Trabalho Individual sobre o GPG

Fernando Paladini, Segurança em Computação (INE5429) 13/09/2016

1) Certificado de Revogação

As chaves criptográficas podem ser revogadas a qualquer momento, desde que o usuário que a criou tenha acesso à chave privada e também a senha utilizada nessa chave privada. A revogação de chaves criptográficas é extremamente importante para casos onde o usuário:

- Tenha sua chave privada comprometida (outras pessoas estejam usando ela);
- Tenha gerado uma nova chave e não queira mais usar a antiga.

Entretanto, às vezes o usuário pode não ter mais acesso à sua chave privada. Foi com o objetivo de fornecer uma segunda camada de proteção para esses casos que os certificados de revogação foram criados. Os "certificados de revogação" permitem revogar uma chave criptográfica mesmo que o usuário não tenha mais acesso à chave privada, de forma que devem ser armazenadas com muito cuidado e geradas quando o usuário ainda tem acesso à chave privada.

Para o experimento de revogação da chave com um certificado, os seguintes comandos foram executados:

Listing 1: Teste de revogação de chaves criptográficas com certificado de revogação.

```
1
2  # Geracao da chave criptografica de teste
3  $ gpg --gen-key
4
5  # Gerando certificado de revogacao
6  $ gpg --list-keys
7  pub   1024R/8316C680 2016-09-13
8  uid   Paladini (revocation certificate test) <fernandopalad@gmail.com>
9  sub   1024R/D6F0C35C 2016-09-13
10
11 $ gpg --output revoke.asc --gen-revoke 8316C680
12
13 # Enviando chave criptografica para o servidor de chaves do RNP
14 $ gpg --keyserver keyserver.cais.rnp.br --send-keys 8316C680
15
16 # Importando certificado de revogacao
17 $ gpg --import revoke.asc
18 $ gpg --list-keys
19 pub   1024R/8316C680 2016-09-13 [revoked: 2016-09-13]
```

```
20 uid   Paladini (revocation certificate test) <fernandopalad@gmail.com>
21
22 # Enviando chave revogada para o servidor de chaves do RNP
23 $ gpg --keyserver keyserver.cais.rnp.br --send-keys 8316C680
```

Durante o processo foi verificado, através da interface web do servidor de chaves do RNP, se os comandos executados estavam causando os efeitos desejados. Por fim, foi possível revogar a chave criptográfica sem maiores problemas. A mesma pode ser visualizada em: https://memoria.rnp.br/keyserver/pks/lookup?search=0x8316C680op=index

2) Revogação de Assinaturas

```
1  # Atualizando as chaves locais a partir do servidor de chaves do RNP
2  $ gpg --refresh-keys --keyserver keyserver.cais.rnp.br
3
4  # Obtendo uma chave criptogr fica de um colega
5  $ gpg --keyserver keyserver.cais.rnp.br --recv-key 85F12A95
6  gpg: requesting key 85F12A95 from hkp server keyserver.cais.rnp.br
7  gpg: key 85F12A95: public key "Bernardo Engelke (Aluno INE) <bernardo.↩
       engelke@grad.ufsc.br>" imported
8  gpg: public key of ultimately trusted key A5313440 not found
9  gpg: public key of ultimately trusted key 2FC2490F not found
10 gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
11 gpg: depth: 0  valid:   5  signed:  11  trust: 0-, 0q, 0n, 0m, 0f, 5u
12 gpg: depth: 1  valid:  11  signed:   1  trust: 11-, 0q, 0n, 0m, 0f, 0u
13 gpg: next trustdb check due at 2017-09-06
14 gpg: Total number processed: 1
15 gpg:               imported: 1  (RSA: 1)
16
17 # Assinando a chave criptogr fica desse colega e enviando ao RNP
18 $ gpg --default-key 622D0665 --sign-key 85F12A95
19 $ gpg --keyserver keyserver.cais.rnp.br --send-key 85F12A95
20
21 # Revogando a assinatura
22 $ gpg --refresh-keys --keyserver keyserver.cais.rnp.br
23 $ gpg --default-key 622D0665 --edit-key 85F12A95
24 $ gpg> revsig
25 $ gpg> save
26 $ gpg --keyserver keyserver.cais.rnp.br --send-key 85F12A95
```

3) Comandos do GPG

Os 10 comandos escolhidos estão abaixo:

```
# Lista, com algumas informacoes, todas as chaves publicas do chaveiro ou apenas as chaves passadas como argumento.
$ gpg --list-keys

# Lista, com algumas informacoes, todas as chaves privadas do chaveiro ou apenas as chaves passadas como argumento.
$ gpg --list-secret-keys

# Importa as chaves informadas (<KEYID1>, <KEYID2>, ..., <KEYIDN>) do servidor de chaves keyserver.cais.rnp.br para o chaveiro local.
$ gpg --keyserver  keyserver.cais.rnp.br --recv-keys <KEYID1> <KEYID2>

# Envia as chaves informadas (<KEYID1>, <KEYID2>, ..., <KEYIDN>) do chaveiro local para o servidor de chaves keyserver.cais.rnp.br.
$ gpg --keyserver  keyserver.cais.rnp.br --send-keys <KEYID1> <KEYID2>

# Deleta as chaves publicas informadas do chaveiro local.
$ gpg --delete-keys <KEYID1> <KEYID2>

# Deleta as chaves privadas informadas do chaveiro local.
$ gpg --delete-secret-keys <KEYID1> <KEYID2>

# Cria um certificado de revogacao para a chave publica informada.
$ gpg --gen-revoke 8316C680

# Importa / funde chaves (aplicavel tambem aos certificados de revogacao).
$ gpg --import revoke.asc

# Gera uma nova chave criptografica
$ gpg --gen-key

# Atualiza o chaveiro local a partir do servidor de chaves keyserver.cais.rnp.br.
$ gpg --refresh-keys --keyserver keyserver.cais.rnp.br
```

4) Opções aos comandos do GPG

```
# Imprime na tela a chave publica do e-mail fnpaladini@gmail.com com armadura ascii (base64).
$ gpg --armor --export fnpaladini@gmail.com
```

```
3
4  # Gera o arquivo meu-arquivo.gpg que esta encriptado com a chave publica ↩
       do recipiente informado.
5  $ gpg --recipient rodrigo_duarte.l@hotmail.com --encrypt meu-arquivo.txt
6
7  # Gera o arquivo meu-arquivo.asc (utilizando armadura ascii, base64) que ↩
       esta encriptado com a chave publica do recipiente informado.
8  $ gpg --recipient rodrigo_duarte.l@hotmail.com --armor --encrypt meu-↩
       arquivo.txt
9
10 # Exporta a chave p blica do usuario informado para o arquivo pubkey.txt ↩
       que tem conteudo codificado com armadura ascii - base 64.
11 $ gpg --armor --output pubkey.txt --export fnpaladini@gmail.com
12
13 # Assina a <KEYID> informada utilizando a chave privada do usuario (no ↩
       caso, 622D0665).
14 gpg -u 622D0665 --sign-key <KEYID>
```

5) Usando Certicados PGP no Webmail do INE (ou outro)

Como não consegui acessar o webmail do INE e o webmail da UFSC (Roundcube) não parece possuir suporte para assinaturas digitais com PGP (pelo menos de acordo com a documentação que encontrei), resolvi fazer de forma mais manual.

```
1  $ cat email_lista_seg.txt
2  Meu primeiro ""e-mail"" assinado digitalmente p/ o t pico 5 do trabalho ↩
       individual sobre GPG.
3
4  Att,
5  Fernando Paladini.
6
7  $ gpg -u 622D0665 --clearsign email_lista_seg.txt
8  $ cat email_lista_seg.txt.asc
9  -----BEGIN PGP SIGNED MESSAGE-----
10 Hash: SHA1
11
12 Meu primeiro ""e-mail"" assinado digitalmente p/ o topico 5 do trabalho ↩
       individual sobre GPG.
13
14 Att,
15 Fernando Paladini.
16 -----BEGIN PGP SIGNATURE-----
17 Version: GnuPG v1
18
```

```
iQIcBAEBAgAGBQJX3C10AAoJENK3VbRiLQZlp9oP/3gX09RGaOT2kbBU3c1Y8WG3
Xt74Din7qYzPNC5Wgry3siG/4M+WcdUyMolpNn5V2t5c/NyzdjAXRWsI+MNP/Kbm
9I89N2lEJZY1qtvIO4tTBkr0CBSroB8fQ6EDQFR9nyyM0OtEhie9e7Yl5GBou2pF
ZaqzygGqngTiPKl6KIS8Uqv3IRfibvPseYfaQIj8JLjXmgiBZWmtxTrs1YP7T7NX
PWz8JMl7kGrK1AJx5lpRtLq996C51uYwysTXEadV8aZlsV/6ZTI+pvEjFV/lRc84
X9VkyTYmdKNrAjKV2c+I51bYwc+aIJP4uC0wh8ur0lsEtku3Zs/9eXpHlE+wu8G+
Kzgr1F1By24jq9RNMoDwTI0YbPu/yeobdAlcyCCSdDxORW6klZLpU6NDuZ3my5fI
o30Wwt0ar48t4aJDaxS6JTFxto7F8bD9H78EfHbfDCSBeMqpO73ICHBI+h2zVe+7
XED52DITcHigG5ZlroXhGG4ErMo6rFrfJJjIbk4R2gCvrK2D7CRHJqEfV73qtXzJ
Dtl5mUauBPJYAzUeG7PCP1hfnAuC95Dml3rhnaYx6zWLfd+ujviYuSP9QmfhZzs7
zCxWKcsbmfwUHanRb/syC8WONYJmu7MkdU60rwLKf7Wqp+5DT1JDhkZ23Bfiqty8
i7IxxrKFckPiUNckztvj
=SKrk
-----END PGP SIGNATURE-----
```

Não compreendi muito bem para onde era para enviar o e-mail, portanto copiei o conteúdo assinado digitalmente e enviei para o fórum da disciplina no CAGR. Não sei qual o endereço da lista de e-mails da disciplina, se ela de fato possui uma. Está disponível em: http://forum.cagr.ufsc.br/listarMensagens.jsf?topicoId=2779920

6) Comando –edit-key

```
# Permite checar a sua assinatura no certificado passado como parametro.
check

# Faz uma assinatura nao revogavel no certificado passado como parametro, ou seja, uma assinatura que nao podera ser removida em algum momento no futuro
nrsign

# Adiciona uma foto para o certificado passado como parametro. Precisa ser JPEG e ter pouca resolucao, alem de baixa qualidade, para nao deixar o certificado muito pesado.
addphoto

# Permite modificar a data de expiracao do certificado passado como parametro.
expire

# Alterna a visualizacao entre chave publica e chave privada.
toggle

# Revoga a assinatura sobre o certificado passado como parametro.
revsig
```

```
18
19 # Mostra a foto que esta associada / junto com o certificado.
20 showphoto
21
22 # Permite modificar a senha utilizada para acessar / usar o certificado.
23 passwd
24
25 # Define o servidor de chaves preferido para o ID de usuario em questao (↩
      similar a um valor default, mas caso o servidor de chaves esteja ↩
      offline, utiliza outros).
26 keyserver
27
28 # Revoga a chave ou sub-chave em questao
29 revkey
```

7) Perguntas gerais

7.1) O que é o anel de chaves privadas? Como este está estruturado? Na sua aplicação GPG onde este anel de chaves é armazenado? Quem pode ter acesso a esse porta chaves?

Este é o local onde estão todas as suas chaves privadas e no meu caso está localizado em "/home/tulio/.gnupg/secring.gpg". Somente o dono do anel de chaves privadas deve ter acesso a ele, mas o arquivo está encriptado.

7.2) Qual a diferença entre assinar uma chave local e assinar no servidor?

Uma chave assinada localmente está disponível apenas para você, no seu chaveiro, enquanto que uma chave assinada no servidor está disponível para todos. Quanto mais assinaturas de chaves estiverem disponíveis nos servidores de chaves, maior é a rede de confiança de que vai existir para a rede como um todo, o que é algo extremamente positivo.

7.3) O que é e como é organizado o banco de dados de confiabilidade?

O banco de dados de confiabilidade é um banco de dados que possui a confiança que você tem em cada certificado. Por exemplo, se você conhece pessoalmente uma pessoa e confia muito nela, você pode assinar o seu certificado de tal forma a garantir mais confiança sua a ela. O bnaco de dados de confiabilidade armazena todas essas relações de forma local.

7.4) O que são e para que servem as sub-chaves?

As sub-chaves são chaves criptográficas que permitem a existência de uma chave-mestre, estando também limitadas / ligadas a ela. Uma sub-chave pode ser usada para encriptação e também para assinatura e é muito interessante pois através de uma chave-mestre é possível criar novas sub-chaves ou revogar as já existentes, o que é muito interessante por motivos de

segurança e também praticidade.

7.5) Coloque sua foto (ou uma figura qualquer) que represente você em seu certificado GPG.
Para colocar uma foto no certificado basta rodar os seguintes comandos:

```
1 $ gpg --edit-key 622D0665
2 addphoto
3 /home/tulio/Pictures/img.jpg
4 save
5 gpg --keyserver keyserver.cais.rnp.br --send-keys 622D0665
```

O certificado pode ser encontrado no servidor de chaves e está com uma imagem jpg extremamente leve.

7.6) O que é preciso para criar e manter um servidor de chaves GPG, sincronizado com os demais servidores existentes?
Para criar seu próprio servidor de chaves GPG que é sincronizado com os demais servidores GPG é necessário a utilização de um servidor SKS. SKS é um acrônimo para Synchronizing Key Server e é um protocolo que foi criado justamente este fim.

7.7) De um exemplo de como tornar sigiloso um arquivo usando o GPG. Envie esse arquivo para um colega e que enviar para você outro arquivo cifrado. Você deve decifrar e recuperar o conteúdo original.
Para fazer isso você precisa possuir a chave pública do colega e ele também deve possuir a sua. Suponha que queremos encriptar o arquivo "meu-arquivo.txt" com a chave pública do Rodrigo Duarte. Para tal, podemos fazer isso:

```
1 $ gpg -u 622D0665 --recipient rodrigo_duarte.l@hotmail.com --encrypt meu-↩
    arquivo.txt --output meu-arquivo.gpg
```

Quando o Rodrigo receber o arquivo encriptado, basta ele usar a sua chave privada para descriptografar o arquivo:

```
1 $ gpg --output meu-arquivo.txt --decrypt meu-arquivo.gpg
```

Depois de inserir a senha da chave privada dele, ele terá exatamente o mesmo arquivo que foi enviado, mas de forma completamente segura.

7.8) De um exemplo de como assinar um arquivo (assinatura anexada e outro com assinatura separada), usando o GPG. Envie uma mensagem assinada para um colega. Esse colega deve enviar para você outra mensagem assinada. Verifique se a assinatura está

correta.

Para assinar um arquivo ("meu-arquivo.txt" e obter um arquivo com a assinatura separada ("meu-arquivo.sig") basta executar:

```
1 $ gpg --output meu-arquivo.sig --detach-sig meu-arquivo.txt
```

Para verificar a assinatura de um arquivo assinado é necessário o comando "–verify" do GPG, passando como argumento tanto a assinatura quanto o arquivo:

```
1 $ gpg --verify meu-arquivo.sig meu-arquivo.txt
```

Para assinar um arquivo utilizando uma assinatura anexada ao mesmo tempo que criptografa o documento basta rodar o seguinte comando:

```
1 $ gpg --output meu-arquivo.sig --sign meu-arquivo.txt
```

Então, para descriptografar o documento ao mesmo tempo que verifica a assinatura basta rodar:

```
1 $ gpg --output meu-arquivo.txt --decrypt meu-arquivo.sig
```