» Issue39503

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/83684

classification

Title: [security][CVE-2020-8492] Denial of service in urllib.request.AbstractBasicAuthHandler	
Type: security	**Stage**: resolved
Components: Library (Lib)	**Versions**: Python 3.11, Python 3.10, Python 3.9, Python 3.8, Python 3.7

process

Status: closed	**Resolution**: fixed
Dependencies:	**Superseder**:
Assigned To:	**Nosy List**: Anselmo Melo, bc, koobs, larry, mgorny, miss-islington, ned.deily, sxt1001, tapakund, vstinner, ware
Priority: normal	**Keywords**: patch

Created on **2020-01-30 15:11** by **vstinner**, last changed **2022-04-11 14:59** by **admin**. This issue is now **closed**.

Files

File name	Uploaded	Description	Edit
bench_parser.py	vstinner, 2020-03-30 17:21		
bench_parser2.py	bc, 2020-04-02 00:46	Benchmark	

Pull Requests

URL	Status	Linked	Edit
PR 18284	merged	vstinner, 2020-01-30 15:20	
PR 19291	closed	miss-islington, 2020-04-02 00:52	
PR 19292	closed	miss-islington, 2020-04-02 00:52	
PR 19296	merged	miss-islington, 2020-04-02 01:39	
PR 19297	merged	miss-islington, 2020-04-02 01:39	
PR 19299	closed	tapakund, 2020-04-02 07:09	
PR 19301	closed	tapakund, 2020-04-02 09:13	
PR 19302	closed	tapakund, 2020-04-02 10:27	
PR 19304	merged	vstinner, 2020-04-02 11:59	
PR 19305	merged	vstinner, 2020-04-02 12:05	

Messages (17)

msg361072 - (view) **Author: STINNER Victor (vstinner) *** **Date: 2020-01-30 15:11**

Copy of an email received on the Python Security Response team, 9 days ago. I consider that it's not worth it to have an embargo on this vulnerability, so I make it public.

Hi there,

I believe I've found a denial-of-service (DoS) bug in urllib.request.AbstractBasicAuthHandler. To start, I'm operating on some background information from this document: HTTP authentication <**https://developer.mozilla.org/en-US/docs/Web/HTTP/Authentication**>. The bug itself is a ReDoS <**https://www.regular-expressions.info/redos.html**> bug causing catastrophic backtracking. To reproduce the issue we can use the following code:

```
from urllib.request import AbstractBasicAuthHandler
auth_handler = AbstractBasicAuthHandler()
auth_handler.http_error_auth_reqed(
    'www-authenticate',
    'unused',
    'unused',
    {
        'www-authenticate': 'Basic ' + ',' * 64 + ' ' + 'foo' + ' ' +
'realm'
    }
)
```

The issue itself is in the following regular expression:

```
rx = re.compile('(?:.*,)*[ \t]*([^ \t]+)[ \t]+'
                'realm=(["\']?)([^"\']*)\\2', re.I)
```

In particular, the (?:.*,)* portion. Since "." and "," overlap and there are nested quantifiers we can cause catastrophic backtracking by repeating a comma. Note that since AbstractBasicAuthHandler is vulnerable, then both HTTPBasicAuthHandler and ProxyBasicAuthHandler are as well because they call http_error_auth_reqed. Building from the HTTP authentication document above, this means a server can send a specially crafted header along with an HTTP 401 or HTTP 407 and cause a DoS on the client.

I won't speculate on the severity of the issue too much - you will surely understand the impact better than I will. Although, the fact that this is client-side as opposed to server-side appears to reduce the severity, however the fact that it's a security-sensitive context (HTTP authentication) may raise the severity.

One possible fix would be changing the rx expression to the following:

```
rx = re.compile('(?:[^,]*,)*[ \t]*([^ \t]+)[ \t]+'
                'realm=(["\']?)([^"\']*)\\2', re.I)
```

This removes the character overlap in the nested quantifier and thus negates the catastrophic backtracking.

Let me know if you have any questions or what the next steps are from here.

Thanks for supporting Python security!

--
Matt Schwager

msg361073 - (view) **Author: STINNER Victor (vstinner) *** **Date: 2020-01-30 15:30**

I added this vulnerability to the following page to track fixes in all Python supported branches:
https://python-security.readthedocs.io/vuln/urllib-basic-auth-regex.html

msg361081 - (view) **Author: STINNER Victor (vstinner) *** **Date: 2020-01-30 22:33**

CVE-2020-8492 has been assigned to this vulnerability:
https://cve.mitre.org/cgi-bin/cvename.cgi?name=2020-8492

msg361343 - (view) **Author: Ben Caller (bc) *** **Date: 2020-02-04 10:53**

Isn't this a duplicate of ~~**bpo-38826**~~ ?

msg364996 - (view) **Author: STINNER Victor (vstinner) *** **Date: 2020-03-25 16:19**

> Isn't this a duplicate of ~~**bpo-38826**~~ ?

Oh right. I marked it as a duplicate of this issue.

msg365335 - (view) **Author: STINNER Victor (vstinner) *** **Date: 2020-03-30 17:21**

bench_parser.py: Benchmark for AbstractBasicAuthHandler.http_error_auth_reqed().

msg365538 - (view) **Author: Ben Caller (bc) *** **Date: 2020-04-02 00:24**

Instead of

```
repeat_10_3 = 'Basic ' + ', ' * (10 ** 3) + simple
```

in the benchmark, try

```
repeat_10_3 = 'Basic ' + ', ' * (10 ** 3) + 'A'
```

msg365541 - (view) **Author: STINNER Victor (vstinner) *** **Date: 2020-04-02 00:44**

Ooooh, I see. I didn't measure the performance of the right header. I re-run a benchmark using the HTTP header (repeat=15):

```
    header = 'Basic ' + ', ' * 15 + 'A'
```

Now I see a major performance difference. Comparison between master ("ref") and ~~**PR 18284**~~ ("fix"):

```
Mean +- std dev: [ref] 88.9 ms +- 2.4 ms -> [fix] 17.5 us +- 0.7 us: 5083.23x faster (-100%)
```

So the worst case is now way faster: more than 5000x faster!

It's even possible to go up to repeat=10**6 characters, it still takes less than 1

seconds: 412 ms +- 19 ms.

On the master branch, repeat=20 already takes around 3 seconds... The slowdown is exponential with repeat increase.

msg365542 - (view) **Author: STINNER Victor (vstinner) *** **Date: 2020-04-02 00:52**

New changeset **0b297d4ff1c0e4480ad33acae793fbaf4bf015b4** by Victor Stinner in branch 'master':
~~**bpo-39503**~~: CVE-2020-8492: Fix AbstractBasicAuthHandler (~~**GH-18284**~~)
https://github.com/python/cpython/commit/0b297d4ff1c0e4480ad33acae793fbaf4bf015b4

msg365578 - (view) **Author: STINNER Victor (vstinner) *** **Date: 2020-04-02 10:15**

New changeset **ea9e240aa02372440be8024acb110371f69c9d41** by Miss Islington (bot) in branch '3.8':
~~**bpo-39503**~~: CVE-2020-8492: Fix AbstractBasicAuthHandler (~~**GH-18284**~~) (~~**GH-19296**~~)
https://github.com/python/cpython/commit/ea9e240aa02372440be8024acb110371f69c9d41

msg365579 - (view) **Author: STINNER Victor (vstinner) *** **Date: 2020-04-02 10:16**

New changeset **b57a73694e26e8b2391731b5ee0b1be59437388e** by Miss Islington (bot) in branch '3.7':
~~**bpo-39503**~~: CVE-2020-8492: Fix AbstractBasicAuthHandler (~~**GH-18284**~~) (~~**GH-19297**~~)
https://github.com/python/cpython/commit/b57a73694e26e8b2391731b5ee0b1be59437388e

msg365663 - (view) **Author: Ned Deily (ned.deily) *** **Date: 2020-04-03 01:16**

New changeset **69cdeeb93e0830004a495ed854022425b93b3f3e** by Victor Stinner in branch '3.6':
~~**bpo-39503**~~: CVE-2020-8492: Fix AbstractBasicAuthHandler (~~**GH-18284**~~) (~~**GH-19304**~~)
https://github.com/python/cpython/commit/69cdeeb93e0830004a495ed854022425b93b3f3e

msg371921 - (view) **Author: Larry Hastings (larry) *** **Date: 2020-06-20 06:27**

New changeset **37fe316479e0b6906a74b0c0a5e495c55037fdfd** by Victor Stinner in branch '3.5':
~~**bpo-39503**~~: CVE-2020-8492: Fix AbstractBasicAuthHandler (~~**GH-18284**~~) (~~**#19305**~~)
https://github.com/python/cpython/commit/37fe316479e0b6906a74b0c0a5e495c55037fdfd

msg401762 - (view) **Author: tongxiaoge (sxt1001)** **Date: 2021-09-14 06:57**

https://github.com/python/cpython/blob/9f93018b69d72cb48d3444554261ae3b0ea00c93/Lib/urllib/request.py#L989
"headers" is a dict object? If so, the dict object does not seem to have no attribute "get_all".

msg401766 - (view) **Author: STINNER Victor (vstinner) *** **Date: 2021-09-14 10:23**

> "headers" is a dict object? If so, the dict object does not seem to have no attribute "get_all".

No, it's not a dict object.

msg401768 - (view) **Author: tongxiaoge (sxt1001)** **Date: 2021-09-14 10:33**

```
At the beginning of the issue, there is the following reproduction code:
from urllib.request import AbstractBasicAuthHandler
auth_handler = AbstractBasicAuthHandler()
auth_handler.http_error_auth_reqed(
    'www-authenticate',
    'unused',
    'unused',
    {
        'www-authenticate': 'Basic ' + ',' * 64 + ' ' + 'foo' + ' ' +
'realm'
    }
)

Here's the headers:

{
        'www-authenticate': 'Basic ' + ',' * 64 + ' ' + 'foo' + ' ' +
'realm'
 }

I think this is a dict object, so the current problem is fixed and no longer
compatible with the previous usage?
```

msg401770 - (view) **Author: STINNER Victor (vstinner) *** **Date: 2021-09-14 10:39**

```
This issue was a security vulnerability. It's now closed, please don't comment
closed issues. If you consider that there is a regression, please open a new issue.
```

History

Date	User	Action	Args
2022-04-11 14:59:26	admin	set	github: 83684
2021-09-14 10:39:30	vstinner	set	messages: + **msg401770**
2021-09-14 10:33:05	sxt1001	set	messages: + **msg401768**
2021-09-14 10:23:50	vstinner	set	messages: + **msg401766**
2021-09-14 06:57:28	sxt1001	set	nosy: + **sxt1001** messages: + **msg401762** versions: + Python 3.10, Python 3.11, - Python 3.5, Python 3.6
2020-06-20 08:33:12	larry	set	status: open -> closed resolution: fixed stage: patch review -> resolved
2020-06-20 08:30:06	koobs	set	nosy: + **koobs**
2020-06-20 06:27:09	larry	set	nosy: + **larry** messages: + **msg371921**
2020-04-03 01:16:03	ned.deily	set	nosy: + **ned.deily** messages: + **msg365663**
2020-04-02 12:05:26	vstinner	set	pull_requests: + **pull_request18667**

2020-04-02 11:59:15	vstinner	set	pull_requests: + **pull_request18666**
2020-04-02 11:43:04	vstinner	set	versions: - Python 2.7
2020-04-02 10:27:09	tapakund	set	pull_requests: + **pull_request18664**
2020-04-02 10:16:20	vstinner	set	messages: + **msg365579**
2020-04-02 10:15:59	vstinner	set	messages: + **msg365578**
2020-04-02 09:13:52	tapakund	set	pull_requests: + **pull_request18663**
2020-04-02 07:09:03	tapakund	set	nosy: + **tapakund** pull_requests: + **pull_request18661**
2020-04-02 01:39:24	miss-islington	set	pull_requests: + **pull_request18656**
2020-04-02 01:39:16	miss-islington	set	pull_requests: + **pull_request18655**
2020-04-02 00:52:45	miss-islington	set	pull_requests: + **pull_request18651**
2020-04-02 00:52:38	miss-islington	set	nosy: + **miss-islington** pull_requests: + **pull_request18650**
2020-04-02 00:52:23	vstinner	set	messages: + **msg365542**
2020-04-02 00:46:16	bc	set	files: - **bench_parser2.py**
2020-04-02 00:46:00	bc	set	files: + **bench_parser2.py**
2020-04-02 00:44:09	vstinner	set	messages: + **msg365541**
2020-04-02 00:25:00	bc	set	files: + **bench_parser2.py**
			messages: + **msg365538**
2020-03-30 17:21:05	vstinner	set	files: + **bench_parser.py**
			messages: + **msg365335**
2020-03-25 16:19:57	vstinner	set	messages: + **msg364996**
2020-03-25 16:19:39	vstinner	link	**issue38826 superseder**
2020-03-04 23:06:47	ware	set	nosy: + **ware**
2020-03-02 09:23:41	mgorny	set	nosy: + **mgorny**
2020-02-04 10:53:53	bc	set	nosy: + **bc** messages: + **msg361343**
2020-01-31 17:24:57	Anselmo Melo	set	nosy: + **Anselmo Melo**
2020-01-30 22:33:12	vstinner	set	messages: + **msg361081** title: [security] Denial of service in urllib.request.AbstractBasicAuthHandler -> [security] [CVE-2020-8492] Denial of service in urllib.request.AbstractBasicAuthHandler
2020-01-30 15:30:40	vstinner	set	messages: + **msg361073**

2020-01-30 15:20:34	vstinner	set	keywords: + **patch** stage: patch review pull_requests: + **pull_request17659**
2020-01-30 15:11:29	vstinner	create	