Encryption with GPG / OpenSSL

Simon M. Balthazar

Research and Education Network for Uganda:
Information Systems and Network Security Workshop

27th - 31st October, 2014

Reminder: Core Security Principals

- Confidentiality
- Integrity
- Authentication
  - Access Control
  - Verification
- Availability

Reminder: Cryptographic Methods

- Critical for Confidentiality, Integrity and Authentication.
- Indirectly leads to better Availability.
- Some methods and tools include:-
  - SSH, TLS/SSL, PGP, MD5/SHA1, Ciphers, DES/3DES/BLOWFISH, Private Keys, Public Keys, Digital Signatures

GPG

- GNU Privacy Guard, a GPL licensed alternative to the PGP suite of cryptographic software.
- GPG is a tool for secure communication.
- Part of Free Software Foundation's GNU Software Project.
- Compatible with RFC 4880, IETF Standards track specification of OpenPGP.

GPG Concepts

- Symmetric Ciphers
  - Uses the same key for both encryption and decryption.
- Public Key Ciphers
  - Invented to solve key exchanges.
  - Reduce the number of keys needed.
  - Uses a pair of keys, Public and a Private key.
- Hybrid Ciphers
  - Symmetric ciphers are stronger from security standing point.
  - Public keys are effective tools in distributing symmetric cipher keys.
  - PGP and GPG are both using hybrid ciphers.

GPG Concepts

- Digital Signatures
  - Using an algorithm such as SHA and MD5, a document is signed by hashing it, and the hash value is the signature.
  - Another person can check the signature by hashing their copy of the document and compare the hash value with the original document.

Key Management

- Its possible for an attacker to temper with your keyring.
- Key management is done by signing keys. Key signing has two purposes
  - It permits you to detect key tempering in your keyring
  - It allows you to certify the key truly belongs to the person named by a uid on the key.

Web of Trust

- Validating other keys in your keyring requires that you check the fingerprint of the key, and you sign his public key with your private key.
- This can be tedious when either you must validate large number of keys or when you communicate with people you don't know personally.
- Web of Trust delegates responsibility of validating public keys to people you trust.

Key Distribution

- Public keys are ideally distributed by giving it to you correspondents, by email or some other electronic correspondents.
- The most effective way of distributing keys is the use of public key servers. Public keys are received by the server is either added to the server's database or merged with the existing key if already present.
- When a key request comes to the server, the server consults its database and returns the requested public key if found.

Example: TLS (SSL) web server with digital certificate

I generate a private key on my webserver

I send my public key plus my identity (my webserver's domain name) to a certificate authority (CA)

The CA manually checks that I am who I say I am, i.e. I own the domain

They sign a certificate containing my public key, my domain name, and an expiration date

I install the certificate on my web server

When a client's web browser connects to me using HTTPS:

They negotiate an encrypted session with me, during which they learn my public key

I send them the certificate

They verify the certificate using the CA's public key, which is built-in to the browser

If the signature is valid, the domain name in the URL matches the domain name in the certificate, and the expiration date has not passed, they know the connection is secure

The security of TLS depends on:

Your webserver being secure

- So nobody else can obtain your private key

The CA's public key being in all browsers

The CA being well managed

- How carefully do they look after their own private keys?

The CA being trustworthy

Do they vet all certificate requests properly?

Could a hacker persuade the CA to sign their key pretending to be someone else? What about a government?

Testing TLS (SSL) Applications

There is an equivalent of telnet you can use: openssl s_client

It opens a TCP connection, negotiates TLS, then lets you type data

$ openssl s_client -connect domain-name:443

END

simon@tznic.or.tz