5 CVE-2022-30115: HSTS bypass via trailing dot

Share:

TIMELINE

haxatron1 submitted a report to curl. May 3rd (7 months ago)

curl allows users to load a HSTS cache which will cause curl to use HTTPS instead of HTTP given a HTTP URL for a given site specified in the HSTS cache.

If the trailing dot is used, the HSTS check will be bypassed.

If a user has a preloaded hsts.txt:

Code 150 Bytes Wrap lines Copy Download

```
# Your HSTS cache. https://curl.se/docs/hsts.html
# This file was generated by libcurl! Edit at your own risk.
accounts.google.com "20230503 08:47:52"
```

Doing the following:

Code 48 Bytes Wrap lines Copy Download

```
curl --hsts hsts.txt http://accounts.google.com.
```

Will cause accounts.google.com to be loaded over HTTP

Code 221 Bytes Wrap lines Copy Download

```
<HTML><HEAD><meta http-equiv="content-type" content="text/html;charset=utf-8">
<TITLE>301 Moved</TITLE></HEAD><BODY>
<H1>301 Moved</H1>
The document has moved
<A HREF="http://accounts.google.com/">here</A>.
</BODY></HTML>
```

This issue has been raised in other HTTP clients before such as in https://bugs.chromium.org/p/chromium/issues/detail?id=461481 and

HSTS bypass

bagder curl staff posted a comment. May 3rd (7 months ago)

Thank you for your report!

We will take some time and investigate your reports and get back to you with details and possible follow-up questions as soon as we can!

bagder curl staff changed the status to ○ **Triaged**. Updated May 3rd (7 months ago)

Confirmed and reproduced. I have a patch pending.

bagder curl staff posted a comment. May 3rd (7 months ago)

This flaw also works the other way around, if the trailing dot is in the file but not in the URL.

bagder curl staff posted a comment. May 3rd (7 months ago)

This patch seems to solve it for me. I also have two new tests that I've used to verify the patch with.

1 attachment:

F1715391: 0001-hsts-ignore-trailing-dots-when-comparing-hosts-names.patch

dgustafsson curl staff posted a comment. May 3rd (7 months ago)

Could there be two trailing dots with one still left after this?

Code 95 Bytes Wrap lines Copy Download

```
1  +  if(duphost[hlen - 1] == '.')
2  +    /* strip off trailing any dot */
3  +    duphost[--hlen] = 0;
```

bagder curl staff posted a comment. Updated May 3rd (7 months ago)

There might be multiple dots since there's nothing that filters away any, but I don't think they are problematic. If there is more than one provided, a normal name resolve will fail:

Code 253 Bytes Wrap lines Copy Download

```
1  $ curl localhost.
2  [server response]
```

```
6  localhost has address 127.0.0.1
7  localhost has IPv6 address ::1
8  host localhost..
9  host: 'localhost..' is not a legal name (empty label)
10
```

bagder curl staff posted a comment. May 3rd (7 months ago)

A question is what other host name comparisons we have that also are this stupid due to the trailing dot. As @haxatron1 also filed https://hackerone.com/reports/1553301 which basically has the exact same source => the host name is now stored *with* the trailing dot which it didn't do before 7.82.0.

haxatron1 posted a comment. May 3rd (7 months ago)

I can confirm for both cases that after the patch that HSTS now works.

These are only the two scenarios I could find where the trailing dot is causing problems.

bagder curl staff posted a comment. May 3rd (7 months ago)

I'm struggling to find an appropriate CWE, but I think CWE-319 might be close enough: **Cleartext Transmission of Sensitive Information**

bagder curl staff updated CVE reference to **CVE-2022-30115**. May 3rd (7 months ago)

May 3rd (7 months ago)

bagder curl staff

changed the report title from **HSTS bypass via trailing dot** to **CVE-2022-30115: HSTS bypass via trailing dot**.

haxatron1 posted a comment. Updated May 3rd (7 months ago)

Well, I believe this is similar to https://curl.se/docs/CVE-2021-22946.html which was "Missing Required Cryptographic Step" so that was what I selected initially.

"Cleartext Transmission of Sensitive Information" seems fine though

bagder curl staff posted a comment. May 3rd (7 months ago)

Advisory

bagder curl staff posted a comment. May 3rd (7 months ago)

Yeah, I don't think the CWE is terribly important so unless someone feels strongly about it, let's just keep this.

haxatron1 posted a comment. May 3rd (7 months ago)

Details LGTM!

haxatron1 posted a comment. May 3rd (7 months ago)

Question, does curl support HPKP (now deprecated in most browsers)? The 2 linked issues also mention that HPKP can be bypassed if a compromised CA issues a certificate for 'example.com.', then when a user connects to https://example.com. the pinning will be bypassed. If it exists, it might be worth checking the parsing logic there as well.

bagder curl staff posted a comment. May 3rd (7 months ago)

> does curl support HPKP

Nope. It supports pinning, but not via HPKP so I don't think there's a host name confusion risk there.

bagder curl staff posted a comment. May 4th (7 months ago)

There is a host name check in altsvc.c that looks like it also needs trailing-dot adjustment, but I don't think it has any security impact to miss out on that.

haxatron1 posted a comment. May 4th (7 months ago)

I don't know if the browsers consider trailing dot for alt-svc or not, but yes the trailing dot should not have any security implications for alt-svc as its not a security feature

bagder curl staff posted a comment. May 5th (7 months ago)

I have notified distros@openwall about this issue now. Set for announcement with the pending release on May 11.

bagder curl staff posted a comment. May 5th (7 months ago)

> I don't know if the browsers consider trailing dot for alt-svc or not.

As it is considered the same host with out without the t-dot in all those other cases it would be very strange to not give alt-svc the same treatment.

release.

bagder curl staff closed the report and changed the status to ○ **Resolved**. May 11th (7 months ago)

Published. This issue is now eligible for a bounty claim from IBB.

bagder curl staff requested to disclose this report. May 11th (7 months ago)

haxatron1 agreed to disclose this report. May 11th (7 months ago)

This report has been disclosed. May 11th (7 months ago)

curl has decided that this report is not eligible for a bounty. May 13th (7 months ago)

Thanks for your work. The actual monetary reward part for this issue is managed by the Internet Bug Bounty so the curl project itself therefor sets the reward amount to **zero USD**. If you haven't already, please submit your reward request to them and refer back to this issue.