DepSecure STYX 管理平台安装手册

2022 年 3 月

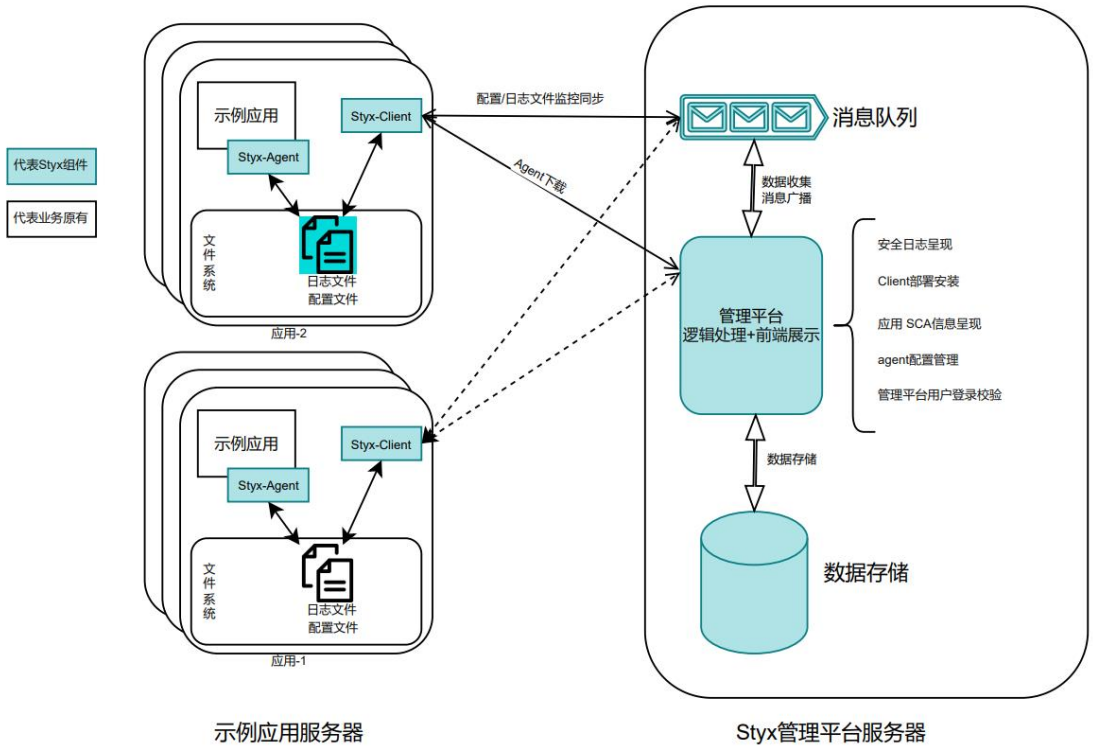
文档修订记录

修订日期	文档版本	agent 版本	更新摘要
2022.04.8	v0.5		

目录

1. 安装规划

1.1. 部署方案规划



1.2. 软硬件配置规划

	架构	CPU	内存	磁盘
管理平台	x86	8C	16G	200G
agent	随业务服务器	随业务服务器	随业务服务器	随业务服务器

环境要求：
管理平台服务器：X86 架构、8CPU、16G 内存、200G 磁盘

1.3. 网络规划

网络规划要求
用于安装管理平台的服务器，可以和承载业务的服务器进行通信即可。

端口规划：

需放行端口			
组件	端口	协议	作用
管理界面	443、80	https/http	访问管理平台 Web 界面
管理平台 MQ	5672	tcp	agent 发送 log 到管理平台

2. 安装流程

2.1. 安装前准备

- 安装管理平台的服务器：Centos、Ubuntu 均可
- 要求：用于安装的管理平台的 Linux 需放开以上要求端口
- STYX 安装镜像文件

2.2. 安装 STYX 安全疫苗管理平台

安装

1、 下载到服务器之后，解压安装包

```
tar -vxf styx-env-<sersion>.tar.gz
```

```
[root@localhost ~]# ls
anaconda-ks.cfg  job  styx-env-2022-03-18_04_03.tar.gz
[root@localhost ~]# ls job/
[root@localhost ~]# tar -vxf styx-env-2022-03-18_04_03.tar.gz -C job/
docker-compose
docker-compose.yml
docker-images.tar
docker.service
docker.socket
docker.tgz
makefile
[root@localhost ~]# ls job/
docker-compose      docker-images.tar  docker.socket  makefile
docker-compose.yml  docker.service     docker.tgz
[root@localhost ~]#
```

2、 安装容器环境（可选）

```
[root@localhost ~]# docker version
-bash: docker: 未找到命令
[root@localhost ~]# docker-compose version
-bash: docker-compose: 未找到命令
[root@localhost ~]#
```

a) 如果服务器没有 docker 和 docker-compose，使用以下命令安装 docker 和 docker-

compose：

```
sudo make install-basic
```

b) 如果服务器有 docker，没有 docker-compose，使用以下命令安装单独 docker-compose：

```
sudo make install-compose
```

```
[root@localhost job]# make install-basic
tar xzvf ./docker.tgz
docker/
docker/containerd-shim-runc-v2
docker/dockerd
docker/docker-proxy
docker/ctr
docker/docker
docker/runc
docker/containerd-shim
docker/docker-init
docker/containerd
sudo cp docker/* /usr/bin/
sudo cp ./docker.socket /etc/systemd/system
sudo cp ./docker.service /etc/systemd/system
sudo systemctl daemon-reload
sudo systemctl restart docker
sudo systemctl enable docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /etc/systemd
/system/docker.service.
mv ./docker-compose /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
[root@localhost job]#
```

◆ 注意：在安装了 docker 和 docker-compose 之后，请尝试使用”docker ps”命令，目的在于如果使用非 root 用户时，没有足够的权限，会导致接下来的安装出问题。 如果显示权限不够，请尝试退出当前用户再进行重新登陆再次执行”docker ps”命令进行尝试，如果依旧不行，则需要重启服务器。

3、导入管理平台镜像

```
make install
```

```
[root@localhost test]# cat test
[root@localhost job]# make install
docker load -i docker-images.tar
8d3ac3489996: Loading layer  5.866MB/5.866MB
e5b50e000158: Loading layer   2.56kB/2.56kB
356bb4aa5e41: Loading layer  1.536kB/1.536kB
019d4257a802: Loading layer   2.56kB/2.56kB
40fc1018ad1a: Loading layer  10.66MB/10.66MB
Loaded image: image.turelljj.com/ture/styx-client:latest
5e03d8cae877: Loading layer  5.855MB/5.855MB
4e4849e08497: Loading layer  400.9kB/400.9kB
06c3de1f7bb3: Loading layer  400.9kB/400.9kB
54a0ca0e4663: Loading layer  784.9kB/784.9kB
a7020aa0c9eb: Loading layer  6.656kB/6.656kB
e2ea73294dc8: Loading layer  120.4MB/120.4MB
903c9cad0cde: Loading layer  83.84MB/83.84MB
......
a71fa97b9fb2: Loading layer  18.43kB/18.43kB
784069e74d35: Loading layer  23.07MB/23.07MB
f7583e6a06d7: Loading layer  4.608kB/4.608kB
bdfbab641900: Loading layer  1.536kB/1.536kB
24c6686d1a68: Loading layer  3.584kB/3.584kB
4bd277c0ad63: Loading layer  4.608kB/4.608kB
9b64657585cb: Loading layer  45.61MB/45.61MB
Loaded image: rabbitmq:3.9.13-management-alpine
[root@localhost test]#
```

4、启动管理平台

```
make up-mgmt-only     #启动管理平台
```

此步骤会提示要求输入 IP 地址，请按照提示输入此服务器的私网 IP 地址。

```
[root@styx job]# make up-mgmt-only
sed -i 's/192.168.2.227/'"$(ip route get 1 | awk '{print $NF;exit}')"'/g' ./docker-compose.yml
docker-compose -f ./docker-compose.yml up -d styxserver mongodb rabbitmq
Creating network "job_styx-network" with driver "bridge"
Creating network "job_default" with the default driver
Creating job_rabbitmq_1    ... done
Creating job_mongodb_1     ... done
Creating job_styxserver_1 ... done
[root@styx job]#
```

5、 验证平台是否启动成功

```
[root@styx job]# docker ps
CONTAINER ID   IMAGE                                  COMMAND                  CREATED              STATUS              PORTS
                                                                                                                                                         NAMES
6db40ef1c336   private404/styxserver:latest           "/bin/sh -c '/wait &…"   About a minute ago   Up About a minute   0.0.0.0:443->443/tcp, :::443->443/tcp
                                                                                                                                                         job_styxserver_1
5479dc45982f   mongo                                  "docker-entrypoint.s…"   About a minute ago   Up About a minute   0.0.0.0:27017->27017/tcp, :::27017->27017/tcp
                                                                                                                                                         job_mongodb_1
0ecfbeb0b07e   rabbitmq:3.9.13-management-alpine      "docker-entrypoint.s…"   About a minute ago   Up About a minute   0.0.0.0:4369->4369/tcp, :::4369->4369/tcp, 0.0.0.0:5671-5672->5671-5672/
tcp, :::5671-5672->5671-5672/tcp, 15671/tcp, 0.0.0.0:15672->15672/tcp, :::15672->15672/tcp, 0.0.0.0:25672->25672/tcp, :::25672->25672/tcp, 15691-15692/tcp   job_rabbitmq_1
[root@styx job]#
```

6、 此安装包内含有两台测试靶机，如果在安装好管理平台之后，需要启用靶机进行测试，可以使用以下命令启动靶机。

```
sudo make up
```

启动之后使用”docker ps”命令啦查看容器的运行及端口情况。

```
[root@styx job2]# make up
Enter primary ip or hostname of management server:192.168.2.146
# sed -i 's/192.168.2.227/'"$(ip route get 1 | awk '{print $NF;exit}')"'/g' ./docker-compose.yml
docker-compose -f ./docker-compose.yml up -d
job2_rabbitmq_1 is up-to-date
Creating job2_acme-daytime_1 ...
Creating job2_styx-client-1_1 ...
Creating job2_acme-daytime_1  ... done
Creating job2_styx-client-1_1 ... done
Creating job2_styx-client-2_1 ... done
Creating job2_jsptest_1       ... done
Creating job2_acme_1          ... done
[root@styx job2]# docker ps
CONTAINER ID   IMAGE                                        COMMAND                  CREATED          STATUS         PORTS
                                                                                                                                                  NAMES
205147f44af3   ture/styx:acme-java.v1                       "catalina.sh run"        7 minutes ago    Up 7 minutes   0.0.0.0:5001->8000/tcp, :::5001->8000/tcp, 0.0.0.0:8010->8080/tcp, :::801
0->8080/tcp                                                                                                                                       job2_acme_1
f86b10102a15   ture/styx:jsptest.v1                         "catalina.sh run"        7 minutes ago    Up 7 minutes   0.0.0.0:8020->8080/tcp, :::8020->8080/tcp
                                                                                                                                                  job2_jsptest_1
447bfa33cc0f   image.turellјj.com/ture/styx-client:latest   "./startup.sh"           7 minutes ago    Up 7 minutes
                                                                                                                                                  job2_styx-client-2_1
063732ebe289   image.turelljj.com/ture/styx-client:latest   "./startup.sh"           7 minutes ago    Up 7 minutes
                                                                                                                                                  job2_styx-client-1_1
d31b2640b7ba   ture/styx:acme-daytime.v1                    "ncat -l 13 --keep-o…"   7 minutes ago    Up 7 minutes   13/tcp
                                                                                                                                                  job2_acme-daytime_1
0f63307ca8a1   mongo                                        "docker-entrypoint.s…"   8 minutes ago    Up 7 minutes   0.0.0.0:27017->27017/tcp, :::27017->27017/tcp
                                                                                                                                                  job2_mongodb_1
263b6a84ec77   private404/styxserver:latest                 "/bin/sh -c '/wait &…"   8 minutes ago    Up 7 minutes   0.0.0.0:443->443/tcp, :::443->443/tcp
                                                                                                                                                  job2_styxserver_1
1de89b20351d   rabbitmq:3.9.13-management-alpine            "docker-entrypoint.s…"   8 minutes ago    Up 7 minutes   0.0.0.0:4369->4369/tcp, :::4369->4369/tcp, 0.0.0.0:5671-5672->5671-5672/t
cp, :::5671-5672->5671-5672/tcp, 15671/tcp, 0.0.0.0:15672->15672/tcp, :::15672->15672/tcp, 0.0.0.0:25672->25672/tcp, :::25672->25672/tcp, 15691-15692/tcp   job2_rabbitmq_1
[root@styx job2]#
```

可以看到 acme 和 jsptest 两个靶机已经被启动，端口分别为 8010 和 8020。

访问靶机：

Acme：http://IP:8010

Jsptest: http://IP:8020

2.3. 初始化安装管理平台

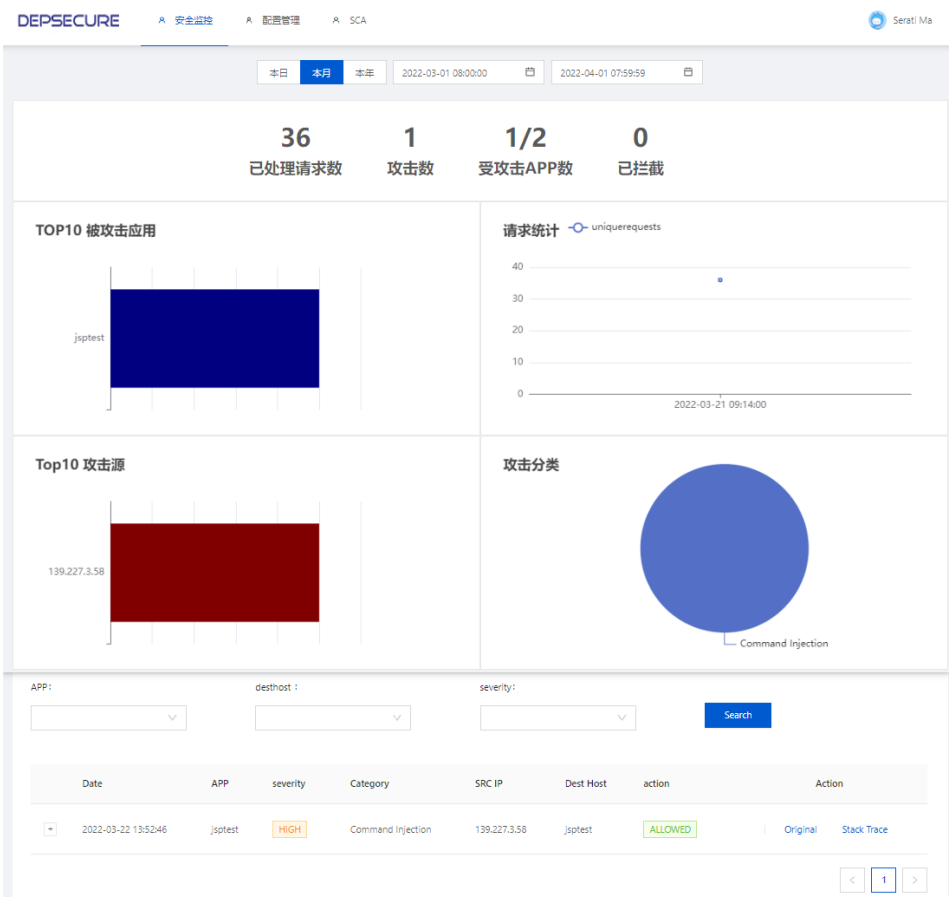
通过地址：https://IP 登录管理平台 web 页面(默认用户：admin 默认密码：Depsecure)

2.3.1.基本常见操作

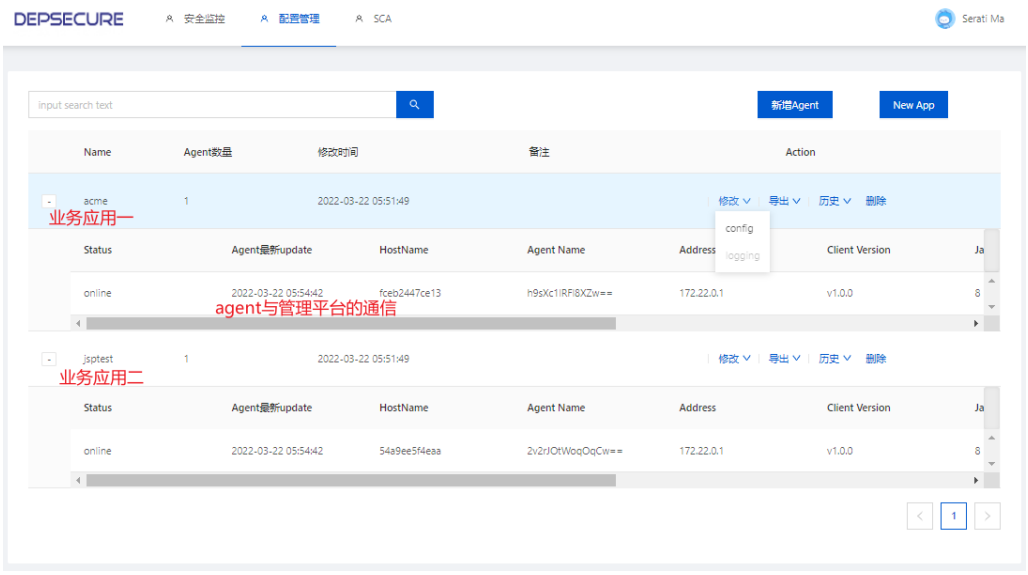
管理平台分为三部分：

- **主页**：用于安全态势监控
- **配置管理**：用于安装 styx 安全插件和配置管理
- SCA：用于“软件成分分析”

- 主页展示：

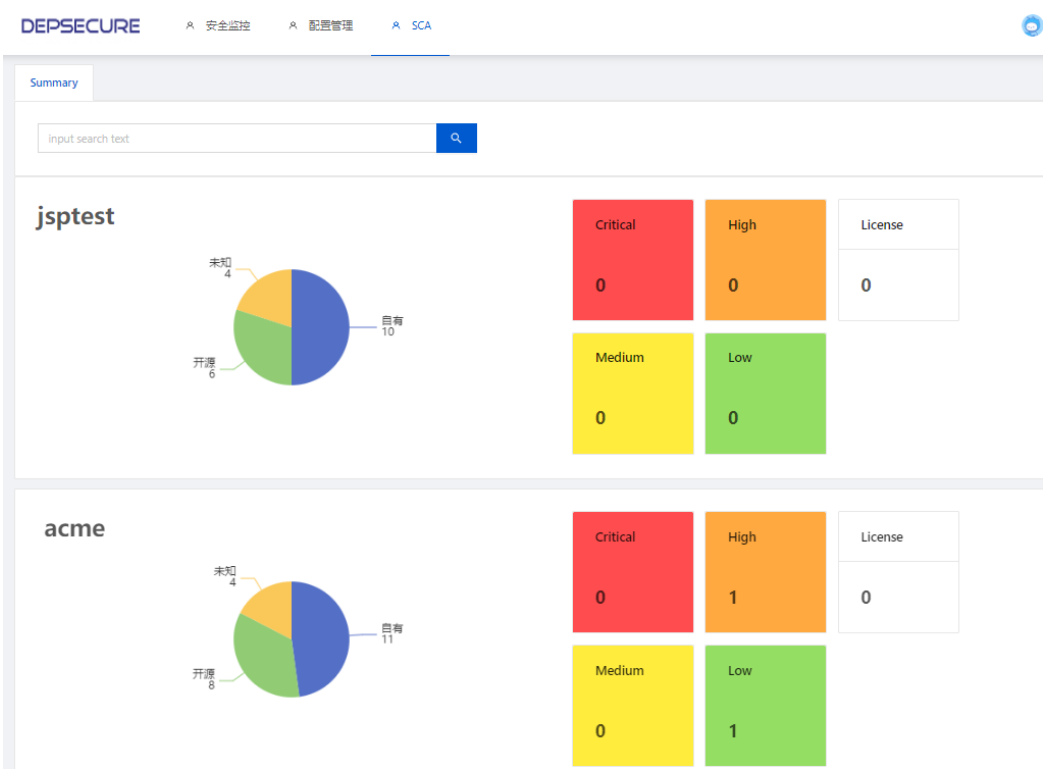


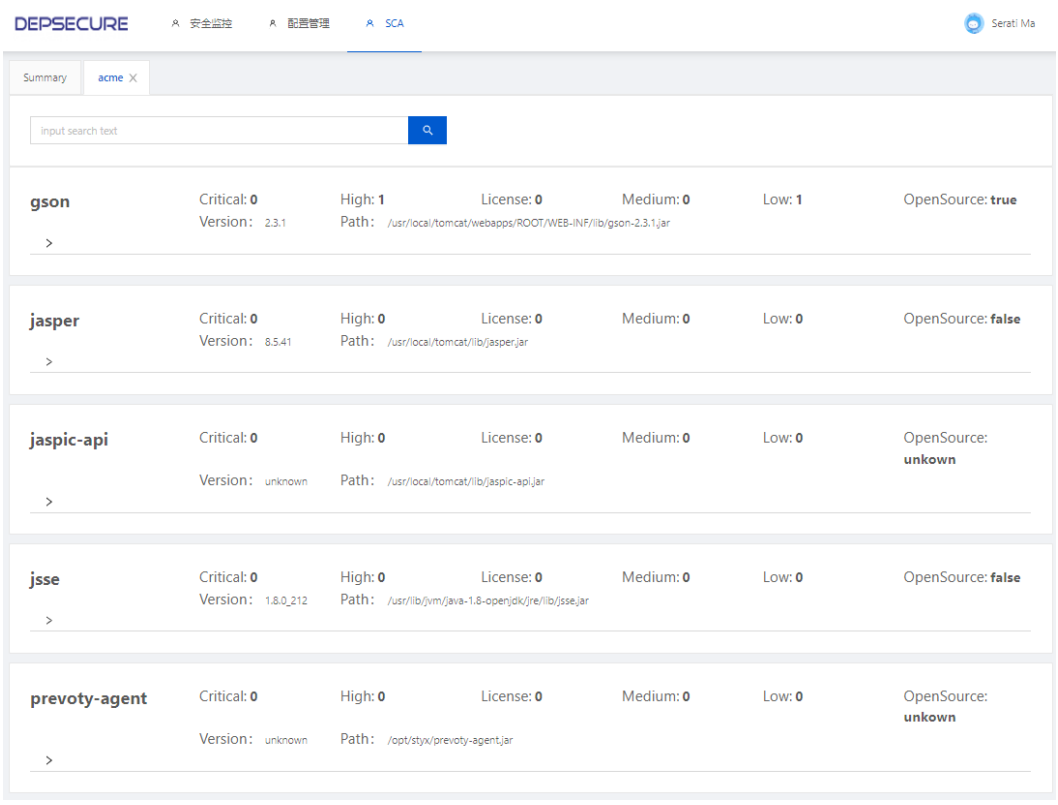
- 配置管理展示：



在配置管理界面，可以清楚的看到被保护的有多少个业务，每个业务中的哪几台服务器受保护，业务的 IP 地址，Agent 的版本信息，业务所使用的 Java 版本，系统信息（Linux/Mac/window），处理器架构(x86、Arm)、Styx-agent 的路径等信息。

- SCA 界面展示：





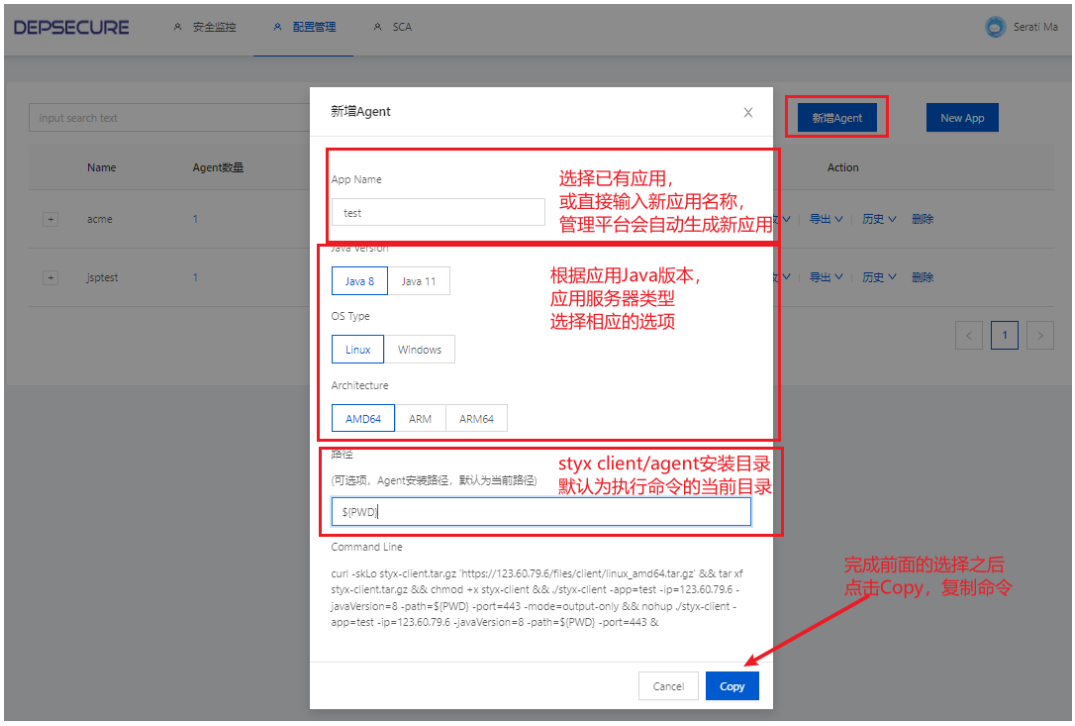
查看具体业务在运行是所调用的所有第三方 jar 包，是否存在漏洞等。

2.3.2. Agent 安装和环境变量修改

- 在配置管理界面，进行添加及删除业务和 agent。

- 新增业务

- 新增 agent



根据使用所需的系统环境进行选择，选择好之后，管理平台会自动生成下发 agent 的命令，将生成的命令在承载业务的服务上进行执行，即可完成 agent 的下发安装过程。

- 安装 Client/agent

连接到应用服务器上，新建一个工作文件夹（这里以/root/job/为例），粘贴执行上一步复制的命令，进行安装 styx-agent。示例：

```
curl -skLo styx-client.tar.gz 'https://123.60.79.6/files/client/linux_amd64.tar.gz' && tar xf styx-
```

```
client.tar.gz && chmod +x styx-client && ./styx-client -app=tomcat -ip=**.**.**.** -javaVersion=8 -path=${PWD} -port=443 -mode=output-only && nohup ./styx-client -app=tomcat -ip=***.***.**.** -javaVersion=8 -path=${PWD} -port=443 &
```

- 使用以下命令查看 agent 所在具体路径

```
cat styx-client.toml | grep path
```

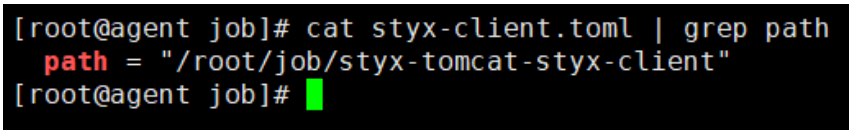
- 参照《快速集成手册》，根据应用服务器的不同，添加-javaagent 参数。这里以 tomcat 应用举例。

环境介绍：

这里是测试环境，所使用的是 Tomcat，版本 9（版本不重要），安装路径”/root/tomcat”,启动脚本路径”/root/tomcat/bin”

新建的工作目录为”/root/job/”

使用命令查看 agent 所在的具体路径（需要在执行“安装 client/agent“命令的工作目录下）。



集成示例：

参考《快速集成手册》，当前环境是 Tomcat 环境，所以使用 setenv.sh(tomcat 启动时，catalina 会自动查找 setenv.sh，并执行其中语句)脚本设置启动服务器去调用 jar 包。

生成生成脚本，赋予权限并移动到“<tomcat_home>/bin/“目录下

```
cat << EOF >> setenv.sh
export JAVA_OPTS="-javaagent:<agent_path>styx-agent.jar ${JAVA_OPTS}"
EOF

chmod +x setenv.sh
mv setenv.sh <tomcat_home>/bin/
```

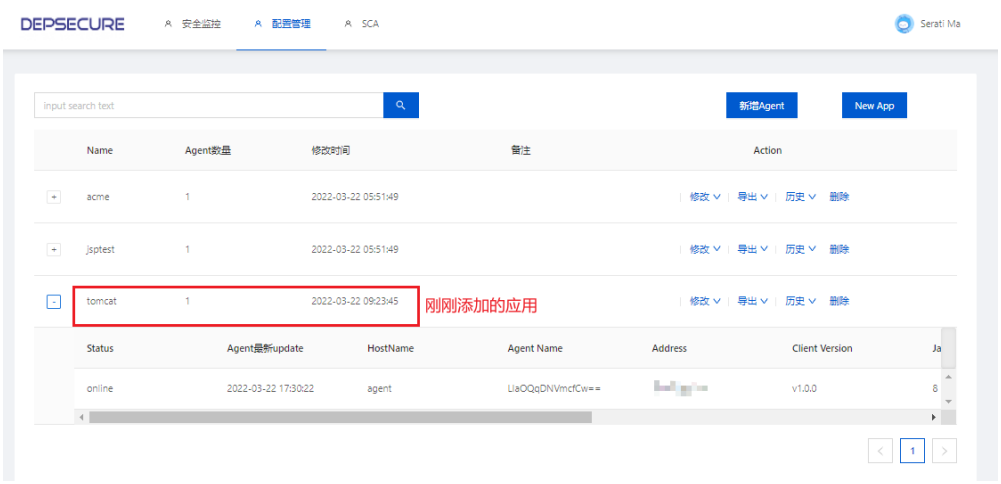
- 重新启动 tomcat

```
<tomcat_home>/bin/shutdown.sh
<tomcat_home>/bin/startup.sh
```

- 卸载只需去除”setenv.sh”，并重新启动 tomcat 即可。

```
rm <tomcat_home>/bin/setenv.sh
<tomcat_home>/bin/shutdown.sh
<tomcat_home>/bin/startup.sh
```

Agent 安装完成，查看管理界面。



应用总数已经由之前的 2 变为了三，说明应用添加成功，已经被监控。

其他环境下 agent 的安装方法类似，具体操作查看《styx-快速集成手册》

3. STYX-快速集成手册

◆ 注意：手册中出现的路径，需要调换成实际环境中，”styx-agent.jat”的绝对路径

3.1. 传统服务器模式

3.1.1. Spring Boot

原始启动命令

```
java -jar app.war
```

修改后的启动命令

```
java -javaagent:/root/styx/styx-agent.jar -jar app.war
```

卸载后的使用命令

```
java -jar app.war
```

3.1.2. Tomcat

使用 setenv.sh 脚本设置（tomcat 启动时，catalina 会自动查找 setenv.sh，并执行其中语句）

```
cat <<EOF >> setenv.sh
```

```
export JAVA_OPTS="-javaagent:/root/styx/styx-agent.jar ${JAVA_OPTS}"
EOF
chmod +x setenv.sh
mv setenv <tomcat_home>/bin/
```

1、 重新启动 tomcat

```
<tomcat_home>/bin/shutdown.sh
<tomcat_home>/bin/startup.sh
```

2、 卸载只需去除 setenv.sh，并重启 tomcat

```
rm <tomcat_home>/bin/setenv.sh
<tomcat_home>/bin/shutdown.sh
<tomcat_home>/bin/starup.sh
```

3.1.3. JBOSS

3.1.3.1. JBOSS 4~6

打开 bin/run.sh，找到任意以 JAVA_OPTS=为起始的行，

```
# Setup JBoss specific properties
JAVA_OPTS="-Dprogram.name=$PROGNAME $JAVA_OPTS"
```

在下面添加 JAVA_OPTS=”-javaagent:/root/styx/styx-agent.jar ${JAVA_OPTS}”

```
# Setup JBoss specific properties
JAVA_OPTS="-Dprogram.name=$PROGNAME $JAVA_OPTS"
JAVA_OPTS="-javaagent:/root/styx/styx-agent.jar ${JAVA_OPTS}"
```

重启

3.1.3.2. JBOSS EAP Standalone 模式

打开 bin/standalone.sh，找到# Display our environment 处，在下面添加：
JAVA_OPTS="-javaagent:/root/styx/styx-agent.jar ${JAVA_OPTS}"
1. 重启

3.1.3.3. JBOSS EAP Domain 模式

如果按照 server-group 配置 styx，此 group 下面所有的服务器都会安装 styx，打开 domain/configuration/domain.xml 文件,找到<server-groups>标签,在需要安装 styx 的 server-group 中找到<jvm>标签添加如下配置：

```
<jvm-options>
    <option value=”-javaagent:/root/styx/styx-agent.jar”/>
<jvm-options>
```

重启

3.1.4.Wildfly

3.1.4.1.Wildfly Standalone 模式

打开 bin/standalone.sh，找到# Display our environment 处，在下面添加：
JAVA_OPTS="-javaagent:/root/styx/styx-agent.jar ${JAVA_OPTS}"
1. 重启

3.1.4.2.Wildfly Domain 模式

如果按照 server-group 配置 styx，此 group 下面所有的服务器都会安装 styx，打开 domain/configuration/domain.xml 文件,找到<server-groups>标签,在需要安装 styx 的 server-group 中找到<jvm-options>标签添加如下配置：

```
<jvm-options>
   <option value=" -javaagent:/root/styx/styx-agent.jar" />
<jvm-options>
```

重启 wildfly

3.1.5.Jetty

原始的启动命令
java -jar app.war
1. 修改后的启动命令
java -javaagent:/root/styx/styx-agent.jar -jar app.war
2. 卸载后的使用命令
java -jar app.war

3.1.6.WebLogic

3.1.6.1.非集群方式

打开 bin/startWebLogic.sh, 在 JAVA_OPTIONS="${SAVE_JAVA_OPTIONS}" 下增加：
JAVA_OPTIONS="-javaagent:/root/styx/styx-agent.jar ${JAVA_OPTIONS}"
1. 重启

3.1.6.2.集群方式

找到<weblogic-home>/user_projects/domains/base_domain/config 目录，打开

config.xml 文件,定位到需要安装 styx 的<server>/<server-start>/<arguments>标签,在<arguments>标签内
添加 javaagent 参数。
如果没有相应的标签可以手动添加，e.g

```
<server>
    <server-start>
        <arguments>-javaagent:/root/styx/styx-agent.jar</arguments>
    </server-start>
</server>
```

重启

3.1.7.WebSphere

1. 以 WAS 8.5 为例，在控制台左侧的导航栏里，选择 Servers -> Server Types ->WebSphere Application Server，进入应用列表界面:
2. 选择你要开启 Styx 的应用（这里是 server1），点击进入管理页面。在新页面向下翻，找到 Server Infrastructure -> Process definition，并点击进入:
3. 之后在右侧，点击 Additional Properties -> Java Virtual Machine 进入 JVM 启动参数编辑界面
4. 最后找到 Generic JVM arguments，加入以下参数
   -javaagent:/root/styx/styx-agent.jar

3.1.8.容器环境集成

容器环境的一大特性是单个应用单个容器，无需担心设置全局环境变量会影响到其他应用。 所以在容器环境中集成 styx 时，可直接设置全局 JAVA_TOOL_OPTIONS

styx-helper 用作与管理平台之间的通讯, 需要环境变量 STYX_MNG, 如 192.168.2.228

styx-init 用作 setup empty_dir volume，由于必须在业务启动前 setup 好，所以单独使用
init-container。 需要环境变量 JAVA_VERSION, 值可为 java8 或 java11 需要环境变量 APP_NAME, 如 styx-test

```
apiVersion: apps/v1
kind: Deployment
metadata:
    creationTimestamp: null
    labels:
        app: test-original
    name: test-original
spec:
    replicas: 1
    selector:
```

```
        matchLabels:
            app: test-original
    strategy: {}
    template:
        metadata:
            creationTimestamp: null
            labels:
                app: test-original
            spec:
                containers:
                - image: ture/styx:jsptest.v1
                  name: styx
                  resources: {}
status: {}

apiVersion: apps/v1
kind: Deployment
metadata:
    creationTimestamp: null
    labels:
        app: test-after
    name: test-after
spec:
    replicas: 1
    selector:
        matchLabels:
            app: test-after
        strategy: {}
        template:
            metadata:
                creationTimestamp: null
                labels:
                    app: test-after
            spec:
                containers:
                - image: ture/styx:jsptest.v1
                    name: jsptest
                    resources: {}
                    volumeMounts: #将 styx 插件文件挂载到业务容器中
                    - name: styx-agents #
                        mountPath: /opt/styx/ #
                    env: #添加环境变量，以便业务启动时带上 styx 参数
                    - name: JAVA_TOOL_OPTIONS #
                        value: "-javaagent:/opt/styx/styx-agent.jar" #
```

```
                - name: styx-helper #styx-helper，用途为与 styx 管理平台交
互
                    image: ture/styx:styx-helper.v1.0.8 #
                    volumeMounts: #
                    - name: styx-agents #
                        mountPath: /opt/styx/ #
                    env: #
                    - name: STYX_MNG #
                        value: "192.168.2.228" #需修改为现场 Styx 管理平台
ip #
                initContainers: #styx-init 用于在业务启动前设置好
                - name: styx-init #
                    image: ture/styx:styx-init.v1.0.8 #
                    volumeMounts: #
                    - name: styx-agents #
                        mountPath: /opt/styx #
                    env: #
                    - name: JAVA_VERSION #
                        value: "java8" #请根据业务 java 版本修改,可为 java8
或者 java11
                    - name: APP_NAME #
                        value: "jsptest-k8s" #请修改为业务名称，为唯一标
识符
                volumes: #
                - name: styx-agents #emptyDir 卷，用户承载 styx 插件。
                    emptyDir: {} #
status: {}
```

如果 K8S 环境只能使用私有仓库可使用如下命令，上传 styx-helper styx-init 到私有仓库。

并相应修改 deployment yml 中 styx-helper 容器和 styx-init 容器到对应 image。

```
docker load -i styx-helper.tar
docker load -i styx-init.tar
docker image tag ture/styx:styx-helper.v1.0.8
new-registry.example.com/styx-helper:v1.0.8
docker image tag ture/styx:styx-init.v1.0.8
new-registry.example.com/styx-init:v1.0.8
docker push new-registry.example.com/styx-helper:v1.0.8
docker push new-registry.example.com/styx-init:v1.0.8
```

3.1.9.验证是否安装成功

可通过验证的一下日志文件与日志产出验证 Styx 是否运行正常。

打开 styx.json 文件，查看启动日志。

如下日志表明 Styx Agent 已经正常加载：

```
RASP Agent: Premain Invoked
```

如下日志表明 Styx RASP jar 文件已经正常加载：

```
[STDOUT] Loaded /native/Linux/x86_64/librasp.so
[STDOUT] [rasp: 1.1.0]
### NOTE: This line varies based on the operating system. Windows
example:
Loaded /native/Windows/x86_64/rasp.dll
```

如下日志表明单独的安全模块已经正常启动：

```
Installing CmdinjectionAgentRuntime
Installing NetworkAgentRuntime
Installing PathTraversalAgentRuntime
Installing QueryAgentRuntime
Installing ServletAPIAgentRuntime
Installing WeakCyptoAgentRuntime
```

3.1.10 插件卸载

请按照以下步骤卸载插件：

1. 去除 JVM 参数(对应设置 JVM 参数的文件中)

```
-javaagent:${STYX_ROOT}/plugins/styx-agent.jar
```

1. 删除 Styx 文件
2. 重启应用服务 重启应用服务，让新的 JVM 参数生效

4 版本升级

升级前的准备：

- 下载好安装包，下载地址同安装时一样。
- 下载好之后，解压，

升级操作：

```
[root@styx ~]# mkdir job2
[root@styx ~]# tar -vxf styx-env-2022-03-28_05.tar.gz -C job2/
docker-compose
docker-compose.yml
docker-images.tar
docker.service
docker.socket
docker.tgz
makefile
[root@styx ~]# ls job2/
docker-compose  docker-compose.yml  docker-images.tar  docker.service  docker.socket  docker.tgz  makefile
[root@styx ~]#
```

- 切换工作目录到”job2”（刚刚解压的目录）中，
- 输入命令“make install“导入一下镜像
- 然后切换工作目录到”job”（之前版本的启动目录），
- 输入命令”make up-mgmt-only”，进行拉起容器（此次启动，使用更新过的镜像，即未更新的镜像所创建的容器，不会改变）

```
Recreating job_rabbitmq_1 ...
job_styx-helper-1_1 is up-to-date
job_mongodb_1 is up-to-date
job_mysql_1 is up-to-date
job_styx-helper-2_1 is up-to-date
Recreating job_styxserver_1 ...
job_acme-daytime_1 is up-to-date
Recreating job_rabbitmq_1    ... done
Recreating job_styxserver_1 ... done
```

- 此四步骤后期有截图麻烦补上

◆ 注意点：这里给出的升级方案在下载更新包时，选择了整体下载最新的安装包，其实在使用时只是使用了最新的 images，所以在“make install“更新镜像后，“job”目录即可弃用。