Certificate hostname check is missing when using secure connection

ADACORE SECURITY ADVISORY

Document Id: SEC.AWS-0031

Version 2 - Jul 10, 2024

AdaCore

Title	Certificate hostname check is missing when using secure connection	
Status	Final	
Author	Frederic Leger	
Reviewed by	Pascal Obry, Olivier Ramonat, Johannes Kliemann	

Revision History

Version	Date	Comments
2	Jul 10, 2024	Issue is fixed on wavefront
1	Jun 06, 2024	Initial version

Contents

1. Preface

1.1. Scope

This document is an advisory describing the security impact of `AWS-0031`. The issue is tracked under the ticket number `AWS-0031` in AdaCore's issue tracking database.

This document also presents possible workarounds and mitigations for the issue.

1.2. Distribution

This advisory is made available in confidence to AdaCore customers under embargo until 2024-08-13 so that they can address the issue it describes before public availability.

Thereafter, it will be available to the general public under the terms of the `CC BY-ND 4.0` licence.

1.3. Contact

For questions on this document, please contact AdaCore support at product-security@adacore.com or using the standard reporting procedures if you are an AdaCore customer.

2. Vulnerability

2.1. Affected Products

The vulnerability described in this document was reported for the following product versions:

- *Ada Web Server* - `aws` *20.00* (20191009)

The vulnerability only affects the client side of the API when using TLS. The server side is not affected.

2.2. Severity and Impact

CVSS v3.1 score: 7.4 (AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N/E:U/RL:O/RC:C)

This issue can possibly cause a *Man In The Middle* attack on an *Ada Web Server* client if an attacker uses a crafted certificate with a modified hostname.

A CVE (*Common Vulnerabilities and Exposures*) has been created for that case, and will be referred as `CVE-2024-37015`.

2.3. Detailed Description

Thanks to a great investigation work by Chris Culnane, a vulnerability on the client side of the *Ada Web Server* has been discovered.

There are indeed three issues for TLS enabled *Ada Web Server* clients. One of them does not have a workaround:

1. By default, expired certificate are not checked
2. By default, self-signed certificate are not checked
3. Certificate hostname cannot be verified

The example below allows to check points 1 and 2.

```
with Ada.Exceptions;
with Ada.Text_IO;
with Ada.Strings.Fixed;

with AWS.Response;
with AWS.Client;
with AWS.Net.SSL.Certificate;

procedure Sslconfig_Example is
    use Ada;
    use Ada.Exceptions;
    use AWS.Client;

    Data    : AWS.Response.Data;
    Conn    : AWS.Client.HTTP_Connection;
    SSL_Cfg : AWS.Net.SSL.Config;
begin
    AWS.Net.SSL.Initialize
       (Config => SSL_Cfg,
        Certificate_Filename => "",
        Security_Mode        => AWS.Net.SSL.TLS_Client,
        Exchange_Certificate => True,
        Trusted_Ca_Filename  => "/etc/ssl/certs/ca-certificates.crt");
```

(continues on next page)

(continued from previous page)

```
    AWS.Client.Create
       (Connection => Conn,
        Host       => "https://wrong.host.badssl.com/",
        SSL_Config => SSL_Cfg);

    if AWS.Net.SSL.Certificate.Verified (AWS.Client.Get_Certificate (Conn)) /= True␣
↪then
        raise Constraint_Error with AWS.Net.SSL.Certificate.Status_Message (AWS.
↪Client.Get_Certificate (Conn));
    end if;

    AWS.Client.Get (Conn, Data);
    Text_IO.Put_Line (AWS.Response.Message_Body (Data));
exception
    when Occurrence : others =>
        Text_IO.Put_Line (Exception_Message (Occurrence));
end Sslconfig_Example;
```

Although the certificate of `https://wrong.host.badssl.com/` has an invalid host name, the above example does not throw an exception as it should.

3. Solution

3.1. Workarounds

Checking for certificate validity is achieved through `AWS.Net.SSL.Certificate.Verified (AWS.Client.Get_Certificate (Conn)) /= True`, but the host name check is not part of the verification.

As an incomplete workaround for hostname validation issue, it is possible to filter the accepted certificates on some appropriate values (on the `issuer` for instance).

3.2. Correction

The vulnerability described in this document is corrected in the following product versions:

- *Ada Web Server* - `aws` > 24.2

Starting from build date `20240627`, the *Ada Web Server* component is fixed on `wavefront`.

4. Appendix

4.1. CVSS Score Justification

Metric	Justification
AV:N	Attack Vector through Network.
AC:H	A successful attack depends on conditions beyond the attacker's control. That is, a successful attack cannot be accomplished at will, but requires the attacker to invest in some measurable amount of effort in preparation or execution against the vulnerable component before a successful attack can be expected.
PR:N	No privilege required to perform an attack.
UI:N	No user interaction required to perform an attack.
S:U	The exploited vulnerability can only affect resources managed by the same authority.
C:H	There is total loss of confidentiality, resulting in all resources within the impacted component being divulged to the attacker. Alternatively, access to only some restricted information is obtained, but the disclosed information presents a direct, serious impact.
I:H	There is a total loss of integrity, or a complete loss of protection. For example, the attacker is able to modify any/all files protected by the impacted component. Alternatively, only some files can be modified, but malicious modification would present a direct, serious consequence to the impacted component.
A:N	There is no impact to availability within the impacted component.
E:U	No exploit code is available, or an exploit is entirely theoretical.
RL:U	There is either no solution available or it is impossible to apply.
RC:C	Detailed reports exist, or functional reproduction is possible (functional exploits may provide this). Source code is available to independently verify the assertions of the research, or the author or vendor of the affected code has confirmed the presence of the vulnerability.