Preview of the syntax for importing from Secure (Encrypted) Eggs

Directions for use:

1). Create your Secure Eggs using the code published at <http://code.google.com/p/pyeggs/downloads/list>.  For now, Windows must be used when creating Secure Eggs however this should not pose any problems as Windows can be used from any OS via VMWare or some variant of VMWare.

Secure Eggs can be used by any OS so long as the Python Source being Secured can run in the target environment.

Secure Eggs support only Python 2.5.x therefore the contents of a Secure Egg can be code for any Python 2.5.x version.

No support is planned for any other Python version at this time however those who support our efforts are encouraged to invest in our Secure Egg technologies to allow us to support additional Python versions going forward.

2). Sample code for importing from a Secure Egg:

    eggname = r'Z:\python projects\pyEggs\@lib\eVyperLogixLib-1.0-py2.5.egg'  
    e = SecureEgg(egg=eggname).module('e.misc').\_import\_(globals={},locals={})

eggname is the name of the single egg from which modules can be imported.  The whole module will be imported based on the module spec which is "'e.misc'" from the above.

Once the import has been completed the globals and locals will be populated per the contents of the module.  Typically the locals will hold the contents (functions) from the module while the globals will contain any globals from the module.

The plain-text source is never exposed to the outside world however remote debugging must be disabled to keep people from being able to see the plain-text source.  Future versions of pyEggs will allow encrypted compiled (sourceless) modules to be used however at this time this is not an option.

3). Execute a function from the module "e.misc" by issuing the following:

e.\_\_locals\_\_.append(foo,10)  # append is the name of the function that resides in the module "e.misc".

4). You can reuse the SecureEgg object instance via the following type of syntax:

e.module('e.misc.\_utils').\_import\_(globals={},locals={})

This gives you a fresh globals and locals for your new module.

e.module('e.misc.\_utils').\_import\_()

This uses the globals and locals that are found in the SecureEgg object instance.

5). When you deploy pyEgg Secured Eggs you will do so for computers you have registered with our Registration System.  Each computer you wish to Register can bee offline at the time the Secured Eggs are used however you will need to execute a specific piece of code to obtain a signature from that computer.  The signature must be used at the time the Secured Egg is used as in the following:

    eggname = r'Z:\python projects\pyEggs\@lib\eVyperLogixLib-1.0-py2.5.egg'  
    e = SecureEgg(egg=eggname).signature('signature-goes-here').module('e.misc').\_import\_(globals={},locals={})

The signature is supplied by our Registration System via a web interface which you can use when pyEggs have been created.  You can obtain the name of the computer for which you want to deploy Secure Eggs using the following Python code:

import socket

socket.gethostbyname\_ex(socket.gethostname())[0] # output from this line of code must be supplied to the Secure Egg Registration System.

This limits where the Secure Eggs can be used.  The Secure Egg is signed with the signature and the code that tries to use the Secure Egg must have the same signature as the Secure Eggs.

Secure Eggs cannot be simply copied from one computer to another unless the signature is changed to match the runtime environment for the specific use of each Secure Egg.  This makes the Secure Eggs a bit more secure than if they were simply Encrypted.

6). It is left up to those who use and deploy Secure Eggs to best determine how they wish to sign their Secure Eggs however our Registration System must be used to obtain a signature for each Secure Egg.

7). Once signed a Secure Egg cannot be changed or it will cease to function.

8). The Secure Egg Registration System will be online within 90 days.  If Secure Eggs must be signed they can be signed in the meantime by via email, details can be worked out whenever the need arises however the process includes having the Secure Egg and the target computer name together at the same time.

9). Additional details may be required going forward and previously established details may require changes however this is the overview for how Secure Eggs can be used.

10). Please direct all questions and comments for the use of Secure Eggs to the attention of [support@vyperlogix.com](mailto:support@vyperlogix.com); we may choose to have one or more of our support techs respond as-required or we may respond ourselves directly.