Individual Plan

Johan Moritz

February 2022

PROJECT INFORMATION

Preliminary title: Decentralized Distribution of .buildinfo Files

Student: Johan Moritz, jmoritz@kth.se

Examiner at KTH: Mads Dam

Supervisor at KTH: Giuseppe Nebbione

Supervisor at Subset: Peter Jonsson, peter.jonsson@subset.se

Current date: February 4, 2022

Keywords: Reproducible builds, Security, Decentralized trust, buildinfo, P2P

BACKGROUND & OBJECTIVE

Discussions on how to verify the lack of malicious code in binaries goes at least as far back as to Ken Thompson's Turing award lecture [1]. In recent years, several attacks on popular packages within the Free Open Source Software ecosystem (FOSS) have been executed **citations needed** where trusted repositories have injected malicious code in their released binaries. This raises the question of how much trust in such dependencies is appropriate. In an attempt to raise the level of trust and security in FOSS, the reproducible builds [2] projects was started within the Debian community. Its goal was to mitigate the risk that a certain package is tampered with by making sure that its builds are deterministic and therefore should be bit-by-bit identical over multiple rebuilds. Any user of a reproducible package can verify that it has indeed been built from its source code and have not been manipulated after the fact simply by rebuilding it from the package's .buildinfo file. These meta data files for reproducible builds include hashes of the produced build artifacts as well as a description of the build environment to allow for verification on the users side. .buildinfo files are by this notion the crucial link to ensure reproducibility, which also means that a great deal of trust is assumed when using them. Current measures for validating .buildinfo files and their corresponding packages involve package repository managers as well as volunteers running rebuilderd **citation** instances that test the reproducibility of every .buildinfo file added to the corresponding package archive. This allows users to audit the separate instances, thus confirming the validity of a particular package. However, because this would be a manual process and the separate instances do not coordinate their work it relies on the user making their own judgement on a case by case basis whether to trust a package or not. This project seek to reduce some of that burden from the user while increasing their trust in the packages they use by investigating possible decentralized solutions for distributing and proving the correctness of .buildinfo files.

The project is carried out at Subset, an IT-security consulting company focusing on critical systems. They see reproducible builds as an important step for raising their trust in FOSS, which is critical for them to be able to use such software and technologies themselves in their work for their customers.

RESEARCH QUESTION & METHOD

Question To what extent can .buildinfo files be provably verified in regards to upholding user trust?

Objectives Explore the research question through a system with the following requirements:

- Multi-parti verification of .buildinfo files
- Public access of provably verified .buildinfo files
- Minimized risk of single-point-of-failure

Such a system would by design increase user trust in .buildinfo files, and so the projects objectives are to research, construct and test such a system through current technologies in order to verify or refute the listed requirements in said system.

Tasks

- Research current state of the art in peer-to-peer and consensus algorithms in order to make a qualified technology choice as the base for the system. This involves understanding not only the functional requirements of the system, but also the technical or practical ones as the system would otherwise be unsuitable as a solution. The main challenge is therefore to understand the consequences of such a technology choice in a production system.
- Construct a decentralized system for distributing .buildinfo files based on the technology choice from the previous task. This is the main production task of the project and while probably quite straightforward could be challenging in terms of time management.

- Setup a simulated production environment for which to test the systems properties. Test scenarios should be easy to describe and setup to allow for easily rerunning tests, which will add to the complexity of the project.

- Run scenarios on the production environment to test whether the system follows the requirements or not. Specifically, scenarios where some nodes in the system are acting maliciously as well as scenarios with reduced availability should be included.

- Stress test the system under its assumed workload were it to be published and used as the primary archive for .buildinfo files. It is probably hard to estimate the actual workload of such a system, so perhaps a general notion of performance will have to do; for example measuring retrieval latency for increasing number of requests.

Method

- To realize the first task of finding a suitable base technology for the system, a literature study as well as a comparison of possible algorithms and solutions is to take place. The comparison should especially consider the requirements of the system to ensure that the technology choice is appropriate.

- The testing of how well the system conforms to the requirements will be done by judging the threat of availability and integrity compromise in the system. As described in the tasks above, this is done by synthetically introducing faults and malicious actors in the testing scenarios. This way, the level of conformity to the system requirements are directly correlated to the results of the tests.

Ethics and Sustainability From an ethics point of view, there is the question of who has the right to validate a .buildinfo file. In other words, who should users trust. The project does not serve to enact any such policies directly but might make the assumption that the parties involved in running the nodes on the system are globally known and trusted by the users, for example institutions or other official organizations.

From a sustainability perspective, there is definitely a consideration to made in that a large distributed system could possibly have a large energy consumption. Care should be taken throughout the project to try and reduce such issues, and especially choose technologies with this point in mind.

Limitations

Risks

EVALUATION & NEWS VALUE

PRE-STUDY

CONDITIONS & SCHEDULE

REFERENCES

References