Smoke

An Android Echo Software Application

About

Smoke is an Android communications research project. The software is composed of a single multi-threaded application. A companion application, SmokeStack, provides mobile server services.

Software sources are available at https://github.com/textbrowser/smoke and https://github.com/textbrowser/smokestack.

Android

Smoke has been successfully tested on Android 4.4, 5.0, 6.0, 7.0, and 7.1. Android 4.4 and 5.0 are not officially supported.

According to https://developer.android.com/about/dashboards/index.html, Smoke supports 89.1% of all Android versions.

Database Containers

Most of the database fields contain authentically-encrypted values. Some fields contain keyed digests, including keyed digests of binary (false / true) values. Values are stored as $E(Data, K_e) \parallel HMAC(E(Data, K_e), K_a)$ and $HMAC(Data, K_a)$.

Discovery via Cryptography

Cryptographic discovery is a mechanism which allows servers to lighten the computational and data responsibilities of mobile devices.

Shortly after a Smoke instance connects to a SmokeStack service, the Smoke instance shares some non-private material. The material allows a SmokeStack server to transfer messages to their correct destinations. SmokeStack instances routinely distribute gathered, non-expired material to other SmokeStack services, thus creating a network of cooperative SmokeStacks.

Cryptographic Discovery assumes a trustworthy network.

To mitigate replay attacks, Smoke offers SmokeStack instances random identity streams during message-retrieval requests. The identity streams self-expire.

Fire

Fire creates a communications network between Smoke and Spot-On.

Inflate

Smoke expands text-messaging data to 8192 bytes. If the provided data exceeds 8192 bytes, Smoke expands the provided data by 1024 + mod(data length, 2) bytes.

Ozone Address

An Ozone address is a pseudo-private string which identifies a virtual entity. Smoke and SmokeStack utilize Ozones as a means of retrieving and storing offline messages and public key pairs. Smoke supports one Ozone while SmokeStack supports infinitely many. Ozone addresses must be shared separately. Retrieved messages are only meaningful within the context of a session. It is possible for multiple Smoke parties to house distinct Ozones if common SmokeStack instances are aware of the distinct Ozone addresses.

Please note that public Ozone addresses will introduce denial of service vulnerabilities.

Private Public Key Server

In addition to housing messages, SmokeStack also serves as a private public key server. A SmokeStack administrator is responsible for coordinating the storage of public key pairs of participants. Participants may request public key pairs of specific participants via Ozone addresses.

SipHash Identities

Exchanging public key pairs is often an involved process. Smoke implements the pseudo-random function SipHash so as to simplify the process. The SipHash function generates outputs of 8 bytes (16 characters hexadecimal). These short strings are easily memorized and/or distributed via other communications applications. SipHash identities are generated as follows:

```
id := siphash(public-encryption-key || public-signature-key,

              pbkdf2(sha512(public-encryption-key || public-signature-key), // Salt

                            public-encryption-key || public-signature-key,

                            4096, // Iteration Count

                            128)) // Bits (16 Bytes)
```

Non-confidential authentication and encryption key streams from SipHash identities are generated as follows:

```
keystream1 := pbkdf2(sha512(id), // Salt

                     id,

                     4096,       // Iteration Count

                     160)        // Bits (20 Bytes)

keystream2 := pbkdf2(sha512(id), // Salt

                     keystream1,

                     1,          // Iteration Count

                     768)        // Bits (96 Bytes)
```

Public Data

Chat Encryption Key
Algorithm: RSA
Fingerprint: 27:1e:f6:31:d7:67:4b:b3:82:00:4b:59:3c:d9:16:41:93:47:a5:a3:c5:17:1e:5f:
70:56:06:cc:a8:de:f2:1d:ea:b4:ca:d7:99:34:a0:a6:8f:27:2e:df:9a:78:7c:43:a1:a1:bc:63:3f:51:e2:9a:83:1f:73:66:22:63:01:f6
Format: X.509
Size: 3072
Chat Signature Key
Algorithm: RSA
Fingerprint: a3:42:81:f1:34:d1:dd:c8:2f:2c:1d:a3:c4:95:31:17:79:73:d7:b2:6c:df:9c:91:8e:1c:07:6f:42:af:16:31:4a:9c:
69:7d:d8:b6:de:5f:ab:8a:b1:58:38:ae:96:ec:37:fd:ef:fc:21:3d:a4:c2:db:36:a3:80:92:fb:ee:5e
Format: X.509
Size: 3072
SipHash Chat ID
@39B8-3DE5-A567-9C6F

RESET SMOKE

The transport keys which are generated from SipHash identities may be used for exchanging public-key data via the Echo Public Key Share (EPKS) protocol.

It is impossible to avoid SipHash collisions as there are infinitely-many inputs and a limited number of outputs.

TCP, UDP Protocols

Smoke supports both the TCP and UDP network protocols. Multicast and unicast UDP varieties are provided. Multiple clients may be defined via Settings. A limit on the number of clients is not imposed. When defining neighbors, one may define SmokeStack and/or Spot-On neighbors. SmokeStack, the companion application of Smoke, offers mobile server services as well as message and public-key storage.