[SIERRA-329] OpApp Discovery Download Failure Created: 2019-11-06 Updated: 2020-03-03

Status:	Awaiting customer
Project:	Sierra
Component/s:	None
Affects Version/s:	N/A
Fix Version/s:	#19, #26, #31
Security Level:	Customer

Type:	Support request	**Priority:**	P1
Reporter:	Matthew Barker	**Assignee:**	Michał Drausowski
Resolution:	Unresolved		
Labels:	opapps		
Remaining Estimate:	Not Specified		
Time Spent:	1 week, 7 hours, 2 minutes		
Original Estimate:	Not Specified		

Attachments:	certificates-2020-01-03.tar.gz, certificates-2020-01-08.tar.gz, certificates-2020-02-11.tar.gz, gen_a.pkg, image-2020-01-24-16-34-20-447.png, log-2020-02-11-hdp-discovery-fail.txt, termPkg.crt, termPkgKey.pem, tivu-OPAPP-CA.crt

Issue Links:				
	Defect			
	created	~~TVSDK-47647~~	[OpApp] Create test package for an ap...	Closed
	Dependency			
	depends on	~~TVSDK-47515~~	Investigate problem with formatting o...	Closed
	depends on	~~TVSDK-47610~~	The tivuon! Operator Signing Certific...	Closed
	depends on	~~TVSDK-48255~~	[OpApp] Add OpAppApplicationsDiscover...	Closed
	Reference			
	is related to	TVSDK-49012	[SIERRA][OpApp] Provide reason why th...	New
	is related to	~~SIERRA-381~~	Crash when Vewd Core tries to use ter...	Verified
	is related to	~~TVSDK-48438~~	Crash when Vewd Core tries to use ter...	Closed
	is related to	~~SIERRA-315~~	Setup OpApps/HD+ feature branch	Resolved

Epic Link:	SIERRA-CR4
Severity:	Significant
Build number:	vewd-core-sdk-4.13.5.431.SIERRA.115_CR4_EB5
Last commented:	2020-03-03 07:44
Account:	Vestel-SDK4-Sierra CR4 (OpApp) (9106150)
Cut-off date:	2019-11-01
Total billable H:	0

Description

While evaluating the discovery, download and installation support in the CR4 EB we tried to use the Tivu servers (for expediency, since we don't have the corresponding arrangement for HD+ or local servers set up yet).

This was done by setting the `Operator discovery_methods_mask` to `CreateOpAppWindow::Operator::DiscoveryMethod::DIRECT_URI` and `discovery_direct_uri` to `https://tivuonopapp.tivu-alchemy.net/opapp.aitx`. However, while the XML AIT was downloaded correctly, the package did not appear to be downloaded.

The following trace was seen after the XML AIT was parsed:

```
[68947:69073:1023/112625.214039:85159979280:VERBOSE1:network_delegate.cc(30)] NetworkDelegate::NotifyBeforeURLRequest: https://tivuonopapp.tivu-alchemy.net/prod/2.4.0-beta.52/opapp-encrypted.cms
```

Then, possibly after several minutes, the following trace can be seen:

```
[68947:69062:1023/112625.319202:85160084443:VERBOSE2:opapp_package_downloader.cc(380)] status: failed on package download
```

It is possible to download the package manually (e.g. using `wget`).

The `Operator` object added to `CreateOpAppWindow::supported_operators` is as follows:

```
    CreateOpAppWindow::Operator tivuon_operator;
    tivuon_operator.org_id = 0x360;
    tivuon_operator.name = "tivuon! Trust Authority";
    tivuon_operator.discovery_methods_mask = CreateOpAppWindow::Operator::DiscoveryMethod::DIRECT_URI;
    tivuon_operator.discovery_direct_uri = "https://tivuonopapp.tivu-alchemy.net/opapp.aitx";
    //tivuon_operator.certificate read from file (see attached)
```

The test Tivu Operator certificate is attached. The corresponding terminal packaging certificate and key passed to `CreateOpAppWindow` are also attached.

A similar attempt was made to download the same (Tivu) package file from a local Apache web server. The same behaviour was seen.
It appears that at least some data is being downloaded from the server, but either it does not complete or there is a problem processing the file but no trace is output indicating why.

Comments

Comment by Matthew Barker [2019-11-12]

We have tried to use binary DER data instead of PEM for certificates and keys, but that made no difference.

It was noticed that the CN attribute of the Tivu certificate does not contain the `organisation_id`, so this might be a factor.

More trace to indicate the current status or if some check has failed would be helpful in diagnosing the problem.

Comment by Filip Brzezniak (Inactive) [2019-11-14 - Visible by: Developers]

Let's try to prepare a dingo tests that replicate the scenario described in this ticket.
Dominik Cięciara, as this is the first time we support client this way, I'll handle this.

Comment by Filip Brzezniak (Inactive) [2019-11-14 - Visible by: Developers]

I created simple test that starts OpApp window with preinstalled app encrypted with provided Terminal Certificate.
The OpApp window fails on importing the Terminal Certificate Private Key:

```
[27874:27888:1114/190238.022701:26468351079:ERROR:opapp_package_decryptor.cc(231)] Error loading terminal packaging private key: security library: improperly formatted DER-encoded message.
```

I'm investigating if the key is properly formatted.

Comment by Dominik Cięciara [2019-11-15 - Visible by: Developers]

Filip Brzezniak Please raise a TVSDK issue for it - thanks.

Comment by Filip Brzezniak (Inactive) [2019-11-15 - Visible by: Developers]

Dominik Cięciara, ticket created and linked: ~~TVSDK-47515~~.

Comment by Szymon Jastrzębski [2019-11-15]

Dear Matthew Barker
we are encountering a problem during importing the Terminal Certificate Private Key:

```
[27874:27888:1114/190238.022701:26468351079:ERROR:opapp_package_decryptor.cc(231)] Error loading terminal packaging private key: security library: improperly formatted DER-encoded message.
```

Can you share the command the key was created with? It might help us debugging the issue.
Regards,
Szymon

Comment by Matthew Barker [2019-11-15]

Hi Szymon,
I didn't generate the key personally, but I understand the key was generated using the following command:

```
openssl genrsa -out termPkgKey.pem 2048
```

(If it matters, I'm not sure what version of `openssl` was used, but I would guess it is: OpenSSL 1.1.1 11 Sep 2018.)
When it was tested using a binary key the following command was used:

```
openssl enc -a -d < termPkgKey.pem > termPkgKey.der
```

Comment by Filip Brzezniak (Inactive) [2019-11-15]

Hi Matthew Barker,

we do generate the Terminal Certificate/Key pair in following way:

```
openssl req -newkey rsa:2048 -nodes -x509 -days 12031 -subj "/O={some_organisation_name}/CN={some_common_name}" -out ./new_cert -keyout new_cert_key
```

They seem to work fine for us.

For OpApps usage we need the cert and key pair, but the provided method seems to yield only the later. Could you please double check it?

Thanks and Best Regards,
Filip

Comment by Matthew Barker [2019-11-15]

Szymon only asked how we generated the key.

The complete process that was originally used to generate the certificate and key is as follows:

```
# Create Root CA Key
openssl genrsa -out termPkgRootCAKey.pem 2048
```

```
# Create Root CA Certificate
openssl req -x509 -new -nodes -key termPkgRootCAKey.pem -sha256 -days 365 -out termPkgRootCA.crt -subj "/CN=Vestel MB230/O=Vestel"

# Create Certificate Signing Key
openssl genrsa -out termPkgKey.pem 2048

# Create Certificate Signing Request
openssl req -new -sha256 -key termPkgKey.pem -out termPkg.csr -subj "/CN=Vestel MB230/O=Vestel"

# Create Terminal Packaging Certificate
openssl x509 -req -extensions v3_req -extfile extensions-x509.cnf -in termPkg.csr -CA termPkgRootCA.crt -CAkey termPkgRootCAKey.pem -CAcreateserial -days 365 -sha256 -out termPkg.crt
```


Where extensions-x509.cnf contains:

```
[ usr_cert ]

extendedKeyUsage = serverAuth, clientAuth, codeSigning
basicConstraints = critical, CA:FALSE

[ v3_req ]

basicConstraints = critical, CA:FALSE
keyUsage = critical, keyEncipherment
authorityKeyIdentifier=keyid:always,issuer:always
```

Note that we were not involved directly in the creation of tivu-OPAPP-CA.crt or the signed and encrypted package.

The package is extracted as follows (as per TS 103 606, 11.3.4.4, 5):

```
# Decrypt the CMS file.
openssl cms -decrypt -binary -inform DER -in opapp-encrypted.cms -keyform PEM -inkey ../termPkgKey.pem -recip ../termPkg.crt -out opapp-decrypted.cms

# Authenticate and create the zip file.
openssl cms -verify -binary -in opapp-decrypted.cms -inform DER -out OpApp.zip -CAfile tivu-OPAPP-CA.crt
```

This process has been proven to work manually.

Comment by Filip Brzezniak (Inactive) [2019-11-18]

Matthew Barker,

thank you for detailed description. I'll investigate the issue further on our side and let you know the results.

Best Regards,
Filip

Comment by Filip Brzezniak (Inactive) [2019-11-18 - Visible by: Developers]

I managed to replicate the scenario described by Matthew. There are three separate issues that were blocking it. I'll describe it in details tomorrow morning.

Comment by Łukasz Romanowski [2019-11-20 - Visible by: Developers]

Filip Brzezniak, can you sum up progress on this for the customer?

Comment by Filip Brzezniak (Inactive) [2019-11-20]

Hi Matthew Barker,

While investigating the issue I found following problems that make the OpApp application discovery fail:

1. The download manager used by Vewd Core to fetch cms package was wrongly configured to limit maximal size of the downloaded item to 1MB. Because of that, it rejected the package in test scenario, as it is of 1.5MB size. The fix for it is currently being prepared by our product team.
2. The current version of Vewd Core supports only the Terminal Packaging Certificate keys in PKCS #8 format. After converting "termPkgKey.pem" to PKCS #8 format with following command, the key was properly imported by Vewd Core.

   ```
   openssl pkcs8 -topk8 -nocrypt -in in.pem -inform PEM -out out.pem -outform PEM
   ```

   Notes:

   - command that generates RSA key in plain format (PKCS #1):

     ```
     openssl genrsa -out genrsa.key 2048
     ```

   - command that generates RSA key in PKCS #8 format:

     ```
     openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -out genpkey.key
     ```

3. The `tivu-OPAPP-CA.crt` is rejected by internal nss library used by Vewd Core to decrypt/verify the cms package. I'm still investigating if the issue is in certificate itself or some wrong configuration of the library. I'll let you know about investigation results in next comments.

Best regards,
Filip

Comment by Filip Brzezniak (Inactive) [2019-11-22]

Hi Matthew Barker,

I followed up on problem number 3 from my previous comment. The conclusion is that the nss implementation used by Vewd Core fails on cms package's signature verification:

```
Verifying certificate of OpApp package opapp-decrypted.cms failed: Certificate invalid: Certificate key usage inadequate for attempted operation.
```

As you've written in your comment, openssl succeeds with it.

I checked the certificate details and found out, that it doesn't fully comply with OpApp specification requirements, defined in p.11.3.2. The certificate is as follows:

```
    Data:
        Version: 3 (0x2)
        Serial Number: 1506803239322248104 9 (0xd11c67a0b61ae499)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=IT, O=tivuon! Trust Authority, OU=HbbTV Operator Applications, CN=tivu Signing Root CA
        Validity
            Not Before: Mar  3 10:45:52 2019 GMT
            Not After : Feb 23 10:45:52 2049 GMT
        Subject: C=IT, O=tivuon! Trust Authority, OU=HbbTV Operator Applications, CN=tivu Signing Root CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (4096 bit)
                Modulus:
                    00:c7:22:f6:c8:fe:63:ac:1e:db:f5:28:8d:ae:fb:
                    58:5a:33:86:b5:c8:3d:b1:9f:fd:f0:6e:f5:15:62:
                    43:46:a4:89:22:26:3f:fc:f8:79:32:5b:58:fe:04:
                    60:7e:1e:0b:6f:93:ab:1c:b2:0e:50:13:b3:63:71:
                    03:34:11:e1:d2:6f:b6:27:f8:a3:d8:19:01:a5:d0:
                    f7:92:6e:63:46:c2:24:aa:66:d0:3b:a0:66:5b:94:
```

But the OpApp spec requires the critical `X509v3 extensions` to be:

```
[ v3_req ]
basicConstraints        = critical, CA:FALSE
keyUsage                = critical, digitalSignature
```

I generated the certificates in out testing framework following above requirements, but they still were rejected by nss lib with the same error as in case of tivuon certificate.

When investigating further I found out, that to pass on nss verification, the `keyCertSign` needs to be added to `keyUsage`. The accepted certificate looks as follows:

```
    Data:
        Version: 3 (0x2)
        Serial Number: 1348228476147095351 5 (0xbb1ab28ef023682b)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN=1, O=Vewd Operator
        Validity
            Not Before: Nov 22 08:41:00 2019 GMT
            Not After : Nov 15 08:41:00 2044 GMT
        Subject: CN=1, O=Vewd Operator
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:bd:44:03:dd:16:fd:59:81:af:5c:6b:c0:35:f8:
                    0d:98:a0:49:30:1d:7d:37:86:f2:20:af:b4:3c:05:
                    9b:d2:ea:87:e0:e5:62:2f:ac:d7:a0:3e:65:45:67:
                    4c:21:28:cb:b2:e1:34:b6:b2:9a:3a:dd:f6:ce:22:
                    dd:a0:d2:d0:eb:fc:f3:8d:89:1d:83:93:ec:fb:a7:
                    c0:5c:3f:6c:26:07:02:c7:7a:e7:a3:ea:16:ad:90:
```

As the certificate is not fully compliant with the OpApp spec, we need to address it somehow. In my opinion a solution could be either:

- update of specification requirements (I'm not sure if it is acceptable),
- fix of nss lib (it still needs to be confirmed if current behavior is wrong).

I'm consulting now the possible solutions with our core team. I will let you know the result as soon as I get some reply.

Best Regards,
Filip

Comment by Matthew Barker [2019-11-22]

Thank you for the updates.

Depending on the outcome of your investigation (or even before it is complete), it may be helpful if you could provide commands for generating the necessary terminal and operator keys and certificates (and possibly for encrypting and signing the package) that are known to work
we could test with.

Comment by Filip Brzezniak (Inactive) [2019-11-22]

Matthew Barker, I talked to Michał Drausowski and he is preparing some sample package for you. He'll provide it on ~~SIERRA-315~~.

Comment by Filip Brzezniak (Inactive) [2019-11-28]

Matthew Barker,

while investigating further I found a bug in the way Vewd Core configures nss lib for verifying OpApp package signature. The fix for the issue it is currently being reviewed by Vewd product team. When it is accepted, it will be ported to Sierra branch.

I also further investigated potential certificate problems and found out that the tivuon! root CA is most likely correct. The problem is rather in Operator Signing Certificate which is signed by this root CA.

I checked `nss` source code path triggered in tested OpApp scenario. I found out that `nss` not only requires proper `keyUsage = digitalSignature` in `X.509 extensions` for signing certificate (as defined in p.11.3.2 of ETSI TS 103 606 V1.1.1), but additionally it requires `extendedKeyUsage = codeSigning` extension for proper detection of certificate type. Only the signing certificates containing this combination of usage and type in `X.509 extensions` are accepted for object signing usage by nss implementation.
For more details, please refer to:

- https://developer.mozilla.org/en-US/docs/Mozilla/Projects/NSS/NSS_Tech_Notes/nss_tech_note3,
- https://www.openssl.org/docs/manmaster/man5/x509v3_config.html#Extended-Key-Usage.

To confirm above findings, I experimented in our testing environment with signing certificates both with and without the `extendedKeyUsage` extension. Only when it was set, the OpApp package was successfully verified. In other cases the following error was logged to console:

```
Certificate invalid: Certificate type not approved for application.
```

The `X.509 extensions` section from config file used to sign working Operator Signing Certificate is as follows:

```
[ signing_cert_extensions ]
authorityKeyIdentifier  = keyid,issuer
basicConstraints        = critical, CA:false
extendedKeyUsage        = codeSigning
keyUsage                = critical, digitalSignature
```

Would it be possible to pass it to tivuon! and ask for re-signing of OpApp package, so we could verify above findings in live environment?

Best Regards,
Filip

Comment by Filip Brzezniak (Inactive) [2019-11-28 - Visible by: Developers]

Marking ~~TVSDK-47610~~ as blocker for current issue, since it introduces necessary fix for nss configuration (https://critic.corp.vewd.com/showcommit?review=123619&filter=files&file=1702013)

Comment by Matthew Barker [2019-11-28]

The example package provided by Michal does not explicitly define any of the X.509 extensions you mention for the operator signing certificate, yet the browser is able to verify the package. Why is this?

To my knowledge Vestel and Tivu are not working on an OpApp product at the moment, so it is unlikely that such a request would be satisfied. It is surprising that the server still responds at all, and we should probably not be using it.

Instead, we should be working with SES to configure their servers for HD+ in the same way. This has been requested, but I am not aware of any progress towards it.

Comment by Filip Brzezniak (Inactive) [2019-11-29]

Matthew Barker,

unfortunately the package provided in ~~SIERRA-315~~ doesn't fully comply with OpApp specification requirements:

- the OpApp package is signed directly by self-signed Root CA instead of Operator Signing Certificate,
- the Root CA doesn't define any `keyUsage` constraints, which means that it is treated as applicable for any usage.

Also as mentioned in my previous comment, the version of Vewd Core you are currently working on doesn't configure nss library properly. Because of that it is over-permissive in terms of the certificate usage/type it accepts for OpApp package signature verification.

After providing you a build with the patch for stricter verification of OpApp package (as required by the OpApp spec), we'll also release a new version of testing package. This way you'll be able to continue integration on your side (hopefully) seamlessly and at the same time the Vewd Core, you use underneath, will be more aligned with specification's requirements.

Thank you for clarifying the Vestel's/Tivu status. Let's wait with more "live" testing until SES prepare their servers then.
In the mean time we can base on the test package provided by us, if you are ok with such an approach.

Thank you and Best Regards,
Filip

Comment by Matthew Barker [2020-01-03]

Would someone provide an update on the progress of this issue?

We tried to use Filip's suggestions regarding X.509 extensions to generate certificates for internal testing (see certificates-2020-01-03.tar.gz), but we get the following error during discovery:

```
[77317:77454:0103/150409.021704:106142833486:ERROR:opapp_package_decryptor.cc(335)] Verifying certificate of OpApp package .org.chromium.Chromium.hhWkzY failed: Certificate invalid: Certificate key usage inadequate for attempted operation.
[77317:77433:0103/150409.022431:106142834210:VERBOSE2:opapp_package_downloader.cc(368)] status: failed on package validation
```

Can you advise what the problem is?

Also, SES are now in a position to accept a test terminal packaging certificate from us and provide an operator signing certificate to allow OpApp discovery from their server. However, I'm not sure what I should tell them regarding how to generate the operator certificate that would work with the latest Vewd SDK.

Should we wait until this issue has been resolved before sending SES our terminal packaging certificate?

Comment by Bliss Bot [2020-01-03 - Visible by: Developers]

Build	tvsdk_4.13_sierra_dev #19
Changes	SIERRA-329: ~~TVSDK-47610~~: Require proper certUsage from certificates used to sign/verify OpApp package's (cms) signature. https://git.corp.vewd.com/tvsdk/opera/work/commit/?h=tvsdk_4.13_sierra_dev&id=496d2439706938 36a9701832e9fa8ed427495f2d

Comment by Alihan Dikici [2020-01-08]

Łukasz Romanowski, can we have an update?
This is critical.

Comment by Filip Brzezniak (Inactive) [2020-01-08]

Alihan Dikici, I'll check provided certificates and provide feedback today. Thank you for your patience.

Comment by Filip Brzezniak (Inactive) [2020-01-08]

Matthew Barker, from what I see in provided package (certificates-2020-01-03.tar.gz), it doesn't include the CA certificate (Root or Intermediate) that could be used to verify the package signed with operator signing certificate (operator.crt in the package).
Could you provide it as well? It seems to be the only missing thing to have complete test setup on my side.

Comment by Matthew Barker [2020-01-08]

Hi Filip,
Sorry. I've uploaded a new package (certificates-2020-01-08.tar.gz), that also contains the root CA certificate for verification.
The error message in my previous comment was due to the use of the operator signing certificate by mistake, but if I use the root CA certificate, discovery does not complete but I get no error message. Again, decryption and verification manually (using openssl) works.

Comment by Filip Brzezniak (Inactive) [2020-01-08]

While using new certificates I got following error:

```
[5395:5449:0108/112554.135406:78406974563:ERROR:opapp_package_decryptor.cc(250)] Error opening encrypted opapp CMS package '.org.chromium.Chromium.FnruL3':Cannot decrypt: you are not a recipient, or matching certificate and private key not found.
```

The nss lib returns this error when terminal certificate is not present on cms package's recipient list.

Since you don't see this error, is it possible that I get different package version?

I use package url: "https://tivuonopapp.tivu-alchemy.net/opapp.aitx". Are you using the same address?
Do I need some specific UA string to be properly recognized by the server?

Comment by Matthew Barker [2020-01-08]

Sorry again, I had assumed simply analysing the certificates would be enough for you to diagnose any issues. These certificates were generated internally for use with our HbbTV test harness on a local server, not for Tivu or SES servers, and I should have made this clear.

Would it be sufficient for us to simply provide a package file (gen_a.pkg) corresponding to these certificates or, for example, should we provide the complete set of certificates we generated to allow you to create your own packages (I've added the operator key to the certificate package)?

As I mentioned before, we could start the process of working with SES to allow us to work with a live server, but we need to provide guidance on how to generate packages that will work with the Vewd SDK or we may have the same problem as with the Tivuon OpApp. I wanted to be sure we had accurate information by using internally generated certificates before starting this process to avoid too many iterations with SES.

Do you think providing the following information would be sufficient?

```
The X.509 extensions section from config file used to sign working Operator Signing Certificate is as follows:

[ signing_cert_extensions ]
authorityKeyIdentifier  = keyid,issuer
basicConstraints        = critical, CA:false
extendedKeyUsage        = codeSigning
keyUsage                = critical, digitalSignature
```

Comment by Filip Brzezniak (Inactive) [2020-01-08]

Matthew Barker, I checked the certificates and they seemed mostly fine to me. Though as I'm not an expert in this area, I preferred to confirm this also by running some basic tests. As you wrote, it usually helps to avoid some extra iterations.

Now with provided package (gen_a.pkg), I managed to fully setup my test environment.

I confirm that the package can be decrypted and verified with the certificates from certificates-2020-01-08.tar.gz. So using the `signing_cert_extensions` from your previous comment should be enough to produce operator signing certificate working with Vewd Core.

Please remember to also mention about requirements from Table 22 in ETSI TS 103 606 V1.1.1. Especially important is proper setting of `'O='` and `'CN='` in certificate's `Subject` field, as they are checked by Vewd Core.

Comment by Matthew Barker [2020-01-08]

Thank you, I will initiate the process with SES.

However, I have so far been unable to get Vewd discovery to complete successfully in a test environment with the certificates and package I provided. The browser downloads the XML AIT and package file (based on the Chromium temporary file in the storage path), but no discovery events or error traces are generated, in fact the browser crashes in `libomi.so` with `SIGTRAP`. This was tested with release DEV.19.

Comment by Filip Brzezniak (Inactive) [2020-01-08]

I did my previous check by pre-installing provided package on device.

I'll try to replicate whole scenario tomorrow. Is the xml ait that you use for discovery exactly the same as in the gen_a.pkg package?

Could you also provide log from crashed test?

Comment by Filip Brzezniak (Inactive) [2020-01-09]

Matthew Barker, I have one more question regarding described crash: is the crashing application exactly the same as the one from gen_a.pkg package? From what I see, it doesn't do much. If not, could you provide the full source code of the application?

Comment by Matthew Barker [2020-01-09]

Hi Filip,
I tried simply pre-installing the package and a crash occurred when I tried to start the OpApp. This appears to be the same crash that occurs during discovery. I also managed to reproduce the crash using the Vewd SDK reference `hbbtv` application.

The critical factor appears to be the presence of a certificate in the NSS database (e.g. `$BROWSER_HOME/.pki/nssdb`) added for TLS authentication of the XML AIT download. If I reset the NSS database the application runs from a pre-installed package as expected without cra

For example, you could try to manually install `rootCA.crt` from the certificates package into the NSS database and re-run your test with the pre-installed application. I used:

```
certutil -d sql:<nss-db-dir> -A -t C -n cabot -i <certificate>
```

Perhaps this crash should be raised as a separate issue.

Comment by Filip Brzezniak (Inactive) [2020-01-09]

Hi Matthew,
thanks for the description, following it I was able to get the crash.

It looks like we fail when asserting that certificate used for package validation is not permanent (we unconditionaly add it as temporary to the nss db, and assert that is not permanent). Since in test procedure the certificate is first manually added to the database as permanent, the assert fails.

I'll file separate ticket for it tomorrow and prepare fix.

Comment by Filip Brzezniak (Inactive) [2020-01-10]

Matthew Barker, I created ~~SIERRA-381~~ for the crash described in your previous comment. Please update it, if you find some information missing.

Comment by Alihan Dikici [2020-01-15]

Filip Brzezniak, if I am not wrong, this ticket is waiting ~~SIERRA-381~~ for verification, right?
If this is the case, set this to 'pending'

Comment by Filip Brzezniak (Inactive) [2020-01-15 - Visible by: Developers]

Łukasz Romanowski, as commented by Alihan this ticket depends on ~~SIERRA-381~~. Could you update its state accordingly? As I don't know what is expected in such a case.

Comment by Michał Drausowski [2020-01-15]

DEV#21:
https://customers.vewd.com/share/page/site/sierra/documentlibrary#filter=path%7CBuilds/DEV/4.13.5.431.SIERRA_DEV.21

Comment by Matthew Barker [2020-01-16]

DEV.21 fixes the crash problem.

Presumably there is still work for Vewd to do to configure NSS library properly and provide another example package, but it seems this is no longer a critical priority issue. We will see how it goes with SES for the HD+ OpApp.

Comment by Matthew Barker [2020-01-24]

With the help of SES I have managed to manually download, decrypt and verify the HD+ OpApp from their servers using command line tools, but I have so far failed to do this with the browser.

The point it currently fails is when processing the XML AIT:

```
[38221:38337:0124/145626.025510:194812405327:INFO:xml_ait_parser.cc(71)] Skipping unsupported AIT Application
[38221:38337:0124/145626.025589:194812405401:VERBOSE2:opapp_package_downloader.cc(368)] status: failed on XML AIT validation
```

I have narrowed this down to the `applicationDescriptor version` field. Oddly, it seems the version in the XML AIT has to be less than 100 or an error is generated (it is 185 in the version served). I can't modify a local copy of the XML AIT since it needs to match the one in the package and I don't have the means to create a package myself, so I can't progress without the version being changed or support from Vewd.

If I use a locally-modified version of the XML AIT, the discovery process appears to stall after tracing out the XML AIT, i.e. there are no error messages (e.g. about mismatching XML AIT) or discovery events generated.

Comment by Dominik Cięciara [2020-01-27 - Visible by: Developers]

Filip Brzezniak Are there any corresponding TVSDK tickets concerting the problems in this issue?

Comment by Filip Brzezniak (Inactive) [2020-01-27]

Matthew Barker, we based our implementation on `mhp::Version` definition from p.C.1.2.24 ETSI TS 102 034 (link):

> C.1.2.24 Version
> <xsd:simpleType name="Version">
> <xsd:restriction base="xsd:string">
> <xsd:pattern value="[0-9a-fA-F]
> Unknown macro: {2}
>
> "/>
> </xsd:restriction>
> </xsd:simpleType>
> A number conveying the version of a table or record. This value will increase with changes to the table or record, modulo 256. This value shall be in hexadecimal.

The format of xml AIT is inherited in following order:

- p.6.1.5.1 ETSI TS 103 606:
  > The terminal shall parse the XML AIT according to the requirements defined in clause 7.2.3.2 of ETSI TS 102 796
- p.7.2.3.2 ETSI TS 102 796:
  > Broadcast-independent applications shall be identified either by the URL of the first page of the application or by the URL of a XML AIT as defined in clause 5.4 of ETSI TS 102 809
- p5.4 ETSI TS 102 809:
  > This clause defines an XML encoding for the AIT in addition to the MPEG-2 table and section based encoding defined in clause 5.3 of the present document. Since the intended use for this XML encoding is in conjunction with the SD&S defined in ETSI TS 102 034 [6], this encoding follows the same format and re-uses already defined elements and types.

Considering above, the `100` (hex) and all higher values overflow allowed range, thus the parsing fails.

> If I use a locally-modified version of the XML AIT, the discovery process appears to stall after tracing out the XML AIT, i.e. there are no error messages (e.g. about mismatching XML AIT) or discovery events generated.

We fixed similar case only recently in Vewd Core by adding `OpAppApplicationsNotDiscovered` event to OMI API. I'll ask my colleges to backport it to yours branch.

Sorry for inconvenience and Best Regards,
Filip

Comment by Filip Brzezniak (Inactive) [2020-01-27 - Visible by: Developers]

Michał Drausowski/Szymon Jastrzębski, Matthew is probably missing patch from ~~TVSDK-48255~~. Could you please make sure he gets it in next released build?

Comment by Dominik Cięciara [2020-01-27 - Visible by: Developers]

Thanks Filip Brzezniak. Michał Drausowski please proceed with porting immediately.

Comment by Matthew Barker [2020-01-28]

> we based our implementation on mhp::Version definition from p.C.1.2.24 ETSI TS 102 034

I have passed on your feedback for correction of their XML AIT.

> We fixed similar case only recently in Vewd Core by adding OpAppApplicationsNotDiscovered event to OMI API. I'll ask my colleges to backport it to yours branch.

A new event to report discovery failures would be useful. Will it indicate the specific error that occurred (e.g. mismatching XML AIT)?
We need some feedback (trace or event) to indicate why the discovery failed to help us to find the cause of the problem.

Comment by Szymon Jastrzębski [2020-01-28]

Matthew Barker
The `OpAppApplicationsNotDiscovered` event is quite simple event:

```
/**
  @brief Informs integration about internal discovery failure.
  Only emitted when internal discovery is enabled.
  This event is emitted when internal discovery was triggered by
  UpdateOpApps OMI message and there isn't any new package.
*/
event OpAppApplicationsNotDiscovered {
  /** The identifier of the window to which this event relates. */
  handle window_id;
}
```

Comment by Bliss Bot [2020-01-28 - Visible by: Developers]

Build	tvsdk_4.13_sierra_dev #26
Changes	SIERRA-329: ~~TVSDK-48255~~: Add OpAppApplicationsNotDiscovered event (reland) https://git.corp.vewd.com/tvsdk/opera/work/commit/?h=tvsdk_4.13_sierra_dev&id=67afd533fd38dfac75749db6f272b9a11ce66792 SIERRA-329: ~~TVSDK-48080~~: Get information about OpApp discovery package download error https://git.corp.vewd.com/tvsdk/opera/work/commit/?h=tvsdk_4.13_sierra_dev&id=df54fb4b73249151e930d6afc3861d0fc0c8bbad

Comment by Szymon Jastrzębski [2020-01-28]

Dear Matthew Barker
I've uploaded build containing the backported event. Please test using 26th build: https://customers.vewd.com/share/page/site/sierra/documentlibrary#filter=path%7C/Builds/DEV/4.13.5.431.SIERRA_DEV.26&page=1
Regards,
Szymon

Comment by Matthew Barker [2020-02-04]

We have evaluated the backported event and it may be useful. It does not address the fact that there is often no indication (e.g. in trace) as to why discovery did not succeed.

Regarding HD+, SES appear to accept that the value of the version field in the XML AIT needs to change and we are awaiting and update to their servers before we can retest (since we cannot update the OpApp package ourselves).

Comment by Filip Brzezniak (Inactive) [2020-02-04]

Matthew Barker,

regarding indication in `OpAppApplicationsNotDiscovered` event: it is not easy to provide considering our current architecture of OpApps. Though we also think it would be useful and we are planning to address it in future releases.

Sorry for inconvenience and Best Regards,
Filip

Comment by Szymon Jastrzębski [2020-02-10 - Visible by: Developers]

Filip Brzezniak Paweł Witkowski Have we raised a ticket for the improvement suggested by Matthew Barker?

Comment by Szymon Jastrzębski [2020-02-10]

Matthew Barker
Can we resolve the issue?

Comment by Paweł Witkowski [2020-02-10 - Visible by: Developers]

There's no task for it. Please, create one.

Comment by Matthew Barker [2020-02-11]

SES have updated their servers to generate a valid version value in the XML AIT.

As before, I have managed to manually download, decrypt and verify the HD+ OpApp from their servers using command line tools, but I have failed to do this with the browser.

Judging from browser trace, the discovery process appears to correctly process the XML AIT and attempts to download the package, but the process fails after several minutes with no trace indicating the cause of the problem and the following:

```
[117789:117905:0211/153733.987420:63160591249 2:VERBOSE2:opapp_package_downloader.cc(368)] status: failed on package download
00:08:05.660 verb> opera/engine: Received event: 0xfa673e1cf46b3254 OpAppApplicationsNotDiscovered:67
```

More details:

Attached are the certificates used certificates-2020-02-11.tar.gz and a log file from a failed run log-2020-02-11-hdp-discovery-fail.txt.

The Operator definition for `CreateOpAppWindow` is as follows:

```
org_id = 0x21
name = "SES-ASTRA"
discovery_methods_mask = FQDN_IN_BROADCAST_NIT_OR_BAT
certificate  = [contents of "SES-ASTRA-CA.der"]
```

The FQDN passed via `NotifyNewNitOrBat` is `dns:hd-plus-cloud.de`.

Vewd will need to provide their public IP address to SES to allow whitelisting to access the development (QA) build from the servers.

Comment by Szymon Jastrzębski [2020-02-12 - Visible by: Developers]

branch used for sending EB:
https://git.corp.vewd.com/tvsdk/opera/work/log/?h=SIERRA-329/EB/2

TODO: commit should be merged into the branch after backporting all the missing features.

Comment by Szymon Jastrzębski [2020-02-12]

Matthew Barker
Would please reproduce the issue again with below EB? Thank you
https://customers.vewd.com/share/page/site/sierra/documentlibrary#filter=path%7C/Builds/Engineering/SIERRA-329&page=1

Comment by Matthew Barker [2020-02-12]

Testing with the EB (with extra verbose logging) shows the following error when trying to download the HD+ OpApp package:

```
[127935:128051:0212/141608.791286:713120716357:VERBOSE2:opapp_downloader.cc(383)] MIME type should be 'application/vnd.hbbtv.opapp.pkg' but is ''
```

This suggests that the server is not correctly setting the `Content-Type` header in the response.
I will provide feedback to SES.

Comment by Matthew Barker [2020-02-14]

SES have updated their servers to correctly report the `Content-Type`. However, validation fails (using DEV.30 EB):

```
[73610:73748:0214/092358.655076:147598774380:ERROR:opapp_package_decryptor.cc(342)] Verifying certificate of OpApp package .org.chromium.Chromium.5EglEx failed: Certificate invalid: Certificate type not approved for application.
[73610:73726:0214/092358.655852:147598775153:WARNING:opapp_discovery_manager.cc(1077)] Downloaded package is invalid. app_id=46 org_id=33 path=buffer/opapp/storage/.org.chromium.Chromium.5EglEx
[73610:73726:0214/092358.655978:147598775278:VERBOSE2:opapp_package_downloader.cc(368)] status: failed on package validation
```

Are you able to determine the cause with the information provided (certificate package and Operator configuration)? Have you been able to reproduce the problem?

Comment by Łukasz Romanowski [2020-02-17]

HI Matthew Barker, Alihan Dikici, can you provide a step by step procedure in separate issue to replicate this on MB230? I have asked Tobi for a whitelist of our IPs so once we have this we should be able to reproduce.

Comment by Alihan Dikici [2020-02-17]

Hi Łukasz Romanowski, testtool commands are finalized yet and working on it to complete by workshop.

Comment by Łukasz Romanowski [2020-02-17]

Matthew Barker, can you check if `Certificate type not approved for application.` is in fact the same as described in this comment? Looking at the latest certificates attached (certificates-2020-02-11.tar.gz) it may well be the issue (missing `extendedKeyUsage = codeSigning`).

Comment by Andrew Dominy [2020-02-18]

Hi Łukasz,
Matthew is on holiday this week. I'll be covering in his absence.
The error looks to be the same as mentioned in the comment. I've tested locally using our test harness Operator signing certificate. A package signed with this certificate is successfully verified with `extendedKeyUsage = codeSigning` included and fails with the same error with the entry removed.
I will mention to SES that the `extendedKeyUsage` entry is required.

Comment by Bliss Bot [2020-02-18 - Visible by: Developers]

Build	tvsdk_4.13_sierra_dev #31
Changes	SIERRA-329: Rename D[V]LOGs to [V]LOGs for easier development https://git.corp.vewd.com/tvsdk/opera/work/commit/?h=tvsdk_4.13_sierra_dev&id=8e793badd782ba2bde77aac5f255e5125beed555

Comment by Alihan Dikici [2020-02-24]

Hi Łukasz Romanowski, Michał Drausowski as discussed in the Workshop , Vewd to investigate the issue and return with fix by mid of next week (HCTV-30)
NSS implementation does not reflect spec. requirement.

Comment by Alihan Dikici [2020-02-28]

Łukasz Romanowski,
Do we have an update?
As we talked; if we do not have a proper solution yet, please share new EB that ignores certificate errors.

Comment by Michał Drausowski [2020-03-02]

Alihan Dikici

Please find the EB_ENV_1 here:
https://customers.vewd.com/share/page/site/sierra/documentlibrary#filter=path%7CBuilds/Engineering/SIERRA-329
To make it work, please set the env variable before launching the browser:
`OPAPP_IGNORE_CERT_VALIDATION_ERR=YES`

If it works properly for you then I can add this change on DEV branch. We can remove it once the issue is solved. Is this approach okay?

edit: please note there is no message printed that we are ignoring the error (I will add it in dev build)

Comment by Alihan Dikici [2020-03-03]

Matthew Barker, can you please check?

Generated at Tue Mar 03 16:38:39 CET 2020 by Mateusz Matejuk using Jira 7.13.11#713011-sha1:bfabf803487815cf20f1284df36170bdf4d336a0.