Pact.online Security Whitepaper

This whitepaper applies to the decentralized file transfer Pact.online

----

October 2018

Version 0.11

Author: David Hawig

Content

1. Introduction

This paper outlines Pact Care's approach to security and compliance for the file transfer tool Pact.online.

Pact.online is a decentralized website without a backend, which runs on the InterPlanetary File System (IPFS). It uses the Web Crypto API (AES256-GCM) to fully encrypt the files inside the browser before they get uploaded as well as the IOTA[1] protocol as a decentralized immutable logging system.

By definition, information security is the balanced protection of the confidentiality, integrity, and availability of data. This is also known as the as the CIA triad2 and the core concept behind ISO/IEC 27001:2013[3]. The aim of Pact.online is to fulfill all three aspects by the technology itself and therefore to be independent both of the structure of the organization as well as of the implementation of a risk management processes. This way we aim to archive the highest possible security and reduce the typical risk associated with cloud computing.

This paper will be divided into four sections. The first three will cover the three pillars of the CIA triad in respect to the file transfer and the logging system features of Pact.online. In the last section, we will list additional security risks, which are not directly related to the CIA triad.

2. Confidentiality

"Confidentiality is the prevention of unauthorized disclosure of information." [4]

2.1 File transfer

To protect the data from unauthorized viewers the file needs to be encrypted. To this end, Pact.online uses the Web Cryptography API[5], a relatively new API published by the World Wide Web Consortium (W3C) on the 26th of January 2017. Among the supported cryptographic algorithms, Pact.online uses the official recommended functions of AES256-GCM[6] with a tag length of 128. This encryption algorithm is, for example, suggested by the German Medical Association and adopted by the U.S. government.

As part of the encryption process, the website randomly generates a new private/public key combination for every uploaded file[7]. The private key is not automatically stored nor shared, and the user is ultimately responsible for taking care of the appropriate measures to keep and share it safely. Pact.online, recommends using end-to-end encrypted messengers, like Telegram or WhatsApp, to share the private key. Additionally, we try to make the sharing process as user-friendly as possible to reduce the security risks even further. However, ultimately the user is always responsible for taking appropriate care of the private key. It is

[1] https://www.iota.org/

[2] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.207.9119&rep=rep1&type=pdf, P. 257

[3] https://www.iso.org/obp/ui/#iso:std:iso-iec:27001:ed-2:v1:en

[4] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.207.9119&rep=rep1&type=pdf, P. 257

[5] https://www.w3.org/TR/WebCryptoAPI/

[6] https://diafygi.github.io/webcrypto-examples/

[7] The main reason therefore is that the website operates without any backend and because of that, there isn't a secure way of storing login credentials or passwords permanent.

important to notice that if the user uses two different channels for sharing the file hash and the private key an attacker needs to have access to both channels.

Another potential attack vector is the receiver of the file. If the receiver gets both messages on the same device, for example, his smartphone, and the device, as well as the messaging services, are not locked with a pin or something similar, anyone with access to this device can also get access to the file itself.

Furthermore, the website can be attacked by malicious browsers or browser extensions, which can read the private key. The constant changing of the key pair reduces the risk insofar, that once the malicious software is installed in the browser it doesn't get automatically access to all previous files.

To reduce the risk of cross-site scripting (XSS) attacks Pact.online uses only a small amount of necessary *npm* packages and external code. The external code is hereby integrated into the HTML page directly and not loaded from a *content delivery network* (CDN) to reduce the security risks even further. Moreover, we constantly try to minimize the integration of code created by external parties. Pact.online for example, doesn't use any *jQuery* code.

2.2 Logging system

A log entry contains the following details:

- Log Id
- File hash
- Time
- IPFS Gateway
- Upload or Download
- Encrypted or not encrypted
- Signature
- Page Signature

As it can been seen, these details do not include any personal information. Additionally, the log entry does not disclose nor allow to derive any information about the actual file being shared. Therefore, the confidentiality of the data is not breached by using this logging system.

3. Integrity

"Integrity is the guarantee that the message that is sent is the same as the message received and that the message is not altered in transit."[8]

3.1 File transfer

On the one hand, the integrity of the data is ensured by the end-to-end encryption, on the other hand, every file uploaded on Pact.online generates a unique hash, which guarantees that the file cannot be altered even if it is not encrypted (i.e. any change in the file itself would result in a different file hash). This hashing system ensures that the receiver will only

[8] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.207.9119&rep=rep1&type=pdf, P. 258

download the file with the hash shared by the sender, and therefore that only the original intended version is accessed.

In this case, the only potential attack vector is during actual sharing of the file hash from the sender to the receiver. If someone can intercept and alter the message (i.e. email), they can previously upload a different file and integrate the hash of this fake file into the message. If the file transfer was also encrypted and sent via two different, hopefully highly secure channels, this threat is close to nonexistent, since an attacker needs to have access to both channels at the same time. If the file is unencrypted and the confidentiality is of high importance, it might be a good idea to share the file hash via multiple channels or public channels. This way it becomes a lot more difficult for an attacker to replace a hash during the transit.

3.2 Logging system

Since the log entries are stored on an immutable ledger called IOTA the integrity of the entry is ensured. Compared to the IPFS based file transfer it's not even possible to delete the entry once it's created. Additionally, as seen above, the log entry contains both a unique file hash and a signature, which can be used to prove the ownership of a transfer as well as the ownership of the file itself.

4. Availability

"Availability is the guarantee that information will be available to the consumer in a timely and uninterrupted manner when it is needed regardless of the location of the user."[9]

4.1 File transfer

As mentioned earlier, Pact.online makes use of the IPFS network.

This means that, in one hand, the website can run on any writable IPFS gateway (i.e. if the domain Pact.online is opened, the user will be redirected to a randomly selected version of the website). This makes it virtually impossible to block its file transfers. Even if someone blocks the main redirect page Pact.online, users can still open any writable IPFS gateway directly and send or access their files.

It is important to note here that even we cannot prevent the user from sharing and receiving files with Pact.online. Compared to almost all other available file transfer solutions, attacks on our servers or other problems on our side cannot prevent users from getting access to their transfers.

On the other hand, IPFS uses a name-based system, instead of the typical location-based system of the world wide web. The name is a unique "hash", which is a combination of letters and numbers. Because information is requested based on names instead of locations, transfers can potentially be a lot faster, if the two parties, which share the files are close to each other. Ultimately, this leads to a higher availability according to the above definition.

[9] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.207.9119&rep=rep1&type=pdf, P. 258

4.2 Logging system

The distributed ledger IOTA, which contains the log entry can be accessed via a huge number of different nodes and therefore the availability of the log entry is ensured and close to unblockable. However, at the moment the log entries on IOTA will be deleted after a certain time and won't be stored indefinitely on the ledger. Therefore, it is highly recommended to download the generated CSV files from time to time, which contain all the information about the logs. It is expected that the IOTA Foundation will release a solution for this problem in the coming month (see the official IOTA Development Roadmap[10]).

5. Additional Security Risks

In this section additional security risks will be discussed. These risks are outside of the CIA triad, but can indirectly impact one or more of its three components.

First, one frequent risk often associated with software development is that users have to trust the developers of the tool. In the case of open source software, which you need to install, the user can check the complete source code first and decide then whether to install the controlled program code. However, this is something the user cannot easily do with regular websites since the provider of the website can change the code without any noticeable changes. With Pact.online it is different. Since the website is running completely on IPFS every version has its own unique hash. This means users can still check the source code of a specific version themselves and in the future keep using this same version.

Second, is the generation of fake log entries. Since in theory, every user is in complete control of the website and its source code, it is also possible to fake the log entry itself with the help of the website. Furthermore, anyone can create similar entries on the distributed ledger IOTA without specifically using Pact.online (e.g.change the timestamp or the hash of the entry). However, because every log entry has a unique signature, the user immediately knows in this case who is responsible for the forgery of the entry. In general, in cases of distrust, it is a good idea to simply check the log entries.

Third, is the false claim of a transfer or file ownership. The key pair of every signature changes automatically with every new log entry. The public key to claim the ownership of the transfer is stored in the local storage of the browser. Because of this. in theory, it is possible to immediately read out the public key of someone else and claim the ownership of the transfer as well as the file. Of course, if the person claiming the ownership doesn't have access to the file itself the claim can easily be disproven.

Finally, it is also important to note that the user of Pact.online can still be attacked by social engineering or simply malicious software on his local device. It is a top priority for PACT Care to reduce this risk by improving the UX and by guiding users during the file transfer process and making it as user-friendly as possible, without being too complex.

[10] https://blog.iota.org/iota-development-roadmap-74741f37ed01