

Bug #3134 CLOSED

mod_extforward plugin has out-of-bounds (OOB) write of 4-byte -1

Added by [povcfe-bug](#) 11 months ago. Updated 11 months ago.

Status: Fixed
Priority: Normal
Category: mod_extforward
Target version: 1.4.64
ASK QUESTIONS IN FORUMS: No

Description

1. OOB write reproduce(lighttpd-1.4.46-1.4.63)

The OOB write covers lighttpd-1.4.46-1.4.63

1.1 lighttpd configuration

```
server.document-root = "/var/www/html/"
server.port = 8080

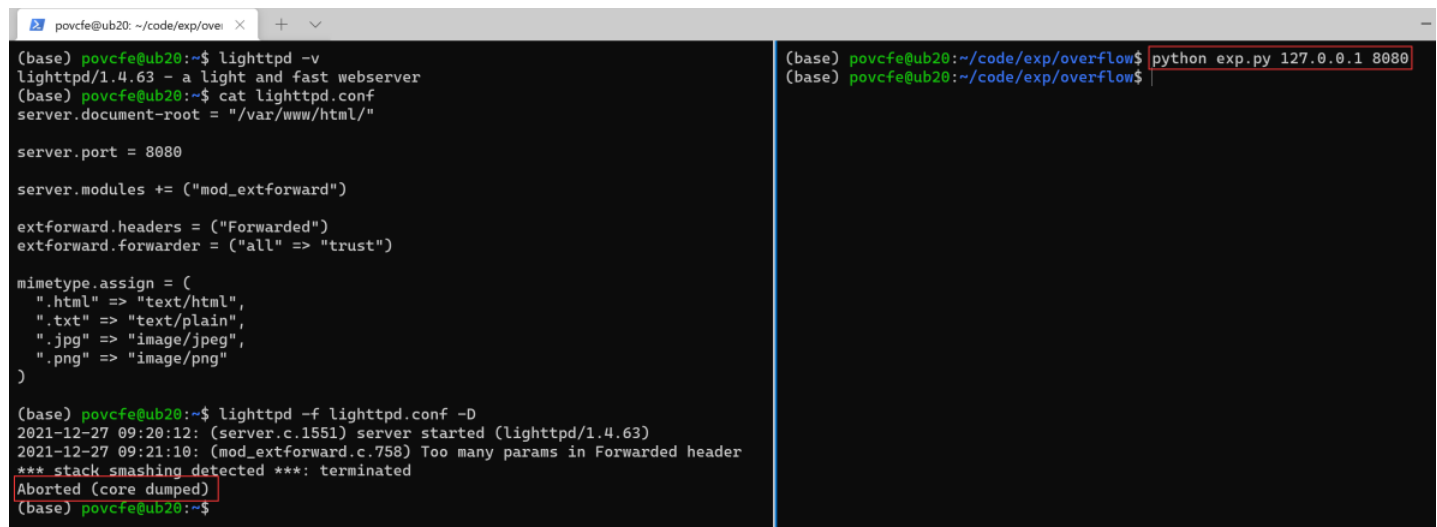
server.modules += ("mod_extforward")
extforward.headers = ("Forwarded")
extforward.forwarder = ("all" => "trust")

mimeassign = (
    ".html" => "text/html",
    ".txt" => "text/plain",
    ".jpg" => "image/jpeg",
    ".png" => "image/png"
)
```

1.2 compilation

```
wget https://download.lighttpd.net/lighttpd/releases-1.4.x/lighttpd-1.4.63.tar.gz
./configure CFLAGS="-m32"
make
make install
```

1.3 remote denial of service



```
(base) povcfe@ub20:~/code/exp/overflow$ python exp.py 127.0.0.1 8080
(base) povcfe@ub20:~/code/exp/overflow$

(base) povcfe@ub20:~$ lighttpd -v
lighttpd/1.4.63 - a light and fast webserver
(base) povcfe@ub20:~$ cat lighttpd.conf
server.document-root = "/var/www/html/"

server.port = 8080

server.modules += ("mod_extforward")
extforward.headers = ("Forwarded")
extforward.forwarder = ("all" => "trust")

mimeassign = (
    ".html" => "text/html",
    ".txt" => "text/plain",
    ".jpg" => "image/jpeg",
    ".png" => "image/png"
)

(base) povcfe@ub20:~$ lighttpd -f lighttpd.conf -D
2021-12-27 09:20:12: (server.c.1551) server started (lighttpd/1.4.63)
2021-12-27 09:21:10: (mod_extforward.c.758) Too many params in Forwarded header
*** stack smashing detected ***: terminated
Aborted (core dumped)
(base) povcfe@ub20:~$
```

2. analysis of the causes of vulnerabilities

The mod_extforward_Forwarded() function of the mod_extforward plugin has a four-byte stack overflow

```
static handler_t mod_extforward_Forwarded (server *srv, connection *con, plugin_data *p, buffer *forwarded) {
```

```

int offsets[256]; /* (~50 params is more than reasonably expected to handle) */
...
while (i < used) {
    ...
    // When "j = 255", "offsets[++j] = -1" means "offsets[256] = -1", causing a stack overflow

    if (s[i] == ';' ) {
        if (j >= (int)(sizeof(offsets)/sizeof(int))) break;
        offsets[++j] = -1; /* ("offset" separating params from next proxy) */
        ++i;
        continue;
    }
}

```

3. patch

```

diff --git a/mod_extforward.c b/mod_extforward-patch.c
index ba957e0..f0a38d4 100644
--- a/mod_extforward.c
+++ b/mod_extforward-patch.c
@@ -715,7 +715,7 @@ static handler_t mod_extforward_Forwarded (request_st * const r, plugin_data * c
     while (s[i] == ' ' || s[i] == '\t') ++i;
     if (s[i] == ';') { ++i; continue; }
     if (s[i] == ',') {
-        if (j >= (int)(sizeof(offsets)/sizeof(int))) break;
+        if (j >= (int)((sizeof(offsets)/sizeof(int)) - 1)) break;
         offsets[++j] = -1; /* ("offset" separating params from next proxy) */
         ++i;
         continue;
     }
}

```

Files

1.png (227 KB)	povcfe-bug, 2022-01-05 05:01
lighttpd.conf (314 Bytes)	povcfe-bug, 2022-01-05 05:09
header.png (105 KB)	povcfe-bug, 2022-01-05 09:00
0001-mod_extforward-fix-out-of-bounds-OOB-write-of-4-byte.patch (937 Bytes)	povcfe-bug, 2022-01-05 11:14



History Notes Property changes Associated revisions

Updated by [povcfe-bug](#) 11 months ago

I uploaded the exp by mistake, please delete it in time

Updated by [gstrauss](#) 11 months ago

- **File** deleted (*exp.py*)

Updated by [gstrauss](#) 11 months ago

- **Subject** changed from *Security - lighttpd mod_extforward plugin has stack overflow vulnerability to lighttpd mod_extforward plugin has out-of-bounds (OOB) write of 4-byte -1*
- **Description** updated (diff)
- **Status** changed from *New* to *Patch Pending*
- **Target version** changed from *1.4.xx* to *1.4.64*

You are correct that there is an out-of-bounds write on the stack. However, this out-of-bounds write is not controlled by the attacker. The value that is written out-of-bounds after the end of offsets[] is -1

However, with the default lighttpd build with gcc -O2, I was unable to trigger a crash in lighttpd on amd64 (64-bit) or on i686 (32-bit).

As with your prior posts, I think your test toolchain is configured with a stack canary which crashes when an out-of-bounds read or write is detected, which is fine for your research, but not necessarily a reflection of real-world impact. When mod_extforward.c is compiled with -fstack-protector-all -fstack-protector-strong, I was able to trigger a crash in lighttpd on i686 (32-bit), but not amd64 (64-bit), using the reproducer (exp.py) you had provided.

Also, the scenario in which this occurs is not enabled by default in lighttpd.

- First, the user must be using `mod_extforward`. (not default)
- Second, the user must explicitly configure `extforward.headers` to contain "Forwarded" (not default)
- Third, the user must have configured `mod_extforward` to trust the upstream proxy server which proxied the request to `lighttpd`, and from which `lighttpd` is receiving the "Forwarded" header. This upstream proxy must allow and use this unusual "Forwarded" header.

Your statement that this issue affects `lighttpd 1.4.41-1.4.63` is incorrect. Support for the "Forwarded" header was added in `lighttpd 1.4.46`.

<https://redmine.lighttpd.net/issues/2703>

I have updated your original post.

Given this initial assessment (not yet complete), I have changed the wording of this issue to tone it down from vulnerability to OOB write. There is a bug that will be fixed, but unless further information comes to light, this does not appear to have security implications for default usage of `lighttpd`. Please help me determine in which scenarios and platforms this might have an impact.

Updated by [gstrauss 11 months ago](#)

- **Subject** changed from *lighttpd mod_extforward plugin has out-of-bounds (OOB) write of 4-byte -1 to mod_extforward plugin has out-of-bounds (OOB) write of 4-byte -1*

Updated by [povcfe-bug 11 months ago](#)

[gstrauss](#) wrote in [#note-3](#):

You are correct that there is an out-of-bounds write on the stack. However, this out-of-bounds write is not controlled by the attacker. The value that is written out-of-bounds after the end of offsets[] is -1

However, with the default lighttpd build with gcc -O2, I was unable to trigger a crash in lighttpd on amd64 (64-bit) or on i686 (32-bit).

As with your prior posts, I think your test toolchain is configured with a stack canary which crashes when an out-of-bounds read or write is detected, which is fine for your research, but not necessarily a reflection of real-world impact. When mod_extforward.c is compiled with -fstack-protector-all -fstack-protector-strong, I was able to trigger a crash in lighttpd on i686 (32-bit), but not amd64 (64-bit), using the reproducer (exp.py) you had provided.

Also, the scenario in which this occurs is not enabled by default in lighttpd.

- First, the user must be using `mod_extforward`. (not default)
- Second, the user must explicitly configure `extforward.headers` to contain "Forwarded" (not default)
- Third, the user must have configured `mod_extforward` to trust the upstream proxy server which proxied the request to `lighttpd`, and from which `lighttpd` is receiving the "Forwarded" header. This upstream proxy must allow and use this unusual "Forwarded" header.

Your statement that this issue affects lighttpd 1.4.41-1.4.63 is incorrect. Support for the "Forwarded" header was added in lighttpd 1.4.46.

<https://redmine.lighttpd.net/issues/2703>

I have updated your original post.

Given this initial assessment (not yet complete), I have changed the the wording of this issue to tone it down from vulnerability to OOB write. There is a bug that will be fixed, but unless further information comes to light, this does not appear to have security implications for default usage of lighttpd. Please help me determine in which scenarios and platforms this might have an impact.

I'm using the ubuntu 20.04 default configuration, where canary is turned on by default

Updated by [povcfe-bug 11 months ago](#)

[povcfe-bug](#) wrote in [#note-5](#):

[gstrauss](#) wrote in [#note-3](#):

You are correct that there is an out-of-bounds write on the stack. However, this out-of-bounds write is not controlled by the attacker. The value that is written out-of-bounds after the end of offsets[] is -1

However, with the default lighttpd build with gcc -O2, I was unable to trigger a crash in lighttpd on amd64 (64-bit) or on i686 (32-bit).

As with your prior posts, I think your test toolchain is configured with a stack canary which crashes when an out-of-bounds read or write is detected, which is fine for your research, but not necessarily a reflection of real-world impact. When `mod_extforward.c` is compiled with `-fstack-protector-all -fstack-protector-strong`, I was able to trigger a crash in `lighttpd` on i686 (32-bit), but not amd64 (64-bit), using the reproducer (`exp.py`) you had provided.

Also, the scenario in which this occurs is not enabled by default in `lighttpd`.

- First, the user must be using `mod_extforward`. (not default)
- Second, the user must explicitly configure `extforward.headers` to contain "Forwarded" (not default)
- Third, the user must have configured `mod_extforward` to trust the upstream proxy server which proxied the request to `lighttpd`, and from which `lighttpd` is receiving the "Forwarded" header. This upstream proxy must allow and use this unusual "Forwarded" header.

Your statement that this issue affects `lighttpd` 1.4.41-1.4.63 is incorrect. Support for the "Forwarded" header was added in `lighttpd` 1.4.46.

<https://redmine.lighttpd.net/issues/2703>

I have updated your original post.

Given this initial assessment (not yet complete), I have changed the wording of this issue to tone it down from vulnerability to OOB write. There is a bug that will be fixed, but unless further information comes to light, this does not appear to have security implications for default usage of `lighttpd`. Please help me determine in which scenarios and platforms this might have an impact.

I'm using the ubuntu 20.04 default configuration, where canary is turned on by default

The reason why 64-bit programs can't crash is that sse 16-bit alignment, when closing intel sse, 64-bit programs will also crash, so please make sure that mips, arm and other architecture programs will not trigger a crash.

Updated by [povcfe-bug](#) 11 months ago

So compiling 32-bit `lighttpd` within an operating system with a higher gcc version, or compiling and using `lighttpd` with an architecture that does not support sse 16-bit byte alignment, will result in a denial of service

Updated by [gstrauss](#) 11 months ago

So compiling 32-bit `lighttpd` within an operating system with a higher gcc version, or compiling and using `lighttpd` with an architecture that does not support sse 16-bit byte alignment, will result in a denial of service

You seem to be overlooking the prerequisites to reaching the bug. Case in point, I wrote:

- Third, the user must have configured `mod_extforward` to trust the upstream proxy server which proxied the request to `lighttpd`, and from which `lighttpd` is receiving the "Forwarded" header. This upstream proxy must allow and use this unusual "Forwarded" header.

For someone to want to configure `lighttpd` this way, they must be using a proxy which supports "Forwarded". Do you know of real-world use in popular CDNs? Most CDNs and cloud providers have their own custom fields for X-Forwarded-For.

I took a quick look and Cloudflare does not currently support Forwarded.

<https://community.cloudflare.com/t/support-for-http-forwarded-header-as-a-replacement-for-x-forwarded-for/320943>

Nor does HAProxy (where the HAProxy "PROXY" protocol is often preferred)

<https://github.com/haproxy/haproxy/issues/575>

I did find that "Forwarded" is implemented in

<https://microsoft.github.io/reverse-proxy/articles/transforms.html#forwarded>

though a developer noted it is not enabled by default

(<https://github.com/dotnet/aspnetcore/issues/5978#issuecomment-668013246>)

and "Forwarded" is implemented in

https://github.com/gorilla/handlers/blob/master/proxy_headers.go

Please keep in mind that my questions here and above are to help gauge potential impact of the bug.

Thus far, based on what I have posted here and above, I am confident that the bug is not reachable in **common** use scenarios of `lighttpd` `mod_extforward`.

Updated by [povcfe-bug](#) 11 months ago

[gstrauss](#) wrote in [#note-8](#):

So compiling 32-bit `lighttpd` within an operating system with a higher gcc version, or compiling and using `lighttpd` with an architecture that does not support sse 16-bit byte alignment, will result in a denial of service

You seem to be overlooking the prerequisites to reaching the bug. Case in point, I wrote:

- Third, the user must have configured `mod_extforward` to trust the upstream proxy server which proxied the request to `lighttpd`, and from which `lighttpd` is receiving the "Forwarded" header. This upstream proxy must allow and use this unusual "Forwarded" header.

For someone to want to configure `lighttpd` this way, they must be using a proxy which supports "Forwarded". Do you know of real-world use in popular CDNs? Most CDNs and cloud providers have their own custom fields for X-Forwarded-For.

I took a quick look and Cloudflare does not currently support Forwarded.

<https://community.cloudflare.com/t/support-for-http-forwarded-header-as-a-replacement-for-x-forwarded-for/320943>

Nor does HAProxy (where the HAProxy "PROXY" protocol is often preferred)

<https://github.com/haproxy/haproxy/issues/575>

I did find that "Forwarded" is implemented in

<https://microsoft.github.io/reverse-proxy/articles/transforms.html#forwarded>

though a developer noted it is not enabled by default

(<https://github.com/dotnet/aspnetcore/issues/5978#issuecomment-668013246>)

and "Forwarded" is implemented in

https://github.com/gorilla/handlers/blob/master/proxy_headers.go

Please keep in mind that my questions here and above are to help gauge potential impact of the bug.

Thus far, based on what I have posted here and above, I am confident that the bug is not reachable in **common** use scenarios of `lighttpd mod_extforward`.

I noticed that "https://redmine.lighttpd.net/projects/lighttpd/wiki/Docs_ModExtForward" mentions a custom setting for "Forwarded", which is not really the default configuration, but I mean that for users with such a configuration, remote denial of service is possible.

Updated by [povcfe-bug](#) 11 months ago

- **File** header.png added



Updated by [gstrauss](#) 11 months ago

This is a bug and the bug has been acknowledged. I am not sure what your posts are trying to say beyond that.

My posts have repeatedly stated that I am trying to gauge the potential impact, and thus far the potential impact in the real world appears to be very low.

Updated by [povcfe-bug](#) 11 months ago

[gstrauss](#) wrote in [#note-11](#):

This is a bug and the bug has been acknowledged. I am not sure what your posts are trying to say beyond that.

My posts have repeatedly stated that I am trying to gauge the potential impact, and thus far the potential impact in the real world appears to be very low.

I agree with you in the comment above that he will not be triggered by default

Updated by [povcfe-bug](#) 11 months ago

[gstrauss](#) wrote in [#note-11](#):

This is a bug and the bug has been acknowledged. I am not sure what your posts are trying to say beyond that.

My posts have repeatedly stated that I am trying to gauge the potential impact, and thus far the potential impact in the real world appears to be very low.

I hope you will listen to me carefully, you said he will not be triggered by default, I agree. Also you said that canary must be turned on to trigger a crash, I told you that canary is turned on by default in higher versions of gcc, and that crashes are possible for system architectures that do not have SSE turned on.

Updated by [povcfe-bug](#) 11 months ago

[povcfe-bug](#) wrote in [#note-12](#):

[gstrauss](#) wrote in [#note-11](#):

This is a bug and the bug has been acknowledged. I am not sure what your posts are trying to say beyond that.

My posts have repeatedly stated that I am trying to gauge the potential impact, and thus far the potential impact in the real world appears to be very low.

I agree with you in the comment above that he will not be triggered by default

Please respect contributors who find problems and submit patches

Updated by [gstrauss 11 months ago](#)

Please respect contributors who find problems and submit patches

I appreciate that you have taken the time to find this and to post it so that it can be fixed. Thank you.

Above, I wrote:

This is a bug and the bug has been acknowledged. I am not sure what your posts are trying to say beyond that.

My posts have repeatedly stated that I am trying to gauge the potential impact, and thus far the potential impact in the real world appears to be very low.

Please help me to understand why you feel disrespected by these grounded statements. Impact analysis is an important part of bug and security triage.

Updated by [povcfe-bug 11 months ago](#)

gstrauss wrote in [#note-15](#):

Please respect contributors who find problems and submit patches

I appreciate that you have taken the time to find this and to post it so that it can be fixed. Thank you.

Above, I wrote:

This is a bug and the bug has been acknowledged. I am not sure what your posts are trying to say beyond that.

My posts have repeatedly stated that I am trying to gauge the potential impact, and thus far the potential impact in the real world appears to be very low.

Please help me to understand why you feel disrespected by these grounded statements. Impact analysis is an important part of bug and security triage.

I don't think you've just listened carefully to my analysis, and there are differences between our two concerns. Impact exclusion is indeed a very important aspect, and I agree with your analysis above

Updated by [gstrauss 11 months ago](#)

This is a bug and the bug has been acknowledged. I am not sure what your posts are trying to say beyond that.

I don't think you've just listened carefully to my analysis, and there are differences between our two concerns.

Would you please describe your concerns in more detail?

Your post is less than 5 hours old and I have spent a considerable amount of time reviewing, reproducing, and analyzing the potential impact, and have tried to be explicit in my posts that these are the steps I am taking. If you feel that I am giving it short thrift, then please describe your expectations in more detail.

Updated by [povcfe-bug 11 months ago](#)

gstrauss wrote in [#note-17](#):

This is a bug and the bug has been acknowledged. I am not sure what your posts are trying to say beyond that.

I don't think you've just listened carefully to my analysis, and there are differences between our two concerns.

Would you please describe your concerns in more detail?

Your post is less than 5 hours old and I have spent a considerable amount of time reviewing, reproducing, and analyzing the potential impact, and have tried to be explicit in my posts that these are the steps I am taking. If you feel that I am giving it short thrift, then please describe your expectations in more detail.

Sorry, my narrative above was just to show you that users using this non-default configuration can trigger crashes with the default compile option. I appreciate your work, and your attention to this issue, and I apologize.

Updated by [gstrauss 11 months ago](#)

Yes, for systems and distros which enable the canary by default, and if the prerequisites to reach the bug are met, then the bug will trigger a crash.

Please attach a patch (e.g. output from `git format-patch`) with the patch in your original post and how you'd like to be credited. The patch looks good, and I'll need to edit the commit message a bit, but will try to preserve what you'd like to include.

It's late here and I'll return to this tomorrow.

Updated by [povcfe-bug 11 months ago](#)

- **File** `0001-mod_extforward-fix-out-of-bounds-OOB-write-of-4-byte.patch` added

patch: "0001-mod_extforward-fix-out-of-bounds-OOB-write-of-4-byte.patch"

you can call me "povcfe" and I would like the commit message to include a personal thx to me

Updated by [povcfe-bug 11 months ago](#)

I have applied for a cve id, please trace

Updated by [gstrauss](#) 11 months ago

You did not include a commit message in the attachment. Please see the lighttpd git history for how I credit contributors and let me know if you would like any different.

I have applied for a cve id, please trace

What do you mean by "please trace"? Is that an incomplete sentence?

I have a feeling that you are disappointed that my analysis indicates that this bug is low severity, as its prevalence in production is likely to be rare and specialized. Even when the bug is reachable, I do not believe it exploitable beyond triggering a crash. I will dispute the CVE if you represent the bug otherwise.

Updated by [povcfe-bug](#) 11 months ago

gstrauss wrote in [#note-22](#):

You did not include a commit message in the attachment. Please see the lighttpd git history for how I credit contributors and let me know if you would like any different.

I have applied for a cve id, please trace

What do you mean by "please trace"? Is that an incomplete sentence?

I have a feeling that you are disappointed that my analysis indicates that this bug is low severity, as its prevalence in production is likely to be rare and specialized. Even when the bug is reachable, I do not believe it exploitable beyond triggering a crash. I will dispute the CVE if you represent the bug otherwise.

I applied for cve with a remote denial of service error type, which I think is reasonable, emm. I would like to know if you agree.

Updated by [povcfe-bug](#) 11 months ago

this is the new patch

```
From 8a91fd45f7f31707a876492b13c0f46845626727 Mon Sep 17 00:00:00 2001
From: povcfe <povcfe@qq.com>
Date: Wed, 5 Jan 2022 12:44:34 +0000
Subject: [PATCH] [mod_extforward] fix out-of-bounds (OOB) write of 4-byte -1

(thx povcfe)
---
src/mod_extforward.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

diff --git a/src/mod_extforward.c b/src/mod_extforward.c
index 733231fd..8dbdfad9 100644
--- a/src/mod_extforward.c
+++ b/src/mod_extforward.c
@@ -715,7 +715,7 @@ static handler_t mod_extforward_Forwarded (request_st * const r, plugin_data * c
     while (s[i] == ' ' || s[i] == '\t') ++i;
     if (s[i] == ';') { ++i; continue; }
     if (s[i] == ',') {
-        if (j >= (int)(sizeof(offsets)/sizeof(int))) break;
+        if (j >= (int)((sizeof(offsets)/sizeof(int)) - 1)) break;
         offsets[++j] = -1; /*("offset" separating params from next proxy)*/
         ++i;
         continue;
     }
--
2.25.1
```

Updated by [gstrauss](#) 11 months ago

I have applied for a cve id, please trace

What do you mean by "please trace"? Is that an incomplete sentence?

What do you mean by "please trace"? Is that an incomplete sentence?

I applied for cve with a remote denial of service error type, which I think is reasonable, emm. I would like to know if you agree.

I do not understand your slang. What is "emm"?

On to the more important issue: context.

I applied for cve with a remote denial of service error type [...]. I would like to know if you agree.

The CVE should be specific, not vague.

The CVE should not be worded in an inflammatory or click-bait manner.

You have found a bug in `lighttpd mod_extforward` when "Forwarded" is configured.
(Thank you for reporting the bug.)

`lighttpd` is a modular web server. The distinction between the `lighttpd` core and `lighttpd` modules is important, as is the popularity of any given `lighttpd` module.

DRAFT:

There is a potential remote denial of service in `lighttpd mod_extforward` under specific, non-default and uncommon 32-bit `lighttpd mod_extforward` configurations.

Under specific, non-default and uncommon `lighttpd mod_extforward` configurations, a remote attacker can trigger a 4-byte out-of-bounds write of value `'-1'` to the stack. This is not believed to be exploitable in any way beyond triggering a crash of the `lighttpd` server on systems where the `lighttpd` server has been built 32-bit and with compiler flags which enable a stack canary `--gcc/clang -fstack-protector-strong` or `-fstack-protector-all`, but bug not visible with only `-fstack-protector`.

With standard `lighttpd` builds using `-O2` optimization on 64-bit `x86_64`, this bug has not been observed to cause adverse behavior, even with `gcc/clang -fstack-protector-strong`.

For the bug to be reachable, the user must be using a non-default `lighttpd` configuration which enables `mod_extforward` and configures `mod_extforward` to accept and parse the "Forwarded" header from a trusted proxy. At this time, support for RFC7239 Forwarded is not common in CDN providers or popular web server reverse proxies. It bears repeating that for the user to desire to configure `lighttpd mod_extforward` to accept "Forwarded", the user must also be using a trusted proxy (in front of `lighttpd`) which understands and actively modifies the "Forwarded" header sent to `lighttpd`.

`lighttpd` natively supports RFC7239 "Forwarded"
`hiawatha` natively supports RFC7239 "Forwarded"

`nginx` can be manually configured to add a "Forwarded" header
<https://www.nginx.com/resources/wiki/start/topics/examples/forwarded/>

A 64-bit build of `lighttpd` on `x86_64` (not known to be affected by this bug) in front of another `lighttpd` will detect and reject a malicious "Forwarded" request header, thereby thwarting an attempt to trigger this bug.

The following servers currently *do not* natively support RFC7239 Forwarded:

`nginx`
`apache2`
`caddy`
`node.js`
`haproxy`
`squid`
`varnish-cache`
`litespeed`

Given the general dearth of support for RFC7239 Forwarded in popular CDNs and web server reverse proxies, and given the prerequisites in `lighttpd mod_extforward` needed to reach this bug, the number of `lighttpd` servers vulnerable to this bug is estimated to be vanishingly small. Large systems using reverse proxies are likely running 64-bit `lighttpd`, which is not known to be adversely affected by this bug.

In the future, it is desirable for more servers to implement RFC7239 Forwarded. `lighttpd` developers would like to thank `povcfe` for reporting this bug so that it can be fixed before more CDNs and web servers implement RFC7239 Forwarded.

[Edit: above has been updated to incorporate some of the additional info below]

Updated by [povcfe-bug 11 months](#) ago

"emm." is a slang word for friendly

Thank you for the detailed analysis of this bug

Updated by [gstrauss 11 months](#) ago

My testing on 32-bit ARM and on 64-bit ARM appears to have the same behavior as `i686` (32-bit) and `x86_64` (64-bit)

My builds use `gcc` and the default build settings in `lighttpd configure.ac`, which include `-O2`

When built from source, `lighttpd` does not crash on 32-bit or 64-bit, on ARM or `i686` or `x86_64`.
When built from source with `--enable-extra-warnings` (which enables `-fstack-protector` in `lighttpd configure.ac`), `lighttpd` does not crash on 32-bit or 64-bit, on ARM or `i686` or `x86_64`.

When built with `-fstack-protector-strong` or `-fstack-protector-all`, then I am able to trigger a crash in lighttpd built **32-bit** on ARM and i686, but 64-bit on ARM or x86_64 builds do not crash.

As expected, the behavior appears to be the same if clang is used with these flags instead of gcc.

Conclusion: in addition to the prerequisites already noted, it appears that to trigger a crash in lighttpd, lighttpd must be built 32-bit and with `-fstack-protector-strong` or `-fstack-protector-all`. Of course, other platforms may differ. However, the majority of lighttpd usage in production is expected to be on platforms using gcc or clang compilers.

Updated by [gstrauss 11 months ago](#)

`-fstack-protector-strong` was introduced in gcc 4.9.0.

`-fstack-protector-strong` is present in clang 5.0.0 documentation, and I found references (unverified) to clang 3.8.

As an aside, I tested OpenWRT 32-bit build of lighttpd and it did not appear to be adversely affected by this bug -- it did not crash. The lighttpd OpenWRT package appears to be built with `-fstack-protector`, possibly due to the prevalence of older compilers for some of openwrt-supported hardware platforms.

Note: this bug will automatically be marked fixed once the commit fixing it is pushed to lighttpd git master branch. I am still subscribed and you can still add comments here.

When assigned, please post the CVE number here.

Updated by [povcfe-bug 11 months ago](#)

gstrauss wrote in [#note-28](#):

-fstack-protector-strong was introduced in gcc 4.9.0.

-fstack-protector-strong is present in clang 5.0.0 documentation, and I found references (unverified) to clang 3.8.

As an aside, I tested OpenWRT 32-bit build of lighttpd and it did not appear to be adversely affected by this bug -- it did not crash. The lighttpd OpenWRT package appears to be built with -fstack-protector, possibly due to the prevalence of older compilers for some of openwrt-supported hardware platforms.

Note: this bug will automatically be marked fixed once the commit fixing it is pushed to lighttpd git master branch. I am still subscribed and you can still add comments here.

When assigned, please post the CVE number here.

the CVE number is CVE-2022-22707

Looking forward to the next collaboration

Updated by [gstrauss 11 months ago](#)

I have lightly edited the draft I posted above. Please coordinate with lighttpd developers (me included) before submitting the CVE to be published.

Updated by [11 months ago](#)

- **Status** changed from *Patch Pending* to *Fixed*

Applied in changeset [8c62a890e23f5853b1a562b03fe3e1bccc6e7664](#).

Updated by [gstrauss 11 months ago](#)

Looking forward to the next collaboration

@povcfe

- You did not "collaborate" with lighttpd developers; you merely reported a bug. I question your understanding of the meaning of the word "collaborate".
- You did not participate in any research to assess the potential impact.
- **You did not coordinate the CVE release with lighttpd developers.** Instead, the CVE was published barely 25 hours after reporting the bug here. That is an astounding lack of professionalism, especially considering that within a few hours of your report I had responded acknowledging the bug and provided an initial assessment. (Also, since you had assumed that this bug was a security issue, the bug should have been reported to the security email address at lighttpd.net, instead of publicly here.)
- The CVE was published before the proposed fix was committed to the official lighttpd git repo.
- The CVE does not clearly state that the impact is limited to 32-bit lighttpd and to specific lighttpd `mod_extforward` configurations expected to be rare. The CVE contains very little information, even though it was published (unbeknownst to me at the time) more than 4 hours after my DRAFT post above.

