1Password for Windows Security Assessment

Project No. 378.2203
Report
FINAL

for

AgileBits Inc dba 1Password
4711 Yonge St., 10th Floor
Toronto, ON M2N 6K8
Canada

Document Versions and Changes

Version	Author	Date	Comment
0.1	Andreas Lindh	2022-09-09	Initial draft
0.2	Johan Rydberg Möller	2022-09-09	Additions
0.3	Lucas Humfeldt	2022-09-12	Technical review
0.4	Andreas Lindh	2022-09-12	Clarifications
0.5	Nico Lindner	2022-09-12	Editorial review
0.6	Florian Grunert	2022-09-29	Update after feedback from AgileBits
0.7	Nico Lindner	2023-02-01	Update after feedback from AgileBits
0.8	Nico Lindner	2023-02-03	Adding feedback from AgileBits to chapters 3.1 and 3.2
0.9	Nico Lindner	2023-02-03	Commenting on the provided feedback in chapters 3.1 and 3.2
1.0	Nico Lindner	2023-02-03	Final version based on v0.9

Table of Contents

Terms and Definitions

Term	Definition
API	Application Programming Interface
CSV	Comma Separated Value
DoS	Denial-of-Service
ID	Identification
PIN	Personal Identification Number
ZIP	ZIP (compressed archive file format)

1 Executive Summary

AgileBits requested consultation by Recurity Labs to perform a security review of 1Password for Windows. The goal of the assessment was to target a set of new functions, as well as the biometric authentication mechanism. Testers were given access to user accounts, source code and documentation, and used multiple techniques to test these functions as thoroughly as possible within the allotted time.

Two issues were noted during this test, neither of them with a risk rating higher than medium. They consist of an observation regarding a possible degradation in the application's security-level when biometric authentication is used, as Windows Hello (by default) allows for a fallback authentication scheme consisting of a 4-digit pin code. The other issue concerns a parsing error when importing CSV files, resulting in a potential leakage of sensitive information to local users. In summary, no critical issues were discovered and the security of the code base appears strong.

1.1 Table of Findings

The following table summarizes the findings Recurity Labs made during the assessment. The individual results were evaluated according to CVSSv3.1[1] on request by AgileBits. The CVSSv3.1 vector used for the calculation can be found in section *Overview* of the respective finding(s), detailed in the sub-chapters of section 3 of this document.

ID	Description	Chapter	CVSS	Severity
Findings in Detail				
378.2203.1	Import Parser Quotation Injection	3.1	4.4	Medium
378.2203.2	Windows Hello Authentication Degradation	3.2	N/A	N/A

1.1.1 Qualitative Severity Rating Scale

All CVSS scores can be mapped to the qualitative ratings defined in the table[2] below:

CVSS Score	Rating
0.0	None
0.1 - 3.9	Low
4.0 - 6.9	Medium
7.0 - 8.9	High
9.0 - 10.0	Critical

1 https://www.first.org/cvss/v3-1/
2 https://www.first.org/cvss/specification-document, chapter 5

2 Project Background

Recurity Labs was tasked by AgileBits to perform a security review of 1Password for Windows, focusing on a number of new features, as well as the biometric authentication mechanism.

2.1 Team

The assessment was performed by Andreas Lind and Johan Rydberg Möller of Recurity Labs between August 29th to September 09th, 2022.

2.2 Analyzed System

1Password for Windows was assessed in version 8.8.0, and was tested on Windows 11, version 10.0.22000. The backend environment was `b5test`.

Source code was provided to the testers as a ZIP with the following sha256 hash:

```
6d4f21a1d780f89358e2cc25e48854c6a535de072bddcbc41b874ccc7931caf2
```

The following output from the tool cloc[3] provides a high-level overview of the source code made available:

```
github.com/AlDanial/cloc v 1.92  T=13.68 s (801.9 files/s, 111081.7 lines/s)
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
JSON                          4113             35              0         615620
Rust                          1703          43809          43274         360069
TypeScript                     969          12991           8317          98720
XML                           1141           4743            216          53479
Swift                          655           9261           7554          50110
Kotlin                         459           5291           1815          37315
YAML                            30           3326            393          30243
JavaScript                      84           5589           4326          26034
Markdown                       220           7702              0          25512
SCSS                           325           3586            546          19695
SVG                            714              2              7          10614
TOML                           254            784            314           5071
HTML                            28            370             21           3783
Go                              38            329            208           2229
C                                5            423            236           1983
Python                          21            445            191           1723
make                            22            427            196           1441
Ruby                            27            343             49           1194
Jupyter Notebook                 5              0            931            881
Bourne Shell                    32            219            161            827
Svelte                          18            119              0            739
Bourne Again Shell              30            211            215            705
Gradle                           6             89             49            545
SQL                             22            116             48            348
C/C++ Header                    20            133            225            345
PowerShell                       9             69             29            268
Objective-C                      3             53             22            216
CSS                              7             26             20            139
DOS Batch                        2             23              2             61
Dockerfile                       3             15             10             28
EJS                              1              3              6             15
Properties                       2              0             18             11
C#                               1              0              0              7
CSV                              1              0              0              3
```

3 https://github.com/AlDanial/cloc

```
Windows Resource File                1             0             0             1
ProGuard                             1             3            18             0
-------------------------------------------------------------------------------
SUM:                             10972        100535         69417       1349974
-------------------------------------------------------------------------------
```

2.3 Procedures

The assessment focused on the access and protection of vaults and their contents, but also on the security of the application from a design standpoint, in terms of input handling, storage, and more. Please note that the backend API has not been in-scope for this review, but has been considered as a vector for attacking the client, and has been utilized as such.

Prior to the assessment, AgileBits provided a list of features, which should be in-scope[4]. The following is a listing of the in-scope features and a high-level representation of the review tasks within the executed assessment.

- Data Import

Parsing of imported data, storage and filesystem interactions related to import and export activities, vault and collection separation after and during imports.

- Secure file attachments

Access protection mechanisms related to secure file attachments, content parsing and rendering of secure file attachments, and storage of file attachments.

- Move item / share items

Access protection mechanisms related to shared and moved items (both in terms of accessing items and sharing or moving them), storage and file system activity during the moving and sharing process.

- Family / Shared Vaults + New sharing details

Access protection mechanisms related to family and shared vaults, storage of shared vaults, and file system activity during the process of sharing and accessing family shared vaults.

- Travel Mode

Access restrictions triggered by travel mode, storage and filesystem activity regarding vaults, which are not set as "safe for travel".

- Password history

Access protection mechanisms related to password history, storage and filesystem activity (cache and log analysis, and more).

- Item archiving and deletion features

Access protection mechanisms related to archived and deleted items, storage of mentioned items, and file system activity during the process of archiving and deleting items.

- Biometric Unlock (Time Allotting)

Implementation of Windows Hello support, both in terms of best practice and from an application-specific design standpoint.

The above-named features represent the main focus items of the present assessment, although other areas have also been explored, when deemed relevant.

Testers were also provided with a tool called `op-internal`, which was used to interact directly with the B5 backend.

4 povided via 'Q3-22_1Password_8_for_Windows-1.pdf'

Static methods, such as source code analysis and disassembly, and dynamic analysis using reverse proxies, debuggers, API monitoring, file system analysis, and more, have been employed for this review. The tests have been performed both on dedicated Windows hardware with support for biometric authentication and in a virtualized environment.

3 Findings in Detail

This section provides technical details on the findings made during this security assessment. Each finding is described and rated according to the following criteria: vulnerability type, CVSSv3.1 base score and CVSSv3.1 vector.

Please note that the finding IDs mentioned in the following chapters do not claim to be sequential, but are solely meant to be unique. Potential gaps in the numbering scheme of finding IDs do not indicate or constitute an error. When providing feedback, please reference the *Finding ID* rather than chapter numbers.

3.1 Import Parser Quotation Injection

Overview

ID	378.2203.1
Type	Code
CVSS Score	4.4 (Medium)
CVSS Metrics	CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:L/I:N/A:L
Location	Import Data

Details

When importing data into 1Password for Windows using a CSV file, the parser will extract the different segments of the file and accordingly place them in the application's database. However, if a "Note" has a single quotation mark (") in it, the parser will treat whatever comes afterwards as part of the note (up until a second quotation mark is encountered), including newlines and subsequent import items. This results in sensitive information, such as passwords, being shown in clear-text within the client.

The following screenshot served as an example for the observed behaviour:

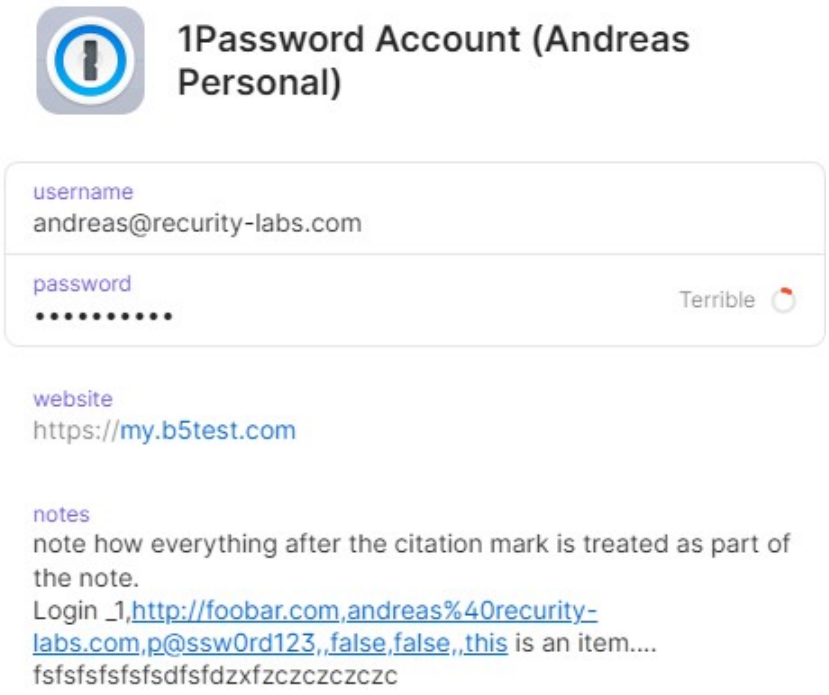

Figure 1 - Import parser quotation injection

The image above is the result of the following import file:

```
Title,Url,Username,Password,OTPAuth,Favorite,Archived,Tags,Notes
1Password Account (Andreas Personal),https://my.b5test.com,andreas@recurity-
labs.com,p@ssw0rd123,,false,false,Starter Kit,"note how everything after the quotation
mark is treated as part of the note.
Login _1,http://foobar.com,andreas@recurity-labs.com,p@ssw0rd123,,false,false,,this is
an item....    fsfsfsfsfsfsdfsfdzxfzczczczczc
```

This introduces the risk of someone being able to "shoulder surf" to read sensitive information, e.g. a password, and also (to some extent) constitutes a Denial-of-Service (DoS) condition, as some items may not be properly imported.

Reproduction Steps

To reproduce the issue, please consider the following steps:

- Within 1Password for Windows, create an export CSV file via the drop-down menu.
- Edit the file and add a single quotation mark to a "Note", which is generally the last field per line.
  - Ensure that there are more items to be imported after this note.
- Import the CSV file and note the result.

Recommendation

It is recommended to treat quotation marks, as well as other special characters, the same as any other character when parsing import data.

Feedback provided by AgileBits (2023-01-20)

```
We accept this finding and immediately implemented a fix within our product.
```

Comment by Recurity Labs (2023-02-03)

Once implemented, a retest of the issue is recommended.

3.2 Windows Hello Authentication Degradation

Overview

ID	378.2203.2
Type	Observation
CVSS Score	N/A (N/A)
CVSS Metrics	N/A
Location	Login

Details

By utilizing Windows Hello for unlocking 1Password, the user is able to use biometrics, such as their fingerprint(s) to unlock the application. However, as biometrics are typically prone to false negatives, Windows requires the user to also register a secondary authentication method, typically a 4-digit PIN code.

The result is that, while the user will be able to use a biometric method to unlock the application, they can also use whatever other Windows Hello authentication methods are configured by the user. This means that the application could be unlocked simply by using a PIN code, which weakens the protection of the application.

Furthermore, the 1Password application has no way of "knowing" if the logged-in application user is the same, who is logged-in to the Windows computer, on which the application is running. Similarly, Windows Hello will simply check if the credentials match the currently logged-in Windows user, when queried by 1Password.

In an edge case scenario, someone could be tricked into logging-on to 1Password on someone else's computer, and then locking it rather than signing-out. In this case, the person logged-in to the Windows computer could unlock the instance of 1Password, simply by choosing to unlock using Windows Hello.

It should be noted that these observations are only applicable when an instance of 1Password is already running on a computer, as the account password must always be entered when starting the application.

Reproduction Steps

The issue can be reproduced by enabling Windows Hello unlocking on a machine running 1Password for Windows and then locking the application.

By clicking the "happy face" in the authentication window, the available Windows Hello authentication methods will be presented.

Recommendation

As the above observations are mainly an issue due to design choices, both in Windows and 1Password, it is difficult to point to a clear and precise recommendation. One path could, however, be to investigate whether it is possible to only allow the use of Windows Hello with biometrics, rather than all available methods.

Feedback provided by AgileBits (2023-01-20)

```
We've reviewed this finding. While we consider the observation valid, we are unable to
fix this finding as there is currently no API present in the Windows Runtime that allows
us to restrict authorization prompts to only use face or fingerprint recognition. We do
address in our support documentation that PINs can be used to access Windows Hello and
recommend users make their PIN strong and memorable, potentially even considering using
the 1Password generator to generate it: https://support.1password.com/windows-hello-
security/#protect-yourself-when-using-windows-hello
```

Comment by Recurity Labs (2023-02-03)

This issue was intentionally filed as an *Observation* since Recurity Labs is well aware and fully agrees with the remarks and facts stated in the above feedback provided by AgileBits.

Since there currently is no straight-forward fix for this issue, Recurity Labs considers addressing it - in the proposed form of related sections in the support documentation - to be a viable solution.