ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 8, Issue 4, pp: (18-20), Month: October - December 2020, Available at: www.researchpublish.com

Why Application Modifications Detection is Important

[1]Nada Al-Noaimi, [2]Sultan Al-Sharif, [3]Bandar Al-Mashari

[1,2,3]Saudi Aramco, Dhahran, Saudi Arabia

Abstract: **This article makes a case for the importance of applying a detection method of source code changes at every phase of the application lifecycle. In which, cybersecurity analysts will have a complete view of what application was modified since their last security assessment. This method will also improve the application security posture and eliminate the duplication of efforts.**

Keyword: **source code, hash, application security, application modification detection.**

I. INTRODUCTION

Cybersecurity is a core value of developed applications in any organization. It is the responsibility of the cybersecurity analyst to check the cyber state of applications and identify all potential risks. Large organizations will have a lot of applications to be analysed; and in many cases, the cybersecurity analyst will analyse applications that haven't been changed by the developer since the previous analysis.

Application developers store their source code in a hosting environment, the cybersecurity analysts access the files and analyse the application every cycle, reporting on relevant security issues. Cybersecurity analysts usually analyse large number of applications and the process can be time consuming.

II. BACKGROUND

A. Application Lifecycle

Most applications go through two phases of testing before the final deployment, and every phase has its hosting environment: development; quality assurance and production. First, the application developers will utilize the development environment, where they can build the application and test its functionality. Second, they will deploy the code to the quality assurance environment, where developers test the final application before moving it to production. Third, once the verification is complete, the new application or changes will be deployed in the production environment.

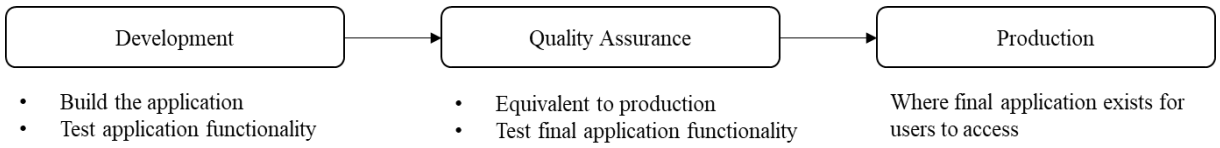

Fig 1: Application Lifecycle

B. Cybersecurity Embedded Throughout Application Lifecycle

Cybersecurity plays a major role throughout the application lifecycle. To check the application security, the cybersecurity analyst must run a source code assessment against the application at every level of the application lifecycle, from development to production environment. Verifying the source code security at early stages of development is vital, as late assessments may report too many vulnerabilities that would be difficult to fix and may delay the project deployment.

III. METHODOLOGY

A. Cybersecurity Specialist Roles

Large organizations host a huge number of applications, and it is difficult for the cybersecurity teams to verify the security of the applications in a timely manner. In which, application developers store their source code in a hosting

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 8, Issue 4, pp: (18-20), Month: October - December 2020, Available at: www.researchpublish.com

environment, the cybersecurity analysts access the files and analyze the application every cycle, reporting on relevant security issues. A method was developed to utilize hash algorithm and direct cybersecurity analysts to any source code changes in the application at any application lifecycle phase. "A hashing algorithm is a cryptographic hash function. It is a mathematical algorithm that maps data of arbitrary size to a hash of a fixed size. It's designed to be a one-way function, infeasible to invert" [1]. There are different hashing algorithms and the most common ones are MD5 and SHA-family.

B. Method Implementation

The method would automatically analyze application repositories in each hosting environment, generating real-time alerts when changes are noted. These alerts would be fed into the cybersecurity analysts work queue to ensure that appropriate control checks are performed. These alerts can be relayed into APIs for existing scanning tools to generate real-time scan of the changes and report on applicable findings of interest. First the solution will generate a hash for every application, as demonstrated in Fig.2.

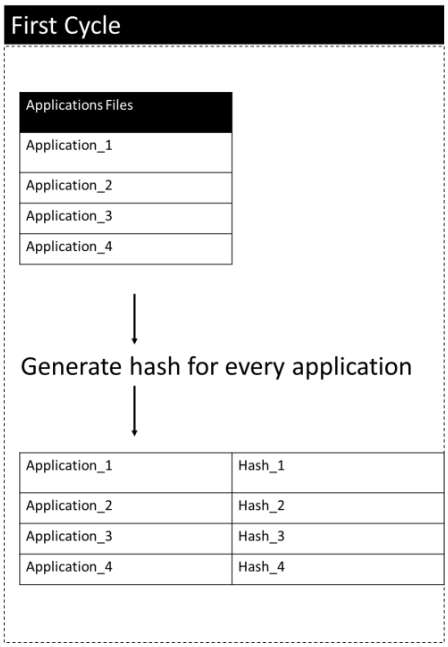

Fig 2:Generate Application Hash Example – 1st assessment

Then, in the second assessment cycle, a new hash will be generated as illustrated in Fig.3.

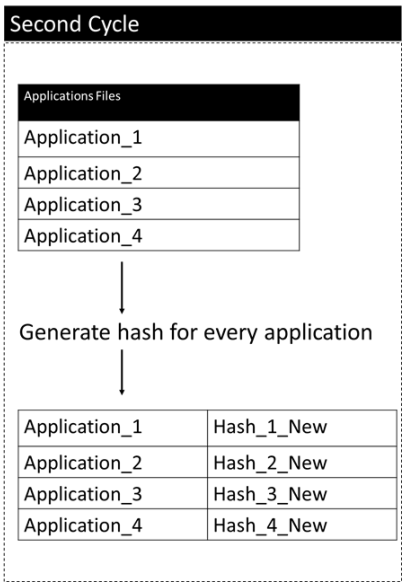

Fig 3: Generate Application Hash Example – 2nd assessment

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 8, Issue 4, pp: (18-20), Month: October - December 2020, Available at: www.researchpublish.com

Finally, the method will run the analysis and inform cybersecurity teams of any changes in the application. For example, the method will verify the changes as follows:

- If Hash_1 = Hash_1_New then Application was not modified
- If Hash_1 ≠ Hash_1_New then Application was modified

Also, this method allows us to discover new applications if new hashes appears, and discover if an application was deleted as illustrated in Fig.4. Moreover, when new or modified applications are identified, the cybersecurity analysts will be informed with the changes to run the security assessment.

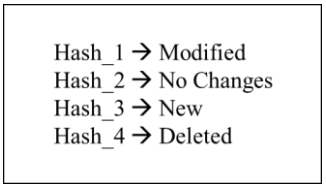

Fig 4: Method Results

IV. CONCLUSION

Finally, using hashing algorithms as one of the means to identify changes in the application source code done by hundreds of application developers to hundreds of applications. The changes occurring on the hosting environments will be monitored using a hashing algorithm as a technique, plus other techniques explained below. The method is about the process of detecting unauthorized/undocumented application modifications. This method will allow cybersecurity analysts to detect and respond to changes, and accordingly perform security assessments only on those modified applications. This method is not about determining the authenticity of the application (origin), but it will trigger an alert to cybersecurity analysts to assess the unauthorized changes.

There are multiple methods to identify changes in production environments, and it is not exclusively reliant on hashing algorithms. These additional methods are:

- Date and time stamp of files: Each operating system stamp date and time for every modified file. This can be used to detect latest changes, but may not be accurate for other operating systems, and therefore a different approach is required to combine with this one.

- Different users modifying the same file: Each file written on the desk will be attributed to a specific user, changes to the same file by different user will result in changing the attributes. If service accounts were used to modify the same file by different users, then the attributes will remain unchanged, therefore a different approach is required to combine with this one.

- Use of updated third-party components and applications configurations: Almost all modern applications consist of source code and third-party packages that will enable reusable functionalities (such as database interfaces, graphs, security, machine learning, etc). Inspecting version number of these packages will indicate changes of previously deployed applications' components.

REFERENCES

[1] Jscrambler, "Hashing Algorithms," Jscrambler. Aug. 2020.