Codesigning Introspection Commands

###Create a CSR (Certificate Signing Request):

First generate a new signing key.

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

Then

```
$ openssl req -new -key MyPrivateSigningKey.key -out MyCertificateSigni
```

###Certificate Signing Request:

Examine the contents of a certificate signing request.

```
$ openssl asn1parse -in <path to CSR file>
```

###Signing Certificate:

Examine the contents of a signing certificate.

```
$ openssl x509 -inform [der|pem] -text -in <path to certificate file>
```

###Provisioning Profile Contents:

Note: the following applies to both `.provisionprofile` and and `.mobileprovision` files.

Examine the contents of the signed plist in a provisioning profile.

```
$ security cms -D -i <path to provisioning profile>
```

Or

```
$ openssl smime -inform der -verify -in <path to provisioning profile>
```

###Provisioning Profile Developer Certificates:

Note: the following applies to both `.provisionprofile` and and `.mobileprovision` files.

Examine the developer certificates that can be used with this provisioning profile.

Dumping the plain text plist version of the provisioning profile.

```
$ openssl smime -inform der -in <path to provisioning profile> -verify
```

Then

```
$ /usr/libexec/PlistBuddy -c 'Print :DeveloperCertificates' -x /tmp/pro
```

This will display all of the signing certificates that can be used with this profile. There may be more than one, to extract the certificates to examine them individually:

```
$ /usr/libexec/PlistBuddy -c 'Print :DeveloperCertificates:<index of ce
```

Note: please see the PlistBuddy manual page to see how to extract other information from the plist version of the provisioning profile.

Then

```
$ openssl x509 -inform der -text -in <path to .cer file>
```

###Provisioning Profile Entitlements:

Note: the following applies to both `.provisionprofile` and and `.mobileprovision` files.

Dumping the plain text plist version of the provisioning profile.

```
$ openssl smime -inform der -in <path to provisioning profile> -verify
```

Then

```
$ /usr/libexec/PlistBuddy -c 'Print :Entitlements' -x /tmp/profiletext.
```

This will display all of the entitlements that are allowed to be used with this provisioning profile.

Note: please see the PlistBuddy manual page to see how to extract other information from the plist version of the provisioning profile.

###Provisioning Profile Signing Certificates:

Note: the following applies to both `.provisionprofile` and and `.mobileprovision` files.

Examine the certificates that were used to sign a provisioning profile.

```
$ openssl pkcs7 -inform der -print_certs -in <path to provisioning prof
```

This will print a couple of certificates in the format of

```
subject=...
issuer=...
----BEGIN CERTIFICATE----
...
----END CERTIFICATE----
```

Copy each certificate (the line starting with `----BEGIN CERTIFICATE----` and ending with `----END CERTIFICATE----`, including both of those lines in the file is important) into a separate file and save it as plain text with a `.pem` extension.

Then

```
$ openssl x509 -inform pem -text -in <path to .pem file>
```

###Find Codesigning Identities in the Keychain:

This will print two lists; one of all found signing identities, the second of only valid signing identities (identities that have certificate and private key).

```
$ security find-identity -p codesigning
```
