Key server (cryptographic)

From Wikipedia, the free encyclopedia

In computer security, a **key server** is a computer that receives and then serves existing cryptographic keys to users or other programs. The users' programs can be working on the same network as the key server or on another networked computer.

The keys distributed by the key server are almost always provided as part of a cryptographically protected identity certificate containing not only the key but also 'entity' information about the owner of the key. The certificate is usually in a standard format, such as the OpenPGP public key format, the X.509 certificate format, or the PKCS format. Further, the key is almost always a public key for use with an asymmetric key encryption algorithm.

Contents

History

Key servers are made possible by the discovery of public key cryptography. In public key cryptography an individual is able to generate a key pair, where one of the keys is kept private while the other is distributed publicly. Knowledge of the public key does not compromise the security of public key cryptography. An individual holding the public key of a key pair can use that key to carry out cryptographic operations that allow secret communications with or strong authentication of the holder of the matching private key. The need to have the public key of a key pair in order to start communication or verify signatures is a bootstrapping problem. Locating keys on the web or writing to the individual asking them to transmit their public keys can be time consuming and insecure. Key servers act as central repositories to alleviate the need to individually transmit public keys and can act as the root of a chain of trust.

The first web-based PGP keyserver was written for a thesis by Marc Horowitz, while he was studying at MIT. Horowitz's keyserver was called the HKP Keyserver after a web-based OpenPGP HTTP Keyserver Protocol (HKP[1]) it used to allow people to interact with the keyserver. Users were able to upload, download, and search keys either through HKP on port 11371, or through web pages which ran CGI scripts. Before the creation of the HKP Keyserver, keyservers relied on email processing scripts for interaction.

A separate key server, known as the PGP Certificate Server, was developed by PGP, Inc. and was used as the software (through version 2.5.x for the server) for the default key server in PGP through version 8.x (for the client software), keyserver.pgp.com. Network Associates was granted a patent co-authored by Jon Callas (United States Patent 6336186)[2] on the key server concept.

To replace the aging Certificate Server, an LDAP-based key server was redesigned at Network Associates in part by Randy Harmon and Len Sassaman, called PGP Keyserver 7.0. With the release of PGP 6.0, LDAP was the preferred key server interface for Network Associates' PGP versions. This LDAP and LDAPS key server (which also spoke HKP for backwards compatibility, though the protocol was (arguably correctly) referred to as "HTTP" or "HTTPS") also formed the basis for the PGP Administration tools for private key servers in corporate settings, along with a schema for Netscape Directory Server. It was later replaced by the new PGP Corporation Global Directory.

Public versus private keyservers

Many publicly accessible key servers, located around the world, are computers which store and provide OpenPGP keys over the Internet for users of that cryptosystem. In this instance, the computers can be, and mostly are, run by individuals as a pro bono service, facilitating the web of trust model PGP uses.

Several publicly accessible S/MIME key servers (http://wiki.cacert.org/KeyServers) are available to publish or retrieve certificates used with the S/MIME cryptosystem.

There are also multiple proprietary public key infrastructure systems which maintain key servers for their users; those may be private or public, and only the participating users are likely to be aware of those keyservers at all.

Privacy concerns

For many individuals, the purpose of using cryptography is to obtain a higher level of privacy in personal interactions and relationships. It has been pointed out that allowing a public key to be uploaded in a key server when using decentralized web of trust based cryptographic systems, like PGP, may reveal a good deal of information that an individual may wish to have kept private. Since PGP relies on signatures on an individual's public key to determine the authenticity of that key, potential relationships can be revealed by analyzing the signers of a given key. In this way, models of entire social networks can be developed.

Problems with keyservers

The OpenPGP keyservers developed in the 1990s suffered from a few problems. Once a public key has been uploaded, it is difficult to remove. Some users stop using their public keys for various reasons, such as when they forget their pass phrase, or if their private key is compromised or lost. In those cases, it was hard to delete a public key from the server, and even if it were deleted, someone else can upload a fresh copy of the same public key to the server. This leads to an accumulation of old fossil public keys that never go away, a form of "keyserver plaque". Another problem is that anyone can upload a bogus public key to the keyserver, bearing the name of a person who in fact does not own that key. The keyserver had no way to check to see if the key was legitimate.

To solve these problems, PGP Corp developed a new generation of key server, called the PGP Global Directory (https://keyserver.pgp.com/). This keyserver sent an email confirmation request to the putative key owner, asking that person to confirm that the key in question is theirs. If they confirm it, the PGP Global Directory accepts the key. This can be renewed periodically, to prevent the accumulation of keyserver plaque. The result is a higher quality collection of public keys, and each key has been vetted by email with the key's apparent owner. However, it should be pointed out that because PGP Global Directory allows key account maintenance and verifies only by email, not cryptographically, anybody having access to the email account could for example delete a key and upload a bogus one.

The last IETF draft for HKP also defines a distributed key server network, based on DNS SRV records: to find the key of *someone@example.com*, one can ask it to *example.com*'s key server.

Keyserver examples

These are some keyservers that are often used for looking up keys with "gpg --recv-key"

- keyserver keys.gnupg.net (http://keys.gnupg.net)
- keyserver hkp://subkeys.pgp.net (http://subkeys.pgp.net:11371) (server pool)
- keyserver hkp://pgp.mit.edu (http://pgp.mit.edu:11371)
- keyserver hkp://pool.sks-keyservers.net (http://pool.sks-keyservers.net:11371) (server pool)
- keyserver hkp://zimmermann.mayfirst.org (http://zimmermann.mayfirst.org/) (also supports secured key requests over TLS)
- keyserver http://keyserver.ubuntu.com

See also

- Lightweight Directory Access Protocol
- GnuPG

References

1. **^** OpenPGP HTTP Keyserver Protocol (HKP) (http://tools.ietf.org/html/draft-shaw-openpgp-hkp-00)
2. **^** United States Patent 6336186 (http://www.pat2pdf.org/patents/pat6336186.pdf)

External links

- Marc Horowitz's Thesis (http://www.mit.edu/afs/net.mit.edu/project/pks/thesis/paper/thesis.html)
- List of Key Servers (http://www.dmoz.org/Computers/Security/Products_and_Tools/Cryptography/PGP/Key_Servers/) at DMOZ

- OpenPGP HTTP Keyserver Protocol (HKP) (http://tools.ietf.org/html/draft-shaw-openpgp-hkp-00)
- OpenPGP Public Key Server (PKS) (http://sourceforge.net/projects/pks/) - an OpenPGP key server software package distributed under a BSD-style license (with advertising clause). It has largely been supplanted by SKS.
- Synchronizing Key Server (SKS) (https://bitbucket.org/skskeyserver/sks-keyserver/wiki/Home) - an OpenPGP key server software package distributed under the GPL.
- PGP Global Directory (https://keyserver.pgp.com/)
- Pool of SKS Keyservers (http://www.sks-keyservers.net/status/)

Retrieved from "http://en.wikipedia.org/w/index.php?title=Key_server_(cryptographic)&oldid=614964105"
Categories: Key management
