HighSide Abbreviated Threat Model

Authored & Maintained by Jonathan Warren, CTO @ HighSide, Inc.
2017-03-08

Security Objectives

- HighSide aims to encrypt and securely transmit data shared between company employees. Data may include messages or files.
- HighSide aims to maintain the non-spoofability of messages; i.e. to prevent malicious actors from sending messages which appear to be sent by a different user.
- HighSide aims to be very difficult to use incorrectly or insecurely.
- In the event of a security breach of a user's keys or data, HighSide aims to restrict the data loss to the one affected user.
- HighSide aims to maintain these security guarantees even if all HighSide servers are compromised.

User Roles

There are three roles:

- Server Operators
- Users
- Administrators

Suppose Acme, Inc. is a company using HighSide. Users and Administrators work for Acme. Server Operators may or may not work for Acme depending on whether or not Acme is self-hosting the HighSide message relay server. Administrators maintain the list of users in the company. The list of users includes the administrators. Server operators may send messages to users but the messages must appear to come from the “Bot” user. This is used for 3rd party integrations. The bot has no way of sending information out of HighSide to anyone. HighSide clients are programmed to reject all messages from any addresses not on the company contact list except for the bot. Server Operators are not capable of changing the company contact list.

Application Overview

HighSide is an end-to-end encrypted messaging client and service. It does *not* depend on SSL for message security but does use SSL for envelope security. The server can and must see the sender and receivers of each message in order to route the messages to the correct destinations.

HighSide is a trust-on-first-use system but has a verification feature built in which all administrators should use. Once an administrator has verified a user, the user will get a twitter-like *verified* icon next to their name in the contact list. Once users have verified the administrator, they will be able to see the various verification icons on other people.

Once a public key is in use, the server cannot change it. If a user loses their private keys, the administrator must remove them from the address book and re-add them. They will effectively be a brand new user who happens to use the same human name as before.

For more details about how the encryption steps works, see this article online: https://HighSide.com/blog/how-HighSide-works/

For more details about the protocol, see the *Protocol Structures* document which we should have sent with this document.

Data is stored locally. On the mobile apps it is stored in plaintext although Operating System-provide user-data encryption should be used. (Apple has a good way of keeping the local encryption key secure.)

In the desktop app, certain fields in the database are encrypted with a server-stored key called the local storage key. The local storage key is delivered on login. This way if an attacker steals a powered-off laptop and the administrator removes the affected user from the address book, the attacker would not be able to read any sensitive data on the laptop because he would not be able to authenticate and retrieve the local storage key.

Threat 1:

Attacker gains administrative access to an employee's computer.

Damage Potential

Score: 4

The damage depends entirely on whether the administrator removes the user from the address book before the thief turns on and uses the laptop. If the admin beats the thief then the damage potential is 0; the attacker can only see that messages were sent between this user and other users. If the thief beats the admin then messages and files sent by or to the user, either in a direct message or in a group, will be compromised. The compromise will continue until it is discovered.

On mobile, if the user is not encrypting their locally stored data, the damage potential score is 5: the users messages and files will be compromised.

Reproducibility

Score: 2
Gaining access to an employee's computer is thought to be generally difficult. The attacker would likely have to be an employee at the same company in order to gain physical access to the computer.

Exploitability

Score: 1
Gaining access remotely would be very difficult. Easier attack methods include simply breaking into the office or home and stealing the computer.

Affected Users

Score: 1
The only user affected would be the user in question. Administrators are not capable of reading messages sent between employees unless the administrator is a normal recipient of each message. Server operators cannot read any messages.

Discoverability

Score: 9
It is common sense that if an attacker can access a running computer then they can read the screen of that computer, access its locally stored files, and log the user’s keystrokes. It is also common sense that if a disk is not encrypted then the attacker would be able to read its contents.

Risk_DREAD = 3.4

Threat 2:

Attacker tricks employee into giving him their import key.

A user's encryption and signing keys are derived from their import key. The user can use their import key to import their messages on another device or after accidentally destroying their primary device.

Damage Potential

Score: 6
Messages and files sent by or to the user, either in a direct message or in a group, will be compromised. The compromise will continue until it is discovered at which point the encrypted keys can be changed.

Reproducibility

Score: 1
This would be a social engineering attack and has no purely technical manifestation. Also, attackers can not contact employees through HighSide; they would have to resort to a different form of communication like email.

Exploitability

Score: 3
The attacker would need to have well-developed social engineering skills to target a particular person. The person would also have to fall for it.

Affected Users

Score: 1
The only user affected would be the user in question. Administrators are not capable of reading messages sent between employees unless the administrator is a normal recipient of each message. Server operators cannot read any messages.

Discoverability

Score: 9
It is common sense to any potential attacker that if they are able to trick a user into sharing their keys then they may be able to access their messages.

Risk_DREAD = 4

Comments:

To try to mitigate this risk, HighSide displays this text in the client when showing the 12-word seed key:

“Here are your twelve words. Keep them private; do not share them with any employee of your company or any employee of HighSide. We will never ask for them.”

Threat 3:

*Attacker hacks the server and, **later,** a user’s machine.*

HighSide does not have perfect forward secrecy. This was a conscious decision; we knew that some people would want it and made a conscious decision not to support it in order to support other features. What we gain is the ability for an individual to send and receive messages on multiple devices without an enormous amount of complexity. If a user inputs his import key into a new device, all of his messages will show up even if his original device is destroyed or turned off. We also can allow users to message each other when the receiving party is not online. This is especially important in group messaging as it is common that not all group members are online at the same time. Ultimately we feel that not having forward secrecy does not come at a large cost and is dramatically outweighed by the benefits. If we provide a service which is useful and functional to many more end-users so that they’ll actually use it, they will be better off overall.

Damage Potential

Score: 4

Messages and files sent by or to the user prior and up to the point of the server hack, either in a direct message or in a group, will be compromised.

Reproducibility

Score: 1

The attacker would have to either hack the message relay server OR the load balancer (to get the private SSL key) and then mount a man-in-the-middle attack against their target. Acquiring a legally invalid but mathematically valid SSL cert would not be sufficient since certificate pinning is used by clients.

Exploitability

Score: 0
Attack requires advanced programming and networking knowledge with custom or advanced tools.

Affected Users

Score: 3
The only user affected would be the user whose machine was hacked although there would then be a larger threat if further users' computers are hacked.

Discoverability

Score: 8
Anyone who knows how the system works could potentially figure out this attack. The fact that the system does not have perfect forward secrecy is mentioned in several places on the website.

Risk_DREAD = 3.2

STRIDE

STRIDE is a classification scheme for characterizing known threats according to the kinds of exploit that are used (or motivation of the attacker). The STRIDE acronym is formed from the first letter of each of the categories below. Here we address all of these threats.

Spoofing Identity

Spoofing an identity is not possible in HighSide. All messages are cryptographically signed. Encryption and signing keys are hashed and displayed as an address. Administrators are tasked with verifying that this address is correct once for each employee.

Tampering with Data

Tampering with data is not possible in HighSide. All messages are cryptographically signed and the signature is checked before any decryption operation is attempted. If a server operator or hacker modifies a message before relaying it to a recipient, the message will be silently rejected. If a

server operator or hacker modifies some attachment data before relaying it to a recipient, the recipient will assume that the error occurred as the result of a transmission error and will re-request the affected file part. The client will never save the tampered data to disk.

Repudiation

HighSide does not offer message non-repudiation; all messages have a sender, and receivers can prove that the sender sent the message. Although the following is not currently implemented, receivers could potentially trivially prove to *other* people that the sender sent the message by either saving the signature ahead of time or by redownload the message from the server and supplying the curious party with the signature.

Information Disclosure

The entire purpose of HighSide is to limit information disclosure. There are no known threats except those described elsewhere in this document.

Denial of Service

HighSide's website, used for managing billing, is behind CloudFlare thus limiting DoS attacks. Unfortunately CloudFlare does not offer DoS protection for non-HTTP connections and thus the message relay server is vulnerable to DoS attacks possibly without even great resources. If the message relay server is self-hosted by a company then an attack of HighSide, Inc.'s server would not affect the operation of said company's server.

Elevation of Privilege

In HighSide, the only way that a user may be elevated to the role of an administrator is for an existing administrator to promote him. This is isn’t currently implemented in the desktop client but *will* be implemented in the form of a cryptographically signed message from the existing administrator which is distributed down to all members of the company saying that this new individual is now an administrator. Thus the user nor even a server operator may promote a user to be an administrator on their own.

Other issues disclosed

The website does not use certificate pinning.

> At this time, although we do have our own valid SSL certificate behind CloudFlare, CloudFlare uses its own cert for users which cannot be pinned. It wouldn't provide much value anyway: most users are likely to only visit the website once. Once they download the client, the security of their messages and files does not depend on the security of the website. Windows and OSX auto-updates are signed by the developers; the security of SSL isn't critical for updates except for Ubuntu users who currently update manually. The only other thing that website certificate pinning could possibly protect is the administrator entering their credit card information for billing purposes. A great many financial websites which collect much more information from users do not use certificate pinning.