Copy Summary ▾ View ▾

Closed Bug 1673240 (CVE-2021-23991) Opened 2 years ago Closed 2 years ago

RNP-01-014 WP1 Thunderbird: Key manipulation via uncertified Auto-Import (Medium)

▾ **Categories**

Product:	MailNews Core ▾	Type:	defect
Component:	Security: OpenPGP ▾	Priority:	P1 Severity: --
Version:	78		

▾ **Tracking**

Status:	RESOLVED FIXED	Tracking Flags:	Tracking Status
Milestone:	88 Branch		thunderbird_esr78 + fixed

▸ **People** (Reporter: wsmwk, Assigned: KaiE)

▸ **References**

▸ **Details** (Keywords: sec-high, Whiteboard: [RNP][fixed-in-rnp][needs tb adjustments])

▾ **Attachments**

Bug 1673240 - Use new RNP APIs for checking key validity. r=mkmelin wsmwk : **approval-comm-esr78+** Details | Review
2 years ago **Kai Engert (:KaiE:)**
48 bytes, text/x-phabricator-request

Bug 1673240 - Add tests using test keys contributed by Neal Walfield. r=mkmelin wsmwk : **approval-comm-esr78+** Details | Review
2 years ago **Kai Engert (:KaiE:)**
48 bytes, text/x-phabricator-request

Show Obsolete

Bottom ↓ Tags ▾ Timeline ▾

Wayne Mery (:wsmwk) Reporter
Description • 2 years ago

It was found that Thunderbird allows importing primary keys and subkeys that are not bound to a valid cryptographically secure signature. Additionally, Thunderbird automatically imports detected, attached PGP primary keys with an already trusted fingerprint, as it has an extended expiry time. This introduces the risk of attackers obtaining a primary key with an extended or unset expiry time of a trusted person, while it has not yet been imported into the PGP key ring of the victim.

The attackers can abuse this by adding a malicious subkey, which has been used by Thunderbird for email encryption, signature verification and key certification. Although the subkey is flagged as invalid by RNP, it is silently accepted and imported by Thunderbird as soon as the user opens the attacker's mail.

Affected File:
comm/mail/extensions/openpgp/content/ui/enigmailMessengerOverlay.js

Affected Code:

```
commonProcessAttachedKey(keyData, isBinaryAutocrypt) {
  let errorMsgObj = {};
  let preview = EnigmailKey.getKeyListFromKeyBlock(
     keyData,
     [...]
  );
  [...]
  for (let newKey of preview) {
     let oldKey = EnigmailKeyRing.getKeyById(newKey.fpr);
     if (!oldKey) {
        [...]
        continue;
      }
      [...]
      let newHasNewValidity =
          oldKey.expiryTime < newKey.expiryTime ||
          (oldKey.keyTrust != "r" && newKey.keyTrust == "r");
      if (!newHasNewValidity)
          continue;
      [...]
      !EnigmailKeyRing.importKeyDataSilent(
         window,
         keyData,
         isBinaryAutocrypt,
         "0x" + newKey.fpr
       )
  [...]
```

It is advised that the validity of the PGP keys returned from the RNP library is respected by Thunderbird and actions ensue accordingly. Additionally, Cure53 recommends for trusted primary keys not getting automatically imported. Instead, the user should be informed and asked about importing the new key. It is crucial to alert the user about every new subkey and user-identity which is about to be newly trusted.

Daniel Veditz [:dveditz]
Updated • 2 years ago

Keywords: sec-high

Magnus Melin [:mkmelin]
Comment 1 • 2 years ago

Attached patch ~~bug1673240_subkey_check.patch~~ (obsolete) — Details — Splinter Review

WIP, though I think this is the gist of it.
I don't know how to generate a key with invalid subKeys, so no idea if it works.

Attachment #9187335 - Flags: feedback?(kaie)

Wayne Mery (:wsmwk) Reporter
Updated • 2 years ago

status-thunderbird_esr78: --- → affected
tracking-thunderbird_esr78: --- → +
Priority: -- → P1

Nickolay Olshevsky
Comment 2 • 2 years ago

New RNP API would be added as this issue will be resolved: https://github.com/rnpgp/rnp/issues/1214
Basically, before doing any manipulations with public keys/subkeys, those should be checked for validity. While we do that internally, still API is not exposed via FFI yet.

Magnus Melin [:mkmelin]
Updated • 2 years ago

See Also: → https://github.com/rnpgp/rnp/issues/1214

Magnus Melin [:mkmelin]
Updated • 2 years ago

Whiteboard: [RNP][fixed-in-rnp][needs tb adjustments]

Kai Engert (:KaiE:) Assignee
Comment 3 • 2 years ago

I've read this problem report many times, and I still think it is confusing.
I think it's necessary that I state my assumptions, and ask questions, and get those clarified, before we can figure out how to fix this issue.

During development, we had the need to increase our compatibility with existing keys. So I had asked the RNP team to relax restrictions, and allow a wider keys to be imported.

I think it's necessary to distinguish between "having a key available in the key store" and "querying key attributes" and "using a key for operations".

My assumptions:

- each public key defines a primary key
- for each subkey, there must be a signature of the primary key, which confirms the subkey was added by the owner of the primary key
- for each user id, there must be a signature of the primary key, which confirms the user id was added by the owner of the primary key
- for each validity time change, there must be a signature of the primary key, which confirms the change was made by the owner of the primary key
- RNP may import/store a public key, even if there are bad signatures
- if the application asks RNP for a list of subkeys, RNP will check the subkey, and only return a subkey if it has a valid signature by the primary key
- if the application asks RNP for a list of used IDs, RNP will check the user IDs, and only return a user ID if it has a valid signature by the primary key
- if the application asks RNP for the key's validity time, RNP will only return information that has been confirmed with a valid signature by the primary key

Are these assumptions correct?
Could you please comment on each item, which ones are correct or not?
For any of the above, did the RNP behavior change after the security audit?

Flags: needinfo?(o.nickolay)

Kai Engert (:KaiE:) Assignee
Comment 4 • 2 years ago

Maybe I will find some answers in the bugs that Neal has reported. I have to study them in detail.

Kai Engert (:KaiE:) Assignee
Comment 5 • 2 years ago

(In reply to Kai Engert (:KaiE:) from ~~comment #3~~)

> - if the application asks RNP for a list of user IDs, RNP will check the user IDs, and only return a user ID if it has a valid signature by the primary key

IIUC, ~~bug 1666236~~ reports that my assumption was incorrect. RNP returned user IDs despite a missing or bad signature.

~~Bug 1666236~~ links two RNP tickets, and both have been marked as fixed in November 2020. We recently updated Thunderbird to v0.14.0 from January 2021.

Nickolay Olshevsky
Comment 6 • 2 years ago

Since this was originally posted RNP's API was updated with a number of functions allowing to check key/subkey/userid/signature validities.

> each public key defines a primary key

And it considered as valid if it has at least one valid self-signature, and it doesn't expire key. There is one edge case, caused by 'stripped' keys, i.e. keys without userids and certifications. In this case key also considered as valid if it has subkey with valid subkey binding signature.
API function to check validity status: `rnp_key_is_valid()` . Also there is a function `rnp_key_valid_till()` , which return timestamp till which key was considered as valid. This is to distinguish the case of emails, signed 3 years ago, with the key which expired or revoked 1 year ago.

> for each subkey, there must be a signature of the primary key, which confirms the subkey was added by the owner of the primary key

Subkey is considered as valid by RNP if it has at least one non-expiring subkey binding signature, and primary key is valid as well. Validity may be checked via the same `rnp_key_is_valid()` function. `rnp_key_valid_till()` also apply here.

> for each user id, there must be a signature of the primary key, which confirms the user id was added by the owner of the primary key

Yeah, this is called self-certification. Also this signature includes information about the key usage - preferred algorithms, key expiration, flags and so on.
This may lead to confusion if the key has multiple userids with different preferences, or direct-key signatures (i.e. signatures which tells some information about the key itself). So, in exotic situations, key preferences would depend on which userid was used to found this key.
To check userid validity you may use function `rnp_uid_is_valid()` . Please note that functions `rnp_key_get_uid_*` will list all uids, including invalid ones. However, `rnp_locate_key()` will search for the key only via valid uids.

> for each validity time change, there must be a signature of the primary key, which confirms the change was made by the owner of the primary key

Yes, usual practice is to add additional signature with extended expiration time. As far as I remember there is no strict MUST in RFC on which signatures take in account in such case, but it is recommended to check the most recent signature. I.e. if one year ago you set key expiration time to 5 years, and yesterday to 1 year, then it should expire in 1 year instead of the 4. While we have `rnp_key_get_expiration()` , you still may walk through all the sigs and override default decision.

> RNP may import/store a public key, even if there are bad signatures

Yes, we do not remove anything automatically giving flexibility to the library user. However, you still may use functions `rnp_key_remove_signature()` , `rnp_key_remove_uid()` or `rnp_key_remove_signatures()` to cleanup the key. This functions were added recently and will be available in v0.15.0 release.

> if the application asks RNP for a list of subkeys, RNP will check the subkey, and only return a subkey if it has a valid signature by the primary key

No, it will return all subkeys and you should check validity via `rnp_key_is_valid()`

> if the application asks RNP for a list of used IDs, RNP will check the user IDs, and only return a user ID if it has a valid signature by the primary key

No, it will return all the uids and you should call `rnp_uid_is_valid()` to check it.

> if the application asks RNP for the key's validity time, RNP will only return information that has been confirmed with a valid signature by the primary key

That's right. But still there may be different edge cases, like different validity times for different userids or direct-key sigs. We have some issues to behave in other way in such cases, but didn't get to those yet. For these cases key validity/expiration may vary depending on which userid was used to find a key.

Hope this helps, feel free to ask for more details. Actually, as for me, OpenPGP standard in some places is too flexible and allows too much freedom of interpretation.

Flags: ~~needinfo?(o.nickolay)~~

Kai Engert (:KaiE:) Assignee
Comment 7 • 2 years ago

Thanks for the explanation.

Regarding user IDs, I think TB should completely ignore user IDs that don't pass signature checks.

However, in its key management user interface, TB wants to show the user ID of an expired key. This can allow the user to identify an expired key, select it, and request to extend the expiration.

This means, I need to be able to ask RNP: Is time the only reason why the user ID is invalid? Does the user ID pass all other validity checks?

rnp_uid_is_valid doesn't answer this question.

If TB hides all user IDs that rnp_uid_is_valid describes as invalid, then the user interface cannot show any user ID for expired keys - which makes it difficult for the user to identify a key that they might want to extend.

For subkeys, you are offering rnp_key_valid_till. I think this provides the answer I need for subkeys. If the timestamp returned by rnp_key_valid_till is nonzero, then in my understanding, the subkey passes all signature checks.

Kai Engert (:KaiE:) Assignee
Comment 8 • 2 years ago

Maybe the following approach is mostly ok. If the primary key is expired or revoked, but the primary key was valid at some time (rnp_key_valid_till returns nonzero), then TB could use rnp_key_get_primary_uid, and show the result, regardless of the uid validity.

Kai Engert (:KaiE:) Assignee
Comment 9 • 2 years ago

The plan from ~~comment 8~~ doesn't work, because rnp_key_get_primary_uid returns failure if all user IDs are expired.

Nickolay Olshevsky
Comment 10 • 2 years ago

Hi Kai,

> This means, I need to be able to ask RNP: Is time the only reason why the user ID is invalid? Does the user ID pass all other validity checks?

To confirm this you may iterate through userid signatures, filter out self-sig, and call `rnp_signature_is_valid()` on it. It will return RNP_SUCCESS if signature is valid (despite the key expiration). The same could be done on the subkey. I have some plans on adding functions `rnp_uid_latest_selfsig()` / `rnp_key_latest_binding()` to make this easier.
I.e. even if uid/key are expired, self-signatures are still valid (however, they may also be marked as expiring).

> The plan from ~~comment 8~~ doesn't work, because rnp_key_get_primary_uid returns failure if all user IDs are expired.

Yeah, sorry, forgot to mention this in the first comment. If there are no valid signatures on the uid, primary flag is dropped.

Kai Engert (:KaiE:) Assignee
Comment 11 • 2 years ago

Thanks Nickolay, that works!

Kai Engert (:KaiE:) Assignee
Comment 12 • 2 years ago

Let's summarize.

This bug reports, we may automatically import keys with invalid data, and use that invalid data.
The reported suggestion was, don't import automatically.
However, not importing automatically is contrary to the intended convenience mechanism, so I'd like to use a different fix.
Also, because the user is allowed to use keys manually, preventing automatic import wouldn't be a complete fix against using manipulated keys.

The report is correct, we automatically import an updated key with new subkeys.
It is correct that the TB code level would automatically accept the use of a new subkey, without checking the subkey's validity at the TB level.
This could be abused to trick TB into using a different encryption key, when sending an encrypted message.
I suggest to fix it by using the recently added new RNP API rnp_valid_till.
We already limit use only subkeys for encryption that are not revoked and are not expired.
By calling rnp_valid_till, and additional requiring that a subkey must have been valid at some time, we ensure that only subkeys certified by the key owner will be used.

Regarding user IDs:
It is correct, we currently display inalid user IDs, based on Nickolay's explanation.
However, we don't automatically trust/accept new user IDs.
At the time the user views the details for a key, we store the acceptance setting for each email address reported on screen.
We don't update the list of accepted email address, unless the user views the key details again (which would then show an updated list of email addresses) and confirms the acceptance settings again.
Nevertheless, we should disallow the use of invalid user IDs (and their email addresses).
The suggestion is to only allow accepting user IDs that carry a valid self signature, as reported by rnp_uid_is_valid.

Kai Engert (:KaiE:) Assignee
Comment 13 • 2 years ago

Comment on ~~attachment 9187335 [details] [diff]~~ [review]
~~bug1673240~~_subkey_check.patch

I think this isn't the right approach to solve the issue. We need to treat keys based on their properties, and the new RNP APIs make that possible. I have a patch in hand that prevents the use of invalid keys, as also reported in ~~bug 1666236~~ and ~~bug 1666360~~. If a key has only bad properties, it won't be imported as a consequence.

Attachment #9187335 - Flags: feedback?(kaie) → feedback-

Kai Engert (:KaiE:) Assignee
Updated • 2 years ago

Assignee: nobody → kaie

Kai Engert (:KaiE:) Assignee
Updated • 2 years ago

Attachment #9187335 - Attachment is obsolete: true

Kai Engert (:KaiE:) Assignee
Comment 14 • 2 years ago

Attached file ***Bug 1673240 - Use new RNP APIs for checking key validity. r=mkmelin*** *— Details*

Kai Engert (:KaiE:) Assignee
Comment 15 • 2 years ago

Adding Neal to CC, because he independently found related issues, and I'd like to fix the issues in this bug.

Kai Engert (:KaiE:) Assignee
Updated • 2 years ago

See Also: → ~~CVE-2021-23993~~

Kai Engert (:KaiE:) Assignee
Comment 16 • 2 years ago

Attached file ***Bug 1673240 - Add tests using test keys contributed by Neal Walfield. r=mkmelin*** *— Details*

Kai Engert (:KaiE:) Assignee
Comment 17 • 2 years ago

Neal, thank you very much for contributing test keys in ~~bug 1666236~~ and ~~bug 1666360~~. I'm adding these keys to the Thunderbird test suite, and have added automated tests.
The tests fail without the fix from this bug, and work with the fix.

Phabricator Automation
Updated • 2 years ago

Attachment #9210268 - Attachment description: Bug 1673240 - Prevent the use of invalid OpenPGP keys and user IDs. r=mkmelin → Bug 1673240 - Use new RNP APIs to check key validity. r=mkmelin

Phabricator Automation
Updated • 2 years ago

Attachment #9210268 - Attachment description: Bug 1673240 - Use new RNP APIs to check key validity. r=mkmelin → Bug 1673240 - Use new RNP APIs for checking key validity. r=mkmelin

Kai Engert (:KaiE:) Assignee
Updated • 2 years ago

Target Milestone: --- → 88 Branch

Kai Engert (:KaiE:) Assignee
Comment 18 • 2 years ago

Comment on attachment 9210268 [details]
~~Bug 1673240~~ - Use new RNP APIs for checking key validity. r=mkmelin

[Approval Request Comment]
Regression caused by (bug #): no
User impact if declined: invalid keys are allowed
Testing completed (on c-c, etc.):
Risk to taking this patch (and alternatives if risky): low, should only affect scenarios that need to fail

Attachment #9210268 - Flags: approval-comm-esr78?

Kai Engert (:KaiE:) Assignee
Comment 19 • 2 years ago

https://hg.mozilla.org/comm-central/rev/5f3efef0a3b5e117e38e2e6b42c4ac5921a65bfe

Status: NEW → RESOLVED
Closed: 2 years ago
Resolution: --- → FIXED

Wayne Mery (:wsmwk) Reporter
Comment 20 • 2 years ago

Comment on attachment 9210268 [details]
~~Bug 1673240~~ - Use new RNP APIs for checking key validity. r=mkmelin

[Triage Comment]
Approved for esr78

Attachment #9210268 - Flags: approval-comm-esr78? → approval-comm-esr78+

Kai Engert (:KaiE:) Assignee
Comment 21 • 2 years ago

https://hg.mozilla.org/releases/comm-esr78/rev/f29328195e9166a13653ba4862a96cd2a7e38a16
78.9.1

status-thunderbird_esr78: affected → fixed

Kai Engert (:KaiE:) Assignee
Comment 22 • 2 years ago • Edited

Suggested CVE text:

```
CVE-??:
     title: An attacker may use Thunderbird's OpenPGP key refresh mechanism to
poison an existing key
     impact: moderate
     reporter: Cure53
     description: |
       If a Thunderbird user has previously imported Alice's OpenPGP key, and
Alice has extended the validity period of her key, but Alice's updated key has
not yet been imported, an attacker may send an email containing a crafted
version of Alice's key with an invalid subkey, Thunderbird might subsequently
attempt to use the invalid subkey, and will fail to send encrypted email to
Alice.
     bugs:
       - url: 1673240
```

Note we still allow the automatic import, but the invalid key details will no longer result in DoS.

Edit: fixed the bug number

Tom Ritter [:tjr]
Updated • 2 years ago

Alias: CVE-2021-23991

Kai Engert (:KaiE:) Assignee
Comment 23 • 2 years ago

Comment on attachment 9210693 [details]
~~Bug 1673240~~ - Add tests using test keys contributed by Neal Walfield. r=mkmelin

tests only

Attachment #9210693 - Flags: approval-comm-esr78?

Kai Engert (:KaiE:) Assignee
Comment 24 • 2 years ago

tests landed into c-c:
https://hg.mozilla.org/comm-central/rev/5d9bce96055f4c4bca69ec574c1597e71280b4c4

Wayne Mery (:wsmwk) Reporter
Comment 25 • 2 years ago

Comment on attachment 9210693 [details]
~~Bug 1673240~~ - Add tests using test keys contributed by Neal Walfield. r=mkmelin

[Triage Comment]
Approved for esr78

Attachment #9210693 - Flags: approval-comm-esr78? → approval-comm-esr78+

Kai Engert (:KaiE:) Assignee
Comment 26 • 2 years ago

tests landed into 78.10
https://hg.mozilla.org/releases/comm-esr78/rev/d25b83da0e46e61f20a149ffe0445cd9899067be

neal
Comment 27 • 2 years ago

I've read this a few times now and, like Kai, I'm having trouble understanding the issue. I apologize for what may appear as nits, but in a discussion like this it is important to be precise and use accepted terminology.

(In reply to Wayne Mery (:wsmwk) from ~~comment #0~~)

> It was found that Thunderbird allows importing primary keys and subkeys that are not bound to a valid cryptographically secure signature.

What does this mean? In OpenPGP primary keys and subkeys are not bound to signatures. A signature binds a component (e.g., a subkey or a User ID) to the primary key. Slightly simplified, a certificate is a primary key (which solely determines the certificate's fingerprint), binding signatures made by the primary key, and components for which there are valid binding signatures made by the primary key. The validity of a primary key is determined by the trust model (web of trust, direct trust, TB's "key acceptance"). The validity of a component is derived from any signatures that the primary key makes over the component; the primary key is the certificate's trust root. In other words, if there is a valid binding signature over a subkey made by the primary key, the subkey is *de jure* associated with the primary key; a signature is the sufficient and necessary proof that the entity that controls the key (certificate plus secret key material) wants the component to be part of the certificate.

Note: subkeys cannot stand alone.

> Additionally, Thunderbird automatically imports detected, attached PGP primary keys with an already trusted fingerprint, as it has an extended expiry time. This introduces the risk of attackers obtaining a primary key with an extended or unset expiry time of a trusted person, while it has not yet been imported into the PGP key ring of the victim.

I don't understand what is being claimed here :/.

Why is the primary important here: "PGP primary keys"? Is the implication that Thunderbird is ignoring the rest of the certificate?

What is a trusted fingerprint? I think what is meant is an authenticated certificate.

What does it mean to have an extended expiry time? Some component has a new binding signature, whose expiration time has been extended?

What does "while it has not yet been imported" mean?

It sounds like: "Thunderbird automatically imports certificates under certain conditions. An attacker may automatically import a new version of the certificate, and a victim may not." Please confirm or correct my understanding. If I understood correctly, what's the risk there?

> The attackers can abuse this by adding a malicious subkey, which has been used by Thunderbird for email encryption, signature verification and key certification.

Guessing again: "An attacker can add an invalid subkey to a certificate, and the victim's Thunderbird may import it, and use it"

> Although the subkey is flagged as invalid by RNP, it is silently accepted and imported by Thunderbird as soon as the user opens the attacker's mail.

Subkeys are not independent objects. If a subkey is not bound to a primary key via a valid binding signature, then the OpenPGP implementation should ignore it. If it doesn't, then the OpenPGP implementation is broken.

Justus reported this to RNP on July 13, 2018:

> Critical vulnerability in RNP, Ribose's fork of Netpgp.
>
> ## Overview
>
> RNP fails to validate subkey bindings, allowing malicious parties to
> add or replace subkeys, even if the authenticity of the key is
> verified by comparing fingerprints, or other means like the web of
> trust. This allows an attacker that is able to control a victims
> network traffic to read messages encrypted using RNP.
>
> ## Impact
>
> OpenPGP implementations need to aquire the transferable public key
> (TPK) of the peers one wants to communicate with. This happens when
> establishing a new peer, and periodically to refresh ones copy of the
> TPK to get new keys, certificates, or revocations.
>
> If an attacker controlling the victims network traffic is able to
> modify the TPK in flight. Due to the missing signature verification,
> she can replace the encryption subkey present in the TPK with her own.
>
> Key verification, like comparing key fingerprints, is ineffective
> because it is done on the primary key, which is not changed.
>
> If the victim now encrypts a message using the attackers encryption
> key, the attacker can intercept the message, read it, re-encrypt it
> with the original encryption key, and send it to the intended
> recipient. The confidentiality of the message has been compromised.
>
> Like subkey bindings, the following signatures are likely also
> affected: user-id bindings, user-attribute bindings, revocations of
> any kind.

Ronald acknowledged the issue and indicated that they would fix it by the end of the week. As far as I know, a report about the vulnerability was never published.

Anyway, that RNP flags a subkey as invalid is good. That Thunderbird has to check if the subkey is valid before using it is deeply dangerous. A context that requires a key should only accept a valid key. Or, in the very least, be opt-in instead of opt-out.

Thunderbird *should* aggressively import certificates. It should be up to the OpenPGP implementation, in this case RNP, to authenticate the components using the primary key as the trust anchor.

Thunderbird should *not* treat the keyring as a curated keyring; it is a cache. If a binding between an identifier (email) and a certificate can be authenticated by the trust model, then the certificate should be used. Otherwise, it should be ignored.

> It is advised that the validity of the PGP keys returned from the RNP library is respected by Thunderbird and actions ensue accordingly.

If RNP exposes invalid subkeys to Thunderbird, then it is indeed Thunderbird's responsibility to add a check that they are valid before using them. But, I don't see what that has to do with importing invalid components, and why that is risky, or what expiry has to do with any of this.

> Additionally, Cure53 recommends for trusted primary keys not getting automatically imported. Instead, the user should be informed and asked about importing the new key. It is crucial to alert the user about every new subkey and user-identity which is about to be newly trusted.

This is horribly, horribly wrong. Do not ask the user about importing new certificates. Subkeys and User IDs are authenticated via binding signatures. The user cannot make an informed choice. Do not make the OpenPGP support less usable then it already is.

Finally, it's disappointing that no proof of concept was provided.

neal
Comment 28 • 2 years ago

(In reply to Kai Engert (:KaiE:) from ~~comment #7~~)

> Regarding user IDs, I think TB should completely ignore user IDs that don't pass signature checks.

This depends on the context.

If you are going to use a User ID in a security-relevant context (pretty much all contexts that matter to users!), the only User IDs that are relevant are those that are authenticated. Interestingly, this normally has *nothing* to do with whether a User ID also has a self signature. Let me repeat that: a self-signed User ID is not more or less authenticated that a User ID that is not self signed. Consider. Let's assume that I'm using a CA as a trust anchor. If that CA certifies the binding between the User ID "<alice@example.org>" and certificate 0xABCD, then that is what I care about; "<alice@example.org>" is authenticated for 0xABCD. The certificate doesn't even need to have a self signature for that User ID! This is perhaps surprising, because current tooling gives an elevated status to self signed User IDs. But, that is primarily due to a deeper problem: the available authentication mechanisms are inadequate not only in Thunderbird, but also in the broader ecosystem.

There are two places in Thunderbird where it makes sense to show self-signed User IDs and not User IDs that are not self signed. First, when accepting a certificate, it makes sense to provide the self-signed user ids as a suggestion similar to how Android's address book uses the name field of a vcard as the suggested contact's name. Second, the keyring manager is a low-level tool, which lists all of the keys in the local cache. If an authenticated User ID is not available for a certificate, it should display: "Allegedly: Self-signed User ID".

Well, we're not there. So, yes, today, Thunderbird should only show self signed User IDs.

> However, in its key management user interface, TB wants to show the user ID of an expired key. This can allow the user to identify an expired key, select it, and request to extend the expiration.
>
> This means, I need to be able to ask RNP: Is time the only reason why the user ID is invalid? Does the user ID pass all other validity checks?

This is spot on and identifies an important issue that the RNP API overlooks again and again: context is essential. See also this bug report.

> rnp_uid_is_valid doesn't answer this question.
>
> If TB hides all user IDs that rnp_uid_is_valid describes as invalid, then the user interface cannot show any user ID for expired keys - which makes it difficult for the user to identify a key that they might want to extend.

Interestingly, we also want to show revoked keys. Say I have an old key, which I've revoked. I still want to be able to use it to decrypt messages. Or, say a certificate has been soft revoked (e.g., retired), and I want to check a signature. That should still be possible.

> For subkeys, you are offering rnp_key_valid_till. I think this provides the answer I need for subkeys. If the timestamp returned by rnp_key_valid_till is nonzero, then in my understanding, the subkey passes all signature checks.

Unfortunately I don't think it does. See the referenced bug.

neal
Comment 29 • 2 years ago

(In reply to Nickolay Olshevsky from ~~comment #10~~)

> Yeah, sorry, forgot to mention this in the first comment. If there are no valid signatures on the uid, primary flag is dropped.

This is unfortunate. A certificate should always have exactly one primary User ID unless there are no self-signed User IDs. If all self signed User IDs are revoked or expired, then a revoked or expired User ID should be used. This is important, because the Primary User IDs self signature provides information about the primary key and certificate, e.g., key expiration, key flags, etc.

Wayne Mery (:wsmwk) Reporter
Updated • 2 years ago

Group: ~~mail-core-security~~

Arvidt
Comment 30 • 2 years ago

(In reply to neal from ~~comment #27~~)

> This is horribly, horribly wrong. Do not ask the user about importing new certificates. Subkeys and User IDs are authenticated via binding signatures. The user cannot make an informed choice. Do not make the OpenPGP support less usable then it already is.

That's what I thought, too, when reading the Cure53 report.

You need to log in before you can comment on or make changes to this bug.

Top ↑