Over-the-air IPhone Setup Using a Signed .mobileconfig File

Note: this does not push your configuration to an iPhone. The user of the iPhone must go to a web address and install a configuration profile.

Suppose that you have a few iPhones that you need to support, but you don't want to spend the time typing in all of the e-mail (IMAP or POP), LDAP, wireless network, or other settings into each phone. Perhaps you have found Apple's Enterprise Deployment Guide but you don't really feel like setting up a whole SCEP Certification Authority to get things done either since your requirements are so simple. But you do realize that it is much easier to tell your user to go to https://example.com/iphone/ on their iPhone than to step them through all the individual setup routines.

Amazingly enough, there is not much documentation out there on how to hand-roll a .mobileconfig file that you can pass out on an HTTPS server to your users. We also want it to be "Verified" by the iPhone so that your users can see it is from you. While they can install untrusted profiles, it sure adds a nice touch to have the green checkmark.

Perhaps you've scoured the Internet since you've read that you can "just use `openssl smime`" to sign your .mobileconfig file, but no one seems to tell you *how*. We'll go over that here as well.

1) Create a configuration (.mobileconfig) file

This file will contain all the configuration you want for your users' iPhones. I believe you can use Apple's iPhone Configuration Utility to create this file. You don't have to, but it'll probably save you some typing.

The Enterprise Deployment Guide defines the syntax of the profiles in Appendix B. You can do some pretty fancy request/response scripting between the phone and your server, but I'll just go over a simpler method that just sends a configuration file from your web server to their phone.

Your .mobileconfig file will end up looking something like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple/DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>PayloadContent</key>
        <array>
                <dict>
                        <key>PayloadDisplayName</key>
                        <string>LDAP Settings</string>
                        <key>PayloadType</key>
                        <string>com.apple.ldap.account</string>
                        <key>PayloadVersion</key>
                        <integer>1</integer>
                        <key>PayloadUUID</key>
                        <string>6df7a612-ce0a-4b4b-bce2-7b844e3c9df0</string>
                        <key>PayloadIdentifier</key>
                        <string>com.example.iPhone.settings.ldap</string>
                        <key>LDAPAccountDescription</key>
                        <string>Company Contacts</string>
                        <key>LDAPAccountHostName</key>
                        <string>ldap.example.com</string>
                        <key>LDAPAccountUseSSL</key>
                        <false />
                        <key>LDAPAccountUserName</key>
                        <string>uid=username,dc=example,dc=com</string>
                        <key>LDAPSearchSettings</key>
                        <array>
                                <dict>
                                        <key>LDAPSearchSettingDescription</key>
                                        <string>Company Contacts</string>
                                        <key>LDAPSearchSettingSearchBase</key>
                                        <string></string>
                                        <key>LDAPSearchSettingScope</key>
                                        <string>LDAPSearchSettingScopeSubtree</string>
                                </dict>
                                <dict>
                                        <key>LDAPSearchSettingDescription</key>
                                        <string>Sales Departments</string>
                                        <key>LDAPSearchSettingSearchBase</key>
                                        <string>ou=Sales,dc=example,dc=com</string>
                                        <key>LDAPSearchSettingScope</key>
                                        <string>LDAPSearchSettingScopeSubtree</string>
                                </dict>
                        </array>
                </dict>
                <dict>
```

```
                        <key>PayloadDisplayName</key>
                        <string>Email Settings</string>
                        <key>PayloadType</key>
                        <string>com.apple.mail.managed</string>
                        <key>PayloadVersion</key>
                        <integer>1</integer>
                        <key>PayloadUUID</key>
                        <string>362e5c11-a332-4dfb-b18b-f6f0aac032fd</string>
                        <key>PayloadIdentifier</key>
                        <string>com.example.iPhone.settings.email</string>
                        <key>EmailAccountDescription</key>
                        <string>Company E-mail</string>
                        <key>EmailAccountName</key>
                        <string>Full Name</string>
                        <key>EmailAccountType</key>
                        <string>EmailTypeIMAP</string>
                        <key>EmailAddress</key>
                        <string>username@example.com</string>
                        <key>IncomingMailServerAuthentication</key>
                        <string>EmailAuthPassword</string>
                        <key>IncomingMailServerHostName</key>
                        <string>imap.example.com</string>
                        <key>IncomingMailServerUseSSL</key>
                        <true />
                        <key>IncomingMailServerUsername</key>
                        <string>username@es2eng.com</string>
                        <key>OutgoingPasswordSameAsIncomingPassword</key>
                        <true />
                        <key>OutgoingMailServerAuthentication</key>
                        <string>EmailAuthPassword</string>
                        <key>OutgoingMailServerHostName</key>
                        <string>smtp.example.com</string>
                        <key>OutgoingMailServerUseSSL</key>
                        <true />
                        <key>OutgoingMailServerUsername</key>
                        <string>username@example.com</string>
                </dict>
        </array>
        <key>PayloadOrganization</key>
        <string>Your Organization's Name</string>
        <key>PayloadDisplayName</key>
        <string>Organization iPhone Settings</string>
        <key>PayloadVersion</key>
        <integer>1</integer>
        <key>PayloadUUID</key>
        <string>954e6e8b-5489-484c-9b1d-0c9b7bf18e32</string>
        <key>PayloadIdentifier</key>
        <string>com.example.iPhone.settings</string>
        <key>PayloadDescription</key>
        <string>Sets up Organization's LDAP directories and email on the iPhone</string>
        <key>PayloadType</key>
        <string>Configuration</string>
</dict>
</plist>
```

I'll talk just briefly about the configuration above. The iPhone, as far as I can tell, uses the UUIDs to know whether or not it is replacing or installing a new profile onto the phone. On a Mac or Linux box, you can generate a UUID with the command `uuidgen`. You'll notice that I did not include any passwords above. With these settings, the iPhone will prompt the user for their e-mail password upon installation of the profile. (The LDAP password will be prompted on first use if logging in fails.)

I actually wrote a PHP script that would take a template .mobileconfig file for me and fill in the username fields for me depending on PHP_AUTH_USER. After you get the basics down, you can go back and do that. There is also a way to encrypt the .mobileconfig files, but we are not covering that here.

Sign the .mobileconfig file

This is the part that no one else seems to go over. Signing your configuration profile is an optional step, but it's not too hard if you already have an X.509 web server or email certificate.

For this step, I'll use the following notations:

- `company.mobileconfig` is your unsigned configuration profile
- `server.crt` is your server's certificate to sign the profile with
- `server.key` is your server's private key
- `cert-chain.crt` is the certificate bundle for the CA that issued your server's certificate. (This may be optional, but I never tested without it)
- `signed.mobileconfig` will be your signed configuration profile

Once you have all the files listed above, you will run a command like the following:

```
openssl smime -sign -in company.mobileconfig -out signed.mobileconfig -
signer server.crt -inkey server.key -certfile cert-chain.crt -outform der -
nodetach
```

The `-outform der` and `-nodetach` are your real tickets here in getting it into a form that the iPhone wants. Now you take `signed.mobileconfig` and move on to the next step!

Help for those that will use PHP scripting: You'll want to look at `openssl_pkcs7_sign()` function with the `$flags` field set to 0. This will create a file that is base-64 encoded. After you strip off the e-mail headers at the top, you can `base64_decode()` to get the same output. For example:

```
$mobileconfig = base64_decode(preg_replace('/(.+\n)+\n/', '', $signed, 1));
```

Serve up the file on your HTTPS server

Okay, it'll probably work on your HTTP server as well. Just another configuration I didn't bother testing.

There is just one caveats when it comes to serving up this file. It needs to be served up with a MIME Content-Type of `application/x-apple-aspen-config`. You may be able to do this by adding a line to your server's configuration or .htaccess file in the folder with:

```
<IfModule mod_mime.c>
        AddType application/x-apple-aspen-config .mobileconfig
</IfModule>
```

If serving the file from within PHP, you may do something like:

```
header('Content-type: application/x-apple-aspen-config; chatset=utf-8');
header('Content-Disposition: attachment; filename="company.mobileconfig"');
echo $mobileconfig;
```

Try it out on your iPhone

Get your iPhone and load up Safari. Go to the web address of where your profile is saved, e.g. https://www.example.com/iphone/. Your phone should prompt you to install the profile.

You can see and remove profiles from Settings > General on your iPhone. Note, that it IS possible to create a profile that cannot be removed except for by the original profile identifier and signed by the same authority. Be careful that you don't lock yourself out.

Finished!

At this point, we are finished. See the Enterprise Deployment Guide for other configuration profiles that you can create. It doesn't let you create or set everything that I wish it did (especially when it comes to setting up IMAP defaults), but it lets you do quite a bit.

I hope that this helps you! This is obviously a very brief guide and I glazed over a few details. If you have any comments, let me know. My e-mail address can be deduced from the very bottom of the document.

See Also

- Retrieving an iPhone response using PHP

I hope this helps someone. Let me know if there are errors above and I'll update this document.
-W Gillespie (wgillespie, es2eng.com)

Last updated: 2010-03-09