The Diffie, Shapiro, and Martin (DSM) Crypto Tool

ActuallyFro

Summer 2016

TL;DR

Using your RSA SSH keypair is 'good enough' security for all persons who are not paranoid about major governments using quantum computers to find their nud3z.

Background

People use SSH to generate millions of key pairs every year[1]. It's more simple to leverage a known public key to encrypt a shared password, which is then used to symmetrically encrypt a desired block of data.

It seems if a person leverages SSH for secure communication needs, then , by extension, the person would tust other secuirty=related uses of the tools. To me it would appear to be sufficient to use for other data/blob encrypting, and would not require large forms of trust chains (I'm looking at you 'PGP/GPG people')

Therefore, instead of using multiple keys. means, methods, and codes a simpler method is being proposed as The Diffie, Shapiro, and Martin (DSM) method[2]. This method seeks to show how existing crypto tools can be used to implement a no non-sense means to secure data and communications. The following tutorial looks at a quick means to leverage the OpenSSH tools to encrypt, decrypt, and abuse the standard convention of crypto ... for pure lolz.

A fact to be pointed out

Many Y-combinator guys insist, very expressly, a-many different and methodical, nuanced interpretation should be utilized supposing to encrypt/ decrypt.

People are not wrong that this solution will probably make no sense, but as noted above: DSM have proposed a non-sense approach to many problems in life.

Topics

The following sections will look at:

- Setting up a key
- Creating and accessible public key
- 0-Crypto 101: A quick example
- 0-Crypto 201: A better use
- The DSM Crypto tool

Setting up a key

- First create a key: `ssh-keygen -t rsa -b <bit strength> -C "<your comment>"`
- Next Create a PEM version of the private key: `openssl rsa -in <private key> -outform pem > <name>.pem`

Creating and accessible public key

- Create a sharable public key PEM: `openssl rsa -in <private key> -pubout -outform pem > <name>.pub.pem`

0-Crypto 101: A quick example

- Encrypt: `echo 'Hello world!' | openssl rsautl -encrypt -inkey <name>.pub.pem -pubin -out Hello.enc`
- Note: you can pass a file with `-in <file>`
- Decrypt: `openssl rsautl -decrypt -inkey <private key>.pem -in Hello.enc -out Hello.txt`

Now the restriction of the total payload will be restricted to the key length of private key. A 1024 bit RSA key can only encrypt ~12 Bytes, but a 4096 key can encrypt ~500 Bytes. (Rough guide: 90% of the key length)

0-Crypto 201: A better use

Alexei Czeskis points out a better means to share secrets is to create a random symmetric key, encrypt it with the above method, and use openssl to AES to encrypt the file to be

shared[3].

He adds:

- Random Key Generation: `openssl rand -base64 32 > key.bin`
- Encrypting the key: `openssl rsautl -encrypt -inkey <private key>.pub.pem -pubin -in key.bin -out key.bin.enc`
- Encrypt the large file: `openssl enc -aes-256-cbc -salt -in SECRET_FILE -out SECRET_FILE.enc -pass file:./key.bin`
- Send the files
- Decrypt Key: `openssl rsautl -decrypt -inkey <private key>.pem -in key.bin.enc -out key.bin`
- Decrypt File: `openssl enc -d -aes-256-cbc -in SECRET_FILE.enc -out SECRET_FILE -pass file:./key.bin`

Note: The Diffie, Shapiro, Martin (DSM) Crypto tool

The DSM Crypto tool is a further, simple implementation of making security work. It's in this repo, use it or don't, I've tried to unbust it...

References

[1] B. Froberg, "It's my opinion, i'll add bogus stats all i want." 2016 [Online]. Available: https://www.youtube.com/watch?v=dQw4w9WgXcQ

[2] J. Diffie, "Live in vegas." 2016 [Online]. Available: https://www.youtube.com/watch?v=Tfd0qzNEZUk

[3] A. Czeskis, "How to encrypt a big file using openssl and someone's public key." september-2014 [Online]. Available: www.czeskis.com/random/openssl-encrypt-file.html