Authentication Bypass in Izanami Docker image 1.10.22 CVE-2023-22495

Security advisory

2023-02-03

Raphaël LOB

Vulnerabilities description

Izanami

Izanami is a shared configuration, feature flipping and A/B testing service perfectly well suited for micro services environments written in scala and developed by the MAIF OSS *team.*

The issues

Synacktiv discovered a way to bypass the authentication in this application when deployed using the official *Docker* image. Because a hard coded secret is used to sign the authentication token (JWT), an attacker could compromise another instance of *Izanami* by doing the following steps :

1. The attacker installs his own *Izanami* application.
2. The attacker logs in and copies the content of the cookie named *Izanami*.
3. The attacker connects to the victim's website and creates a cookie named *Izanami* with the previous value.
4. **The attacker is successfully log-in, even if his user does not exist.**

Affected versions

At the time this report is written, the version 1.10.22 was proven to be affected. Previous versions are likely to be vulnerable too.

Timeline

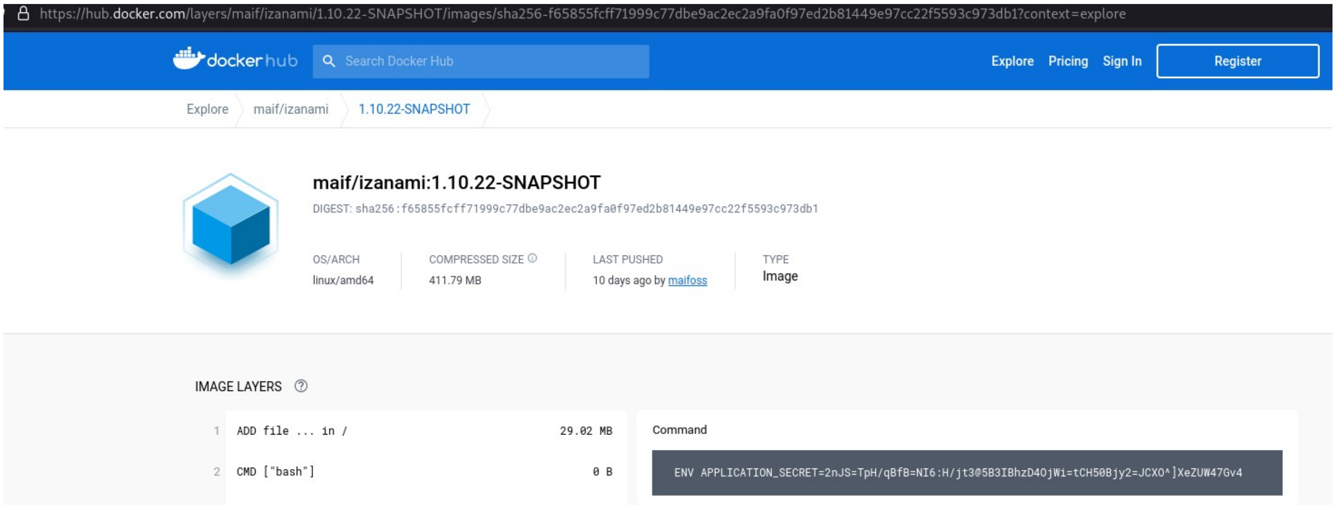
Date	Action
2022-12-21	Advisory sent to oss@maif.fr
2023-01-13	CVE-2023-22495 assigned
2023-02-03	Public release

Technical description and proof-of-concept

Authentication bypass

The following hard coded secret is set in the official *Izanami Docker* image. The latest release can be downloaded with the following link:

- https://hub.docker.com/layers/maif/izanami/1.10.22-SNAPSHOT/images/sha256-f65855fcff71999c77dbe9ac2ec2a9fa0f97ed2b81449e97cc22f5593c973db1?context=explore



https://hub.docker.com/layers/maif/izanami/1.10.22-SNAPSHOT/images/sha256-f65855fcff71999c77dbe9ac2ec2a9fa0f97ed2b81449e97cc22f5593c973db1?context=explore

maif/izanami:1.10.22-SNAPSHOT

DIGEST: sha256:f65855fcff71999c77dbe9ac2ec2a9fa0f97ed2b81449e97cc22f5593c973db1

OS/ARCH	COMPRESSED SIZE	LAST PUSHED	TYPE
linux/amd64	411.79 MB	10 days ago by maifoss	Image

IMAGE LAYERS

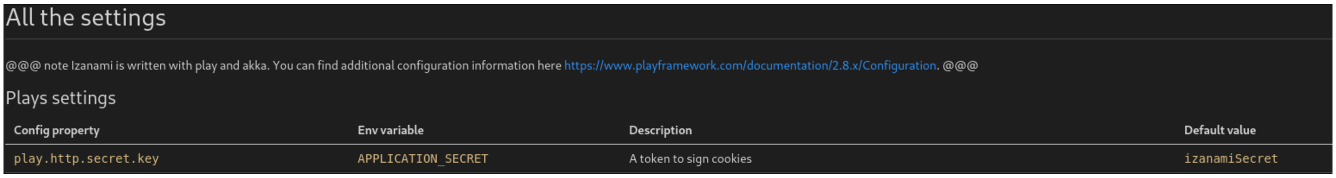
```
1  ADD file ... in /        29.02 MB
2  CMD ["bash"]             0 B
```

Command

```
ENV APPLICATION_SECRET=2nJS=TpH/qBfB=NI6:H/jt3@5B3IBhzD4OjWi=tCH50Bjy2=JCXO^]XeZUW47Gv4
```

Illustration 1: *APPLICATION_SECRET* is set to a fixed value.

The *APPLICATION_SECRET* variable allows the application to sign cookies:



All the settings

@@@ note Izanami is written with play and akka. You can find additional configuration information here https://www.playframework.com/documentation/2.8.x/Configuration. @@@

Plays settings

Config property	Env variable	Description	Default value
play.http.secret.key	APPLICATION_SECRET	A token to sign cookies	izanamiSecret

Illustration 2: Extract of the documentation (*izanami-documentation/src/main/paradox/settings/settings.md*).

Extract of *izanami-server/build.sbt*:

```
dockerCommands ++= Seq(
  Cmd("ENV", "APP_NAME izanami"),
  Cmd("ENV", "APP_VERSION 1.0.6-SNAPSHOT"),
  Cmd("ENV", "LEVEL_DB_PARENT_PATH /leveldb"),
  Cmd("ENV", "REDIS_PORT 6379"),
  Cmd("ENV", "REDIS_HOST redis"),
  Cmd("ENV", "KAFKA_HOST kafka"),
  Cmd("ENV", "KAFKA_PORT 9092"),
  Cmd("ENV", "HTTP_PORT 8080"),
  Cmd("ENV", "APPLICATION_SECRET
2nJS=TpH/qBfB=NI6:H/jt3@5B3IBhzD4OjWi=tCH5OBjy2=JCXO^]XeZUW47Gv4")
)
dockerExposedVolumes ++= Seq(
  "/leveldb",
  "/data"
)

dockerUsername := Some("maif")

dockerEntrypoint := Seq("/opt/docker/bin/start.sh")
```

The security experts identified a script that could fix this issue. Extract of *izanami-server/docker/start.sh*:

```
#!/bin/bash -e

APPLICATION_SECRET=$(head -c 64 /dev/urandom | base64)

HOST=$(awk 'END{print $1}' /etc/hosts)

exec /opt/docker/bin/izanami -Dlogger.file=./conf/docker-logger.xml -
Dcluster.akka.remote.netty.tcp.hostname="${HOST}" -Dcluster.akka.remote.netty.tcp.bind-
hostname="${HOST}" -Dplay.server.pidfile.path=/dev/null $@
```

Recommendation

Randomize the value of the *APPLICATION_SECRET* variable.