Decentralized package signing

niekt0@hysteria.sk

August 15, 2015

Current situation

- Modern (Linux) distribution have typically some kind of a package system
- Packages are signed by some central server,
- Signatures are (should be) verified by a client before installing a package.

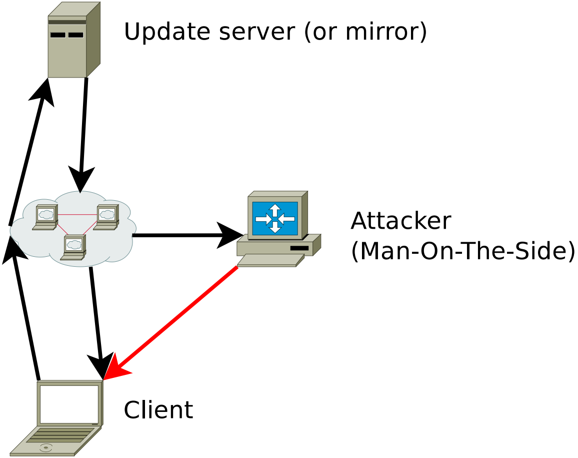
Strong attacker

- Strong attacker (NSA, Chinesee intelligence, GRU, Mossad, ... choose your side:)
- May be able to compromise the repository signing key
- By technical ways (physical access to the signing server, use of valuable zero day, hw backdoor, ...)
- By human factor (Attacker can provide person with enough time, zeal and resources to a community project, lacking these)
- By "legal" ways (release they key and be quiet about it, or you will go to jail (or worse...))
- Typically can do MITM (or "man on the side") on SSL. Have one of the global "authorities" certificate, and required network access
- This can be hard, but the reward (ability to neatly compromise almost any linux (bsd, ...) machine) is sweet enough to try

Attack

- Attacker compromise signing key for a distribution (Debian, CentOS, ...)
- Attacker wait until targeted victim update packages
- Attacker step into SSL communication (if SSL is even used) to place his own package
- Attacker insert his own validly signed package
- Package get installed on a victims system, place a backdoor, clean the evidence

Attack



Update server (or mirror)

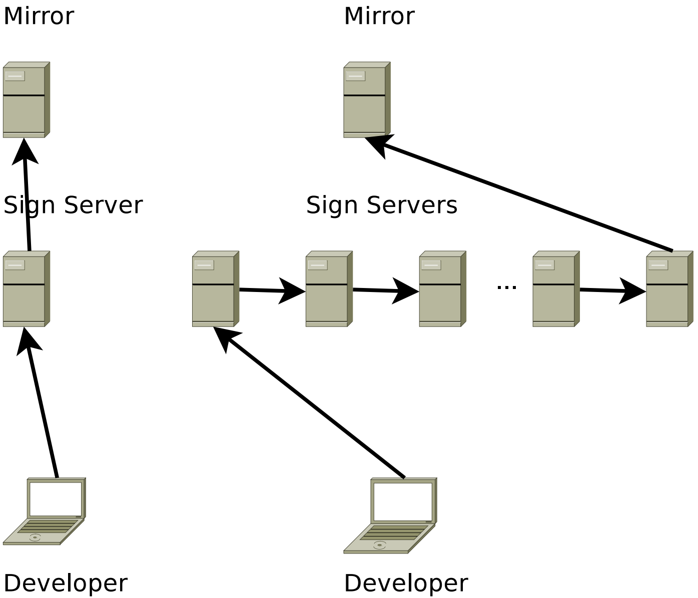
Attacker
(Man-On-The-Side)

Client

Schema

- Multisig schema (for example 5 from 7)
- Each package would be required to went through the signing "round"
- Each of the signing servers would be on a different continent, under different legislative
- For example 1xUSA, 1xEU, 1xRF, 1xChina, 1xBrasil, ...
- Let's assume that each world power can somehow obtain control over signing server on their territory
- Each world power should be also able to prevent other powers from taking control over signing server on their territory
- Or at least it will be hard to capture enough keys to sign an arbitrary package

Schema



Mirror

Mirror

Sign Server

Sign Servers

...

Developer

Developer

Comparison

- Need to compromise single computer vs. need to compromise several computer all around the world (inside ”enemy” territory)
- There is no single entity able to solely make a new valid package
- multisig overhead:
- When signing, package is sent to several computer instead of just one. (negligible slower, a more sophisticated code is needed)
- Verification of signature on client is slightly slower (negligible, when compared to shell-hell of package installers)
- ?