strongSwan - Bug #3586

unsupported key type [Ed25519] in swanctl --load-creds

07.10.2020 18:03 - Daniel Schuermann

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	swanctl		
Target version:	5.9.1		
Affected version:	5.9.0	**Resolution:**	Fixed

Description

I use

```
pki --gen --type ed25519 --outform pem > private/vpnClientKey.pem
```

to generate an EdDSA key, but strongswan fails to load it:

```
unsupported key type in '/etc/swanctl/private/vpnClientKey.pem'
loaded private key from '/etc/swanctl/private/vpnClientKey.pem'
```

The client fails to authenticate itself (via EAP-TLS):

```
[TLS] received TLS cert request for 'C=CH, O=strongSwan, CN=strongSwan Root CA
[TLS] no TLS peer certificate found for 'C=CH, O=strongSwan, CN=vpnclient@<tld.net>', skipping cli
ent authentication
```

I tried RSA and ECDSA keys and they worked fine. I've got the following plugins loaded:

```
loaded plugins: charon-systemd ldap pkcs11 aesni aes des rc2 sha2 sha3 sha1 md5 mgf1 random nonce
x509 revocation constraints pubkey pkcs1 pkcs7 pkcs8 pkcs12 pgp dnskey sshkey pem openssl fips-prf
 gmp curve25519 agent chapoly xcbc cmac hmac ntru drbg newhope bliss curl mysql sqlite attr kernel
-netlink resolve socket-default bypass-lan connmark forecast farp stroke vici updown eap-identity
eap-sim eap-aka eap-aka-3gpp2 eap-simaka-pseudonym eap-simaka-reauth eap-md5 eap-gtc eap-mschapv2
eap-dynamic eap-radius eap-tls eap-ttls eap-peap xauth-generic xauth-eap xauth-pam xauth-noauth dh
cp radattr unity counters
```

Am I missing a plugin? openssl and curve25519 are loaded. Maybe someone can try to reproduce my problem.

Associated revisions

Revision 68392567 - 27.10.2020 11:17 - Tobias Brunner

vici: Support all defined key types

References #3586.

Revision 30d47ea4 - 27.10.2020 11:17 - Tobias Brunner

swanctl: Support any key type for decrypted keys

The previous code required explicit support for a particular key type,
of which Ed25519 and Ed448 were missing. While a fallback to `any` would
have been possible (this is already the case for unencrypted keys in the
`private` and `pkcs8` directories, which are not parsed by swanctl), it's
not necessary (as long as swanctl and the daemon are from the same release)
and does not require the daemon to detect the key type again.

Fixes #3586.

History

#1 - 08.10.2020 13:47 - Tobias Brunner

- Category set to swanctl

- Status changed from New to Feedback

> to generate an EdDSA key, but strongswan fails to load it:

Looks like swanctl is missing explicit support for EdDSA keys (the vici plugin too, actually), which is currently required if there is a secret defined for the loaded key (there are two different code paths for encrypted and unencrypted keys, the latter are passed to the daemon without parsing and detecting the type). Did you configure a secret in swanctl.conf even though the key is unencrypted (*pki* does not produce encrypted keys)?

Anyway, I pushed fixes to the *3586-swanctl-keytype* branch so it should also work if a secret is defined for a key.

> The client fails to authenticate itself (via EAP-TLS):
> [...]

That would not work anyway because our TLS stack currently doesn't support EdDSA keys (support for TLS 1.3 and indirectly EdDSA is currently being developed by a student, although I'm not sure if EdDSA support for older TLS versions will also be part of that).

#2 - 08.10.2020 21:12 - Daniel Schuermann

I've used unencrypted keys and I've no secrets defined in swanctl.conf.
As suggested I tested using encrypted keys.
For RSA or ECDSA keys I successfully tested encrypted keys using

```
openssl rsa -aes256 -in private/vpnClientKeyRSA.pem -out private/vpnClientKeyRSAEnc.pem
openssl ec -aes256 -in private/vpnClientKeyECDSA.pem -out private/vpnClientKeyECDSAEnc.pem
```

but I've no idea how encrypt an existing ed25519 key. I can generate a new one using

```
openssl genpkey -algorithm X25519 -aes256 -out private/vpnClientKeyEDDSAEnc.pem
```

but swanctl can't open the encrypted key file and keeps asking for the password.

#3 - 09.10.2020 15:17 - Tobias Brunner

> I've used unencrypted keys and I've no secrets defined in swanctl.conf.

Are you prompted for a password by the swanctl command? Because there is no code path that leads to the unsupported key type error message without a secret being available (either from the config or prompt).

> As suggested I tested using encrypted keys.

If a secret is required to load the key, it definitely won't work as there will only be the code path that leads to the error above. Looking back at the log messages above and the code, the key is actually loaded successfully (loaded private key from '/etc/swanctl/private/vpnClientKey.pem'). There is a fallback that tries to load the key as is (i.e. without parsing) if loading it with a secret fails.

So in your case, the key will actually be available. But as mentioned, you currently won't be able to use it with EAP-TLS.

> but I've no idea how encrypt an existing ed25519 key.

Could probably be done via openssl pkcs8 -topk8 command.

> I can generate a new one using
> [...]
> but swanctl can't open the encrypted key file and keeps asking for the password.

That command generates an x25519 DH key, not an Ed255196 key (i.e. use -algorithm ed25519 instead). But again, that definitely won't work without the patches in the *3586-swanctl-keytype* branch.

#4 - 10.10.2020 10:21 - Daniel Schuermann

> Are you prompted for a password by the swanctl command?

No, the key is unencrypted. I've got no secrets in swanctl.conf and swanctl -q isn't asking for a password either.
Nevertheless I get the "unsupported key type" error, but swanctl --list-certs confirms the key is loaded anyway.

> Because there is no code path that leads to the unsupported key type error message without a secret being available (either from the config or prompt).

I can only tell you what I configured.

> So in your case, the key will actually be available. But as mentioned, you currently won't be able to use it with EAP-TLS.

I tried sucessfully using pubkey auth instead.

> Could probably be done via openssl pkcs8 -topk8 command

Thanks for the hint. When I use encrypted keys I get a different error message:

```
unsupported key type in '/etc/swanctl/private/vpnClientKeyEnc.pem'
loading '/etc/swanctl/private/vpnClientKeyEnc.pem' failed: parsing ANY private key failed
```

and the key actually isn't loaded.
Anyway, thank you very much for your help!

#5 - 12.10.2020 14:19 - Tobias Brunner

- Tracker changed from Issue to Bug

- Target version set to 5.9.1

> > Are you prompted for a password by the swanctl command?
>
> No, the key is unencrypted. I've got no secrets in swanctl.conf and swanctl -q isn't asking for a password either.
> Nevertheless I get the "unsupported key type" error, but swanctl --list-certs confirms the key is loaded anyway.
>
> > Because there is no code path that leads to the unsupported key type error message without a secret being available (either from the config or prompt).
>
> I can only tell you what I configured.

Yeah, I saw that now. Sorry for the confusion. A password is really only requested if required, so the unencrypted key is parsed fine without any but then you run into the issue with the unsupported key type, followed by the fallback to not parsing the key at all, which succeeds.

> So in your case, the key will actually be available. But as mentioned, you currently won't be able to use it with EAP-TLS.

> I tried sucessfully using pubkey auth instead.

OK, great.

> Could probably be done via openssl pkcs8 -topk8 command

> Thanks for the hint. When I use encrypted keys I get a different error message:
> [...]
> and the key actually isn't loaded.

Yes, that's what I expected.

#6 - 27.10.2020 11:19 - Tobias Brunner

- Subject changed from unsupported key type [Ed25519] to unsupported key type [Ed25519] in swanctl --load-creds

- Status changed from Feedback to Closed

- Assignee set to Tobias Brunner

- Resolution set to Fixed