Pact.online Security Whitepaper

This whitepaper applies to the decentralized file transfer Pact.online

---

October 2018

Version 0.1

Author: David Hawig

Inhalt

Pact.

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.

Traditionally 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 independent of the structure of the organization or 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. Therefore, the paper will be structured according to the three pillars of the CIA triad. Each topic will be divided into the file transfer part and the logging system of Pact.online.

Nevertheless, the user of the tool of course still can be attacked by social engineering or simply malicious software on his local device. The goal of Pact.online is therefore also to reduce this risk by making the file transfer easy and guide users during the process.

At the end of the paper, 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. Therefore Pact.online uses the Web Cryptography API[5]. The API is relatively new and the World Wide Web Consortium (W3C) released its recommendation on 26 January 2017. Of all the supported cryptographic algorithms Pact.online only uses the official recommended functions of AES256-GCM[6] with a tag length of 128. The same encryption algorithm is, for example, recommended by the German Medical Association and adopted by the U.S. government.

The website randomly generates a new private/public key combination for every uploaded file[7]. The private key is not automatically stored or shared, and the user is ultimately responsible to take care of the appropriate measures to keep the private key safe.

---

[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.

Pact.online, therefore, 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's 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 information:

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

Since it's impossible to derive any information from the file hash about the actual file without having access to this data, the log entry doesn't contain any personal information and because of this doesn't disclose any private information.

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]

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

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 ensures you cannot alter the file even if it is not encrypted. The reason, therefore, is any change in the file itself would result in a different file hash. However, since the receiver requests the file based on the shared file hash, he always downloads the originally intended version.

In this case, the only potential attack vector is the sharing of the file hash from one user to another user. If someone can intercept and alter the message, 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. Furthermore, as seen above the log entry not only can be used to prove the ownership of a transfer but also to prove the ownership of the file itself. The reason, therefore, is that the log entry contains the unique file hash as well as a signature.

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 before Pact.online makes use of the IPFS network and can run on any writable IPFS gateway. If you open the domain pact.online you will be redirected to a random select version of the website. This way it's basically impossible to block the file transfers. Even if someone blocks the main redirect page pact.online, you could still open any writable IPFS gateway directly and this way access your files.

It's important to mention at this stage that even we cannot prevent you from sharing and receiving files with Pact.online. Compared to almost all other available file transfers, attacks on our servers or other problems on our side cannot prevent you from getting access to your transfers.

Apart from this, 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

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

locations, transfers can 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.

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's highly recommended to download the generated CSV files from time to time, which contain all the information about the logs. It's expected that the IOTA foundation will release a solution for this problem in the coming month.

5. Additional Security Risks

One security risk that almost always exists is that you have to trust the developers of the tool. In the case of open source software, which you need to install, you can check the complete source code first and then install this specific version of the controlled program code. This is something you usually cannot easily do with regular websites since the provider of the website can change the code without you immediately noticing any change. However, with Pact.online it's different. Since the website is running completely on IPFS every version has its own unique hash. This means you still can check the source code of a specific version yourself and in the future keep using this same version.

Another security risk is the generation of fake log entries. Since in theory, every user is in complete control of the website and its source code, it's 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. For example, change the timestamp or the hash of the entry. However, because every log entry has a unique signature, you immediately know in this case who is responsible for the forgery of the entry. In general, it's a good idea in cases where you do not fully trust one party to simply check the log entries.

The key pay 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's possible to immediately read out the public key of somebody 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.