

## Bug 580018 - Denial-of-Service vulnerability in the DTLS stack

**Status:** NEW

**Alias:** None

**Product:** Californium

**Component:** Scandium ([show other bugs](#))

**Version:** unspecified 

**Hardware:** PC Linux

**Importance:** P3 normal

**Target** --- 

**Milestone:**

**Assignee:** Security vulnerabilitied reported against Eclipse projects

**QA Contact:**

**URL:**

**Whiteboard:**

**Keywords:** security

**Depends on:**



**Blocks:**

**Reported:** 2022-05-26 06:14 EDT by Nurullah Erinola 

**Modified:** 2022-07-29 09:24 EDT ([History](#))


**CC List:** 5 users ([show](#))

**See Also:**

Attachments		
<a href="#">PCAP file with the interaction</a> (4.62 KB, application/vnd.tcpdump.pcap) 	<i>no flags</i>	<a href="#">Details</a>
<a href="#">2022-05-26 06:14 EDT</a> , Nurullah Erinola 		
<a href="#">Add an attachment</a> (proposed patch, testcase, etc.)		<a href="#">View All</a>

Note

You need to [log in](#) before you can comment on or make changes to this bug.

Nurullah Erinola  2022-05-26 06:14:29 EDT

[Description](#)

Created [attachment 288540](#) [\[details\]](#).

PCAP file with the interaction

Our systematic analysis of DTLS implementations indicated that your DTLS stack is vulnerable to a Denial-of-Service attack.

Tested build

=====

v3.5.0

Affected protocol versions

=====

DTLS 1.2

Expected behavior

=====

According to the DTLS 1.2 standard in chapter 4.2.1 [1], the server may respond with a HelloVerifyRequest message when the client sends its ClientHello message.

This mechanism should provide defense against the following two DoS attacks:

1. An attacker can consume excessive resources on the server by transmitting a series of handshake initiation requests, causing the server to allocate state and potentially to perform expensive cryptographic operations.
2. An attacker can use the server as an amplifier by sending connection initiation messages with a forged source of the victim. The server then sends its next message (in DTLS, a Certificate message, which can be quite large) to the victim machine, thus flooding it.

#### Observed behavior

=====

The DTLS stack does not perform a stateless cookie exchange during session resumption with Session IDs. If the server receives a ClientHello message with a valid Session ID, it responds directly with the ServerHello, ChangeCipherSpec, and Finished messages.

For an attack, an attacker only needs to collect valid Session IDs and then flood the server with ClientHello messages that contain one of the collected parameters.

The main problem here is that an attacker can force the server to send ServerHello, Certificate, ..., ServerHelloDone messages. As a result, the amplification attack is greatly increased.

A properly implemented server should not allow skipping the stateless cookie exchange.

#### Steps to Reproduce

=====

To reproduce the bug, OpenSSL can be used.

1. Launch an server. (We used for our tests the example DTLS server.)

2. Launch the OpenSSL client with:

```
openssl s_client -dtls -connect localhost:4433 -sess_out sess.pem -no_ticket -  
cipher ECDHE-ECDSA-AES256-GCM-SHA384
```

3. Close the connection.

4. Launch the OpenSSL client again with:

```
openssl s_client -dtls -connect localhost:4433 -sess_in sess.pem -no_ticket -cipher  
ECDHE-ECDSA-AES128-GCM-SHA256
```

The most important point here is that the cipher suite in the second handshake must be different from the negotiated in the first handshake. If everything works as planned, Wireshark should show an interaction similar to that in the attached PCAP files.

#### Links

=====

- [1] <https://datatracker.ietf.org/doc/html/rfc6347#section-4.2.1>

Achim Kraus  2022-07-05 06:42:50 EDT

[Comment 1](#)

Thanks for reporting this.

I change my e-mail address end of May and was in vacation.

Is see two issues:

- if the server falls back to an full-handshake, it must use the HelloVerifyRequest

- if not, then there have been some discussions about the in the IETF mailing list, see <https://mailarchive.ietf.org/arch/msg/tls/hlW5oeiZ-ZsrThTG2FgrlFhA-8s/> .

The outcome of that is to configure the behavior according your requirements. See

<https://github.com/eclipse/californium/blob/main/scandium-core/src/main/java/org/eclipse/californium/scandium/config/DtlsConfig.java#L691>

and

<https://github.com/eclipse/californium/blob/main/demo-apps/cf-plugtest-server/src/main/java/org/eclipse/californium/plugin/PlugtestServer.java#L142>

how to change the default.

Achim Kraus  2022-07-05 11:57:15 EDT

[Comment 2](#)

I prepared a PR to fix it.

<https://github.com/eclipse/californium/pull/2039>

maybe you can retest with that.

I also added your test to

<https://github.com/eclipse/californium/tree/main/californium-tests/californium-interopability-tests/src/test/java/org/eclipse/californium/interopability/test/openssl>

but I prefer to publish that test after a 3.6 release.


Currently I plan the 3.6 release for end of next week, 14. July. 2022.

Simon Bernard  2022-07-06 08:26:44 EDT

[Comment 3](#)

Some questions :

1. Just to be sure, If we set DTLS\_VERIFY\_PEERS\_ON\_RESUMPTION\_THRESHOLD to 0, we are no more affected by this ?
2. Do you plan to create a CVE for this ?
3. I see you plan to release a 3.6 version with a fix, do you plan to do a security fix for 2.x or 2.7.x ?

Achim Kraus  2022-07-06 08:37:18 EDT

[Comment 4](#)

> 1. Just to be sure, If we set DTLS\_VERIFY\_PEERS\_ON\_RESUMPTION\_THRESHOLD to 0, we are



Yes, with that, the HelloVerifyRequest is always used.

> 2. Do you plan to create a CVE for this ?

I consider it.

> 3. I see you plan to release a 3.6 version with a fix, do you plan to do a security




The patch uses a new API introduced with 3.x. Therefore it can't be applied to a 2.7.x directly.

To skip the HelloVerifyRequest was introduced when the resumption handshake was intended to be used for frequent address changes. With RFC9146 it gets less

frequently useful. So I think, use a 0 will do it in a 2.7.

Anyway, if you feel, it's important to use it with a 2.7. I will prepare a fix for that as well.

Simon Bernard  2022-07-06 08:49:23 EDT

[Comment 5](#)

> Anyway, if you feel, it's important to use it with a 2.7. I will prepare a fix for




At Sierra we don't use it, so we don't really need it.

But I'm not so comfortable with the idea to have this issue in Leshan 1.x with the default configuration.

Maybe for cf 2.x :

- the default behavior should be set to 0
- if user activate it, a warn about the security issue should be raised ?

(or any other idea ?)


Achim Kraus  2022-07-06 08:53:28 EDT

[Comment 6](#)

> (or any other idea ?)

Then fixing it will be easier ;-).

(Changing the defaults isn't that effective, because old "Californium.properties" will keep the old value.)

Simon Bernard  2022-07-06 09:03:37 EDT

[Comment 7](#)

> (Changing the defaults isn't that effective, because old "Californium.properties"



That's why I propose the warn log in addition.

> Then fixing it will be easier ;-).

If fixing is easy, I'm ok with it too.

Achim Kraus  2022-07-06 13:37:35 EDT

[Comment 8](#)

About Californium 2.7:

A ClientHello, which causes in 3.0 a fallback to a full handshake, causes with the 2.7 an Alert "Illegal Parameter".

So affect only 3.0-3.5.

Achim Kraus  2022-07-07 01:39:28 EDT

[Comment 9](#)

CVE:

I would go for <https://cwe.mitre.org/data/definitions/440.html>

In my opinion, the basic idea, that a resumption handshake doesn't require a HelloVerifyRequest because the reply flight is very small and inexpensive, isn't wrong.

But the check, if a resumption or the fallback to a full-handshake is used instead, was wrong. This weakness then caused the vulnerabilities:

- possible amplification (DDoS others)
- possible CPU exhausting (Dos affected server)

Any other opinions/proposals?

Kai Hudalla  2022-07-07 02:41:53 EDT

[Comment 10](#)

(In reply to Achim Kraus from [comment #9](#))

> CVE:

>

> I would go for <https://cwe.mitre.org/data/definitions/440.html>

>

FMPOV <https://cwe.mitre.org/data/definitions/408.html> would be a more specific fit.

> In my opinion, the basic idea, that a resumption handshake doesn't require a  
> HelloVerifyRequest because the reply flight is very small and inexpensive,  
> isn't wrong.

I agree

>


> But the check, if a resumption or the fallback to a full-handshake is used  
> instead, was wrong. This weakness then caused the vulnerabilities:

>

- > - possible amplification (DDoS others)  
> - possible CPU exhausting (Dos affected server)

>

So, the proposed fix is to ALWAYS send the HelloVerifyRequest when falling back to a full handshake but act according to the value set in DTLS\_VERIFY\_PEERS\_ON\_RESUMPTION\_THRESHOLD otherwise, right? If that is the case then I also agree here :-)

Achim Kraus  2022-07-07 02:57:47 EDT

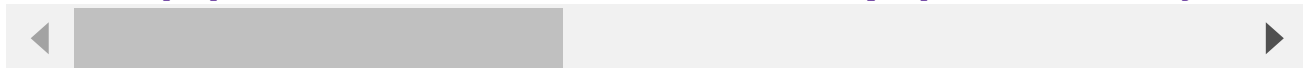
[Comment 11](#)

I also read the 408, my understanding of it is:

1. "do expensive processing"
2. "do the security check"

With the bug would cause no "security check" (HelloVerifyRequest) at all.


> So, the proposed fix is to ALWAYS send the HelloVerifyRequest when falling back to



Yes, that's the proposed fix.

The tricky thing is, that it requires a callback API to implement it. Therefore I introduced "ExtendedResumptionVerifier". The background is, that if the "session-store" (former cache) is a "remote implementation", the processing maybe very specific.

(Just to mention: because hono uses a cluster, I disables the session resumption in hono at all.)


Kai Hudalla  2022-07-07 03:38:24 EDT

[Comment 12](#)

(In reply to Achim Kraus from [comment #11](#))  
> I also read the 408, my understanding of it is:  
>  
> 1. "do expensive processing"  
> 2. "do the security check"  
>  
> With the bug would cause no "security check" (HelloVerifyRequest) at all.  
>

"The software allows an entity to perform a legitimate but expensive operation before authentication or authorization has taken place."

Isn't that exactly what is happening when falling back to a full handshake without sending the ClientVerifyRequest?

Achim Kraus  2022-07-07 04:01:49 EDT

[Comment 13](#)

> "The software allows an entity to perform a legitimate but expensive operation be:



> Isn't that exactly what is happening when falling back to a full handshake without



Yes, if "before" also includes "not to authenticate".  
"Incorrect Behavior Order" points for me to the order.

Right way:

1. check
2. process

Wrong way:

1. process
2. check

But the bug does:

1. process

(The check isn't done at all. Therefore the misbehavior is not based on the wrong order, its based on the missing check.)

Simon Bernard  2022-07-07 04:32:12 EDT

[Comment 14](#)


> About Californium 2.7:

> A ClientHello, which causes in 3.0 a fallback to a full handshake, causes with > 1

> So affect only 3.0-3.5.

This doesn't really respect the RFC as it should fallback to fullhandshake, right ?

Does it impact opened connection/session attached to this Session ID ?

Simon Bernard  2022-07-07 04:40:07 EDT

[Comment 15](#)

My 2 cts, I also think that <https://cwe.mitre.org/data/definitions/408.html> fit better but maybe I don't understand the issue enough.

About :


> But the bug does:  
> 1. process

Could we consider that the check after "process" could be the fullhandshake itself ?

By the way, 408 seems to match better for Technical Impacts too.

440 Impact: Quality Degradation; Varies by Context

408 Impacts : DoS: Amplification; DoS: Crash, Exit, or Restart; DoS: Resource Consumption (CPU); DoS: Resource Consumption (Memory)

Achim Kraus  2022-07-07 05:16:35 EDT

[Comment 16](#)

> This doesn't really respect the RFC as it should fallback to fullhandshake, right

At least I didn't found something in RFC 5246, except the MUST in


> If the  
> session\_id field is not empty (implying a session resumption  
> request), this vector MUST include at least the cipher\_suite from  
> that session.

But that doesn't say "fallback" nor "fail".

Only RFC 6066 states for SNI

> A server that implements this extension MUST NOT accept the request  
> to resume the session if the server\_name extension contains a  
> different name. Instead, it proceeds with a full handshake to  
> establish a new session.

therefore I changed the behavior in 3.0.

Achim Kraus  2022-07-07 05:22:24 EDT

[Comment 17](#)

About CWE 408:


I was a little fixed at "server hello, certificate" and "hello verify request". But if I change to:

- check for full handshake
- use either "hello verify request"
- or "server hello, certificate"

[- (over simplified, ineffective) check for "hello verify request"]

- use either "hello verify request"
- check for full handshake
- "server hello, certificate" or "server hello, server hello done"

Then I guess we can also go for 408.

Simon Bernard  2022-07-07 05:33:12 EDT

[Comment 18](#)

> At least I didn't found something in RFC 5246

I was thinking this part of the spec about :

- > If a Session ID match is not
- > found, the server generates a new session ID, and the TLS client and
- > server perform a full handshake.


(<https://www.rfc-editor.org/rfc/rfc5246#section-7.3>)

Achim Kraus  2022-07-07 05:40:54 EDT

[Comment 19](#)

> If a Session ID match is not found

So that depends on how to interpret "Session ID match". Only the ID? Or also the other parameter? Anyway, though it's a MUST, a violation is always "out of spec". So I don't think, it pays off that much, to change that behavior for 2.7.

Simon Bernard  2022-07-07 06:05:26 EDT

[Comment 20](#)

> So that depends on how to interpret "Session ID match". Only the ID? Or also the c



For me, it's pretty clear "Session ID" means "Session ID"

> Anyway, though it's a MUST, a violation is always "out of spec".

I don't get your point. "How a violation is handled" can be part of a spec.




Anyway here this is not a violation, this is just a possible common use case OR I totally missed something.

The client tries to resume a session, it send a session ID it want to resume. If server lost the session (for any reason, restart, session lifetime) the handshake turn in full handshak.

If I understand you correctly the behavior in 2.7 is to send an Alert instead of fallback on fullhandshake, correct ?

(You didn't answer about : "Does it impact opened connection/session attached to this Session ID ?" don't know if it's on purpose or just oversight)

Achim Kraus  2022-07-07 06:17:16 EDT

[Comment 21](#)

> I don't get your point. "How a violation is handled" can be part of a spec.

Sure, but, as I wrote, I miss that part explaining, what to do, if the "MUST" in

> If the  
> session\_id field is not empty (implying a session resumption  
> request), this vector MUST include at least the cipher\_suite from  
> that session.

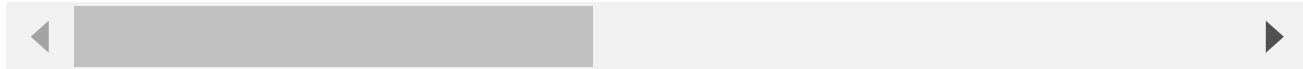
fails.

Basically, this reported bug does exactly this, it doesn't include the cipher\_suite and so breaks the MUST.

Achim Kraus  2022-07-07 06:21:07 EDT

[Comment 22](#)

> Anyway here this is not a violation, this is just a possible common use case OR I  
> The client tries to resume a session, it send a session ID it want to resume. If s



No, only in the cases, where the cipher\_suite or compression\_method doesn't comply to MUST. In the "common case", where the session for the session ID is missing, the 2.7 falls back to the full-handshake. (You may verify that on your own and there are some unit tests to ensure that.)

Achim Kraus  2022-07-07 06:23:50 EDT

[Comment 23](#)

> (You didn't answer about : "Does it impact opened connection/session attached to t



Oversight.

If there is an other session on that "unverified address resuming address", scandium uses a "HelloVerifyRequest".


Achim Kraus  2022-07-07 07:28:17 EDT

[Comment 24](#)

2.7:

I think, if that attack uses a SNI change, the attack will work as well with a 2.7.  
;-).

Unfortunately, there is much more to do for 3.6 (and tinydtls), so I guess, the fix for 2.7 must wait until end of July.

Simon Bernard  2022-07-07 08:28:47 EDT

[Comment 25](#)


> Basically, this reported bug does exactly this, it doesn't include the cipher\_suit



OK I missed this part about different cipher\_suite.

And so I understand now why you say RFC doesn't define the behavior.

I guess in that case an ALERT could be OK.

Achim Kraus  2022-07-11 06:18:19 EDT

[Comment 26](#)

Project name: Eclipse Californium

Versions affected: [2.0.0, 2.7.2] and [3.0.0, 3.5.0]

Common Weakness Enumeration:

- CWE-408: Incorrect Behavior Order: Early Amplification

Summary:

In Eclipse Californium version 2.0.0 to 2.7.2 and 3.0.0-3.5.0 a resumption handshake falls back to a full handshake on a parameter mismatch without using a HelloVerifyRequest. Especially, if used with certificated based cipher suites, that results in message amplification (DDoS other peers) and high CPU load (DoS peer).

Links:

- <https://bugs.eclipse.org/580018>

Achim Kraus  2022-07-11 06:20:51 EDT

[Comment 27](#)

@Simon

@Kai

please check, if that description is OK for you.

For 3.6.0 the PR with the fix is already merged to main.

For 2.7.x the PR #2041 is pending.

If no objection or other preferred schedules, I would like to release

2.7.3 on Wednesday 13. July

3.6.0 on Thursday 14. July

Kai Hudalla  2022-07-11 06:51:34 EDT

[Comment 28](#)

(In reply to Achim Kraus from [comment #26](#))

> Project name: Eclipse Californium


>

> Versions affected: [2.0.0, 2.7.2] and [3.0.0, 3.5.0]

>  
> Common Weakness Enumeration:  
>  
> - CWE-408: Incorrect Behavior Order: Early Amplification  
>  
> Summary:  
>  
> In Eclipse Californium version 2.0.0 to 2.7.2 and 3.0.0-3.5.0 a resumption  
> handshake falls back to a full handshake on a parameter mismatch without  
> using a HelloVerifyRequest. Especially, if used with certificated based  
> cipher suites, that results in message amplification (DDoS other peers) and  
> high CPU load (DoS peer).  
>  
> Links:  
>  
> - <https://bugs.eclipse.org/580018>

Just a small typo


> Especially, if used with certificate based ...

Kai Hudalla  2022-07-11 06:52:02 EDT

[Comment 29](#)

(In reply to Achim Kraus from [comment #27](#))  
> @Simon  
> @Kai  
>  
> please check, if that description is OK for you.  
>  
> For 3.6.0 the PR with the fix is already merged to main.  
> For 2.7.x the PR #2041 is pending.  
>  
> If no objection or other preferred schedules, I would like to release  
>  
> 2.7.3 on Wednesday 13. July  
> 3.6.0 on Thursday 14. July

Sounds good to me. Thanks for handling this vulnerability, Achim :-)


Simon Bernard  2022-07-11 08:50:36 EDT

[Comment 30](#)

LGTM, Except maybe :

I understand that this security issue only concerns users who are using  
DTLS\_VERIFY\_PEERS\_ON\_RESUMPTION\_THRESHOLD > 0.

If you think this is still true, I think this could be a good addition to say  
something about this. (just to let user know if there are concerned and also that  
they can use this workaround instead of upgrade)

Achim Kraus  2022-07-11 08:56:34 EDT

[Comment 31](#)

Project name: Eclipse Californium

Versions affected: [2.0.0, 2.7.2] and [3.0.0, 3.5.0]

Common Weakness Enumeration:


- CWE-408: Incorrect Behavior Order: Early Amplification

## Summary:

In Eclipse Californium version 2.0.0 to 2.7.2 and 3.0.0-3.5.0 a DTLS resumption handshake falls back to a DTLS full handshake on a parameter mismatch without using a HelloVerifyRequest. Especially, if used with certificate based cipher suites, that results in message amplification (DDoS other peers) and high CPU load (DoS own peer). The misbehavior occurs only with DTLS\_VERIFY\_PEERS\_ON\_RESUMPTION\_THRESHOLD values larger than 0.

## Links:

- <https://bugs.eclipse.org/580018>

Simon Bernard  2022-07-11 09:04:50 EDT

[Comment 32](#)

LGTM

Achim Kraus  2022-07-14 08:04:05 EDT


[Comment 33](#)

The fix is now successfully released with 2.7.3 and 3.6.0.

@Simon

Do you plan to provide Leshan releases as well?

If not, then I would ask Wayne (or Mikael) to publish the CVE next week.

Simon Bernard  2022-07-18 05:42:07 EDT

[Comment 34](#)

> Do you plan to provide Leshan releases as well?

I don't know yet because there is nothing to release for Leshan 1.x except update the californium dependency.

I'm still a bit frustrated with the way maven handle version dependency because I feel this is not so clean to have strict defined version dependency for a library...

I remember when I was coding eclipse plugins, the way dependencies was managed was really cleaner. You can say "my plugin depends on given plugin v1.2 or later" so you depends to a version of API and bug fix release is handle by each project.

> If not, then I would ask Wayne (or Mikael) to publish the CVE next week.

Yes please do it, don't wait for us.

Kai Hudalla  2022-07-18 07:37:21 EDT

[Comment 35](#)

(In reply to Simon Bernard from [comment #34](#))

>  
> I'm still a bit frustrated with the way maven handle version dependency  
> because I feel this is not so clean to have strict defined version  
> dependency for a library...  
>  
> I remember when I was coding eclipse plugins, the way dependencies was  
> managed was really cleaner. You can say "my plugin depends on given plugin  
> v1.2 or later" so you depends to a version of API and bug fix release is  
> handle by each project.  
>

you should be able to do that with Maven as well if I am not mistaken:

<https://books.sonatype.com/mvnref-book/reference/pom-relationships-sect-project-dependencies.html#pom-relationships-sect-version-ranges>

Simon Bernard  2022-07-18 08:55:59 EDT

[Comment 36](#)

@Kai, Thx for sharing this. 🙌

I was aware about this but I had the vague memory that it was not advised or even deprecated to use it with Maven. (I don't remember why I thought this 🤔...)

Maybe I should give it a try.

Do you guys already use it ? any feedbacks ? (or maybe reason why you don't use it)

Kai Hudalla  2022-07-18 09:37:32 EDT

[Comment 37](#)

(In reply to Simon Bernard from [comment #36](#))

> @Kai, Thx for sharing this. 🙌

> Maybe I should give it a try.

>

> Do you guys already use it ? any feedbacks ? (or maybe reason why you don't use it)

I am currently not using this but for no particular reason. There is one thing that IMHO needs to be considered: not every third party dependency uses semantic versioning or if it claims to do so, actually implements the semantics correctly and reliably. So, in many cases you will need to test manually with every new (minor) version and if you need to do that anyway, specifying a version range for a dependency doesn't seem beneficial or appropriate at all.

Achim Kraus  2022-07-19 03:29:10 EDT

[Comment 38](#)

@Wayne  
@Mikael

from my side the team agreed on having a CVE created with:

-----  
Project name: Eclipse Californium

Versions affected: [2.0.0, 2.7.2] and [3.0.0, 3.5.0]

Common Weakness Enumeration:

- CWE-408: Incorrect Behavior Order: Early Amplification

Summary:


In Eclipse Californium version 2.0.0 to 2.7.2 and 3.0.0-3.5.0 a DTLS resumption handshake falls back to a DTLS full handshake on a parameter mismatch without using a HelloVerifyRequest. Especially, if used with certificate based cipher suites, that results in message amplification (DDoS other peers) and high CPU load (DoS own peer). The misbehavior occurs only with DTLS\_VERIFY\_PEERS\_ON\_RESUMPTION\_THRESHOLD values larger than 0.

Links:

- <https://bugs.eclipse.org/580018>

-----


If nothing is missing, please create it.

Simon Bernard  2022-07-26 03:57:52 EDT

[Comment 39](#)


I released :

- leshan v1.4.1 with californium v2.7.3
- leshan v2.0.0-M8 with californium v3.6.0

Mikaël Barbero  2022-07-29 07:54:06 EDT

[Comment 40](#)

I'm on it.

Mikaël Barbero  2022-07-29 08:39:29 EDT

[Comment 41](#)

I've submitted <https://github.com/CVEProject/cvelist/pull/6693>, but it's not yet merged as my account is still under verification. I will keep you posted as soon as it's done.

Achim Kraus  2022-07-29 08:43:54 EDT

[Comment 42](#)

Thanks for the update!

Mikaël Barbero  2022-07-29 09:24:57 EDT

[Comment 43](#)

CVE has been merged, you should be all set to socialize about CVE-2022-2576. It will appear soon at <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-2576>