-12 07:09

@vstinner sorry to bother you, I have a quick question.

the request(...) method is like this:

```
def request(self, method, url, body=None, headers={}, *,
            encode_chunked=False):
    """Send a complete request to the server."""
    self._send_request(method, url, body, headers, encode_chunked)
```

'request' calls '_send_request' method and '_send_request' calls 'putrequest' inside itself.

So is it good if I encode 'method' parameter to ASCII inside 'putrequest'??!

msg361896 - (view) Author: Senthil Kumaran (orsenthil) * Date: 2020-02-12 14:35

Welcome to work on the patch, Amir.

* We shouldn't be encoding anything.
* Create reject for Unicode control characters and reject the request if the request contains any control character. Write tests for this.

It will similar to one of the examples Victor has shared.

msg362239 - (view) Author: Maor Kleinberger (kmaork) * Date: 2020-02-18 23:17

Hey, it's been a week since the last activity here...
Amir, if you are not working on it I'd be glad to work on it as well :)

msg373915 - (view) Author: miss-islington (miss-islington) Date: 2020-07-18 20:16

New changeset **8ca8a2e8fb068863c1138f07e3098478ef8be12e** by AMIR in branch 'master':
~~**bpo-39603**~~: Prevent header injection in http methods (~~**GH-10405**~~)
https://github.com/python/cpython/commit/8ca8a2e8fb068863c1138f07e3098478ef8be12e

msg373916 - (view) Author: Guido van Rossum (gvanrossum) * Date: 2020-07-18 20:23

The 3.9 and 3.8 backports are waiting for tests to complete. The 3.7 and 3.6 backports need to be merged by the RM (Ned). Then someone can close this issue.

msg373917 - (view) Author: miss-islington (miss-islington) Date: 2020-07-18 20:39

New changeset **668d321476d974c4f51476b33aaca870272523bf** by Miss Islington (bot) in branch '3.8':
~~**bpo-39603**~~: Prevent header injection in http methods (~~**GH-10405**~~)
https://github.com/python/cpython/commit/668d321476d974c4f51476b33aaca870272523bf

msg373918 - (view) Author: miss-islington (miss-islington) Date: 2020-07-18 20:41

New changeset **27b811057ff5e93b68798e278c88358123efdc71** by Miss Islington (bot) in branch '3.9':
~~**bpo-39603**~~: Prevent header injection in http methods (~~**GH-10405**~~)
https://github.com/python/cpython/commit/27b811057ff5e93b68798e278c88358123efdc71

msg373944 - (view) Author: Ned Deily (ned.deily) * Date: 2020-07-19 09:27

New changeset **ca75fec1ed358f7324272608ca952b2d8226d11a** by Miss Islington (bot) in branch '3.7':
~~**bpo-39603**~~: Prevent header injection in http methods (~~**GH-10405**~~) (~~**GH-21530**~~)
https://github.com/python/cpython/commit/ca75fec1ed358f7324272608ca952b2d8226d11a

msg373945 - (view) Author: Ned Deily (ned.deily) * Date: 2020-07-19 09:28

New changeset **f02de961b9f19a5db0ead56305fe0057a78787ae** by Miss Islington (bot) in branch '3.6':
~~**bpo-39603**~~: Prevent header injection in http methods (~~**GH-10405**~~) (~~**GH-21539**~~)
https://github.com/python/cpython/commit/f02de961b9f19a5db0ead56305fe0057a78787ae

msg373946 - (view) Author: Ned Deily (ned.deily) * Date: 2020-07-19 09:32

Merged for release in 3.9.0b5, 3.8.5, 3.7.9, and 3.6.12. Thanks, everyone!

msg374020 - (view) Author: Łukasz Langa (lukasz.langa) * Date: 2020-07-20 17:24

New changeset **580fbb018fd0844806119614d752b41fc69660f9** by Łukasz Langa in branch '3.8':
Python 3.8.5
https://github.com/python/cpython/commit/580fbb018fd0844806119614d752b41fc69660f9

msg374093 - (view) Author: Max (maxpl0it) Date: 2020-07-22 15:33

I've just noticed an issue with the current version of the patch. It should also include 0x20 (space) since that can also be used to manipulate the request.

msg374095 - (view) Author: Guido van Rossum (gvanrossum) * Date: 2020-07-22 16:01

> It should also include 0x20 (space) since that can also be used to manipulate the request.

Can you indicate how to use a space in the HTTP verb as part of an attack?

msg376335 - (view) Author: Larry Hastings (larry) * Date: 2020-09-04 00:54

New changeset **524b8de630036a29ca340bc2ae6fd6dc7dda8f40** by Victor Stinner in branch '3.5':
~~**bpo-39603**~~: Prevent header injection in http methods (~~**GH-18405**~~) (~~**#21946**~~)
https://github.com/python/cpython/commit/524b8de630036a29ca340bc2ae6fd6dc7dda8f40

msg377586 - (view) **Author: Mauro Matteo Cascella (mcascella)** **Date: 2020-09-28 07:05**

Hello,

CVE-2020-26116 has been requested/assigned for this flaw via MITRE form: **https://cveform.mitre.org/**

I suggest mentioning it in the related vulnerability page: **https://python-security.readthedocs.io/vuln/http-header-injection-method.html**

Also note that httplib (python-2.7.18) seems to be affected too. Any particular reason for it not to be listed in the same vulnerability page?

Thank you,

msg377607 - (view) **Author: Larry Hastings (larry) *** **Date: 2020-09-28 16:20**

> Also note that httplib (python-2.7.18) seems to be affected too. Any particular reason for it not to be listed in the same vulnerability page?

Yes: 2.7 has been end-of-lifed and is no longer supported.

msg377643 - (view) **Author: STINNER Victor (vstinner) *** **Date: 2020-09-28 22:42**

Mauro Matteo Cascella: "CVE-2020-26116 has been requested/assigned for this flaw via MITRE form: **https://cveform.mitre.org/** I suggest mentioning it in the related vulnerability page: **https://python-security.readthedocs.io/vuln/http-header-injection-method.html**"

Thanks, done.

History

Date	User	Action	Args
2022-04-11 14:59:26	admin	set	github: 83784
2020-09-28 22:42:24	vstinner	set	messages: + **msg377643** title: [security] http.client: HTTP Header Injection in the HTTP method -> [security][CVE-2020-26116] http.client: HTTP Header Injection in the HTTP method
2020-09-28 16:20:00	larry	set	messages: + **msg377607**
2020-09-28 07:05:14	mcascella	set	nosy: + **mcascella** messages: + **msg377586**
2020-09-04 00:54:24	larry	set	nosy: + **larry** messages: + **msg376335**
2020-08-24 15:52:33	vstinner	set	pull_requests: + **pull_request21056**
2020-07-22 16:01:52	gvanrossum	set	messages: + **msg374095**
2020-07-22 15:33:33	maxpl0it	set	messages: + **msg374093**
2020-07-20 17:24:33	lukasz.langa	set	nosy: + **lukasz.langa** messages: + **msg374020**
2020-07-19 09:32:11	ned.deily	set	status: open -> closed versions: + Python 3.10, - Python 2.7 messages: + **msg373946** resolution: fixed stage: patch review -> resolved
2020-07-19 09:28:48	ned.deily	set	messages: + **msg373945**
2020-07-19 09:27:39	ned.deily	set	nosy: + **ned.deily** messages: + **msg373944**
2020-07-18 22:33:16	M W2	set	nosy: + **christian.heimes**, **M W2** components: + SSL assignee: **christian.heimes**
2020-07-18 20:41:59	miss-islington	set	messages: + **msg373918**
2020-07-18 20:39:19	miss-islington	set	messages: + **msg373917**
2020-07-18 20:23:27	gvanrossum	set	nosy: + **gvanrossum** messages: + **msg373916**
2020-07-18 20:17:00	miss-islington	set	pull_requests: + **pull_request20681**
2020-07-18 20:16:51	miss-islington	set	pull_requests: + **pull_request20680**
2020-07-18 20:16:45	miss-islington	set	pull_requests: + **pull_request20679**
2020-07-18 20:16:37	miss-islington	set	pull_requests: + **pull_request20678**
2020-07-18 20:16:18	miss-islington	set	nosy: + **miss-islington** messages: + **msg373915**
2020-02-18 23:17:06	kmaork	set	nosy: + **kmaork** messages: + **msg362239**
2020-02-12 14:35:43	orsenthil	set	messages: + **msg361896**
2020-02-12 13:54:56	Amir	set	pull_requests: + **pull_request17858**
2020-02-12 11:05:08	Amir	set	keywords: + **patch** stage: patch review pull_requests: + **pull_request17850**
2020-02-12 07:09:21	Amir	set	messages: + **msg361865**
2020-02-11 18:34:30	Amir	set	nosy: + **Amir** messages: + **msg361828**
2020-02-11 14:13:39	maxpl0it	set	messages: + **msg361818**
2020-02-11 12:54:59	xtreak	set	nosy: + **xtreak**
2020-02-11 12:29:20	vstinner	set	nosy: + **vstinner**, **orsenthil** messages: + **msg361808**
2020-02-11 12:27:34	vstinner	set	title: Injection in http.client -> [security] http.client: HTTP Header Injection in the HTTP method
2020-02-10 19:29:35	maxpl0it	create	