Recurity Labs GmbH
https://www.recurity-labs.com

Secrets Automation Security Assessment

Project No. 378.2208
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	Bruno Kirschner	2023-05-23	Initial draft
0.2	Stefan Seefeldt	2023-05-30	Additions
0.3	Nico Lindner	2023-05-31	Editorial review
0.4	Nico Lindner	2023-07-06	Incorporating feedback
1.0	Nico Lindner	2023-07-06	Final version

Table of Contents

Terms and Definitions

Term	Definition
HTTPS	HTTP over SSL
ID	Identification
TLS	Transport Layer Security

1 Executive Summary

AgileBits tasked Recurity Labs to perform a time-boxed best-effort security assessment of the *Secrets Automation* feature of the B5 solution. Specifically, the *Service Accounts* feature and the newly developed *Connect Server* feature was put in-scope by AgileBits.

This assessment was performed in the context of the regular review policy followed by AgileBits. In conformance with this policy, Recurity Labs put the focus on the tooling and APIs related to the new *Connect Server* feature. Due to the large size and complexity of the solution in-scope, and the limited time available for this project, Recurity Labs did not assess the *Service Accounts* feature.

Summarizing the assessment, the security posture of the solution was found to be in good condition. Only one issue has been identified, resulting from the design of an offline-capability of the *Connect Server*, as documented in chapter 3.1.

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
378.2208.001	Missing Synchronization Enforcement	3.1	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

AgileBits requested Recurity Labs to perform a security review of the *Secrets Automation* feature set within the ecosystem of the *B5* Cloud solution, and the related *Connect Server* application. This assessment was conducted with a time-boxed, best-effort, risk-based approach within the regular review process established by AgileBits.

2.1 Team

The security assessment has been conducted between May 15th and May 26th in 2023 by Stefan Seefeldt and Bruno Kirschner of Recurity Labs. Support was provided by a dedicated team at AgileBits, and Florian Grunert of Recurity Labs as responsible project manager.

2.2 Analyzed System

The review was performed against the *B5* test environment available at the domain `b5test.com`, local setups of the new *Connect Server* solution, including the `op-connect` and `op-sync` applications, as well as the `op` command line tool.

User accounts were created by Recurity Labs at the *B5* test environment at `b5test.com`, utilizing the following email addresses, provided in the present report to aid AgileBits in their cleanup tasks:

- `stefan1@recurity-labs.com` and `stefan2@recurity-labs.com`
- `bruno@recurity-labs.com` to `bruno5@recurity-labs.com`

The source code review portion of the assessment utilized the provided source code labeled with the following tags and commit IDs

- B5 Web application: `b5-b5app-release-1507`
- op cli: `f04931c110bc6665c9021923f141deea8714daa3`

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

```
> scc ./source/

───────────────────────────────────────────────────────────────────────────────
Language                 Files     Lines   Blanks  Comments     Code Complexity
───────────────────────────────────────────────────────────────────────────────
Go                       23721   7096062   703692   1024318  5368052     780730
TypeScript                2948    413545    40653     21138   351754      27384
Go Template               1232    121645    11030       941   109674       4557
Markdown                  1212    142849    38357         0   104492          0
License                    737     60293    10513         0    49780          0
SVG                        584       892       29         1      862          0
Sass                       496     43712     7437       518    35757         33
JSON                       460    183345      137         0   183208          0
YAML                       404     21639     1715      1345    18579          0
Assembly                   372    103883    19241         0    84642       1078
SQL                        349     14626     2072       780    11774        489
[...]
───────────────────────────────────────────────────────────────────────────────
Total                    33801   8618121   860192   1138260  6619669     839913
───────────────────────────────────────────────────────────────────────────────
```

3 https://github.com/boyter/scc

Additional project-related documentation was not made available, but instead, AgileBits provided the following set of source code, tools and documents to simplify the scoping of the assessment and to provide further insight into the reasoning behind certain changes:

File Name	SHA256 Checksum
`[RFD 0056] Activedefense for SaaS platforms with IP addresses shared between customers.pdf`	`1f56bb935f9edef72c0f7b22546055a4 015811374b2c694ced890b4017e8cd13`
`[RFD 0059] Scalable architecture for service account.pdf`	`195c02921419193226938cf544c75f1c 22ce6b956f175d26b84192f589b66fb4`
`b5-b5app-release-1507.zip`	`167000deae38f035b42ddba8c71268a4 c0d7d3647b05665ce66a083c3ed09a7f`
`Connect_Security_Documentation.pdf`	`7de36630321b8f518813ee5437e8450b 7583e2c7afe4c8d504a5f3d4879a4f9b`
`op-f04931c110bc6665c9021923f141deea8714daa3 .zip`	`d7cd25a0700db009f9912ef251c769c8 c5a344af1e733ca5f4043fefe6d9df97`
`Q2-23_Secrets_Automation.pdf`	`b74e5b6cfbd3c0e3e90068ae06e53c36 b5e9fb8daecad8b195658de30d1d7ba1`
`Service_Account_Security_Documentation_.pdf`	`608decd03d7607335edc85564996f389 4f5de1c3f8b492de0faa3374fa44582f`

Additionally, a Slack channel was provided by AgileBits ensuring an efficient communication between the consultants and the development team.

2.3 Procedures

The audit was performed in the timeframe of May 15th to 26th in 2023.

The assessment followed a mixed approach, where both a source code review and dynamic testing were performed, with the main objective to uncover weaknesses and vulnerabilities within the integration of the *Security Automation* tooling into the *B5* solution and the *OP* command line application.

Based on the time-boxed, best-effort and risk-based approach, AgileBits requested Recurity Labs to self-determine the focus within this feature set based on the available documentation. Consequently, Recurity Labs decided to focus on the tooling and APIs related to the new *Connect Server* feature and accordingly, the *Service Account* feature was not further assessed.

In context of the *Connect Server*, focus was placed upon, but not limited to, the following vulnerability categories:

- Access control
- Logical flaws
- Session management
- Sensitive data exposure

2.3.1 Areas of Concern

Included within the provided documents, AgileBits defined an overview of the *Areas of Concern* related to the assessed feature set. This subsection provides an overview of the information gathered by Recurity Labs during the present assessments, focusing on the concerns applicable to the *Connect Server* feature.

```
(1) Can users successfully create and use tokens that go beyond their own privileges?
```

Based on the results of the static assessment, it should not be possible to create tokens with escalated privileges. An exhaustive dynamic assessment of this feature was not possible within the given timeframe of the assessment.

```
(2) Can users continue to use revoke tokens?
```

It is only possible to utilize revoked tokens if the underlying synchronization service of the *Connect Server* application is no longer able to communicate with the related *B5* backend. For further details, please refer to section 3.1.

```
(3) Can non admin/owner users gain access to managing service accounts? Only admin and
owner users should be able to generate a service account.
```

During the timeframe of the assessment, Recurity Labs was unable to identify any bypass for the restrictions in-place, neither via static source code analysis nor dynamic testing.

```
(4) Is it possible to break any of the following guarantees?

* A connect server can only read items from vaults it has been given READ access to.
* A connect server can only update, delete and create items for vaults it has been
  given WRITE access to.
* A connect server can only be given access to vaults that the creator of the service
  account has access to.
* A connect server associated with a deleted service account will not be allowed to
  authenticate.
* A connect server can be used to authorize another connect server.
```

Within the course of the assessment, Recurity Labs was unable to break any of the previously stated guarantees. Also, it was not possible to identify any hints on a potential violation of the present guarantees, neither during the dynamic application testing nor the static source code analysis.

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 Missing Synchronization Enforcement

Overview

ID	378.2208.001
Type	Design
CVSS Score	N/A (N/A)
CVSS Metrics	N/A
Location	Protocol Design

Details

The *Connect Server* maintains an encrypted local copy of each of the vaults it is officially allowed to manage. It also stores information about any existing connect token related to the present *Connect Server* instance. This information is automatically kept up-to-date through the `connect-sync` service, which regularly connects with the *B5* Cloud environment to ensure that the local copy is consistent with the actual state of the underlying account.

Such, a local mirror is required, as the *Connect Server* is expected to work even if the *B5* Cloud environment is currently unreachable. This offline-capability automatically opens a time window, which might be leveraged by an attacker to access credentials, even if the token in use has been revoked since the last successful synchronization.

During the course of the present assessment, Recurity Labs intercepted the communication between the `connect-sync` binary and the *B5* test environment. Since such an interception typically utilizes a custom TLS certificate, the application was no longer able to perform the synchronisation of the local database mirror due to the existing certificate validation checks in-place. Accordingly, the server started to log error messages, such as the following:

```
{
  "log_message": "(E) Server: (unable to get credentials and initialize API, retrying in 500ms), Wrapped: (failed to NewAPI), Authentication: (failed to auth.LookupAuth), Network: (failed to request.DoUnencrypted), Get \"https://my.b5test.com/api/v2/auth/pvje343cklv4c@1passwordserviceaccounts.com/A3/MWS98G/w6vc4fyb2t2ia7ssmsjcnvkhki\": tls: failed to verify certificate: x509: certificate signed by unknown authority",
  "timestamp": "2023-05-23T14:58:24.032653459+02:00",
  "level": 1
}
```

As the `connect-sync` client was no longer able to keep the local mirror consistent with the upstream copy of the account, the revocation of existing tokens could not be propagated to the *Connect Server*. Due to this circumstance, any token the server expected to be valid could be utilized to access the local copy of the vault, even if the token was revoked through the *B5* Web interface. This was possible even if the `connect-sync` was unable to perform a successful synchronization for more than 24 hours, which might be an unexpectedly long timeframe.

Nevertheless, it should be emphasized that this is a design issue, which inherently comes with any implementation of an offline-capability. As the service has to rely on external information to update the local mirror, the only way to circumvent this is to limit the timeframe the *Connect Server* is allowed to function without a successful synchronization. Furthermore, for these reasons, Recurity Labs intentionally refrained from assigning a CVSS vector/score.

Reproduction Steps

To verify if this behaviour still exists, it is necessary to set-up a local *Connect Server* test environment and to forward any outgoing communication of the `connect-sync` service through an HTTPS interception proxy, such as ZAP[4] or Burp[5].

This kind of interception requires an adjustment of the provided docker and docker-compose setup, or through the `HTTPS_PROXY` environment variable supported by most Unix-based Operating Systems, to ensure that the interception proxy is able to replace the HTTPS certificate, e.g. as shown within the following code snippet:

```
env \
  HTTP_PROXY=8078 \
  HTTPS_PROXY=8078 \
  OP_HTTP_PORT=8081 \
  OP_LOG_LEVEL=info \
  OP_SESSION=${SESSION} \
  XDG_DATA_HOME=${CUSTOM_HOME} \
  connect-sync &
```

Recommendation

Even if the behaviour described within the present finding is actually intended, Recurity Labs still recommends to analyse if the unexpectedly long timeframe the *Connect Server* keeps working without successful synchronization is actually required or if it can be (significantly) shortened, as it allows to circumvent the revocation mechanism as described above.

Furthermore, it should be ensured that public documentation emphasizes the potential risk inherently included within the present offline-capability of the feature, as well as the timeframe the server is expected to function without a successful remote update.

As an additional defense-in-depth mechanism, it might be helpful to produce further warnings if an existing *Connect Server* has not been synchronized within a *configurable* timeframe.

Feedback provided by AgileBits (2023-07-05)

```
1Password accepts the observation made by Recurity Labs and intends to fix the behavior
in the future.
```

4 https://www.zaproxy.org/
5 https://portswigger.net/