E-mail Encryption with GnuPG

A hands-on tutorial for the Washington, DC #cryptoparty

By: The Doctor [412/724/301/703] [ZS|Media]
drwho@virtadpt.net
https://drwho.virtadpt.net/
PGP key ID: 0x807B17C1
PGP fingerprint: 7960 1CDC 85C9 0B63 8D9F DD89 3BD8 FF2B 807B 17C1

V1.0

What is e-mail encryption?

- Obscuring the content of e-mail so that it cannot be read
  - On the wire
  - While stored on disk
- Encrypted so that only the recipients(s) can decrypt it
- Often, client-to-server traffic is encrypted with SSL or TLS
- Rarely, server-to-server traffic is encrypted

What ISN'T e-mail encryption?

- SMTP headers aren't encrypted
- Source and destination addresses aren't encrypted
- Subject line isn't encrypted
- If they were, mail couldn't be sent

Problem: Key Distribution

- How do the people in the conversation know the key to encrypt and decrypt messages?
  - Send them in unencrypted e-mail?
    - ...not so much.
  - Meet face to face?
    - Then why bother with e-mail?
  - Choose a phrase in a book that both can get access to?
    - Eavesdroppers can torrent ebooks, too.

Solution: Public Key Cryptography

- Every user has two keys
  - Public key used to encrypt
    - Posted everywhere
    - Easy to find and add
  - Secret key used to decrypt
    - Kept secret
    - Kept encrypted with a passphrase for safety
- What is encrypted to a public key is decrypted with a private key
  - The other way around, too
- Messages can be encrypted to multiple keys

How do we actually do this?

- Generate a keypair
- Publish your public key
- Publish the fingerprint of your public key
- Configure your mail client to do the heavy lifting

What is GnuPG?

- http://www.gnupg.org/
- Available from finer package repositories everywhere
- An open source implementation of RFC 4880 (PGP message format)
- Carries out the same tasks as PGP but is truly open (PGP is owned by Symantec)
- Cross platform command line crypto tool that encrypts, decrypts, signs, checks signatures, etc.
- Implements public key crypto, symmetric crypto, digital signatures, message digest functions...
- Designed so other applications can use it to perform crypto tasks
- Front ends and interfaces for many apps exit
- Standard in Linux (used by package managers)

Generate A Keypair

- MacOSX: GPGtools (http://v.gd/MUjZcv)
- Linux and MacOSX - gpg
  - gpg –gen-key
  - Select “(1) RSA and RSA (default)”
  - Keysize: 4096 (no reason not to)
  - Time to expire: 0 (never)
  - Enter a name
  - Enter an e-mail address (important)
  - Enter a comment (optional)
  - Enter a strong passphrase

Generate A Keypair (cont'd)

- Windows – Gpg4win (http://www.gpg4win.org/)
  - Run Kleopatra
  - File → New Certificate
  - “Create a personal OpenPGP key pair”
  - Fill out the form, then click “Advanced Settings”
  - For RSA, DSA, and ElGamal, turn the key sizes all the way up
  - Click “Create Key”
  - Enter a passphrase to protect the private key
  - “Make A Backup Of Your Key Pair…”

Generate A Keypair (cont'd)

- Windows (continued...)
  - Save the backup to removable media
  - “Upload Certificate to Directory Service”
    - Will automatically export your public key to a public server
  - Or, right-click on key, “Export Certificates”
    - Then post key to key servers, websites

Export Your Public Key

- So other people can find and use it
- Copies it into a text file for distribution
- Linux and MacOSX:
  - gpg --armor --output <filename> --export "you@example.com"
- Windows:
  - Right-click on key in GPA or Kleopatra
  - GPA: “Export Keys”
  - Kleopatra: “Export Certificates”
  - Pick a folder, enter a filename
  - Kleopatra: “Upload Certificate to Directory Service”

Extract Key ID

- Key ID is an eight digit hex value that identifies a key
- One key, one key ID, potentially multiple e-mail addresses
- Linux/MacOSX
  - gpg –list-keys “me@example.com” | grep 'pub'
- Windows (GPA and Kleopatra)
  - It's in the “Key ID” column

Publish Key and Fingerprint

Is like “Eliminate moose and squirrel,” but not.

- You can post a key just about anywhere
  - On a website
  - To a key server
  - In a QR code on a business card
- Fingerprints verify that the key hasn't been altered
  - In your e-mail signature
    - Helps tie a public key to an active address
- Cryptographic hash of the public key
  - Linux/MacOSX: gpg –fingerprint “you@example.com”
  - Windows (GPA): Click on key, read fingerprint
  - Windows (Kleopatra): Right-click on key, Details

Encrypt A File

- Linux and MacOSX
  - gpg –recipient “someone@example.com” --encrypt <filename>
- Windows (Gpg4win)
  - Right-click on file, “Sign and encrypt”
  - Optional: “Remove unencrypted original file when done”
  - Pick (a) key(s) to encrypt to, click “Add”
  - Click “Encrypt”
- Result: <filename>.gpg

Decrypt A File

- Linux and MacOSX
  - gpg –output <filename> --decrypt <filename>.gpg
- Windows (Gpg4win)
  - Right-click, “Decrypt and verify”
  - “Decrypt/verify”
- Enter your passphrase when prompted
- Result: <filename>

Configure Your Mail Client

- Allows your client to do the heavy lifting of encrypting and decrypting messages
- Allows your client to verify cryptographic signatures on messages to verify authenticity
- Means you don't have to do it manually
- It will ask you for the passphrase on your secret key beforehand

Clients - Linux

- Mutt
  - Built in
  - Configuration directives are well known and documented
  - Copy-and-paste them into ~/.muttrc
  - http://wiki.mutt.org/?MuttGuide/UseGPG
  - Some distros may package Mutt already configured
  - After writing message, hit 'p', select 'sign and encrypt', give passphrase when it asks
  - Mail is automatically decrypt when received
  - Signatures are automatically checked

Clients: Thunderbird and Seamonkey

- Cross-platform mail clients
- Does it all automatically on a per-account basis
  - Encrypt and decrypt
  - Sign and verify
- Plugin: Enigmail (http://www.enigmail.net/)
- Download: https://addons.mozilla.org/en-us/thunderbird/addon/enigmail/
- Platform dependent (*nix, OSX, Windows) (32-/64-bit)

Enigmail

- http://www.enigmail.net/documentation/quickstart.php
- Tools → Addons → Get Add-ons
- Search for “Enigmail”
- Install from Mozilla Add-On Archive
- Exception: 64-bit Linux
  - Download the .xpi file
  - “Install from file”

Thunderbird: Configure Account

- Edit → Account Settings
- For each configured account, “OpenPGP Security”
- “Enable OpenPGP support”
- “Use specific OpenPGP key ID”
  - Enter your PGP key ID
- “Sign (non-)encrypted messages by default”
- Turn off “Use PGP/MIME by default”

Windows: Outlook

- Gpg4win (http://www.gpg4win.org/): GpgOL
- Plugin for Outlook 2003 and 2007
- Installed automatically
- Serious drawbacks at this time
  - Does not work with Exchange server
  - Outlook 2007 crashes after opening attachments
  - Drafts get saved on server unencrypted
- USE THUNDERBIRD. SORRY.

Authenticity – Did You Really Send That?

- E-mail is trivial to forge
- Encryption doesn't prove identity
  - Anyone can generate a key for anyone
- The user of a public key has to build a verifiable reputation over time
  - Posting signed files
  - Posting signed e-mails to public lists
  - Posting identical copies of public key all over the place (unlikely that every copy will be a fake)

Digital Signatures

- Verifies that a message was signed with a particular public key
  - Ties into the rep that a key has built up
- A message is run through a hash
- Hash is encrypted by the user's private key
- If the hash can be decrypted by the sender's public key, chances are the message was sent by the person behind that public key
- Done automatically by mail client plug-in
- Will prompt you for passphrase

Digital Signatures (cont'd)

- Client will sign your outgoing message
- Client will detect signed messages and verify them
- You'll get a message saying “Good signature” or “Bad signature.” Pay attention.

Best Practices

- The identity associated with your key doesn't have to be your real identity
  - Identities on keys have reputations
  - Post key on websites, keyservers so that as many as possible can find it
- At a minimum, sign everything
- Check signatures on messages received
- If signatures don't match, be concerned
- Encrypt ALL messages to people who encrypt, no matter how trivial
  - Makes it difficult to tell sensitive messages from trivial messages

That's it!

Comments?

Questions?

Suggestions?