pfSense Packages - Bug #6339

OpenVPN Client Export package option for "Use Microsoft Certificate Storage" does not specify which certificate to use

05/10/2016 12:18 AM - Scott Bradner

Status:	New	**Start date:**	05/10/2016
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	OpenVPN Client Export	**Estimated time:**	0.00 hour
Target version:			
Affected Version:		**Affected Architecture:**	All

Description

Just spent a while tracking this down, please disregard if it's a PEBKAC issue. :)

I tried the option in the client export for "Use Microsoft Certificate Storage instead of local files.", which does have the installer correctly add the certificate to the microsoft certificate storage. However, when I try to connect with the included configuration, it seems to pick a certificate to connect with at random (which for me happened to be a domain auth certificate for my work domain, rather than the OpenVPN cert that had just gotten installed). After a bit of searching, I found that the correct certificate was being located by the following configuration directive:

```
cryptoapicert "SUBJ:"
```

Which as I understand it searches the Microsoft Certificate Storage for any certificate with a subject...rather than the specific one just installed. :) I found ticket 386 (https://community.openvpn.net/openvpn/ticket/386) over in the OpenVPN issue tracker, which documents what the submitter was able to figure out as far as what the subject format needed to be to properly locate the cert. tl;dr version:

1) Run the cert through "openssl x509 -in foo.cer -noout -subject"
2) Replace each container label other than the first (i.e. "ST=") with ", "

In my case openssl gave me a subject of:

```
subject= /C=US/ST=Washington/L=Newcastle/O=Superlime Industries/emailAddress=xxx@xxx.xxx/CN=OpenVPNAuth"
```

which transformed into a config directive of:

```
cryptoapicert "SUBJ:US, Washington, Newcastle, Superlime Industries, xxx@xxx.xxx, OpenVPNAuth"
```

With the proper config directive in place, selecting the cert worked exactly as expected and I was able to connect successfully. Should the export package do all that transformation for the user? Otherwise, it seems fairly unlikely that the right cert would end up getting selected for the user automatically.

History

#1 - 04/19/2018 03:39 PM - Caleb Hornbeck

Not sure if it would be easier to implement, but using this works well for me:

```
cryptoapicert "THUMB:<cert thumbprint>"
```