Pseudonymous Software Development and Strong Distribution

V. Alex Brennen
codepoet@dublin.ie
What is Pseudonymous Development?
- Tied to a cyberspace identity
- Not tied to a meatspace identity

What is Strong Distribution?
- A distribution model which is cryptographically strong
- Software and Communications protected through the use of strong public key cryptography
- Most commonly PGP or x509
Why do we need It?

- Patent System has long been broken
- Developers may now face imprisonment
  - DMCA (Reverse Engineering)
  - DRM
  - SkipJack
- Developers may now face lawsuits
  - Trade Secrets
  - DMCA
    - RIAA
    - MPAA
  - Unintended Usage
• Other Sources of Chilling Effects
  – Employer Mission Confusion
    • Public University or Private Incubator?
    • Researcher or Entrepreneur?
    • 9-5 Employee or Slave Property?
  – American Entitlement Liability
    • Grand Theft Auto
    • Columbine High School
• Two adversaries
  – The US Gov't
    • The US Gov't is evil, ever present, and ever monitoring
      – You can't win against them
      – There are no secrets from them
  – Corporations
    • Limited by their own property
    • Limited by borders
    • Attempt to get the Gov't to act for them, but it's difficult
• Choose Just Not To Contribute
  – Don't write any code
  – Send small patches and ask not to be named

• Try and Contribute in Secret
  – Separate your on-line identity
  – Contribute under a pseudonym
  – But what if you're discovered and exposed?
  – The subject of this talk!
http://www.freestateproject.org/
• Core Components Necessary
  – Key Server Infrastructure
    • A few different networks
  – Anonymous Email Infrastructure
    • Mixmaster anonymous remailers
  – Anonymous Posting Infrastructure
    • email to usenet gateways
  – Onion Routing Downloading Infrastructure
    • Tor
• Create an Identity
  – openPGP key with no contact information
  – Integrate it with a Web of Trust
• Step 1: Write the Software
• Step 2: Sign A Software Archive
• Step 3: Post Your Public Key
• Step 4: Distribute Software over Pseudonymous Network
• Step 5: Publish a Method of Communicating With You
• Step 6: Rinse and Repeat
• Step 1: Write The Software
  – That's the hard part
• Step 2: Sign a Software Archive
  - `gpg -a -o prog.tar.bz2.sig --detach-sig prog.tar.bz2`
  - Provides Protection on the Server
  - Provides Protection in Transit
  - Provides Protection for the Customer
• Step 3: Post Your Public Key
  – Many Keyserver Networks
    • MIT Keyserver
    • subkeys
    • CryptNET
  – Single Web Site
  – p2p Network
  – Newsgroup
• Step 4: Distribute Software over Pseudonymous Network
  – Tor
  – email to usenet
  – Mirrors to pick it up
  – Major distribution sites
    • SourceForge
• Step 5: Publish a Method of Communicating With You
  – alt.anonymous
  – Steganography
  – Avoid Personal Email
  – Email Lists with Archives OK
• Step 6: Rinse and Repeat
  – New releases made the same way
  – Safely use method for years
  – Alternative method available if a system gets shut down
  – Sneaker net leaks
The patch life cycle

- User gets software and public key
- Writes patch
  - Encrypts with public key
  - Posts cyphertext in public place
- Developer Discovers Cyphertext
  - Decrypt
  - Processes patch
  - Signed security advisory
  - New release w/ fix
• Egoboo
  – Developer asks Potential Employer to Encrypt Secret with project public key and provide cyphertext
  – Developer decrypts secret
  – Message Identifying Developer Signed with Project Keypair
- Arbitrary Statements about Arbitrary Content
  - A hash representation of anything can be digitally signed
  - Signatures can be circulated in detached form
  - Pseudonymous Security Audits
  - Non-Pseudonymous Security Audits
• Project Forking
  – Easy as Generating New Keypair
  – Signature on Keypair Lend Credibility
    • Old project keypair signature
    • Developer signature
• Compromise
  – Identity
  – Project Keypair
  – Developer Keypair
  – Keys in Web of Trust
  – Keypair Revocation is Important
  – Possession of Keypair proof of involvement
• Breaking a Keypair with Factoring
  – Government can do it

\[ O \left( \exp \left( \left( \frac{64}{9} n \right)^{\frac{1}{3}} \left( \log n \right)^{\frac{2}{3}} \right) \right) \]
• Ring Signatures
  – Someone in a Group
    • Signer Hard to Identifiable
  – Secret Leaking Protocols Can Be Instructive
The Strong Distribution HOWTO
http://cryptnet.net/fdp/crypto/strong_distro.html

Guerrilla Software Development HOWTO
http://cryptnet.net/fdp/crypto/guerrilla-devl.html

The Keysigning Party HOWTO
http://cryptnet.net/fdp/crypto/gpg-party.html
V. Alex Brennen
codepoet@dublin.ie