SCIENCE CHINA
Information Sciences

• LETTER •

March 2023, Vol. 66 139105:1–139105:2
https://doi.org/10.1007/s11432-021-3408-y

SMAF: a secure and makespan-aware framework for executing serverless workflows

Shuai ZHANG[1], Yunfei GUO[1], Zehua GUO[2*], Hongchao HU[1,3] & Guozhen CHEN[1,3]

[1]*Institute of Information Technology, Information Engineering University, Zhengzhou* 450002*, China;*
[2]*School of Automation, Beijing Institute of Technology, Beijing* 100081*, China;*
[3]*Purple Mountain Laboratories, Nanjing* 211111*, China*

Received 15 July 2021/Revised 4 January 2022/Accepted 8 January 2022/Published online 18 January 2023

Citation Zhang S, Guo Y F, Guo Z H, et al. SMAF: a secure and makespan-aware framework for executing serverless workflows. Sci China Inf Sci, 2023, 66(3): 139105, https://doi.org/10.1007/s11432-021-3408-y

Serverless computing, or function-as-a-service (FaaS), is an emerging service provision paradigm in cloud computing. Workflow, which comprises a set of tasks with data-dependent on each other, is a typical application abstract model in clouds. Generally, applications in serverless computing can be modeled by serverless workflow, in which multiple functions are orchestrated together as a workflow to provide an integrated service [1]. However, serverless workflows face severe security threats. Specifically, attackers can exploit the vulnerabilities of the function to achieve its goals, such as stealing sensitive data, damaging the execution of the functions, and tampering with the intermediate results. Furthermore, the lightweight nature of serverless functions makes it difficult to deploy the traditional "reinforced" and "stacked" defense mechanisms, which makes its security problem more challenging [2]. Inspired by a proactive defense mechanism, such as moving target defense (MTD), enabling dynamic at the serverless functions can increase the security by weakening the cyber kill chain (CKC) for attackers. However, the dynamic-based solutions inevitably influence the execution of the serverless workflow and lead to additional makespan costs. Therefore, it is challenging to maintain high security and low makespan for executing serverless workflows.

We propose a secure and makespan-aware framework for executing serverless workflows named SMAF. We analyze the attackers' behavior pattern in the execution of the serverless workflow and propose the running environment refresh mechanism to protect against attackers' lateral movement. In order to balance the tradeoff between security and makespan, we formulate the HAG (holistic attack graph) model to characterize the security performance of serverless workflow and selectively apply our defense mechanism to serverless workflow, taking both security and makespan into account. Experimental results show that SMAF can significantly increase the security of serverless workflow with small makespan cost.

Threat model and defense mechanism. Attackers can launch cyberattacks according to the cyber kill chain. Specifically, attackers first perform various reconnaissance actions to find vulnerabilities of the functions, and then launch their cyberattacks according to the vulnerabilities of the targets. We assume that the attackers are outside the cloud, and all existing functions may become the targets of attackers. The entry functions of the workflow will first be attacked. Then, there are two available ways for attackers to move laterally in the serverless workflow to achieve the attack goal: (1) data transmission based attack path; (2) running environment based attack path. Data transmission based attack path is determined by the dependency among serverless functions, as attackers can exploit the legitimate interface among functions to launch cyberattacks. Running environment based attack path is introduced by running environment reuse. For example, the same container is used for multiple functions, attackers can hide in the container and move from one function to another.

Based on the threat model, we find that the reuse of the running environment provides an easy way for attackers to move laterally to the targets. In order to interrupt the attack phase of attackers, we consider using a running environment refresh mechanism. With this mechanism, we will create a brand-new running environment and delete the old running environment periodically. In this way, the attackers' existence in the old running environment can be cleaned, such as the invoked virus, and the running environment based attack path can be blocked.

HAG model. Given a serverless workflow with K functions $\{f_1, f_2, \ldots, f_K\}$, HAG model is applied to represent the overall attack events for the serverless workflow. HAG can be seen as a directed graph $G = (N, C)$, where $N = \{n_1, n_2, \ldots, n_K\}$ denotes the set of function instances while C is a set of attack paths between the function instances. A function instance n_p can be seen as the combination of function f_p and its environment $r_{H(p)}$, where $H(p)$ indicates the index of function environment allocated to f_p. We use $c_{p,q} \in C$ to represent the attack path from function instance n_p to function n_q. Given an edge $c_{p,q} = (n_p \to n_q) \in C$, we define the DF (c) as the weight of edge c, which represents

* Corresponding author (email: guolizihao@hotmail.com)

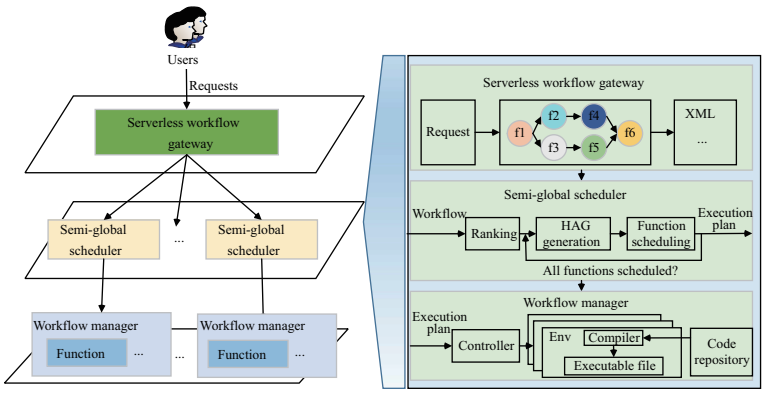

Figure 1 (Color online) SMAF framework overview.

the attack difficulty from the compromised function instance n_p to the target function instance n_q.

In order to characterize the attack difficulty quantitatively, exploitability metric AD defined in Common Vulnerability Scoring System (CVSS) is used to quantify the exploiting difficulty for a specific vulnerability. Considering the target with multiple vulnerabilities, the attackers can compromise the target if any of these vulnerabilities are exploited. However, we have no idea about the ability and behavior preferences of attackers. Therefore, all vulnerabilities in the target are likely to be exploited. In this case, we use Exploit Code Maturity AP in CVSS to evaluate the probability of a vulnerability being exploited. Given a compromised function instance n_p and the target function instance n_q, the attackers can only compromise n_q based on vulnerability exploitation or running environment reuse. Thus the weight of edge $c_{p,q}$ can be expressed as

$$\mathrm{DF}\left(c_{p,q}\right)=\begin{cases}0, & H_p=H_q,\\ \frac{\sum\left(\mathrm{AP}\times\mathrm{AD}\right)}{\sum\mathrm{AP}}, & \left(n_p\rightarrow n_q\right)\in C \text{ and } H_p\neq H_q,\\ +\infty, & \left(n_p\rightarrow n_q\right)\notin C \text{ and } H_p\neq H_q.\end{cases} \tag{1}$$

We assume that the attackers are rational and will launch cyberattack along the shortest path to the target function instances. Therefore, the attack difficulty for target function instances can be represented by the length of the shortest path from attackers to the target in G. As all the function instances may be the target of attackers and we cannot obtain any information about the attack target, we use the average attack difficulty among all function instances to evaluate the security performance of the workflow.

SMAF overview. In this study, we provide the design overview of the SMAF framework, which is responsible for the execution of the serverless workflow. Figure 1 shows the overall framework architecture of SMAF, which consists of three modules, serverless workflow gateway, semi-global scheduler, and workflow manager.

The serverless workflow gateway is the point of access to our framework, which acts as a reverse proxy and distributes the workflow execution requests across a pool of semi-global scheduler. Besides, the information about the serverless workflow will be parsed into XML format and sent to semi-global scheduler.

The semi-global scheduler is the core module of our framework, which will decide how to execute the workflow based on the HAG module. There are three process steps in the semi-global scheduler. First, the scheduling priorities of functions will be decided in the ranking phase. The upward ranking method is adopted for the ordering function in the serverless workflow [2]. Then, the HAG model can be built based on the information of workflow and CVSS metrics. Finally, we can iterate through all the functions based on the priorities to build our workflow defense strategy and function scheduling strategy. In every iteration, the scheduler will decide which running environment to be allocated and whether to refresh the running environment, taking both makespan and security into account. Specifically, the security performance can be calculated based on HAG model, and the makespan can be obtained based on the earliest finish time.

Workflow manager is the daemon for workflow execution. According to the scheduling plan, Workflow execution engine can schedule functions to the selected running environment, trigger the execution of functions, collect the functions response and decide whether to refresh the running environments after the execution of functions.

Experimental results. We set up a serverless cloud environment and evaluate our SMAF framework. Results show that the security performance has around 15 folds increase, with around 0.44% makespan cost compared with the traditional HEFT algorithm. For detailed results of the experiment, please refer to Appendix A

Conclusion. We propose a secure and makespan-aware framework for executing serverless workflows named SMAF. SMAF can significantly help in improving the security performance in the execution of serverless workflow with little makespan cost.

Acknowledgements This work was partly supported by National Natural Science Foundation of China (Grant Nos. 62072467, 62002019, 62002383), National Key Research and Development Plan of China (Grant Nos. 2021YFB1006200, 2021YFB1006201), and Beijing Institute of Technology Research Fund Program for Young Scholars.

Supporting information Appendix A. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Arnav S, Pubali D, Adam B. Workflow integration alleviates identity and access management in serverless computing. In: Proceedings of Annual Computer Security Applications Conference, New York, 2020. 496–509

2 Wang Y W, Guo Y F, Guo Z H, et al. CLOSURE: a cloud scientific workflow scheduling algorithm based on attack-defense game model. Future Gener Comput Syst, 2020, 111: 460–474