Pretty Good Privacy (PGP)

Introduction

Check the GnuPG web page for documentation on the GnuPG package:

> http://www.gnupg.org/

In particular, look at the documentation, and the "Mini HOWTO" for lots of good information:

Public Key Cryptography

We'll use pkg_add to install GNUPG. A local copy is on our noc box:

```
# pkg_add ftp://noc/pub/FreeBSD/6.2-RELEASE/i386/packages/All/gnupg-1.4.5_1.tbz
```

Creating a Public/Private Key Pair

Now that GnuPG is installed you can use the `gpg` command to use the Gnu version of PGP. In these steps you should do this as your user on your system (i.e. admin), *not* as the root user!:

```
$ gpg --gen-key
```

You will be asked quite a few questions. Picking the defaults for key type and size should be fine.

Be sure to set an expiry date when prompted.

Your PGP information will be stored in your user's account home directory under the ".gnupg" directory.

Extracting your Public Key

To extract your public key as text do:

```
$ gpg -a --export <your key id>
```

Note: "your key id" is probably your email address in this case.

To see what public keys you have installed, you can always type

```
$ gpg --list-keys
```

To place your newly generated public key on the MIT (Massachussetts Institute of Technology in Boston, Massachussetts, United States) PGP server do the following:

```
$ gpg --list-keys yourUserid
```

Look for the line that says "pub" - the second set of numbers after the "/" is your key-id. A sample such line is:

```
pub 1024D/E947C3B4 2007-11-28 [expires: 2008-11-27]
```

The string "E947C3B4" is the key-id in this case. So, to send your newly created public key to a keyserver you would do:

```
$ gpg --keyserver pgp.mit.edu --send-keys key-id
```

That's it! Your public key is now available to anyone who goes to the MIT PGP server and searches on your name, email address, etc. Your key information will propagate to all the pgp key-servers worldwide within a few hours.

Generating your Public Key's Fingerprint

You calculate the fingerprint for a local copy of a public key like this:

```
$ gpg --fingerprint <key id>
```

Importing Someone Else's Public Key

Once you have obtained a public key, you can import it to your local keyring so that you can use it like this:

```
$ gpg --import <filename>
```

Note, you can, also, import public keys via email plugins. We'll be doing this later on in these exercises.

Signing a Public Key

If you have a copy of someone else's public key on your keyring and you have decided that you trust it (e.g. by verifying the fingerprint with the key's owner) and you have also decided that you trust the identity of the key's owner (e.g. by checking a passport) you can sign it. This does two things:

To sign a key:

```
$ gpg --sign-key <key id>
```

PGP with Thunderbird and Enigmail

Now that you have installed with PGP, created your own key and played with some of its features let's use PGP via an email client like Thunderbird. First we'll need to install Thunderbird on your workstations. To do this type (you need to be root):

```
# pkg_add
ftp://noc/pub/FreeBSD/6.2-RELEASE/i386/packages/All/thunderbird-1.5.0.7_1.tbz
```

Once Thunderbird is installed let's install the enigmail extension, which adds PGP support to Thunderbird. First you'll need to get the enigmail Thunderbird extension. We have a local copy of this on the noc. Make sure you are "admin" for this exercise:

```
$ cd
$ ftp noc
username: anonymous
password: admin@pcN
ftp> cd pub/FreeBSD/configs
ftp> lcd /home/admin
ftp> get enigmail.xpi
ftp> exit
```

OK, now you have all the bits and pieces needed to make Thunderbird work with your created pgp keys. You can find Thunderbird under the Application ==> Internet menu in your desktop.

First open Thunderbird, then you will be prompted with a New Account Setup dialogue. Answer the following:

1. POP and server is localhost
2. Your Name: "WhateverYouWant"
   Email Address: admin@pcN.cctld.eu.org
3. SMTP Server: localhost
4. Account Name: whatever you like

   And, that should do it for account setup.

5. Now, once you are done with this you need to go to the Tools menu, then pick Extensions. Click on the Install button and choose the enigmail.xpi file. This will install Enigmail 0.94.1 in to Thunderbird. You will need to restart Thunderbird at this point.
6. Now the Enigmail extension will ask you if you wish to configure OpenPGP security for the selected identity. Say "Yes" to this.
7. Click the Enable OpenPGP support (Enigmail) for this identity.
8. The defaults should work. In the next window check the "Encrypt Message" dialogue and

press OK. You may get a warning about using html-formatted email. You can turn off html formatting in the Thunderbird options to avoid this in the future.

9. Choose to send email to someone in the class who has generated their PGP key and has uploaded it to the pgp.mit.edu server.
10. You won't have that person's key, but in the dialogue that appears you can choose to download it. Pick the "pgp.mit.edu" server from the drop-down list, click OK once the person's public key is found, then check their key-id, and click OK again to finally send the message encrypted. The next time you send to that person you will not have to go through so many steps.

More Information

There are many more things you can do with GnuPG than those described in these notes. For more information, see:

http://www.gnupg.org/