

Informe Auditoría: “La Rodilla”



Universidad
Francisco de Vitoria
UFV Madrid

*Ciclos Formativos de
Grado Superior · CETYS*

**Gonzalo Pascual
Lucas Gavin
Luca Sancho
Adrián Tenorio
Tomás Rodríguez-Mata**

RODILLA

DESDE 1939



ÍNDICE DEL DOCUMENTO

1. Introducción	2
2. Alcance	2
3. Metodología	3
4. Resultados	9
5. Alcance Económico	31



1. Introducción

En respuesta a la creciente importancia de la seguridad informática en entornos empresariales, se ha llevado a cabo una auditoría integral al establecimiento de "La Rodilla", ubicada en el campus universitario de la Universidad Francisco de Vitoria. La seguridad de la información es esencial para salvaguardar los datos, la privacidad y la continuidad de las operaciones.

La presente auditoría se ha centrado en evaluar la infraestructura tecnológica, las prácticas de seguridad de "La Rodilla" y las vulnerabilidades del software más sensible para la organización. Se ha empleado la herramienta JFrog para la detección de vulnerabilidades, y Docker como contenedor para evaluar las aplicaciones críticas y asegurar la integridad del entorno informático.

2. Alcance

La auditoría se centra en evaluar la seguridad informática de "La Rodilla", tienda ubicada en el campus universitario, con un enfoque específico en el análisis de vulnerabilidades en aplicaciones críticas para su operación. Se dará especial atención a las siguientes aplicaciones:

- **Nginx:**
 - Evaluación de configuraciones de seguridad.
 - Detección de posibles vulnerabilidades en la implementación del servidor web Nginx.
- **Python:**
 - Análisis de la seguridad en la ejecución de scripts y aplicaciones basadas en Python.
 - Identificación de posibles debilidades en las prácticas de desarrollo y configuración.
- **Httpd (Apache):**
 - Revisión de configuraciones de seguridad en el servidor web Apache.
 - Detección de vulnerabilidades potenciales que puedan afectar la seguridad de "La Rodilla".
- **MariaDB:**
 - Evaluación de la seguridad en la base de datos MariaDB.
 - Identificación de posibles vulnerabilidades y configuraciones subóptimas.
- **PHP:**
 - Análisis de la seguridad en el uso del lenguaje de programación PHP.
 - Detección de vulnerabilidades comunes y buenas prácticas de seguridad en el desarrollo de aplicaciones basadas en PHP.



El enfoque en estas aplicaciones críticas permitirá una evaluación detallada de la seguridad en la tecnología utilizada por "La Rodilla". La utilización de JFrog y Docker facilitará la identificación de posibles amenazas y la propuesta de soluciones con el objetivo de fortalecer la seguridad de las aplicaciones clave en el entorno.

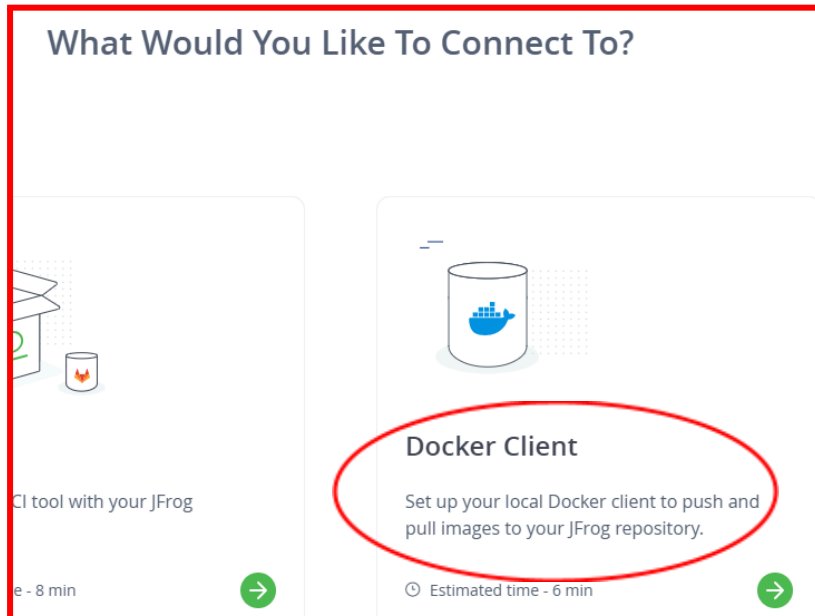
3. Metodología

La auditoría se realizó en una máquina virtual instalando Docker y añadiendo las debidas imágenes y se conectó Jfrog siguiendo los siguientes pasos:

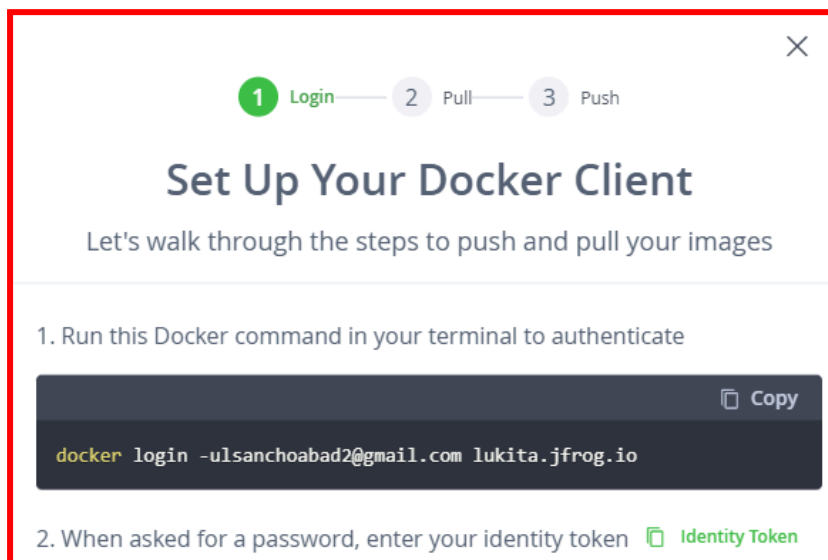
Una vez entremos en Jfrog con la sesión, lo primero que hemos hecho es crear el primer repositorio, que en este caso va a ser de docker virtual

Get Started with JFrog
We recommend these steps to get you started
0%
Set up your workflow 0/3
1 Create your first repository
Create a repository to host your favorite package type
Docker
Create Repositories
Assign a name to your new repositories by adding a meaningful prefix identifier
Repositories prefix ② dockervirtual 13/20

Una vez hayamos creado el repositorio, que en nuestro caso es docker virtual, lo conectamos como cliente al docker que tenemos dentro de la máquina virtual del entorno del laboratorio



Para conectarlo, vinculamos mediante el comando que nos proporciona JFrog el docker de la máquina virtual



```
root@Produccion:/home/user# docker login -ulsanchoabad2@gmail.com lukita.jfrog.io
Password: █
```



```
root@Produccion:/home/user# docker login -ulsanchoabad2@gmail.com lukita.jfrog.io
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

Una vez vinculado ya el docker, el propio JFrog nos enseña a coger una imagen y “descargarla” en el propio repositorio creado, mediante los comandos pull, tag y push con el ejemplo de “Hello-world”

Pull Your First Image

Pull an image using Artifactory as a proxy to Docker Hub

Run this Docker command in your terminal to pull an image

```
docker pull lukita.jfrog.io/dockervirtual-docker/hello-world:latest
```

```
root@Produccion:/home/user# docker pull lukita.jfrog.io/dockervirtual-docker/hello-world:latest
latest: Pulling from dockervirtual-docker/hello-world
Digest: sha256:88ec0acaa3ec199d3b7eaf73588f4518c25f9d34f58ce9a0df68429c5af48e8d
Status: Downloaded newer image for lukita.jfrog.io/dockervirtual-docker/hello-world:latest
lukita.jfrog.io/dockervirtual-docker/hello-world:latest
```

Retag and Push Your Image

Last step: re-tag and push this Docker image

1. Tag the image

```
docker tag lukita.jfrog.io/dockervirtual-docker/hello-world
lukita.jfrog.io/dockervirtual-docker/hello-world:1.0.0
```

2. Push the image

```
docker push lukita.jfrog.io/dockervirtual-docker/hello-world:1.0.0
```



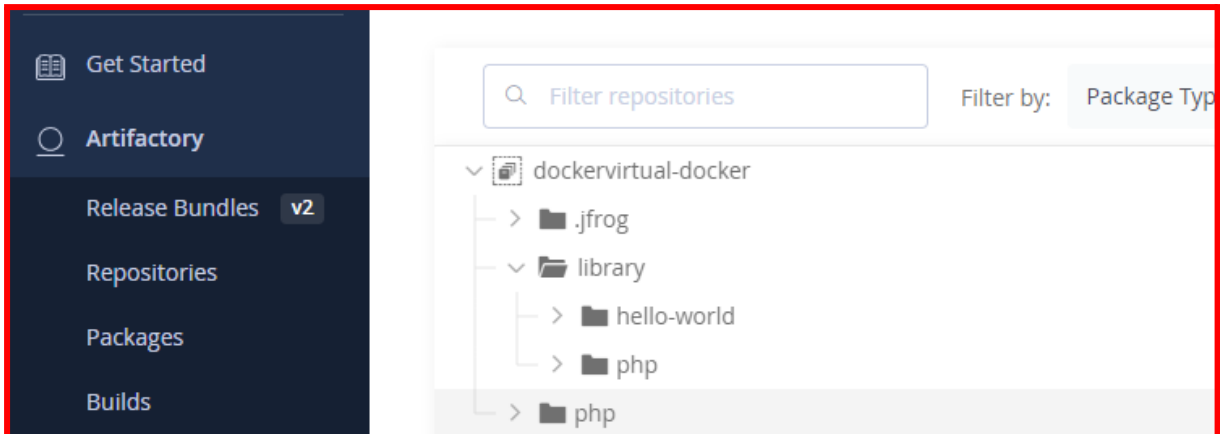
```
root@Produccion:/home/user# docker tag lukita.jfrog.io/dockervirtual-docker/hello-world lukita.jfrog.io/dockervirtual-docker/hello-world:1.0.0
root@Produccion:/home/user# docker push lukita.jfrog.io/dockervirtual-docker/hello-world:1.0.0
The push refers to repository [lukita.jfrog.io/dockervirtual-docker/hello-world]
01bb4fce3eb1: Pushed
1.0.0: digest: sha256:7e9b6e7ba2842c91cf49f3e214d04a7a496f8214356f41d81a6e6dcad11f11e3 size: 525
```

El ejemplo de JFrog anterior fue sobre la imagen HelloWorld, así que repetimos el mismo proceso para las siguientes imágenes, en todas se hace de la misma manera:

- Php:

```
root@Produccion:/home/user# docker pull lukita.jfrog.io/dockervirtual-docker/php:latest
latest: Pulling from dockervirtual-docker/php
Digest: sha256:c4350e495ee03126f49794a49dacde6d0c0730e68b9ec7b3eaca36be4a7dc04a
Status: Downloaded newer image for lukita.jfrog.io/dockervirtual-docker/php:latest
```

```
root@Produccion:/home/user# docker tag lukita.jfrog.io/dockervirtual-docker/php lukita.jfrog.io/dockervirtual-docker/php:latest
root@Produccion:/home/user# docker push lukita.jfrog.io/dockervirtual-docker/php:latest
The push refers to repository [lukita.jfrog.io/dockervirtual-docker/php]
7730d5a1100c: Pushed
e3ded4b3cf95: Pushed
57deefe6f004: Pushed
dade862fc2fd: Pushed
20b71db9d728: Pushed
3be4dadb5906: Pushed
8f84dd0e178d: Pushed
a49bca1f41af: Pushed
ec983b166360: Pushed
latest: digest: sha256:0bc8a9e0d830fd2f3131976cb309a8a084c3cc29864588568ee2e0dab0005c5f size: 2202
```



Podemos observar que se ha añadido la imagen de php, así que seguimos con las demás



- **Nginx:**

```
root@Produccion:/home/user# docker pull lukita.jfrog.io/dockervirtual-docker/nginx:latest
latest: Pulling from dockervirtual-docker/nginx
Digest: sha256:86e53c4c16a6a276b204b0fd3a8143d86547c967dc8258b3d47c3a21bb68d3c6
Status: Downloaded newer image for lukita.jfrog.io/dockervirtual-docker/nginx:latest
lukita.jfrog.io/dockervirtual-docker/nginx:latest
```

```
root@Produccion:/home/user# docker tag lukita.jfrog.io/dockervirtual-docker/nginx lukita.jfrog.io/docker
virtual-docker/nginx:latest
root@Produccion:/home/user# docker push lukita.jfrog.io/dockervirtual-docker/nginx:latest
The push refers to repository [lukita.jfrog.io/dockervirtual-docker/nginx]
505f49f13fbe: Pushed
9920f1ebf52b: Pushed
768e28a222fd: Pushed
715b32fa0f12: Pushed
e503754c9a26: Pushed
609f2a18d224: Pushed
ec983b166360: Layer already exists
latest: digest: sha256:d2e65182b5fd330470eca9b8e23e8a1a0d87cc9b820eb1fb3f034bf8248d37ee size: 1778
```

- **Httpd:**

```
root@Produccion:/home/user# docker pull lukita.jfrog.io/dockervirtual-docker/httpd:latest
latest: Pulling from dockervirtual-docker/httpd
Digest: sha256:4e24356b4b0aa7a961e7dfb9e1e5025ca3874c532fa5d999f13f8fc33c09d1b7
Status: Downloaded newer image for lukita.jfrog.io/dockervirtual-docker/httpd:latest
lukita.jfrog.io/dockervirtual-docker/httpd:latest
```

```
root@Produccion:/home/user# docker tag lukita.jfrog.io/dockervirtual-docker/httpd lukita.jfrog.io/docker
virtual-docker/httpd:latest
root@Produccion:/home/user# docker push lukita.jfrog.io/dockervirtual-docker/httpd:latest
The push refers to repository [lukita.jfrog.io/dockervirtual-docker/httpd]
cdae29f197b2: Pushed
5ca84525a215: Pushed
e0ba343b5cab: Pushed
2905795cb5b8: Pushed
ec983b166360: Layer already exists
latest: digest: sha256:d70861224a52d0175c4b485ef7b70f56125f2c74d8a489a9a8ca10e3656929c9 size: 1366
```




- **Python:**

```
root@Produccion:/home/user# docker pull lukita.jfrog.io/dockervirtual-docker/python:latest
latest: Pulling from dockervirtual-docker/python
Digest: sha256:7b8d65a924f596eb65306214f559253c468336bcae09fd575429774563460caf
Status: Downloaded newer image for lukita.jfrog.io/dockervirtual-docker/python:latest
lukita.jfrog.io/dockervirtual-docker/python:latest
```

```
root@Produccion:/home/user# docker tag lukita.jfrog.io/dockervirtual-docker/python lukita.jfrog.io/dockervirtual-docker/python:latest
root@Produccion:/home/user# docker push lukita.jfrog.io/dockervirtual-docker/python:latest
The push refers to repository [lukita.jfrog.io/dockervirtual-docker/python]
701d0b971f5f: Pushed
619584b251c8: Pushed
ac630c4fd960: Pushed
86e50e0709ee: Pushed
12b956927ba2: Pushed
266def75d28e: Pushed
29e49b59edda: Pushed
1777ac7d307b: Pushed
latest: digest: sha256:5a2936b50ea64ce3e090c862d2482d5d90ed19ee2ceba5cf96ea171bd1dcba67 size: 2007
```

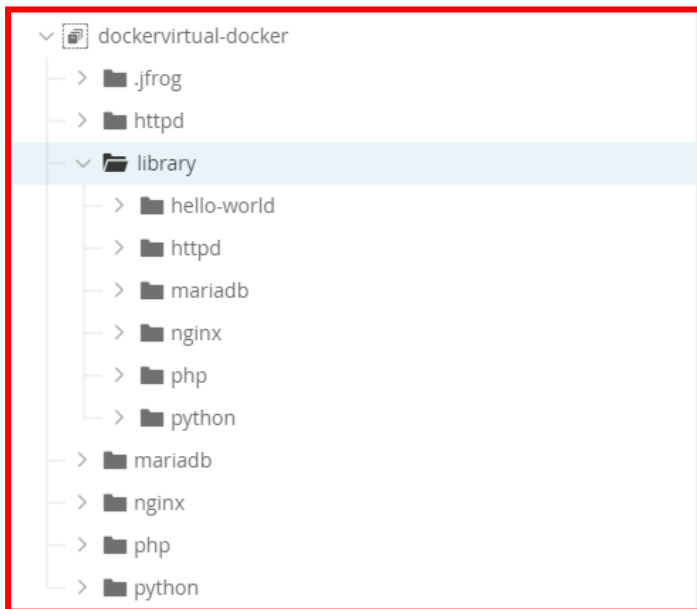
- **Mariadb:**

```
root@Produccion:/home/user# docker pull lukita.jfrog.io/dockervirtual-docker/mariadb:latest
latest: Pulling from dockervirtual-docker/mariadb
Digest: sha256:2403cc521634162f743b5179ff5b35520daf72df5d9e7e397192af685d9148fd
Status: Downloaded newer image for lukita.jfrog.io/dockervirtual-docker/mariadb:latest
lukita.jfrog.io/dockervirtual-docker/mariadb:latest
```

```
root@Produccion:/home/user# docker tag lukita.jfrog.io/dockervirtual-docker/mariadb lukita.jfrog.io/dockervirtual-docker/mariadb:latest
root@Produccion:/home/user# docker push lukita.jfrog.io/dockervirtual-docker/mariadb:latest
The push refers to repository [lukita.jfrog.io/dockervirtual-docker/mariadb]
43ac6280a320: Pushed
4492981ce68f: Pushed
35eb6f9aef4e: Pushed
1735b46dd2eb: Pushed
cd6359feb8c8: Pushed
bd769c7980ae: Pushed
9ba16167b8ea: Pushed
256d88da4185: Pushed
latest: digest: sha256:e51c275914b2da5e8e8e0ed9eaecf1e4d5142b5e570f231224320001cf5c86cf size: 1991
```



En el propio JFrog podemos comprobar que se han añadido todas las imágenes para su posterior análisis de vulnerabilidades



4. Resultados

Dentro del apartado de Xray, scan list, podemos identificar las vulnerabilidades de las imágenes anteriormente añadidas

Artifact Name	Violations	Malicious Packages	Vulnerabilities	Exposures	Repository Path
mariadb/latest	8	Not Found	77	Not Scanned	/mariadb/latest/manifest.json
python/latest	0	Not Found	70	Not Scanned	/python/latest/manifest.json
httpd/latest	0	Not Found	16	Not Scanned	/httpd/latest/manifest.json
nginx/latest	0	Not Found	33	Not Scanned	/nginx/latest/manifest.json
php/latest	0	Not Found	18	Not Scanned	/php/latest/manifest.json

A continuación estudiaremos las vulnerabilidades más críticas de cada paquete ordenadas por el CVVv3 (Sistema de Puntuación de Vulnerabilidades Comunes) que es un estándar utilizado para evaluar y puntuar la gravedad de las vulnerabilidades de seguridad, proporcionando una métrica numérica para medir el impacto de una vulnerabilidad.



- Mariadb

Mariadb cuenta con una gran lista de vulnerabilidades, cuyas más importantes son las de nivel crítico que se han detectado 8 y de nivel alto que se han detectado 35

Xray > Scans List > dockervirtual-docker-local > mariadb/latest

Scan Name
dockervirtual-docker-local/latest

Overview

- Policy Violations 8
- SBOM 461
- Security Issues 77
- Vulnerabilities 77
- Malicious Packages 0
- Secrets
- Services
- Applications
- Descendants
- Ancestors

mariadb/latest

Repository Path
dockervirtual-docker-local/mariadb/latest/manifes...

Created on
08 Nov 2023 16:28 (GMT+0100)

Created by
lsanchoabad2@gmail.com

Downloads
0

Vulnerabilities

by Severity

- Critical 8
- High 35
- Medium 16
- Low 18
- Unknown 0

Critical & High Vulnerabilities by Applicability

Applicability Not Scanned

Scan Now

Policy Violations

by Severity

- Critical 8
- High 0
- Medium 0
- Low 0
- Unknown 0

by Type

- Security 8
- License 0
- Operational 0

Xray > Scans List > dockervirtual-docker-local > mariadb/latest

Scan Name
dockervirtual-docker-local/latest

Overview

- Policy Violations 8
- SBOM 461
- Security Issues 77
- Vulnerabilities 77
- Malicious Packages 0
- Secrets
- Services
- Applications
- Descendants
- Ancestors

Critical & High Vulnerabilities
43

Includes Fix Version
61

Enriched by JFrog
42

Component With Most Vulnerabilities
51 | github.com/golang/go 1.16.7

8 Vulnerabilities

Severity	ID	Contextual Analysis	Component	Fix Version	CVSS v3
Critical	CVE-2022-23806	UNDETERMINED	github.com/golang/go:1.16.7	1.16.14, 1.17.7	9.1
Critical	CVE-2023-24538	UNDETERMINED	github.com/golang/go:1.16.7	1.19.8, 1.20.3	9.8
Critical	CVE-2023-29402	UNDETERMINED	github.com/golang/go:1.16.7	1.19.10, 1.20.5	9.8
Critical	CVE-2023-29405	UNDETERMINED	github.com/golang/go:1.16.7	1.19.10, 1.20.5	9.8
Critical	CVE-2023-29404	UNDETERMINED	github.com/golang/go:1.16.7	1.19.10, 1.20.5	9.8
Critical	CVE-2023-39323	UNDETERMINED	github.com/golang/go:1.16.7	1.20.9, 1.21.2	9.8
Critical	CVE-2023-24540	UNDETERMINED	github.com/golang/go:1.16.7	1.19.9, 1.20.4	9.8
Critical	CVE-2021-38297	UNDETERMINED	github.com/golang/go:1.16.7	1.16.9, 1.17.2	9.8

Dentro de las vulnerabilidades críticas, vamos a destacar 4 de ellas



- CVE-2023-24538

Esta vulnerabilidad trata de que, la falta del delimitador que se escapa en Go `html/template` provoca la inyección de código JavaScript cuando se utilizan acciones de plantilla de Go dentro de literales de plantilla de JavaScript.

CVE-2023-24538

JFrog research last updated on 19 Apr 2023 4:02 PM
This CVE is enriched by JFrog research and provides more accurate information

Xray ID: XRAY-513412

Contextual Analysis: Not Scanned

JFrog Severity: Medium

Components (1)

Name: github.com/golang/go:1.16.7

Version: 1.16.7

Upgrade to: 1.19.8, 1.20.3

CVSS Score: **9.8 (v3)**

CWE: CWE-94

Show Less

JFrog Research | Public Sources | Impact Paths | References

Summary

Missing delimiter escaping in Go `html/template` leads to JavaScript code injection when using Go template actions inside JavaScript template literals.

Remedio: El paquete `html/template` de Go implementa plantillas basadas en datos para generar resultados HTML seguros contra la inyección de código. Se descubrió que el paquete de plantilla no considera correctamente las comillas invertidas como delimitadores de cadenas JavaScript y no las escapa como se esperaba.

Por lo tanto, si una plantilla de Go contiene una acción de plantilla de Go (por ejemplo, `{{.}}`) dentro de un literal de plantilla de JavaScript (cualquier literal delimitado por un carácter ``` de comilla invertida) y un atacante puede controlar la salida de la acción de plantilla de Go, entonces el atacante puede generar el carácter de comilla invertida de la acción que escapará del literal de la plantilla de JavaScript y permitirá la inyección de código JavaScript arbitrario.



- CVE-2023-29404

Esta vulnerabilidad consta de que hay una validación insuficiente en el vinculador de Go que conduce a la ejecución de código en tiempo de compilación al compilar código fuente que no es de confianza.

Se puede remediar deshabilitando el soporte "cgo" ejecutando la herramienta go con `CGO_ENABLED=0`

CVE-2023-29404

JFrog research last updated on 31 Dec 0 11:45 PM
This CVE is enriched by JFrog research and provides more accurate information

Xray ID XRAY-521542

Contextual Analysis Not Scanned

JFrog Severity High

Components (1)

Name github.com/golang/go:1.16.7

Version 1.16.7

Upgrade to 1.19.10, 1.20.5

CVSS Score **9.8 (v3)**

CWE CWE-94

Show Less

JFrog Research Public Sources Impact Paths References

Summary

Insufficient validation in Go's linker leads to build-time code execution when compiling untrusted source code.

[Remedio:](#) Se puede remediar deshabilitando el soporte "cgo" ejecutando la herramienta go con `CGO_ENABLED=0`

- **CVE-2021-38297**

Esta vulnerabilidad consta de que hay un desbordamiento del búfer en el módulo Wasm de Go, que podría provocar la ejecución remota de código en un espacio aislado al analizar argumentos de línea de comando maliciosos.

CVE-2021-38297

JFrog research last updated on 08 Jan 2023 8:28 PM
This CVE is enriched by JFrog research and provides more accurate information

Xray ID

XRAY-187759

Contextual Analysis

Not Scanned

JFrog Severity

M Medium

Components (1)

Name

github.com/golang/go:1.16.7

Version

1.16.7

Upgrade to

1.16.9, 1.17.2

CVSS Score

7.5 (v2) 9.8 (v3)

CWE

CWE-120

Show Less

JFrog Research

Public Sources

Impact Paths

References

Summary

A buffer overflow in Go's Wasm module could lead to sandboxed remote code execution when parsing malicious command line arguments.

Asegúrese de que al ejecutar el binario Wasm, verifique que la longitud total de *go.argv* sea menor que 4k

JFrog ResearchPublic SourcesImpact PathsReferences

When running the Wasm binary - check that the total length of `go.argv` is smaller than 4k -

```
const maxSize = 4096;
function checkWasmCmd(arr) {
  var len = arr.reduce(
    (cur, elem) => {
      //pad each element to 8 bytes
      return cur + elem.length + (8 - (elem.length % 8))
    }, 0
  );
}
```



- CVE-2022-23806

En esta vulnerabilidad, se ha detectado que hay una valoración inadecuada de argumentos en la función *IsOnCurve()* del paquete criptográfico/elíptico de Go puede hacer que devuelva resultados incorrectos

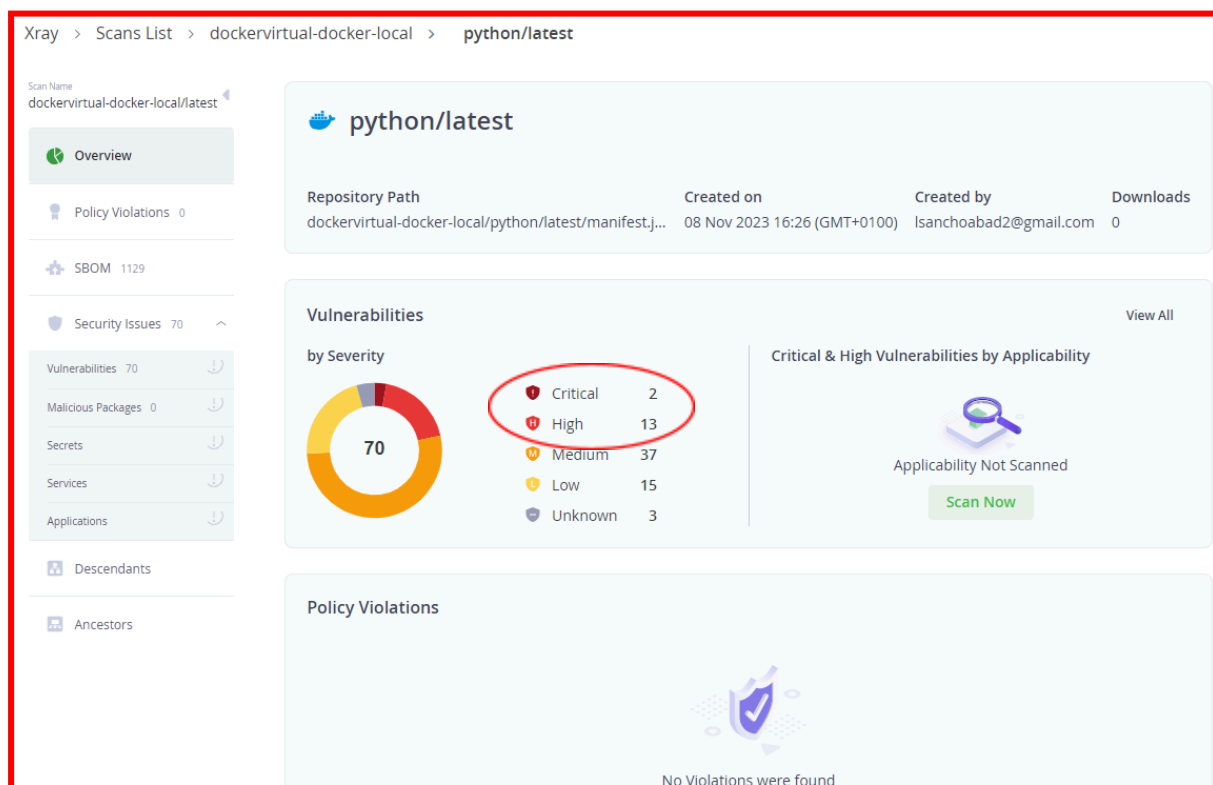
Para la llamada manual a *Curve.IsOnCurve()*, agregue la siguiente verificación antes de la llamada a la función:

```
if x.Sign() < 0 || x.Cmp(curve.P) >= 0 ||  
    y.Sign() < 0 || y.Cmp(curve.P) >= 0 {  
    return nil, fmt.Errorf("Point is not on the curve")  
}
```



- Python

Python cuenta con una menor lista de vulnerabilidades que mariadb, con un total de 70 vulnerabilidades, 2 vulnerabilidades de nivel crítico y 13 vulnerabilidades de nivel alto.





Vamos a centrarnos en las dos de nivel crítico y alguna de nivel alto:




- CVE-2023-45853

Se ha encontrado que hay un desbordamiento del búfer de montón en zlib que puede provocar la ejecución remota de código al analizar un archivo malicioso

 **CVE-2023-45853** 

JFrog research last updated on 31 Dec 0 11:45 PM
This CVE is enriched by JFrog research and provides more accurate information


Xray ID

XRAY-533715 

Contextual Analysis


Not Scanned

JFrog Severity

 High

Components (1)

Name

 debian:bookworm:zlib1g:1:1.2.13.dfsg-1

Version

1:1.2.13.dfsg-1

Upgrade to

None

CVSS Score

9.8 (v3)

CWE

CWE-190

^ Show Less

JFrog Research

Public Sources

Impact Paths

References

Summary

A heap buffer overflow in zlib may lead to remote code execution when parsing a malicious archive.

Remedio: Asegúrese de que los archivos con nombres de más de 65536 caracteres no se analicen con zlib. Además, actualmente existe una solución en la rama de desarrollo de zlib y se puede implementar manualmente.



- CVE-2023-28531

ssh-add en OpenSSH anterior a 9.3 agrega claves de tarjeta inteligente a ssh-agent sin las restricciones de destino por salto previstas

The screenshot shows the Xray tool interface for CVE-2023-28531. It includes fields for Xray ID (XRAY-427962), Contextual Analysis (Not Scanned), Components (1), Name (debian:bookworm:openssh-client:1:9.2p1-2+deb12u1), Version (1:9.2p1-2+deb12u1), Upgrade to (None), CVSS Score (9.8 (v3)), and CWE (NVD-CWE-noinfo). A 'Show Less' button is visible. Below the details, there are tabs for 'Public Sources', 'Impact Paths', and 'References'. The 'Summary' tab is active, showing a description of the vulnerability: 'ssh-add in OpenSSH before 9.3 adds smartcard keys to ssh-agent without the intended per-hop destination constraints. The earliest affected version is 8.9.'

Remedio: afecta a las versiones de OpenSSH entre la 8.9 y la 9.2 ambas incluidas y se puede solucionar actualizando a la versión 9.3

- CVE-2023-39417

Este CVE de nivel alto, indica que en el script de extensión, se encontró una vulnerabilidad de inyección SQL en PostgreSQL si usa @extowner@, @extschema@ o @extschema:...@ dentro de una construcción de cotización (cotización de dólares, " o "). Si un administrador ha instalado archivos de una extensión vulnerable, confiable y no incluida, un atacante con privilegio CREATE a nivel de base de datos puede ejecutar código arbitrario como superusuario de arranque.

The screenshot shows the Xray tool interface for CVE-2023-39417. It includes fields for Xray ID (XRAY-527710), Contextual Analysis (Not Scanned), Components (1), Name (debian:bookworm:libpq5:15.3-0+deb12u1), Version (15.3-0+deb12u1), Upgrade to (None), CVSS Score (8.8 (v3)), and CWE (CWE-89). A 'Show Less' button is visible. Below the details, there are tabs for 'Public Sources', 'Impact Paths', and 'References'. The 'Summary' tab is active, showing a description of the vulnerability: 'IN THE EXTENSION SCRIPT, a SQL Injection vulnerability was found in PostgreSQL if it uses @extowner@, @extschema@, or @extschema:...@ inside a quoting construct (dollar quoting, ", or "). If an administrator has installed files of a vulnerable, trusted, non-bundled extension, an attacker with database-level CREATE privilege can execute arbitrary code as the bootstrap superuser.'



- CVE-2023-27103

Esta vulnerabilidad es sobre el descubrimiento de que Libde265 v1.0.11 contiene un desbordamiento de búfer en el montón a través de la función `derive_collocated_motion_vectors` en `motion.cc`.

CVE-2023-27103

Xray ID

XRAY-427848

Contextual Analysis

Not Scanned

Components (1)

Name

debian:bookworm:libde265-0:1.0.11-1

Version

1.0.11-1

Upgrade to

None

CVSS Score

8.8 (v3)

CWE

CWE-787

^ Show Less

Public Sources

Impact Paths

References

Summary

Libde265 v1.0.11 was discovered to contain a heap buffer overflow via the function `derive_collocated_motion_vectors` at `motion.cc`.



- HTTPD

En httpd se han encontrado menos vulnerabilidades, con un total de 16 vulnerabilidades, 1 vulnerabilidad de nivel critico y 2 vulnerabilidades de nivel alto

Xray > Scans List > dockervirtual-docker-local > httpd/latest

Scan Name
dockervirtual-docker-local/latest

Overview

Policy Violations 0

SBOM 329

Security Issues 16

Vulnerabilities 16

Malicious Packages 0

Secrets

Services

Applications

Descendants

Ancestors

Repository Path
dockervirtual-docker-local/httpd/latest/manifest.js...

Created on
08 Nov 2023 16:23 (GMT+0100)

Created by
lsanchoabad2@gmail.com

Downloads
0

Vulnerabilities

by Severity

16

Critical 1

High 2

Medium 4

Low 7

Unknown 2

Critical & High Vulnerabilities by Applicability

Applicability Not Scanned

Scan Now



Policy Violations



No Violations were found



- CVE-2023-45853




Es la misma vulnerabilidad que la encontrada en Python. Se ha detectado un desbordamiento del búfer de montón en zlib que puede provocar la ejecución remota de código al analizar un archivo malicioso.


 **CVE-2023-45853** 

JFrog research last updated on 31 Dec 0 11:45 PM

This CVE is enriched by JFrog research and provides more accurate information

Xray ID	XRAY-533715 
Contextual Analysis	Not Scanned
JFrog Severity	 High
Components (1)	
Name	 debian:bookworm:zlib1g:1:1.2.13.dfsg-1
Version	1:1.2.13.dfsg-1
Upgrade to	None
CVSS Score	<div>9.8 (v3)</div>
CWE	CWE-190


 Show Less

JFrog Research

Public Sources

Impact Paths

References

Summary 

A heap buffer overflow in zlib may lead to remote code execution when parsing a malicious archive.

Remedio: Asegúrese de que los archivos con nombres de más de 65536 caracteres no se analicen con zlib. Además, actualmente existe una solución en la rama de desarrollo de zlib y se puede implementar manualmente.

- **CVE-2023-2953**

Se encontró una vulnerabilidad en openldap. Este fallo de seguridad provoca una desreferencia del puntero nulo en la función *ber_memalloc_x()*.

H

CVE-2023-2953

Xray IDXRAY-520865

Contextual AnalysisNot Scanned

Components (2)

Name

debian:bookworm:libldap-2.5-0

Version

2.5.13+dfsg-5

Upgrade to

None

CVSS Score

7.5 (v3)

CWE

CWE-476

Show Less

Public Sources

Impact Paths

References

Summary

A vulnerability was found in openldap. This security flaw causes a null pointer dereference in ber_memalloc_x() function.



- CVE-2023-31484

La falta de verificación de TLS en CPAN.pm permite ataques de intermediario al descargar paquetes y puede provocar la ejecución de código.

CVE-2023-31484

JFrog research last updated on 31 Dec 0 11:45 PM
This CVE is enriched by JFrog research and provides more accurate information

Xray ID	XRAY-515823
Contextual Analysis	Not Scanned
JFrog Severity	High
Components (1)	
Name	debian:bookworm:perl-base:5.36.0-7
Version	5.36.0-7
Upgrade to	None
CVSS Score	8.1 (v3)
CWE	CWE-295

Show Less

JFrog ResearchPublic SourcesImpact PathsReferences

Summary

Missing TLS check in CPAN.pm allows man-in-the-middle attacks when downloading packages and may lead to code execution.



- Nginx

Se han encontrado un total de 33 vulnerabilidades en la imagen, con 1 de nivel crítico y 5 de nivel alto

Xray > Scans List > dockervirtual-docker-local > nginx/latest

Scan Name
dockervirtual-docker-local/latest

Overview

Policy Violations 0

SBOM 351

Security Issues 33

Vulnerabilities 33

Malicious Packages 0

Secrets

Services

Applications

Descendants

Ancestors

nginx/latest

Repository Path
dockervirtual-docker-local/nginx/latest/manifest.js...

Created on
08 Nov 2023 16:21 (GMT+0100)

Created by
lsanchoabad2@gmail.com

Downloads
0

Vulnerabilities [View All](#)

by Severity

Critical	1
High	5
Medium	16
Low	9
Unknown	2

Critical & High Vulnerabilities by Applicability

APPLICABLE	0
UNDETERMINED	4
NOT APPLICABLE	2



Policy Violations

No Violations were found




- CVE-2023-45853

Es la misma vulnerabilidad que la encontrada en Python. Se ha detectado un desbordamiento del búfer de montón en zlib que puede provocar la ejecución remota de código al analizar un archivo malicioso.

 **CVE-2023-45853** 

JFrog research last updated on 31 Dec 0 11:45 PM
This CVE is enriched by JFrog research and provides more accurate information


Xray ID

XRAY-533715 

Contextual Analysis


Not Scanned

JFrog Severity

 High

Components (1)

Name

 debian:bookworm:zlib1g:1:1.2.13.dfsg-1


Version

1:1.2.13.dfsg-1

Upgrade to

None

CVSS Score

 9.8 (v3)

CWE

CWE-190

^ Show Less


JFrog Research

Public Sources

Impact Paths

References

Summary




A heap buffer overflow in zlib may lead to remote code execution when parsing a malicious archive.

Asegúrese de que los archivos con nombres de más de 65536 caracteres no se analicen con zlib. Además, actualmente existe una solución en la rama de desarrollo de zlib y se puede implementar manualmente.



- CVE-2023-27103

Se descubrió que Libde265 v1.0.11 contiene un desbordamiento del búfer de montón mediante la función *derive_collocated_motion_vectors* en *motion.cc*


 **CVE-2023-27103**

Xray IDXRAY-427848

Contextual AnalysisUNDETERMINED

Components (1)

Name

 debian:bookworm:libde265-0:1.0.11-1

Version

1.0.11-1

Upgrade to

None

CVSS Score

8.8 (v3)

CWE

CWE-787

^ Show Less

Public Sources

Impact Paths

References



Summary

Libde265 v1.0.11 was discovered to contain a heap buffer overflow via the function *derive_collocated_motion_vectors* at *motion.cc*.




- **CVE-2023-31484**

Esta vulnerabilidad lo que indica es que hay una falta de verificación de TLS en CPAN.pm permite ataques de intermediarios al descargar paquetes y podría dar lugar a la ejecución de código.

 **CVE-2023-31484** 

JFrog research last updated on 31 Dec 0 11:45 PM
This CVE is enriched by JFrog research and provides more accurate information


Xray ID

XRAY-515823 

Contextual Analysis


Not Scanned

JFrog Severity

 High

Components (1)

Name

 debian:bookworm:perl-base:5.36.0-7

Version

5.36.0-7

Upgrade to

None

CVSS Score

8.1 (v3)

CWE

CWE-295

^ Show Less

JFrog Research

Public Sources

Impact Paths

References

Summary

Missing TLS check in CPAN.pm allows man-in-the-middle attacks when downloading packages and may lead to code execution.



- **CVE-2023-39616**

La desreferencia de puntero no válida en libaom conduce a la denegación de servicio al codificar datos de vídeo creados con opciones de configuración no predeterminadas establecidas

CVE-2023-39616

JFrog research last updated on 31 Dec 0 11:45 PM
This CVE is enriched by JFrog research and provides more accurate information

Xray ID

XRAY-529506

Contextual Analysis

UNDETERMINED

JFrog Severity

Low

Components (1)

Name

debian:bookworm:libaom3:3.6.0-1

Version

3.6.0-1

Upgrade to

None

CVSS Score

7.5 (v3)

CWE

CWE-119

^ Show Less

JFrog Research

Public Sources

Impact Paths

References

Summary

Invalid pointer dereference in libaom leads to denial of service when encoding crafted video data with nondefault configuration options set



- Php

Php cuenta con un total de 18 vulnerabilidades, siendo 1 de ellas de nivel critico y 2 de ellas de nivel alto

Xray > Scans List > dockervirtual-docker-local > php/latest

Scan Name
dockervirtual-docker-local/latest

Overview

Policy Violations 0

SBOM 562

Security Issues 18

Vulnerabilities 18

Malicious Packages 0

Secrets

Services

Applications

Descendants

Ancestors

php/latest

Repository Path
dockervirtual-docker-local/php/latest/manifest.json

Created on
08 Nov 2023 16:17 (GMT+0100)

Created by
lsanchoabad2@gmail.com

Downloads
0

Vulnerabilities

by Severity

18

Critical	1
High	2
Medium	4
Low	9
Unknown	2

Critical & High Vulnerabilities by Applicability

Applicability Not Scanned

Scan Now



Policy Violations



No Violations were found



- CVE-2023-45853

Es la misma vulnerabilidad que la encontrada en Python. Se ha detectado un desbordamiento del búfer de montón en zlib que puede provocar la ejecución remota de código al analizar un archivo malicioso.


 **CVE-2023-45853** 

JFrog research last updated on 31 Dec 0 11:45 PM

This CVE is enriched by JFrog research and provides more accurate information


Xray ID

XRAY-533715 

Contextual Analysis


Not Scanned

JFrog Severity

 High

Components (1)

Name

 debian:bookworm:zlib1g:1:1.2.13.dfsg-1

Version

1:1.2.13.dfsg-1

Upgrade to

None

CVSS Score

9.8 (v3)

CWE

CWE-190


^ Show Less

JFrog Research

Public Sources

Impact Paths

References

Summary 

A heap buffer overflow in zlib may lead to remote code execution when parsing a malicious archive.

Asegúrese de que los archivos con nombres de más de 65536 caracteres no se analicen con zlib. Además, actualmente existe una solución en la rama de desarrollo de zlib y se puede implementar manualmente



- CVE-2023-2953

Es la misma vulnerabilidad encontrada en httpd. Se encontró una vulnerabilidad en openldap. Este fallo de seguridad provoca una desreferencia del puntero nulo en la función *ber_memalloc_x()*.

CVE-2023-2953

Xray ID

XRAY-520865

Contextual Analysis

Not Scanned

Components (2)

Name

debian:bookworm:libldap-2.5-0

Version

2.5.13+dfsg-5

Upgrade to

None

CVSS Score

7.5 (v3)

CWE

CWE-476

Show Less

Public Sources

Impact Paths

References



Summary



A vulnerability was found in openldap. This security flaw causes a null pointer dereference in ber_memalloc_x() function.



- **CVE-2023-31484**

Es la misma vulnerabilidad encontrada en httpd. La falta de verificación de TLS en CPAN.pm permite ataques de intermediario al descargar paquetes y puede provocar la ejecución de código.


 **CVE-2023-31484** 

JFrog research last updated on 31 Dec 0 11:45 PM

This CVE is enriched by JFrog research and provides more accurate information


Xray ID

XRAY-515823 

Contextual Analysis


Not Scanned

JFrog Severity

 High

Components (1)

Name

 debian:bookworm:perl-base:5.36.0-7


Version

5.36.0-7

Upgrade to


None

CVSS Score

 8.1 (v3)

CWE

CWE-295


 Show Less

JFrog Research

Public Sources

Impact Paths

References

Summary 

Missing TLS check in CPAN.pm allows man-in-the-middle attacks when downloading packages and may lead to code execution.



5. Alcance Económico

El alcance económico de una empresa se suele referir a la extensión y a la magnitud de las actividades económicas en las que la empresa está involucrada , así como su impacto en el entorno económico.

La gestión de las vulnerabilidades identificadas en este informe puede requerir inversiones significativas en términos de tiempo y recursos. Los costos asociados pueden incluir:

1. Actualización y parcheo de las bibliotecas y dependencias afectadas.
2. Mejoras en las configuraciones de jFrog Xray para prevenir futuras vulnerabilidades.
3. Capacitación del personal en prácticas de seguridad y gestión de vulnerabilidades.

Se recomienda que "La Rodilla" designe un presupuesto y un equipo de seguridad de la información para abordar de manera efectiva las vulnerabilidades detectadas y mitigar los riesgos.

Hemos tardado en realizar este proyecto unas 3 horas aproximadamente , que afecta a nuestro salario , debido a que tenemos un tiempo estimado para realizar este proyecto.

En todo proyecto existen muchas restricciones, pero hay tres que se suelen considerar muy importante y suelen ser las más comunes para cualquier proyecto; Estos son: el costo , el tiempo y el alcance y conforman lo que se suele denominar , la triple restricción de un proyecto.

La restricción de coste , se suele referir a la cantidad presupuestada necesaria para alcanzar los objetivos del proyecto.

La restricción del tiempo , se suele referir normalmente a la cantidad del tiempo que disponemos para la realización de dicho proyecto.

La restricción de alcance , se refiere a lo que se debe hacer para producir el resultado final del proyecto

El por qué escogemos estas tres variables , el costo, el tiempo, y el alcance, está en relación a que todas las decisiones tomadas con respecto a las áreas relacionadas con cualquiera de estos tres aspectos al final se suelen traducir en que terminan afectando o modificando a alguna de las otras dos variables.



La herramienta utilizada por nuestra empresa es jfrog que nos sirve para ver las vulnerabilidades; Esta herramienta es una herramienta privada que es de pago , debido a que la herramienta utilizada es de pago, se añadiría al costo del proyecto . El trabajo está valorado en 2.000 euros.