Global Information Assurance Certification Paper

PGP in a Networked, Multi-user Environment
Mark Fennig
September 10, 2000

Most discussions of PGP (Pretty Good Privacy) seem to follow the notion that a user has a non-networked, single-user PC or laptop that is solely in their possession and control. That was the model PGP was designed for so the risks associated with the theft of private-keys and pass phrases were minimized or at least under the control of the user. However, the reality for many users is that they are on a networked, multi-user environment where much of the data storage is provided via remote file servers. Typically a business or educational institution owns the computing resources in these environments and the end-user has little or no ownership and control over any of it. Assuming this is the environment you are operating in, let's see how we can use PGP in a reasonably safe way.

PGP Risks

First, what are the risks? In networked, multi-user environments, it mostly comes down to stealing private-keys and pass phrases through various types of system compromises, system snooping, and network sniffing. You can create procedures and take precautions to reduce these risks. However, keep in mind that systems in this kind of environment will have users with root or administrator capabilities. These users will often have the ability to circumvent many of the precautions you might take.

Minimizing the Risks

So, what can we do to reduce the risks?

1. Check out the administrators of your desktop devices and servers. Do they run a tight ship with adequate security and controls in place or is the environment an open door with poor availability and so on? If you are not comfortable with the integrity of your systems or administrators, you probably shouldn't use PGP on them.
2. Learn to use PGP and its features correctly. If you don't have a good understanding of PGP, you may implement or use it in a way that unnecessarily compromises its integrity.
3. Store your PGP key-ring(s) on a removable, write-protected media like floppy, zip, or CD-ROM disk. You can control where key-rings are stored by setting PGP's PGPPATH variable. Only mount your key-ring disk as needed and always remove it when you are done. Be sure to have a backup copy of your key-ring stored in a safety deposit box or other secure location.
4. If you are on a system with no access to a removable media drive, you'll have to consider committing your key-rings to disk. If this is the case, refer back to item 1 above and make sure you are comfortable with the integrity of the system and it's administrators. If you decide to put your key-ring on disk, make sure it is to a filesystem where ownership and permissions can be enforced (no FAT filesystems, please!) And make sure YOU are the owner and that only YOU have read/write access to the key-ring(s).
5. Memorize your pass phrase. Do not write it to a file or on paper.
6. Avoid PGP's undocumented **-z** option, especially on a multi-user system like Unix/Linux. The –z option allows you to specify your pass phrase on the command-line to help automate PGP processing. To see how revealing this can be on a Unix system, assume my pass phrase is "*my secret code*" and that I use the –z option to issue the PGP command **pgp –z "my secret code" –s readme.txt**. While the command is executing, any other user on the system could use the **ps** command and see my pass phrase as shown here:

   command: **ps –ef | grep pgp**

   pgpuser 26554 26528 1 13:55:15 pts/2 0:00 pgp **–**z **my secret code** -s readme.txt

7. PGP's PGPPASS variable can be set to hold your pass phrase. This will eliminate prompting for the pass phrase and is useful if you have many PGP operations to perform. When you are through with your PGP commands, set the value of PGPPASS to null or some other value to prevent accidentally displaying your pass phrase.
8. If you use PGPPASS in a script to automate pass phrase responses, set its value by prompting the user for it. Do not set this variable by hard-coding the pass phrase in the script.

Automating PGP

One special case of using PGP in a networked, multi-user environment is that of having to execute PGP commands in the background (i.e., batch environment). The challenge here is that the userid owning the batch process and data may not be available to provide the passphrase at 3:00am when the batch job runs. Or, the userid may not be associated with an individual, but rather an application or business process (e.g., payroll).

All of the precautions mentioned in the previous section still apply except for memorizing the pass phrase. Since there is no individual available to provide the pass phrase, we must find some other reasonably secure way of providing it. In this case, consider the following method for handling the pass phrase.

1. Choose a pass phrase that is fairly long and random. In this scenario, we don't intend to commit the pass phrase to any person's memory. Write this pass phrase to a file on a removable media like CD-ROM, making sure the file's ownership is for the appropriate userid and read/write access only for owner. This should be a very controlled procedure, maybe managed by an organization's IT Security staff or other "trusted" entity.
2. The disk containing the pass phrase should be kept in a secure, preferably locked safe or container at your data- or operations-center. A duplicate of this disk should also be kept offsite in a secure, locked location.
3. Insert code into your batch script or program to prompt an operator to mount the appropriate pass phrase disk, before the execution of your pgp commands.
4. Assign the value contained in the pass phrase file to the PGPPASS variable, or preferably, utilize the PGPPASSFD variable to set which file descriptor to read the pass phrase file from. Using PGPPASS is less desirable because of the chance that the pass phrase could be inadvertently exposed on the console or logs when running scripts or programs in debug mode.

Following is an example of a simple UNIX bourne-shell script utilizing a PGP pass phrase disk as described above. I've inserted comments to document the key commands in the script.

```
#!/bin/sh -xv

# Check for pass phrase file. Issue mount request if necessary.

ppdiskmounted=0

while [ $ppdiskmounted = "0" ]

do

if [ -f /cdrom/cdrom0/payroll/pp ]; then

ppdiskmounted=1;

else

echo "Place Disk# payroll-pp in cdrom drive on `hostname`"

echo "Press ENTER to continue..."

read response

fi

done

# Set PGPPASSFD to read pass phrase from file descriptor 3
```

```
PGPPASSFD=3; export PGPPASSFD

# Set location of key-ring

PGPPATH=/secretkeys/payroll; export PGPPATH

cd /data/payroll

# Encrypt and sign file. Get pass phrase from file descriptor 3

/usr/bin/pgp +force +batchmode -seat payroll.data \

"John Doe <jdoe@someorg.com>" \

-u "payroll <payroll@mycompany.com>" 3 < /cdrom/cdrom0/payroll/pp
```

Note the use of the +force and +batchmode options in the pgp command. These are needed to force a "yes" response to some questions and to eliminate others. If you prefer to have an operator interactively respond to these questions, eliminate these options. This script also uses the PGPPASSFD variable to obtain the pass phrase from a file. I could have used the PGPPASS variable instead by setting it with the following command.

```
PGPPASS=`cat /cdrom/cdrom0/payroll/pp`
```

But, since the shell script is set to run with the –x option (see line 1 of the script) this would result in the value of PGPPASS being displayed via STDOUT.

Conclusion

Although PGP is primarily intended for use by individuals and their personally owned computers, its use on networked and multi-user systems is necessary at times. PGP has the features to make this work, but not without risk. If you need to use PGP on a networked, multi-user system, you must recognize what these risks are and then manage those risks to an acceptable level.

References

- infiNity [daemon9@netcom.com /route@infonexus.com], "PGP Attacks." v.50, Feb 1996. URL: http://axion.physics.ubc.ca/pgp-attack.html . 4 Sept, 2000.
- Jackson, David S. "PGP and Linux." 29 Nov, 1998. URL: http://www.dsj.net/compedge/pgp4linux.html . 4 Sept, 2000.
- Zimmermann, Philip. "PGP User's Guide,Volume 2: Special Topics." 11 Oct, 1994. URL: http:// www.chemistry.mcmaster.ca/pgp/pgpdoc2/pgpdoc2.html . 2 Sept, 2000.
- Engelfriet, Arnoud. "The comp.security.pgp FAQ." v1.5. 22 Oct, 1998. URL: http://www.au.pgp.net/pgpnet/pgp-faq/faq-02.html . 6 Sept, 2000.
- Network Associates, Inc. "PGP Command Line Guide." URL: ftp://ftp.pgpi.org/pub/pgp/6.5/docs/english/PGPCmdLineGuide.pdf
- Williams, Randall T. "The passphrase FAQ." v1.04. 23 Mar, 1997. URL: http://www.stack.nl/~galactus/remailers/passphrase-faq.html