We Won’t Borg You With The Details

Make SA Great Again
Dalton Ellis, Gib Filter, Neil Hartje

Introduction

The team, "Make Software Assurance Great Again" has chosen an open source project to contribute to for the semester project. The project is called Borg Backup and it has many qualities that will help make the team successful when it comes time to make recommendations, fix issues, and make pull requests.

Project Description:

“Borg Backup (short: Borg) is a deduplicating backup program. Optionally, it supports compression and authenticated encryption. The main goal of Borg is to provide an efficient and secure way to backup data. The data deduplication technique used makes Borg suitable for daily backups since only changes are stored. The authenticated encryption technique makes it suitable for backups to not fully trusted targets.” [1] The project is led by 4 main contributors, but has been worked on by 80 different people. [2] In the past week there were 5 merged pulls and 8 total pull requests. On top of that, there are currently 193 open issues for the project. [3] The majority of the project is written in python, but there is some underlying C. Parts of the project use the “Cython” c extensions for python. The project is multi-platform, currently there is support for Linux, Mac OS X, FreeBSD, Open BSD and NetBSD. There is a portion of contributors that are attempting to make the project viable on Windows. [4]

Functional security requirements

Borg Backup is inherently a computer security project. The project requires secure functionality to safely create and store system backups. We have compiled a list of functional security requirements for this project, they are as follows:

- Ensure the confidentiality, integrity, and availability of data during and after backup process has begun.
- Ensure encryption algorithms are implemented correctly
- Ensure encryption keys are properly generated/stored
- Ensure off-site backups communicate the data securely
- Ensure sufficiently secure functions are used in the encryption/decryption/and storage process

Licenses and procedures

Borg is distributed under a 3-clause BSD License. [5] The source code and binary format redistributions must contain the copyright notice. There are several guidelines that contributors should follow [6]:

- Discuss about changes on GitHub issue tracker, IRC, or mailing list.
- Choose the branch you base your changesets on wisely.
- Do clean changesets.
- If you write new code, please include tests and documentation for it.
- Run the tests, fix anything that comes up.
- Make a pull request and wait for review.

Security related history

The project deals heavily with encrypting backups and authenticating those who access them. Therefore, many of the closed and open security issues are related to these processes. These topics include secure random key generation, proper IV generation, aes and hmac implementation. For each of these topics there is insightful discussion on potential methods to proceed with certain issues. A few of these were closed very recently and many are currently open to be solved. [7]

Motivation

There are several reasons for selecting Borg Backup as the software assurance project. The most important of these reasons is the fact that the project is developed in Python. Python is probably the easiest language to understand for new programmers. It also happens to be the one that the group is collectively most experienced in.

The project has become very active in the past year, compared to other years. This demonstrates that the project is alive and well and that most pull requests will at least be acknowledged. Additionally, Borg Backup is not overwhelmingly large that the number of potential contributors will be challenging to compete with for updates. Borg backup has 80 total contributors whereas Django, another GitHub project, has had 1,256. Similarly, the community is fairly accepting and in the past week, there have been 5 pull requests that were accepted. Given this information, it is safe to say that the friendly GitHub requirement has been met.

Another box that can be checked off with Borg Backup is the security related checkbox. Borg Backup can be considered a security tool of sorts because it deals with backups which satisfies an availability component of the CIA acronym. Additionally, it handles authentication and encryption for the tool and this could arguably satisfy both confidentiality and integrity. Therefore, the tool is somewhat developed with security in mind and for secure purposes. The flip-side to this is that it will also be able to be analysed for security bugs. With the tool being related to security and also requiring checks for safe development, the project will open up more opportunities to contribute. Currently, there sits 194 open issues that could be considered in addition to what our group might find through our own investigation.

Conclusion

Borg Backup fits the skillset and experience of the team quite well. The language used is appropriate for what the team is comfortable with and the community surrounding the project seems to be accepting of new developers and engages in discussion with different contributors. Because the community is not too large, the team will be able to make a meaningful contribution by fixing security related issues and by adding security related features in the form of enhancements for the project. There are a number of good opportunities to interact with this project and therefore it is a good candidate for a semester project for Software Assurance. [8]

References

[1] https://github.com/borgbackup/borg/blob/master/README.rst
[2] https://github.com/borgbackup/borg/graphs/contributors
[3] https://github.com/borgbackup/borg/pulse
[4] https://borgbackup.readthedocs.io/en/stable/
[5] https://borgbackup.readthedocs.io/en/stable/authors.html#license
[6] https://github.com/borgbackup/borg/blob/master/docs/development.rst
[7] https://github.com/borgbackup/borg/issues?q=is%3Aissue+is%3Aclosed+label%3Asecurity
[8] https://github.com/borgbackup/borg