



GPG mass-cleaning and the PGP Corp. Global Directory

 The article was obtained at the following URL: <http://www.kfwebs.net/articles/article/17>
 The article might be distributed further as long as it is provided as it is, with the credits stated.
 The Article was written and first published by KF Webs, at <http://www.kfwebs.net>
 #####

Are you plagued by a number of expired and revoked signatures, and especially expired signatures from PGP Corporation's global directory? Please read on.
Added: 2005-08-04 02:25:16 - Modified: 2006-09-23 23:20:54 - Level: Beginner

Ok, so after PGP Corp started the Global Directory keyserver, it has caused some, if not problems, so at least estetical issues for users of other keyservers. The thing is that the Global Directory, which will be refered to as GD from now on, in addition to be a central storage of keys, try to verify that the email address of the user id's, from now on shortened as uid, is authentic.

There are a couple of flaws in regards to their attempt to do so. First of all the email is sent un-encrypted, so anyone able to snoop the network can get the verification email. But back to the main issue; the GD issue a signature valid for two weeks each time a user download the key from it. That way, if the user has removed the key from the global directory, it will take maximum two weeks from it happend till all signatures are expired.

At this point I would like to urge you all not to sign the Global Directory verification key, as it in my opinion isn't good enough to authenticate anyone. Therefore if you have signed the global directory key, you won't be signed by me, and if you sign afterwards and I find out, I will most likely revoke the signature, so that my Web of Trust isn't weakened, for users placing trust in me with their signature.

The problem with the global directory signatures is that if anyone download the key from GD, then upload it to an ordinary keyserver, the signature is added, and won't be removed, therefore, you get a new signature every time this happend, which might make the key look something like:

```
sig sig CA57AD7C 2005-01-01 2005-01-15 _____ PGP Global Directory Verification Key
sig sig CA57AD7C 2005-01-14 2005-01-28 _____ PGP Global Directory Verification Key
sig sig CA57AD7C 2005-01-27 2005-02-10 _____ PGP Global Directory Verification Key
sig sig CA57AD7C 2005-02-09 2005-02-23 _____ PGP Global Directory Verification Key
sig sig CA57AD7C 2005-02-19 2005-03-05 _____ PGP Global Directory Verification Key
sig sig CA57AD7C 2005-03-28 2005-04-11 _____ PGP Global Directory Verification Key
sig sig CA57AD7C 2005-04-10 2005-04-24 _____ PGP Global Directory Verification Key
sig sig CA57AD7C 2005-04-23 2005-05-07 _____ PGP Global Directory Verification Key
sig sig CA57AD7C 2005-05-07 2005-05-21 _____ PGP Global Directory Verification Key
sig sig CA57AD7C 2005-05-20 2005-06-03 _____ PGP Global Directory Verification Key
```

An example of how this can happend is: Alice upload her key to GD. The day after Bob download the key, which expire in 14 days. The day after that, Charlie download the key from the GD with its own 14-day



signature. Now, both Bob and Charlie upload the key to another keyserver, it suddenly has two signatures, one expiring the day after the other.

Now, you can probably see why this is annoying in the long term. So the developers of GnuPG have been nice enough to add a "clean" function to the edit-key menu, and import and export options to clean the key as of version 1.4.2. If you want to clean keys on import and when refreshing from a keyserver you'd put the following in the configuration file

```
keyserver-options import-clean  
import-options import-clean
```

If you on the other hand want to perform this manually against keys already on the keyring, you can use a custom script that loop through all the keys. Such a script will look like:

```
#!/bin/bash  
for i in `gpg --list-keys | grep "^pub" | awk '{split($2,a,"/");print a[2]}`;  
do  
gpg --edit-key $i clean save;  
done;
```

A very simple for-loop in bash, but it come handy.

Related articles: