CVE-2021-22557

SLO-Generator 2.0.0 - Code Execution

Introduction

To manage services efficiently, 3 indicators are used to measure relevant properties and metrics and to define actions to take when services no longer function as expected. These indicators are called Service-Level Indicators (SLIs), Service-Level Objectives (SLOs) and Service-Level Agreements (SLAs) [1], [2]. SLIs are quantitative measures of different relevant aspects of the provided service. Examples of such indicators are request latency, throughput and error rate. SLOs are target values for SLIs. Taking throughput as an example, an SLO would be a minimum desired throughput, `target_throughput`. Thus, the service must achieve `SLI ≥ target_throughput`. SLAs are explicit or implicit contracts that specify which objectives are to be met together with consequences and actions to take in case said SLOs are not achieved.

SLO Generator

Google developed SLO Generator [3] as a command-line tool for creating, manipulating and converting SLOs (hence the name), Burn Rates and Error Budgets. It is written in Python3 and installed with Python's package manager, `pip`. It uses YAML or JSON configuration files provided by the user.

Once SLO Generator v2 was implemented, one of its new features was to migrate YAML configuration files from the format used by v1 to that used by v2. For this, the `migrations/migrator.py` module [4] is employed.

Vulnerability

SLO Generator parses and loads its configuration file using a dedicated Python library for handling YAML files, called `ruamel.yaml` [5]. The parser must first be constructed using one of the available constructors, defined in `constructor.py` [6]: `BaseConstructor`, `SafeConstructor`, and `Constructor`. One of the key differences between `Constructor` and `SafeConstructor` is that `Constructor` allows executing commands from the parsed YAML files, whereas `SafeConstructor` does not. This difference is also highlighted in `ruamel.yaml`'s official documentation [7].

On 4 October 2021, CVE-2021-22557 [8], [9] was published. It showed that the migrator module was vulnerable to a command execution attack via the YAML file it uses. A very simple way to achieve command execution is to create a YAML file that contains the following line:

```
!!python/object/apply:os.system ["id;whoami"]
```

as described in [10]. The line above invokes the python interpreter and calls `os.system("id;whoami")`. By using the `os.system` function, an attacker can execute arbitrary commands with the permissions of the user that invokes `slo-generator`.

Therefore, SLO Generator was found to be vulnerable to a code injection and execution attack. If deployed as a remote service that users could query by providing a YAML file, the vulnerability could result in a remote code execution attack. Such an attack can compromise the service provider's server by leaking private information, such as certificates, passwords and databases or by stopping it entirely.

A docker container with a proof-of-concept of this vulnerability is available here [11]. The YAML file containing the exploit is found in `/root`. Alternatively, you can build the container yourself by cloning this repository [12] and running `make run`.

Patching the Vulnerability

This vulnerability was patched by [13]. As this Pull Request shows, the vulnerability was caused by using `Constructor` instead of `SafeConstructor` [14], i.e. creating the YAML loader with `Loader=Loader` instead of `Loader=SafeLoader`. The aforementioned Pull Request [13] fixes this issue and removes the vulnerability.

References

[1] https://sre.google/sre-book/service-level-objectives/
[2] https://www.atlassian.com/incident-management/kpis/sla-vs-slo-vs-sli
[3] https://github.com/google/slo-generator
[4] https://github.com/google/slo-generator/blob/master/slo_generator/migrations/migrator.py
[5] https://sourceforge.net/projects/ruamel-yaml
[6] https://sourceforge.net/p/ruamel-yaml/code/ci/default/tree/constructor.py
[7] https://yaml.readthedocs.io/en/latest/api.html#departure-from-previous-api
[8] https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-22557
[9] https://nvd.nist.gov/vuln/detail/CVE-2021-22557
[10] https://www.exploit-db.com/exploits/50385
[11] https://hub.docker.com/repository/docker/teodordutu/slo-generator-2.0.0-exploit
[12] https://github.com/teodutu/CDCI/tree/master/Assignments/Assignment-1
[13] https://github.com/google/slo-generator/pull/173
[14] https://sourceforge.net/p/ruamel-yaml/code/ci/default/tree/constructor.py#l352