» Issue43223



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/87389

classification

Title:	[security] http.server: Open Redirection if the URL path starts with //		
Type:	security	**Stage**:	patch review
Components:	Library (Lib)	**Versions**:	Python 3.11, Python 3.10, Python 3.9, Python 3.8, Python 3.7, Python 3.6

process

Status:	open	**Resolution**:	
Dependencies:		**Superseder**:	
Assigned To:		**Nosy List**:	christian.heimes, hamzaavvan, lukasz.langa, ned.deily, paul.moore, steve.dower, tim.golden, vstinner, zach.ware
Priority:	deferred blocker	**Keywords**:	patch

Created on **2021-02-14 11:42** by **hamzaavvan**, last changed **2022-04-11 14:59** by **admin**.

Files

File name	Uploaded	Description	Edit
Capture.PNG	hamzaavvan, 2021-02-14 11:42		

Pull Requests

URL	Status	Linked	Edit
PR 24848	open	hamzaavvan, 2021-03-13 21:13	

Repositories containing patches

https://github.com/hamzaavvan/cpython/tree/fix-issue-43223

Messages (5)

msg386945 - (view) **Author: Hamza Avvan (hamzaavvan) *** **Date: 2021-02-14 11:42**

```
The provided version of python distros 3.8.7 and 3.7.4 are vulnerable to open
redirection while traversing to an existing directory.

# PAYLOAD
http://127.0.0.1:8000//attacker.com/..%2f..%2f..%2f..%2f..%2f../%0a%0d/../.ssh
```

In this case, the actual path of .ssh was:
http://127.0.0.1:8000/.ssh

Upon visiting the payload URI the python server will respond back with a Location header instead of serving the directory contents directly which triggers the redirection to attacker.com

Server: SimpleHTTP/0.6 **Python/3.8.7**

msg387193 - (view) **Author: STINNER Victor (vstinner) *** **Date: 2021-02-17 22:26**

I can only reproduce the issue if the current directory (directory used by the HTTP server, see --directory command line option) contains a .ssh/ subdirectory.

The problem is that the HTTP Header Location starts with "//domain/" and such URL is interpreted as an absolute URL of a new domain name ("domain"), rather than a relative path of the same domain ("localhost").

Maybe we should simply strip all additional leading slashes to only keep one. Replace "//path" or "/////path" with "/path" for example.

By the way, http.server uses urllib.parse.urlsplit() on the request URL without passing its own domain, and urllib.parse.urlsplit() interprets "//attacker.com/path" as if attacker.com is a host with no scheme:

```
>>> urllib.parse.urlsplit('//attacker.com/path')
SplitResult(scheme='', netloc='attacker.com', path='/path', query='', fragment='')
```

Maybe parse_qs() should be used instead? Or we should reinject the server domain and port number? I am not sure that it's an issue in practice.

SimpleHTTPRequestHandler.translate_path('//attacker.com/..%2f..%2f..%2f..%2f..%2f../.ssh') returns os.path.join(self.directory, ".ssh"). I don't think that it's an issue, it sounds like the expected behavior. We don't attempt to reject ".." in URL.

To reproduce the issue, I used two terminals.

Terminal 1:

```
$ python3.8 -V
Python 3.8.7
$ python3.8 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
127.0.0.1 - - [15/Feb/2021 09:18:20] "GET
//attacker.com/..%2f..%2f..%2f..%2f..%2f../.ssh HTTP/1.1" 301 -
```

Terminal 2:

```
$ wget 'http://127.0.0.1:8000//attacker.com/..%2f..%2f..%2f..%2f..%2f../.ssh'
(...)
```

```
HTTP request sent, awaiting response... 301 Moved Permanently
Location: //attacker.com/..%2f..%2f..%2f..%2f..%2f../.ssh/ [following]

--2021-02-15 09:18:20--  http://attacker.com/..%2f..%2f..%2f..%2f..%2f../.ssh/
Resolving attacker.com (attacker.com)... 45.88.202.115
Connecting to attacker.com (attacker.com)|45.88.202.115|:80... connected.

(...)
```

wget is redirected and connects to attacker.com.

The HTTP redirection comes from **Lib/http/server.py**:

```
    def send_head(self):
        path = self.translate_path(self.path)
        f = None
        if os.path.isdir(path):
            parts = urllib.parse.urlsplit(self.path)
            if not parts.path.endswith('/'):
                # redirect browser - doing basically what apache does
                self.send_response(HTTPStatus.MOVED_PERMANENTLY)
                new_parts = (parts[0], parts[1], parts[2] + '/',
                             parts[3], parts[4])
                new_url = urllib.parse.urlunsplit(new_parts)
                self.send_header("Location", new_url)
                self.end_headers()
                return None
            ...
        ...
```

The problem is that the "Location" header starts with "//".

msg387284 - (view) **Author: Hamza Avvan (hamzaavvan) *** **Date: 2021-02-19 05:59**

As for the directory issue, not only .ssh but an attacker can use any directory to make the open redirection exploitable.

And as for the HTTP Header Location, the server does not remove extra trailing slash from the PAYLOAD uri, which seems to be the cause of vulnerability getting exploited.

http://127.0.0.1:8000//attacker.com/..%2f..%2f..%2f..%2f..%2f../%0a%0d/../.ssh

So I believe the server should check for multiple slashes and remove them from the path.

Additionally, as you've mentioned it should also prepend the host:port/ to the new_url variable before writing the HTTP Header Location because if an attacker bypasses the protection and add an extra slash the server will still redirect to the host which is getting inserted into the Location header. But honestly I need your opinion as concatenating host to the url may lead to Host Header Injection but it'll then require a different context.

Please watch the POC video.
POC Video: **https://youtu.be/rLfOoEu1XXg**

msg390047 - (view) **Author: Łukasz Langa (lukasz.langa) *** **Date: 2021-04-02 09:35**

Deferred the blocker to a regular release due to lack of activity in time for the current expedited releases.

msg394193 - (view) **Author: Hamza Avvan (hamzaavvan) *** **Date: 2021-05-23 08:10**

Requested review for the unit test few days ago. Please check.
https://github.com/python/cpython/pull/24848

History

Date	User	Action	Args
2022-04-11 14:59:41	admin	set	github: 87389
2021-05-23 08:10:57	hamzaavvan	set	messages: + **msg394193**
2021-05-21 23:10:46	ned.deily	link	**issue32084 superseder**
2021-05-18 09:24:46	kj	set	type: crash -> security title: [security] http.server: Open Redirection if the URL path starts with //racha999coperation -> [security] http.server: Open Redirection if the URL path starts with //
2021-05-18 09:15:07	azura9999999999	set	versions: + Python 3.11 type: security -> crash title: [security] http.server: Open Redirection if the URL path starts with // -> [security] http.server: Open Redirection if the URL path starts with //racha999coperation
2021-04-02 09:35:23	lukasz.langa	set	priority: release blocker -> deferred blocker messages: + **msg390047**
2021-03-31 19:59:03	christian.heimes	set	priority: normal -> release blocker nosy: + **christian.heimes**, **ned.deily**, **lukasz.langa**
2021-03-15 12:51:31	hamzaavvan	set	hgrepos: + hgrepo404
2021-03-13 21:13:55	hamzaavvan	set	keywords: + **patch** stage: patch review pull_requests: + **pull_request23609**
2021-02-19 05:59:56	hamzaavvan	set	messages: + **msg387284**
2021-02-17 22:27:18	vstinner	set	title: [SECURITY] Open Redirection In Python 3.7 & 3.8 -> [security] http.server: Open Redirection if the URL path starts with // components: + Library (Lib), - Windows versions: + Python 3.6, Python 3.9, Python 3.10
2021-02-17 22:26:33	vstinner	set	nosy: + **vstinner** messages: + **msg387193**
2021-02-15 12:50:46	hamzaavvan	set	title: Open Redirection In Python 3.7 & 3.8 -> [SECURITY] Open Redirection In Python 3.7 & 3.8
2021-02-14 11:42:36	hamzaavvan	create	