

# ARIMA

100% SOFTWARE DESIGN

**06 JUNIO, 2024**

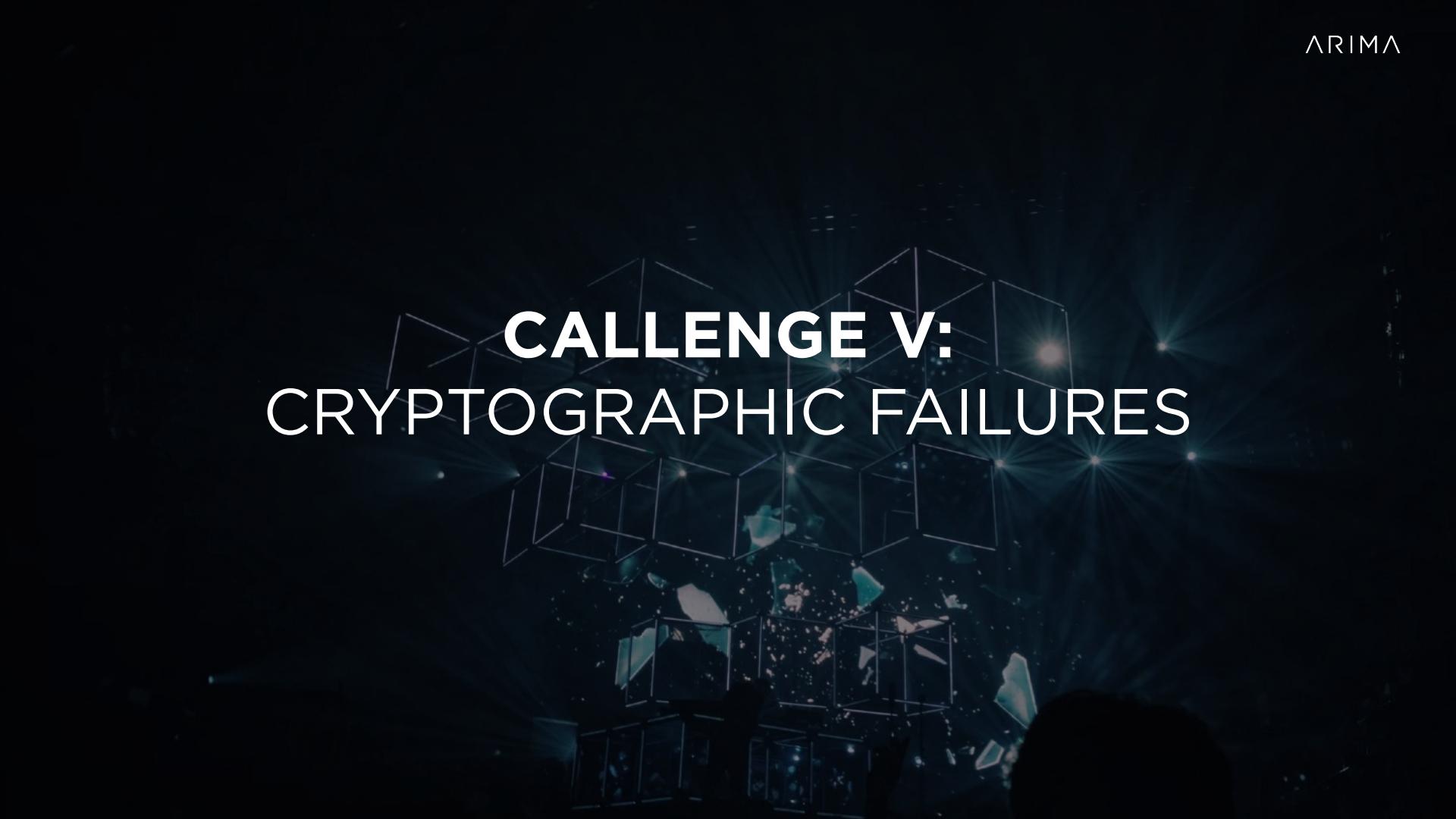
DESARROLLO  
SEGURO WEB



SSID: CR  
PWD: OngietorriTknikara22

JUICE SHOP URL  
[http://192.168.214.\[202-230\]](http://192.168.214.202-230)

# CHALLENGE V: CRYPTOGRAPHIC FAILURES

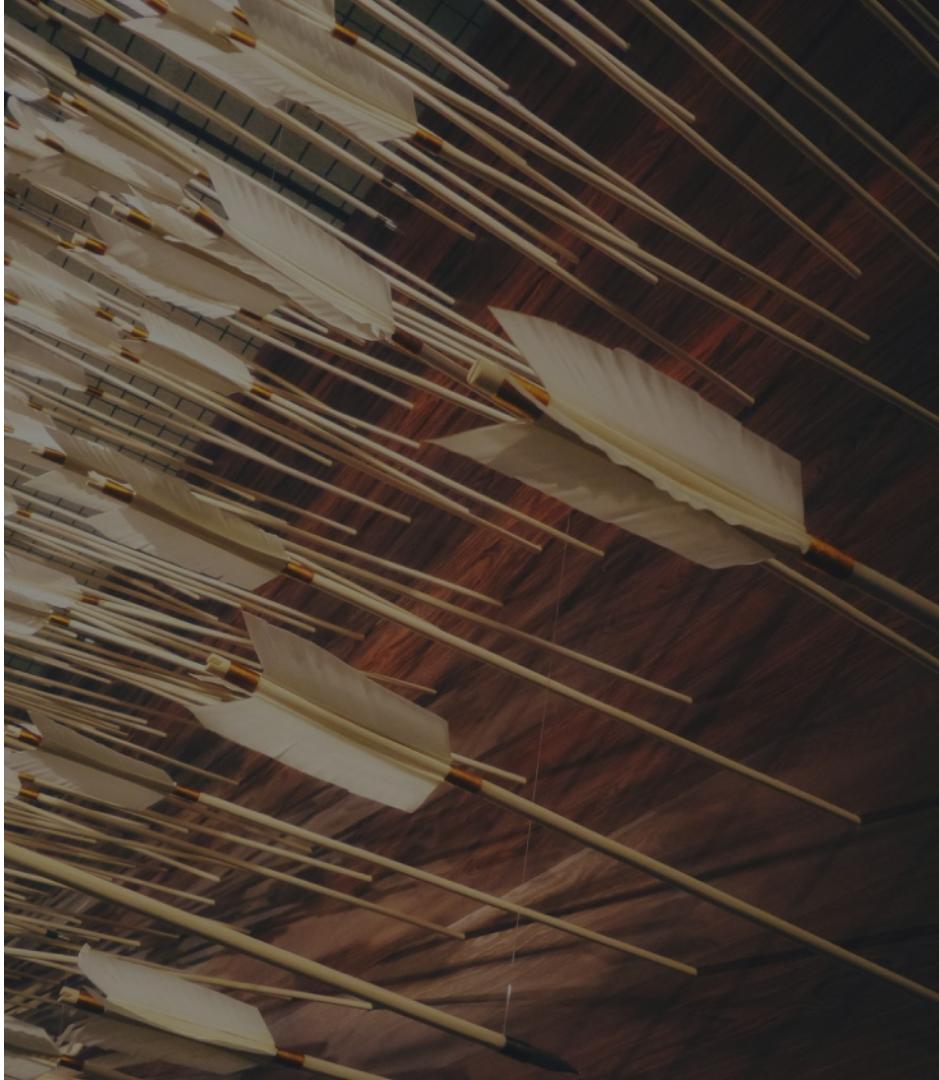


# A02:2021 CRYPTOGRAPHIC FAILURES

Antes era conocido como “Sensitive Data Exposure”, ahora lo han agrupado en esta nueva categoría.

Entran el uso de password hardcodeados o por defecto, algoritmos criptográficos rotos o entropía insuficiente.

Esto puede llevar a la pérdida de datos sensibles, suplantación de identidad, etc.



LET'S HACK!



# Null Byte Injection

# NULL BYTE INJECTION

Esta técnica consiste en incluir un byte nulo (0x00) a un input para manipular la forma en la que se comporta la aplicación.

Por ejemplo supongamos que un atacante quiere subir al servidor un archivo malicious.php pero que solo se permite subir PDFs.

Podemos añadir el byte null al final del nombre y a continuación, la extensión PDF. De esta forma, saltaremos la validación pero al leer el archivo, el sistema operativo ignorará el byte nulo y todo lo que viene a continuación.

malicious.php **%00.pdf**

LET'S TRY IT!



No funciona porque tenemos  
que codificar la URL:

**package.json.bak%2500.pdf**

# ALGUNOS CASOS POPULARES

# ADULT FRIEND FINDER

En 2016 fueron expuestos 412 millones de cuentas de la web de citas adult friend finder.

Se filtraron emails y contraseñas que estaban o bien en texto plano, o hasheadas con SHA-1 que es un algoritmo inseguro.

## Up to 400 million accounts in Adult Friend Finder breach

⌚ 14 November 2016



The screenshot shows the homepage of AdultFriendFinder. At the top, there's a navigation bar with links for 'Join Now!', 'Home', 'Browse', 'Hookup', 'Dating Forums', and 'Live Chat'. On the right side of the header, there are fields for 'Username' and 'password' with a 'Forgot password?' link and a 'Login' button. Below the header, there's a large banner featuring a woman with blonde hair. To the right of the banner, a red call-to-action button says 'Sign Up Now! Start Hooking Up Tonight!'. A registration form is overlaid on the banner, asking for 'I am/We are a:' (with 'Man' selected), 'Interested in meeting:' (with 'Men' and 'Women' checked), 'My birthdate:', 'Country:' (set to 'United Kingdom'), and 'County:' (set to 'Any'). A 'Register Now' button is at the bottom of the form. At the very bottom of the page, a banner reads 'Join the World's Largest Sex & Swinger Community'.

¿CÓMO NOS PODEMOS  
PROTEGER?



Revisar la configuración por defecto de los frameworks y productos que utilizamos.

Revisar los algoritmos criptográficos que utilizamos y aplicar el más apropiado para cada caso.

[https://cheatsheetseries.owasp.org/cheatsheets/Password\\_Storage\\_Cheat\\_Sheet.html#password-storage-cheat-sheet](https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html#password-storage-cheat-sheet)

# INSECURE DESIGN



# A04:2021 INSECURE DESIGN

Es una nueva categoría de 2021 dirigida a los riesgos asociados al diseño de soluciones software que no incluyen la seguridad como parte del diseño desde el principio.

Dentro de esta categoría entrarían aspectos como:

- / No controlar errores y dejar que se muestren trazas de error con información sensible.
- / Almacenamiento de credenciales inadecuado.
- / Etc.

## Server Error in '/IDI' Application.

### Error al descargar el documento

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.ArgumentException: Error al descargar el documento

Source Error:

An unhandled exception was generated during the execution of the current web request. Information regarding the origin and location of the exception can be identified using the exception stack trace below.

Stack Trace:

```
[ArgumentException: Error al descargar el documento]
IDI.ObtenerDocumento.Page_Load(Object sender, EventArgs e) +2408
System.Web.UI.Control.LoadRecursive() +71
System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint) +3178
```

Version Information: Microsoft .NET Framework Version:4.0.30319; ASP.NET Version:4.0.30319.34274

¿CÓMO NOS PODEMOS  
PROTEGER?

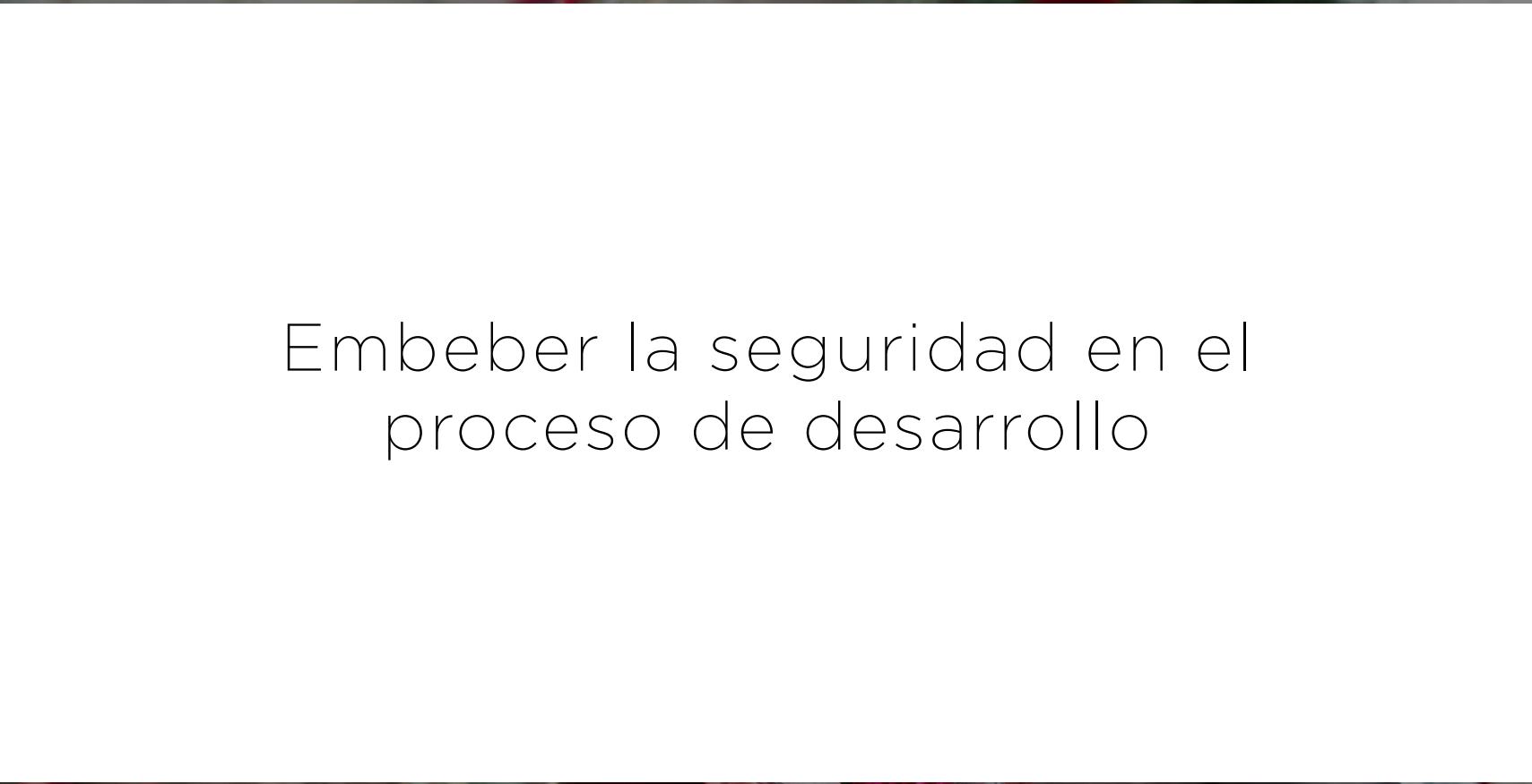




Seguir las buenas prácticas establecidas en la industria.



Establecer en las  
organizaciones arquitecturas de  
referencia y fijar procesos que  
garanticen su cumplimiento.



Embeber la seguridad en el  
proceso de desarrollo

# CHALLENGE VII: SECURITY MISCONFIGURATION



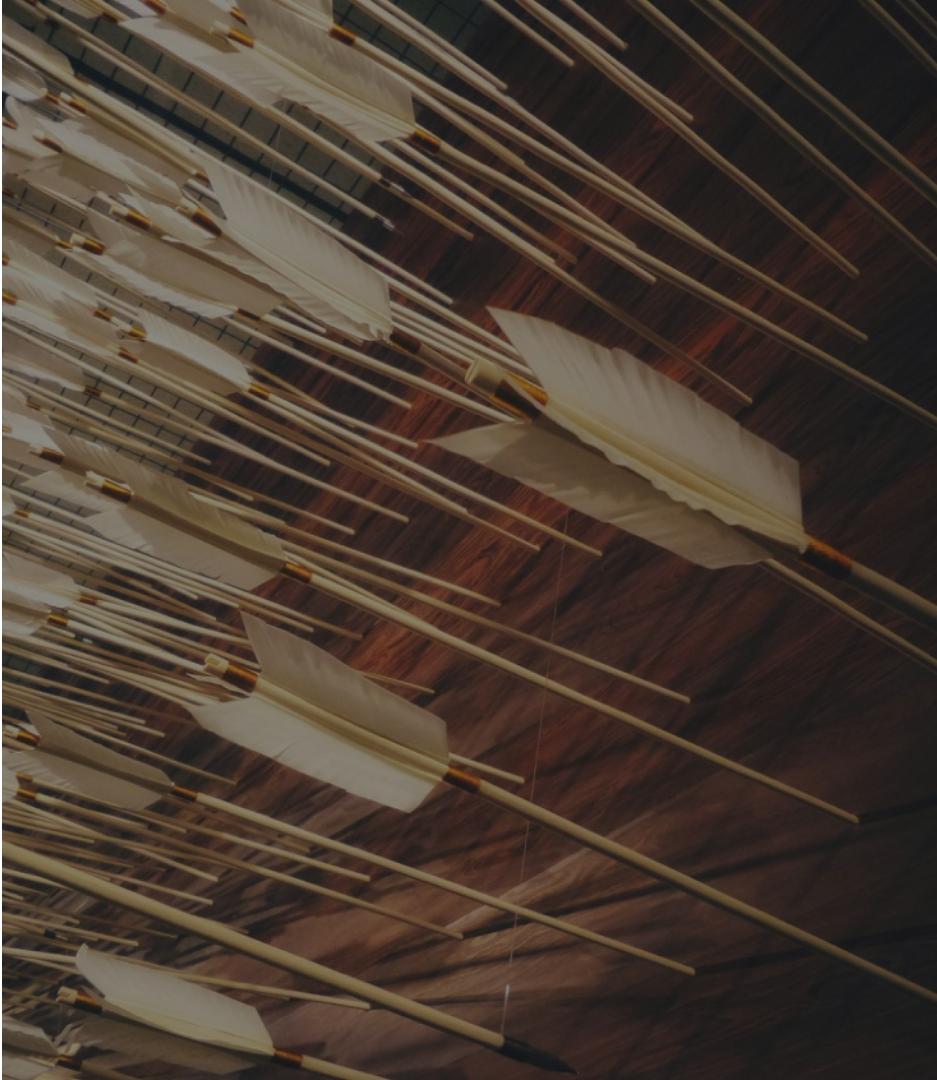
# A05:2021 SECURITY MISCONFIGURATION

Cada vez utilizamos más software y productos de terceros y tenemos que ser cuidadosos con las configuraciones de estos.

Esto tiene especial importancia en los nuevos productos cloud que utilizamos.

Se utilizan configuraciones por defecto de frameworks o herramientas que no son las más seguras.

También son parte de esta categoría el no eliminar el código obsoleto, dejar abiertos puertos o endpoints que no son necesarios,...



LET'S HACK!



# XXE

Tradicionalmente se ha utilizado el formato XML muy frecuentemente para intercambiar información entre sistemas.

Los XML tienen una característica que permite definir entidades en archivos externos.

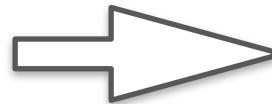
De esta forma, se lee el archivo externo y se inyecta su contenido en la entidad del XML.

A dark background with colorful, blurred text representing XML code. The text includes various XML tags like <array>, <?php>, <a href=>, <h3>, and <p>. The colors used for the text are cyan, magenta, yellow, and green, creating a rainbow effect against the black background.

# USO LEGÍTIMO

A continuación se muestra un ejemplo en el que se hace uso de entidades externas de manera legítima:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo [
  <!ENTITY info SYSTEM "file:///path/to/secure/directory/info.txt">
]>
<document>
  <title>Información Importante</title>
  <content>&info;</content>
</document>
```



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo [
  <!ENTITY info SYSTEM "file:///path/to/secure/directory/info.txt">
]>
<document>
  <title>Información Importante</title>
  <content>Nuestro nuevo producto, el EcoBottle, ofrece una solución sostenible
</document>
```

Nuestro nuevo producto, el EcoBottle, ofrece una solución sostenible

LET'S HACK!



# VULNERABILIDAD

A continuación se muestra un ejemplo en el que se vale de esta característica para acceder a información confidencial.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
  <!ENTITY xxe SYSTEM "file:///etc/passwd">
]>
<foo>&xxe;</foo>
```



```
<foo>
root:x:0:0:root:/root:/bin/bash
johndoe:x:1000:1000:John Doe:/home/johndoe:/bin/bash
janedoe:x:1001:1001:Jane Doe:/home/janedoe:/bin/bash
guest:x:1002:1002:Guest Account:/home/guest:/usr/sbin/nologin
</foo>
```

```
root:x:0:0:root:/root:/bin/bash
johndoe:x:1000:1000:John Doe:/home/johndoe:/bin/bash
janedoe:x:1001:1001:Jane Doe:/home/janedoe:/bin/bash
guest:x:1002:1002:Guest Account:/home/guest:/usr/sbin/nologin
```

# ALGUNOS CASOS POPULARES

# VERIZON

En 2017, un bucket de S3 mal configurado por una empresa subcontratada (NICE Systems) provocó la exposición de los datos de 14 millones de usuarios de Verizon.

Los datos filtrados incluían información sensible como nombres de clientes, números de teléfono y PINs.



## AWS S3 bucket leak exposes millions of Verizon customers' data

News roundup: An AWS S3 bucket leak containing personal data of millions of Verizon customers was exposed to the public. Plus, DNC hack victims are suing the Trump campaign, and more.

By [Madelyn Bacon](#), TechTarget

Published: 14 Jul 2017

The personal data of millions of Verizon customers was exposed because of a misconfigured [Amazon Web Services S3 bucket leak](#).

A researcher at security firm UpGuard reportedly discovered a repository containing the names, addresses, account details and account PINs of 14 million Verizon customers in the U.S. The AWS S3 bucket is owned and run by Nice Systems, a third-party vendor based in Israel that Verizon uses to handle its back-office and call center operations.

# UBER

En 2014, un atacante pudo acceder a un bucket de S3 en el que había datos de usuarios almacenados en texto plano.

El atacante pudo acceder a esta información gracias a que un ingeniero de Uber había publicado una API Key en un repositorio público de Github.

La información a la el atacante pudo acceder contenía información sensible de unos 100.000 usuarios: nombres, carnets de conducir, cuentas corrientes, etc.

## Developers keep leaving secret keys to corporate data out in the open for anyone to take

We've found 7,448 code results

Sort: Best matches

repositories

- de [1]
- us [35]
- ers [2]

iges

- Notebook
- ipt

vn

- 368
- 163
- 153
- 145
- 132
- 125
- 101
- 100
- 71
- 59

slack: [REDACTED] Showing the top match. Last indexed 4 days ago.  
1 slack:  
2 api\_token: "xoxp-????????????????????????????????????"

[REDACTED] Showing the top match. Last indexed on Mar 28.  
1 xoxp-[REDACTED]

[REDACTED] Showing the top match. Last indexed on Mar 28.  
1 SLACK\_API\_TOKEN="xoxp-hogehoghe"

[REDACTED] Showing the top match. Last indexed on Mar 29.  
1 {  
2 "SLACK\_TOKEN": "xoxp-[REDACTED]"  
3 }

A code search on GitHub

Image: GitHub

By Keith Collins Published May 4, 2016



The hackers who stole data on 50,000 Uber drivers in 2014 didn't have to do much hacking at all. They got into the company's database using login credentials they'd found on GitHub, the code-sharing website used by more than 14 million developers. An Uber employee had uploaded the credentials to GitHub by accident, and left them on a public page for months.

# CAPITAL ONE

En 2019 una ex empleada de AWS se aprovechó de que un firewall no estaba bien configurado para acceder a buckets S3 de la compañía.

Esta brecha de seguridad afectó a unos 100 millones de americanos y 6 millones de canadienses.

Se filtraron datos sensibles incluidos números de la seguridad social y números de cuentas corrientes.

## Capital One Attacker Exploited Misconfigured AWS Databases

After bragging in underground forums, the woman who stole 100 million credit applications from Capital One has been found guilty.



Tara Seals, Managing Editor, News, Dark Reading  
June 20, 2022

4 Min Read



SOURCE: PONGPHAN RUENGCHAI VIA ALAMY STOCK PHOTO



The 36-year-old Seattle tech worker behind the infamous 2019 Capital One data breach has

¿CÓMO NOS PODEMOS  
PROTEGER?



# ENTORNOS REPRODUCIBLES

Tener entornos reproducibles e idénticos ayuda a tener mejor controladas las distintas piezas que componen la infraestructura y la configuración a aplicar a cada una de ellas.

Hacer uso de plataformas mínimas que solo incluyan las características esenciales.

Realizar revisiones periódicas de todas las configuraciones aplicadas.

Utilizar herramientas de revisión de código que ayudan a identificar secretos en el código.

Utilizar cabeceras de seguridad como CSP o CORS lo más restrictivas posibles.



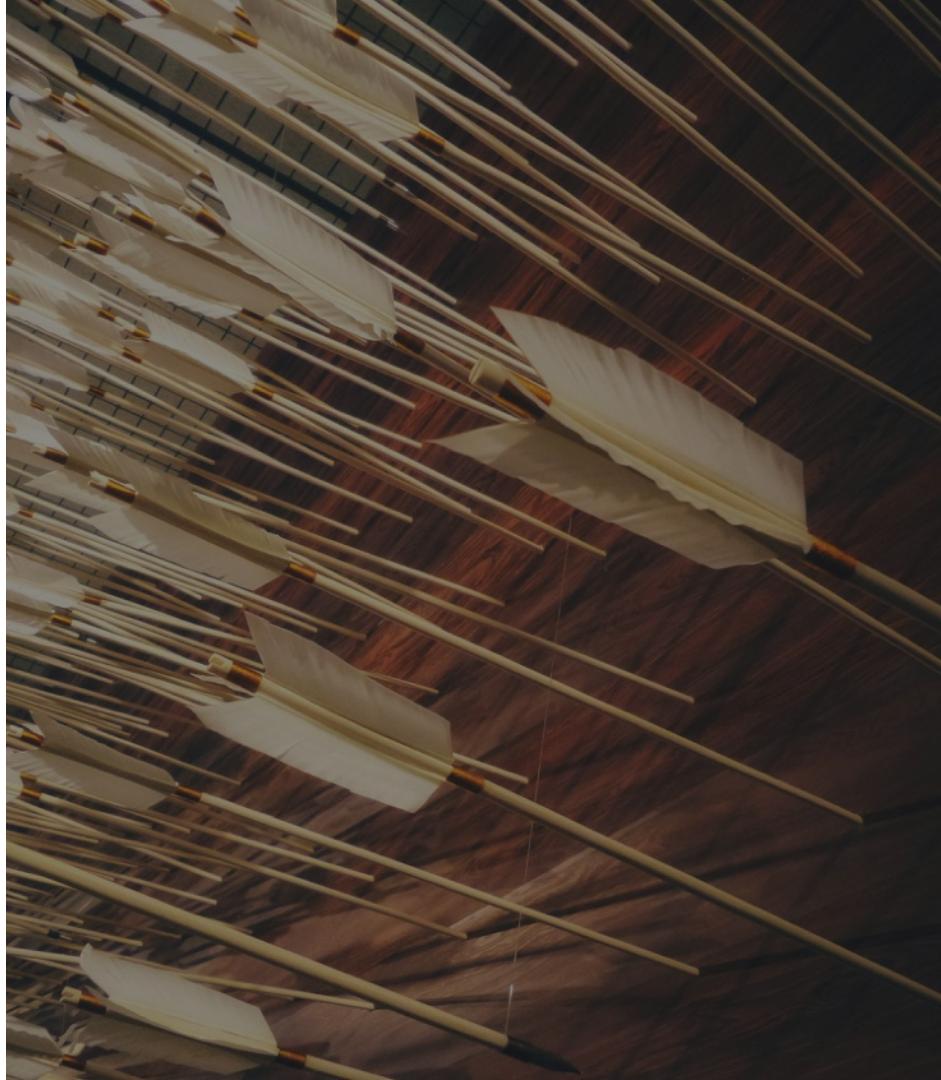
# CHALLENGE VII:

## VULNERABLE AND OUTDATED COMPONENTS

# A06:2021 VULNERABLE AND OUTDATED COMPONENTS

Esto ocurre cuando no se tiene un control sobre las librerías que utilizan nuestras aplicaciones directamente o de manera transitiva.

Cuando no se monitoriza si aparecen vulnerabilidades conocidas para alguna de estas librerías o si han dejado de estar soportadas.



LET'S HACK!



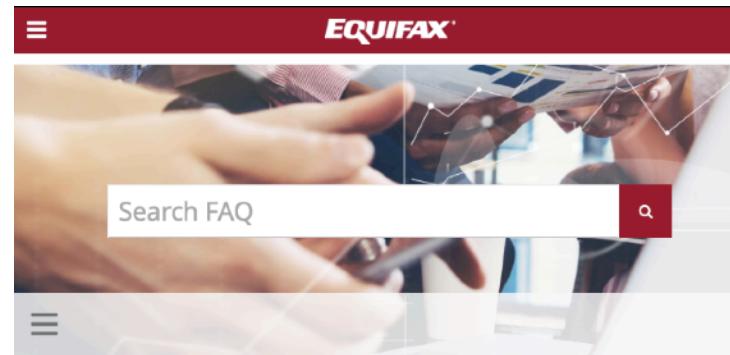
# ALGUNOS CASOS POPULARES

# EQUIFAX

En 2017 Equifax, una de las mayores agencias de información crediticia del mundo, sufrió la fuga de información personal de unos 147 millones de americanos.

La fuga de información se dio aprovechando una vulnerabilidad en el framework web Apache Struts que permitía la ejecución de código remoto.

Esta vulnerabilidad había sido reportada y corregida meses antes ([CVE-2017-5638](#)) pero Equifax no había aplicado el parche.



## Cybersecurity Incident & Important Consumer Information

On September 7, 2017 Equifax Inc., our U.S. parent company, announced a cybersecurity incident. On October 2, 2017 we announced we completed our investigation into the incident. With respect to potentially impacted Canadian citizens, we shared in a news release on October 2 and with customers the same week that personal information of approximately 8,000 Canadian consumers was impacted. Also, it was determined that an additional 11,670 credit card numbers of Canadian consumers may have been impacted.

Equifax Inc. learned of the incident on July 29, 2017, and acted immediately to stop the intrusion and conduct a forensic review. During that review, we identified unauthorized access to limited personal information for certain Canadian consumers. Based on our investigation, the unauthorized access occurred from mid-May through July 2017.

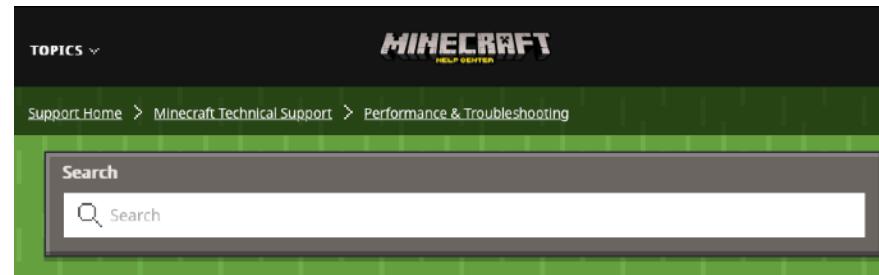
The potentially impacted information includes names, addresses, Social Insurance Numbers and credit card numbers. Other potentially impacted information includes login credentials for our direct-to-consumer website, such as username and password, and secret question/answer, which we believe are several years old.

# LOG4J2

En 2021 se hizo pública una vulnerabilidad crítica en la librería Java Log4j2.

Esto tuvo un gran impacto en la comunidad Java ya que esta es una librería de Logging que era utilizada en la mayoría de las aplicaciones desarrolladas en este lenguaje de programación.

Esta vulnerabilidad afectaba a todas las versiones anteriores a la 2.17.0.



*This article only applies to Minecraft: Java Edition.*

We have identified a vulnerability in the form of an exploit within Log4j – a common Java logging library.

This exploit affects many services – including Minecraft: Java Edition.

This vulnerability poses a potential risk of your computer being compromised, and while this exploit has been addressed thanks to a recent patch to the game client, you still need to take the following steps to secure your game and your servers.

## WHAT YOU NEED TO DO:

These steps vary based on how you're interacting with the game. Please follow the steps most relevant to your situation.

## OFFICIAL GAME CLIENT

If you play Minecraft: Java Edition but aren't hosting your own server, you will need to take the following steps: Close all running instances of the game and the Minecraft Launcher. Start the Launcher again – the patched version will download automatically.

# LOG4J2

Dos años después de que se hiciera pública la vulnerabilidad, Veracode publicó un informe en el que se indicaba que 2 de cada 5 aplicaciones todavía hacían uso de versiones de Apache Log4j vulnerables.

Una de cada tres aplicaciones analizadas utilizaban Log4j 1.2.x que no se mantiene desde 2015.

Un 3.8% de las aplicaciones utilizaba la versión 2.17.0 que contiene el parche pero que es vulnerable a [CVE-2021-44832](#), otra vulnerabilidad grave que permite la ejecución de código remoto.

## DIVE BRIEF

### 2 years on, Log4j still haunts the security community

Research from Veracode shows nearly 2 in 5 applications are still running vulnerable versions.

Published Dec. 8, 2023



David Jones

Reporter



¿CÓMO NOS PODEMOS  
PROTEGER?



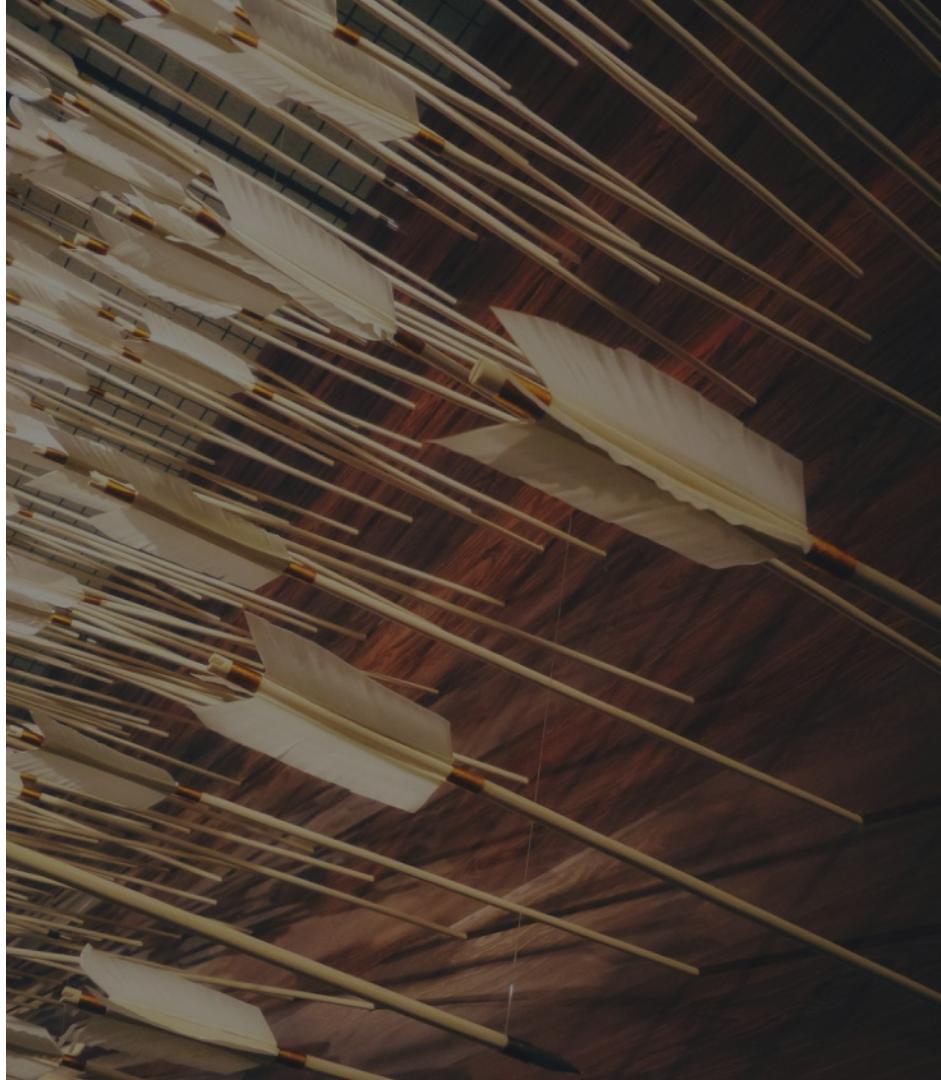
# AUDITORÍAS AUTOMÁTICAS

Hoy en día hay multitud de herramientas que analizan las dependencias que tiene un proyecto y avisan si alguna es vulnerable a problemas conocidos.

Algunas soluciones de gestión de paquetes como npm, tienen esta funcionalidad integrada.

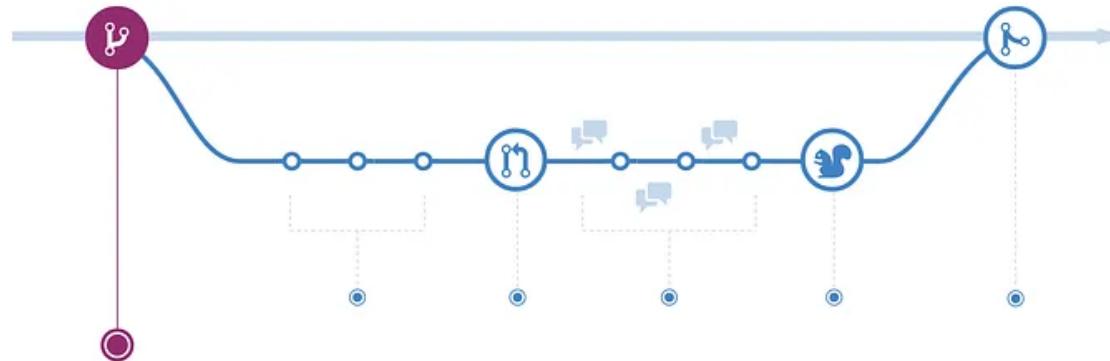
Existen multitud de herramientas comerciales y no comerciales como [OWASP Dependency-Check](#) que hacen este trabajo.

Conviene integrar este trabajo en el ciclo de vida del desarrollo.



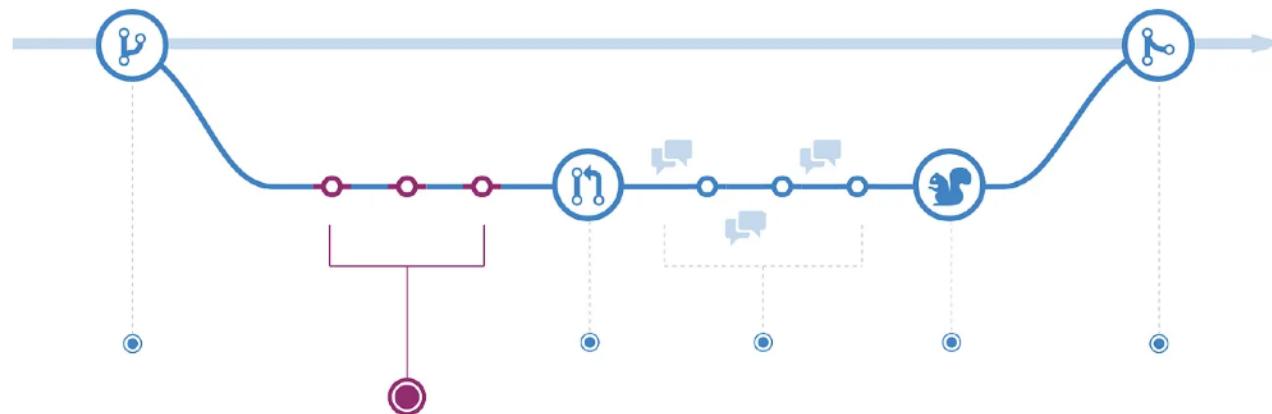
# DESARROLLO BASADO EN PR-S

Los desarrolladores no suben los cambios en el código directamente a la rama principal de GIT.



# DESARROLLO EN UNA RAMA

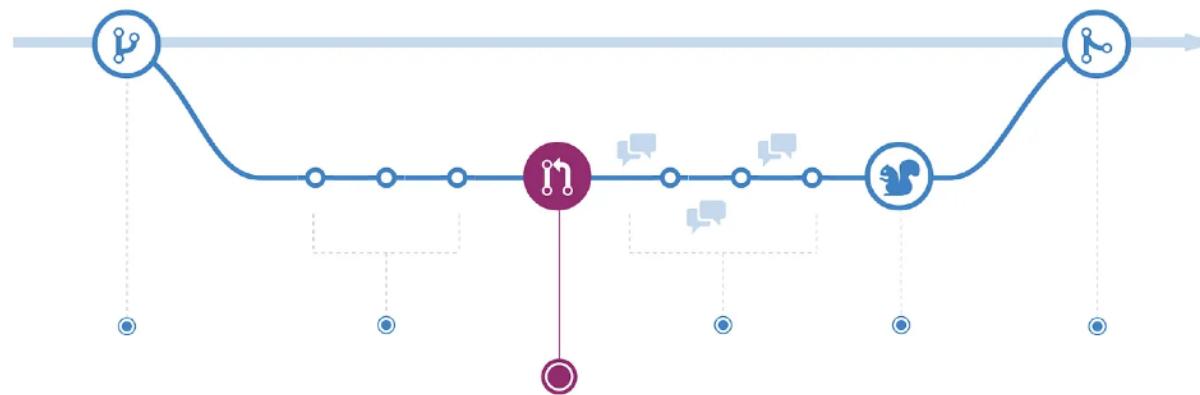
Cada desarrollador implementa su tarea en una rama independiente.



# CREACIÓN DE PULL REQUEST

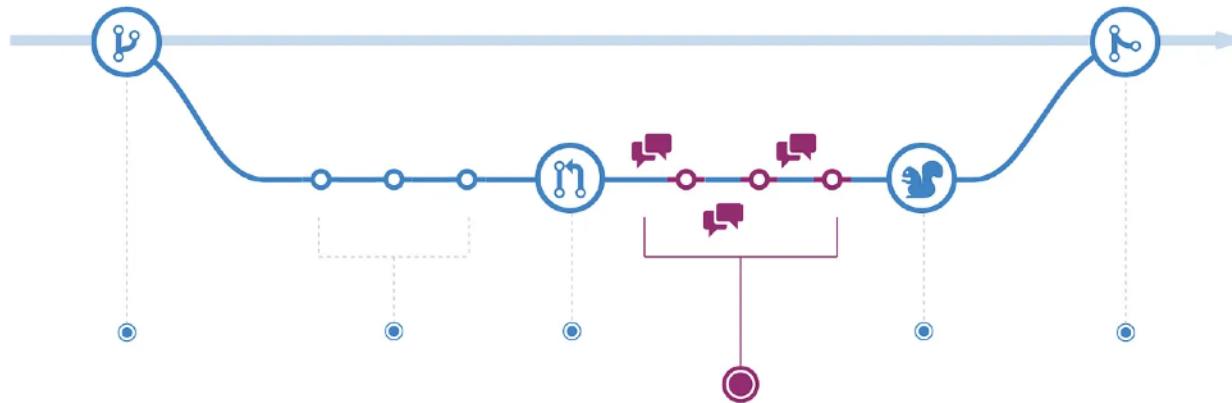
Cuando considera que el trabajo ya está listo para ser revisado, realiza un PR en el repositorio Git.

Esto dispara que se ejecute un proceso en el servidor CI en el que se construye el proyecto, se ejecutan los tests y se realizan una serie de verificaciones (npm audit por ejemplo). Si alguna de estas fases no se ejecuta de forma satisfactoria, se impide el "merge".



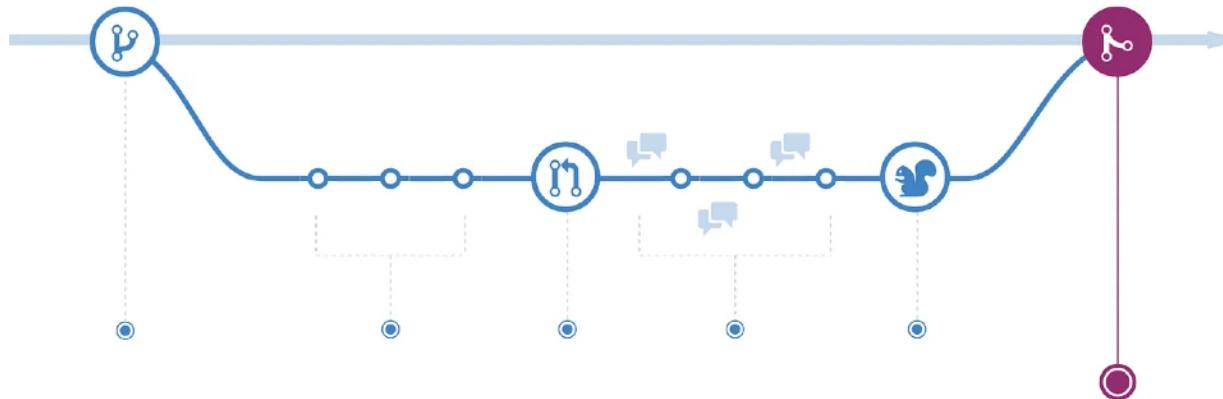
# REVISIÓN CRUZADA

Otra persona del equipo realiza una revisión al pull-request y se discuten sobre el mismo posibles errores o propuestas de mejora que se hayan podido detectar.



# MERGE

Una vez que se ha llegado a un acuerdo, se procede a realizar el “merge” de forma que los cambios realizados en la rama, pasan a formar parte de la rama principal.



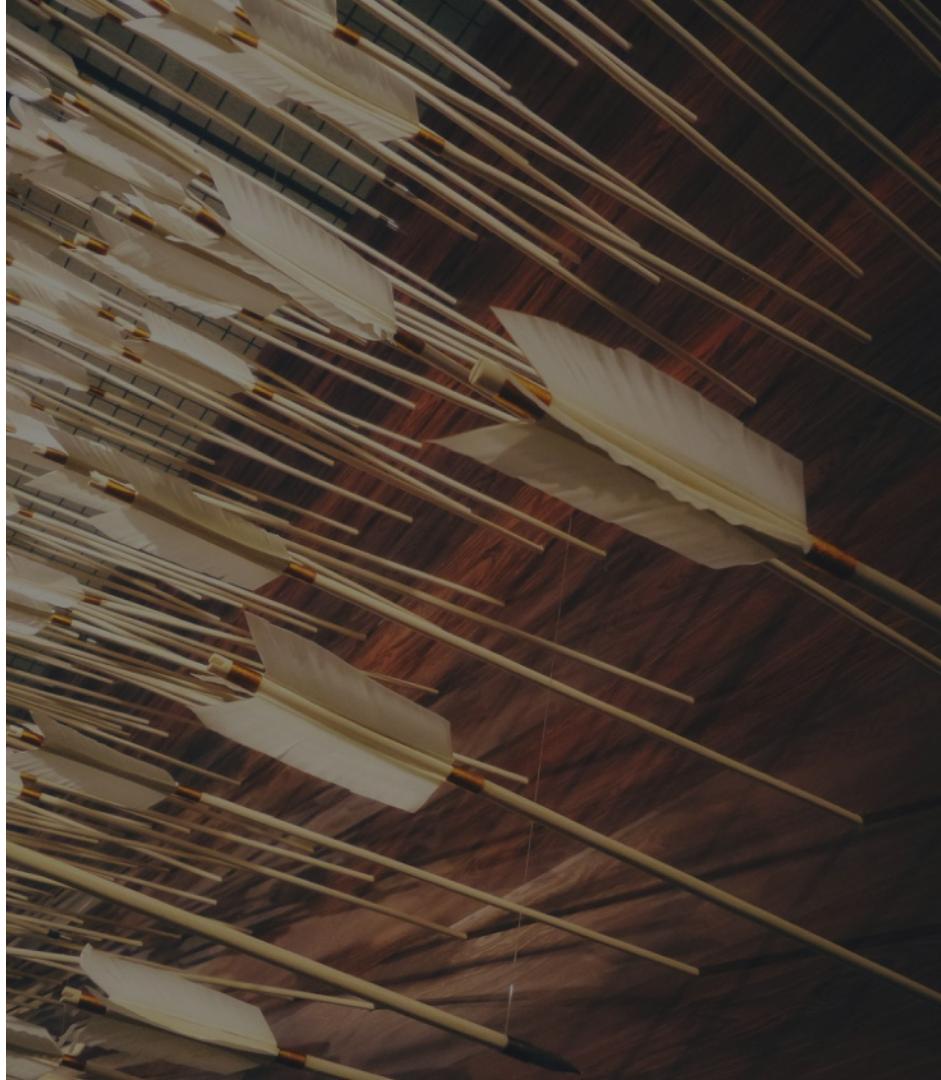
# **CALLENGE VIII:** IDENTIFICATION AND AUTHENTICATION FAILURES

# A07:2021 ID AND AUTHN FAILURES

En esta categoría se engloban las vulnerabilidades relacionadas con la gestión de las identidades de los usuarios, la sesión, etc.

Se da cuando la aplicación permite:

- / El uso de passwords débiles.
- / Almacena los passwords en claro o con algoritmos criptográficos débiles.
- / No realiza las verificaciones necesarias sobre las credenciales / tokens presentados.



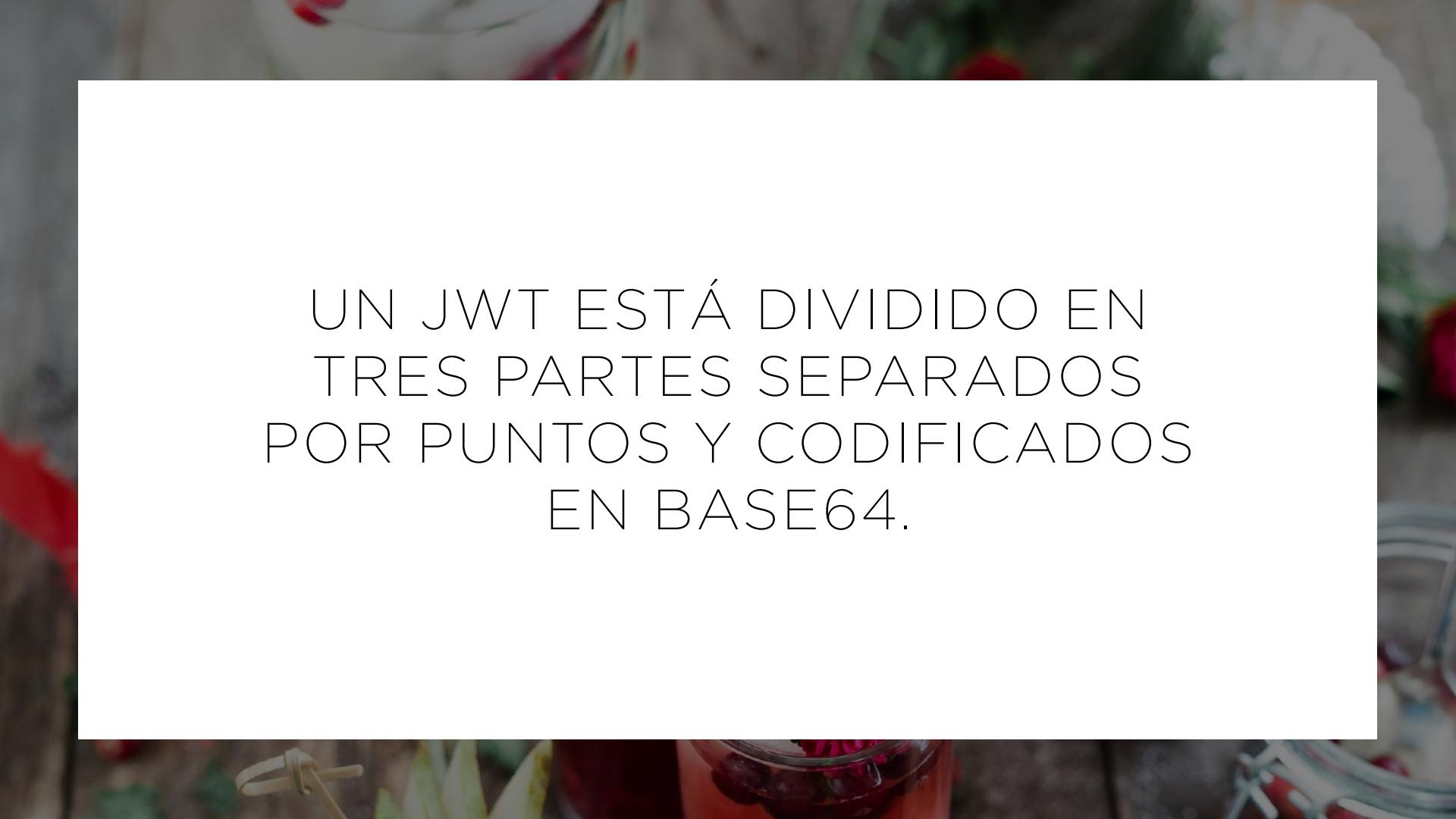
# JWT

JWT (JSON Web Token) es un tipo de token que define una forma compacta de transmitir información utilizando un objeto JSON.

Los JWT están firmados digitalmente para que puedan ser verificados. Se pueden utilizar diferentes tipos de firma:

- / Las basadas en un secreto compartido (HMAC).
- / Firmas basadas en claves pública/privada utilizando RSA o ECDSA.

eyJ0eXAi0iJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMi0iJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MjIsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJhcm10ekBhcmltYS5ldSIsInBhc3N3b3JkIjoiNDK1M2I1Mjh0DA2ZDBj0DEyYmFj0WViYjRkNjgy0TUilCJyb2x1IjoiY3VzdG9tZXIiLCJkZWx1eGVUb2t1biI6IiIsImxhc3RMb2dpbk1wIjoiMTI3LjAuMC4xIiwickHJvZmlsZUltYWdlIjoiL2Fzc2V0cy9wdWJsaWMvaW1hZ2VzL3VwbG9hZHMyZGVmYXVsdc5zdmciLCJ0b3RwU2VjcmV0IjoiIiwiXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjQtMDYtMDQgMjA6MTY6MDUuNDA0ICswMDowMCIsInVwZGF0ZWRBdCI6IjIwMjQtMDYtMDUgMTI6MzM6MDMuNTkyICswMDowMCIsImR1bGV0ZWRBdCI6bnVsbH0sImlhdcI6MTcxNzU5MDgwMn0.MkFw-Z02zJFvM0am9AHigAgsJHeKX1Nay0\_CgZW561yW\_IxiuBqWs5jUlh10QY-wH0mg1DgU-H-uk-ozf4mHGmZT4vXJIx2DvT7JRQi26tpd0\_VhvterZ1C\_dW3kSTZ-Ba9tr6w5\_TAhLWQpm0m6z0Yyg3WEjftFpLu-NWMFjKk



UN JWT ESTÁ DIVIDIDO EN  
TRES PARTES SEPARADOS  
POR PUNTOS Y CODIFICADOS  
EN BASE64.

# HEADER

En esta parte se define el tipo de token (JWT) y el algoritmo de hashing utilizado.

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9
```

```
{
  "typ": "JWT",
  "alg": "RS256"
}
```

# PAYLOAD

Contiene las declaraciones (claims) de una entidad (usualmente, el usuario) y datos adicionales. Hay una serie de propiedades que son estándar como: iss (issuer), exp (expiration time), sub (subject), ...

```
ey  
JzdGF0dXMi0iJzdWNjZXNzIiwiZGF0YSI6eyJpZ  
CI6MjIsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJh  
cm10ekBhcmltYS5ldSIsInBhc3N3b3JkIjoiNDk  
1M2I1Mjh0DA2ZDBj0DEyYmFj0WViYjRkNjgyOT  
UiLCJyb2x1IjoiY3VzdG9tZXIiLCJkZWx1eGVUb  
2tlbiI6IiIsImxhc3RMb2dpbkIwIjoiMTI3LjAu  
MC4xIiwicHJvZmlsZU1tYWd1IjoiL2Fzc2V0cy9  
wdWJsaWMvaW1hZ2VzL3VwbG9hZHmvZGVmYXVsdc  
5zdmciLCJ0b3RwU2VjcmV0IjoiIiwiXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjQtMDYt  
MDQgMjA6MTY6MDUuNDA0ICswMDowMCIsInVwZGF  
0ZWRBdCI6IjIwMjQtMDYtMDUgMTI6MzM6MDMuNT  
kyICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbH0sIm  
lhcdCI6MTcxNzU5MDgwMn0
```

```
{  
  "status": "success",  
  "data": {  
    "id": 22,  
    "username": "",  
    "email": "aritz@arima.eu",  
    "password": "4953b528f806d0c812bac9ebb4d68295",  
    "role": "customer",  
    "deluxeToken": "",  
    "lastLoginIp": "127.0.0.1",  
    "profileImage":  
      "/assets/public/images/uploads/default.svg",  
    "totpSecret": "",  
    "isActive": true,  
    "createdAt": "2024-06-04 20:16:05.404 +00:00",  
    "updatedAt": "2024-06-05 12:33:03.592 +00:00",  
    "deletedAt": null  
  },  
  "iat": 1717590802  
}
```

# SIGNATURE

Para crear la firma se utiliza el header codificado en base64, el payload codificado en base64 y se firma con el algoritmo especificado en la cabecera.

MkFw-

Z02zJFvM0am9AHigAgsJHeKX1NayO\_CgZW561yW  
\_IxieuBqWs5jUlh10QY-wH0mg1DgU-H-uk-  
ozf4mHGmZT4vXJIX2DvT7JRQi26tpd0\_VhvterZ  
1C\_dW3kSTZ-  
Ba9tr6w5\_TAhLWQpm0m6z0Yyg3WEjftFpLu-  
NWMFjKk

# VALIDACIÓN DE JWT-S

Los JWT se han convertido en tokens de autenticación y autorización muy populares debido a su naturaleza autocontenido.

Suele enviarse en la cabecera HTTP "Authorization" con el prefijo "Bearer".

La validación de este tipo de token consiste en verificar que no ha expirado (propiedad exp del payload vigente), que la firma se corresponde con el contenido recibido y que proviene de un origen de confianza.

eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MjIsInVzZXJuYWlIjoiIiwiZW1haWwiOiJhcm10ekBhcmltYS5ldSIsInBhc3N3b3JkIjoiNDK1M2I1Mjh0DA2ZDBj0DEyYmFjOWViYjRkNjgy0TUiLCJyb2x1IjoiY3VzdG9tZXIiLCJkZWx1eGVUb2tlbiI6IiIsImxhc3RMb2dpbk1wIjoiMTI3LjAuMC4xIiwiCHJvZmlsZUltYWdlIjoiL2Fzc2V0cy9wdWJsaWMvaW1hZ2VzL3VwbG9hZHMyZGVmYXVsdc5zdmciLCJ0b3RwU2VjcmV0IjoiIiwiXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjQtMDYtMDQgMjA6MTY6MDUuNDA0ICswMDowMCIsInVwZGF0ZWRBdCI6IjIwMjQtMDYtMDUgMTI6MzM6MDMuNTkyICswMDowMCIsImR1bGV0ZWRBdCI6bnVsbH0sImlhdcI6MTcxNzU5MDgwMn0.MkFw-Z02zJFvM0am9AHigAgsJHeKX1Nay0\_CgZW561yW\_IxiuBqWs5jUlh10QY-wH0mg1DgU-H-uk-ozf4mHGMZT4vXJIx2DvT7JRQi26tpd0\_VhvterZ1C\_dW3kSTZ-Ba9tr6w5\_TAhLWQpm0m6z0Yyg3WEjftFpLu-NWMFjKk

LET'S HACK!



¿CÓMO NOS PODEMOS  
PROTEGER?



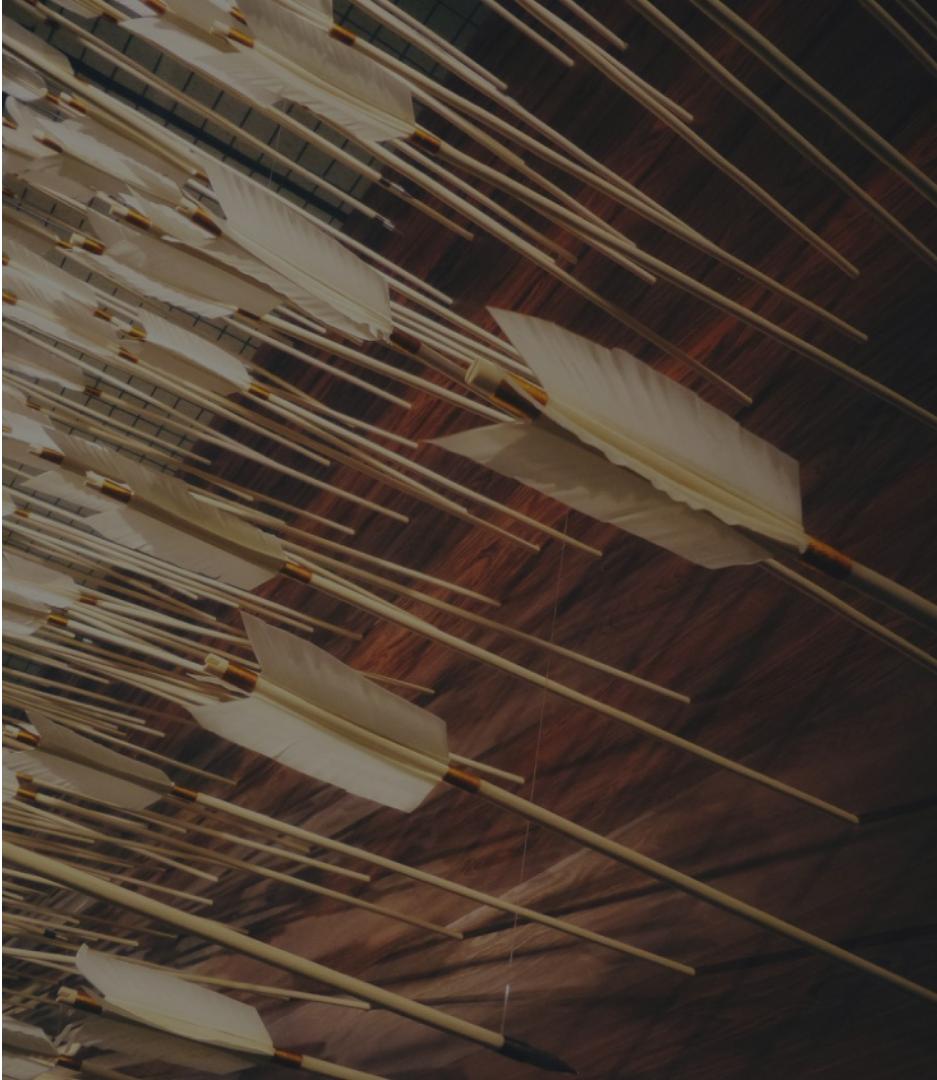
# BUENAS PRÁCTICAS

Implementar autenticación con multi factor.

No desplegar herramientas con credenciales por defecto, especialmente para usuarios admin.

Implementar verificaciones para detectar passwords débiles o filtrados.

Limitar el número de logins fallidos.



# CALLenge IX: SOFTWARE AND DATA INTEGRITY FAILURES

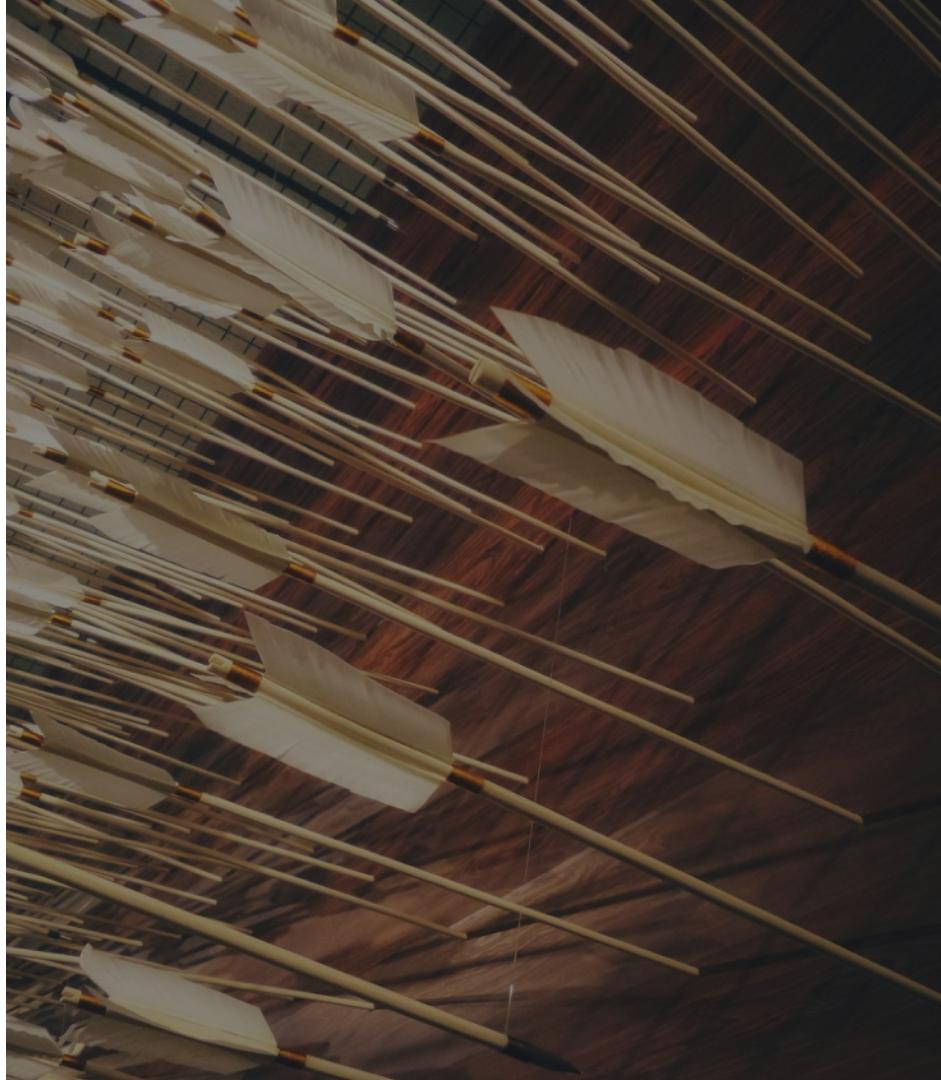


# A08:2021 SOFTWARE AND DATA INTEGRITY

Es una nueva categoría introducida en el ranking del 2021.

Se basa en las asunciones que hacemos a la hora de llevar a cabo actualizaciones de software y pipelines CI/CD sin verificar la integridad.

Esto afecta también a los sistemas que se actualizan automáticamente ya que deben garantizar que las actualizaciones que se están aplicando automáticamente son legítimas.



# PACKAGE LOCK

En NPM tenemos el archivo **package-lock.json** además del package.json que es donde se establecen las dependencias del proyecto.

Este archivo tiene una doble función:

- / Fijar las versiones de las dependencias para hacer que el build sea reproducible.
- / Añadir el SHA de cada dependencia para verificar que no ha habido manipulaciones.

```
"dependencies": {  
    "@angular/animations": {  
        "version": "4.2.6",  
        "resolved": "https://registry.npmjs.org/@angular/animations/-/animations-4.2.6.tgz",  
        "integrity": "sha1-nZyAoRmwDaTy9I7uvcosVMf/8c="  
    },  
    "@angular/common": {  
        "version": "4.2.6",  
        "resolved": "https://registry.npmjs.org/@angular/common/-/common-4.2.6.tgz",  
        "integrity": "sha1-IQzOS9JON1+LQbpS/rNLGKiHdo="  
    },  
    "@angular/compiler": {  
        "version": "4.2.6",  
        "resolved": "https://registry.npmjs.org/@angular/compiler/-/compiler-4.2.6.tgz",  
        "integrity": "sha1-ZndW1JXKDUXSBhJooQisr40fr/Q="  
    },  
    "@angular/core": {  
        "version": "4.2.6",  
        "resolved": "https://registry.npmjs.org/@angular/core/-/core-4.2.6.tgz",  
        "integrity": "sha1-DByP8BV/B29KfAtyHKFCPxu+Fk4="  
    },  
    "@angular/forms": {  
        "version": "4.2.6",  
        "resolved": "https://registry.npmjs.org/@angular/forms/-/forms-4.2.6.tgz",  
        "integrity": "sha1-nTl6lgjkYDu/GXQXqluU6Ap0frA="  
    },  
    "@angular/http": {  
        "version": "4.2.6",  
        "resolved": "https://registry.npmjs.org/@angular/http/-/http-4.2.6.tgz",  
        "integrity": "sha1-SZ4rolvB89cbdt6+wOTJWMrxE04="  
    },  
    "@angular/platform-browser": {  
        "version": "4.2.6",  
        "resolved": "https://registry.npmjs.org/@angular/platform-browser/-/platform-browser-4.2.6.tgz",  
        "integrity": "sha1-oTH/WSil/mSwvKLJz/YSpNvd9Dc="  
    },  
    "core-js": {  
        "version": "2.4.1",  
        "resolved": "https://registry.npmjs.org/core-js/-/core-js-2.4.1.tgz",  
        "integrity": "sha1-1dOgqfjXQV9oCwvzXnqFmzXuXWU="  
    },  
    "zone.js": {  
        "version": "0.8.4",  
        "resolved": "https://registry.npmjs.org/zone.js/-/zone.js-0.8.4.tgz",  
        "integrity": "sha1-1dOgqfjXQV9oCwvzXnqFmzXuXWU="  
    }  
}
```

# GO.SUM

En el gestor de paquetes de Golang existe un concepto similar al de NPM: por un lado está el go.mod que es donde se establecen las dependencias y por otro, el go.sum que es donde están los checksums de estos.

Al descargar las dependencias se realiza una validación de las firmas y si no coinciden, se interrumpe la instalación lanzando un error.

```
cloud.google.com/go v0.26.0/go.mod h1:aQUYkXzVsufM+DwF1aE+0xfcU+56JwCaLick0ClmMTw=
cloud.google.com/go v0.34.0/go.mod h1:aQUYkXzVsufM+DwF1aE+0xfcU+56JwCaLick0ClmMTw=
cloud.google.com/go v0.38.0/go.mod h1:990N+gFupTy94rShfmMCWGDN0LpTmnzTp2qbd1dvSRU=
cloud.google.com/go v0.44.1/go.mod h1:iSa0KzasP4Uvy3f1mN/7Pi0bzGgflwredwwASm/v6AU=
cloud.google.com/go v0.44.2/go.mod h1:60680Gw3Yr4ikxnPRS/oxxxBccT6SA1yMk63TGekxKY=
cloud.google.com/go v0.44.3/go.mod h1:60680Gw3Yr4ikxnPRS/oxxxBccT6SA1yMk63TGekxKY=
cloud.google.com/go v0.45.1/go.mod h1:RpBamKRgapWjb87xiFsdk4g1CME7QZg3uwTez+TSTjc=
cloud.google.com/go v0.46.3/go.mod h1:a6bKKbmY7er1mI7TEI4lsAktks/mkhTSZK8w33B4RAg0=
cloud.google.com/go v0.50.0/go.mod h1:r9sluTvynVuxRIOHXQEHMffphuXHOMZMycpNR5e6To=
cloud.google.com/go v0.52.0/go.mod h1:pXajvRH/6o3+F9jDHZWQ5PbGhn+o8w9qiu/CffaVd04=
cloud.google.com/go v0.53.0/go.mod h1:fp/UouUEsRkN6ryDKNW/Upv/JBKnv6WDthjR6+vze6M=
cloud.google.com/go v0.54.0/go.mod h1:1rq20EkV3YMF6n/9ZvGWI3GWw0VodH/1x2nd8Is/bPc=
cloud.google.com/go v0.56.0/go.mod h1:jz7tqZxxKOVYizybht9+26Z/gUq7tiRzu+ACVAMBKVk=
cloud.google.com/go v0.57.0/go.mod h1:oXiQ6Rzq3RAkkY7N6t3TcE6jE+CIBBbA36lwQ1JyzZs=
cloud.google.com/go v0.62.0/go.mod h1:jmCYTdRCQuc1PHIIJ/maLInMho30T/Y0M4hTdTSh0Yc=
cloud.google.com/go v0.65.0/go.mod h1:05N8zS7uWy9vkA9vayVHs65eM1ubvY4h553ofrNHObY=
cloud.google.com/go v0.72.0/go.mod h1:M+5Vjvlc2wnp6tjzE102Dw08nGShTscUx2nZMufOKPI=
cloud.google.com/go v0.74.0/go.mod h1:VV1xSbzvo+9QJOxLDaJfTjx5e+MePCpCwvftOeQmWk=
cloud.google.com/go v0.75.0/go.mod h1:VGuuCn7PG0dw5dXPVm2Mm3w1h3EL55/79EKB6h1PTY=
cloud.google.com/go v0.110.10 h1:Lxy9GEO+timppncPIAZo0j3158LIU9k+kn48AN7I03Y=
cloud.google.com/go v0.110.10/go.mod h1:v10oFqYxiBkUrruItNM3eT41LByNjxmJSV/xDKJNnic=
cloud.google.com/go/bigquery v1.0.1/go.mod h1:i:xbL2U1R5RvWAURpBYZTtm/cXjCha91bfpx4poX+o=
cloud.google.com/go/bigquery v1.3.0/go.mod h1:PjpwJns1EMmckchkHFfq+HTD2DmtT67aNFKH1/VBDHE=
```

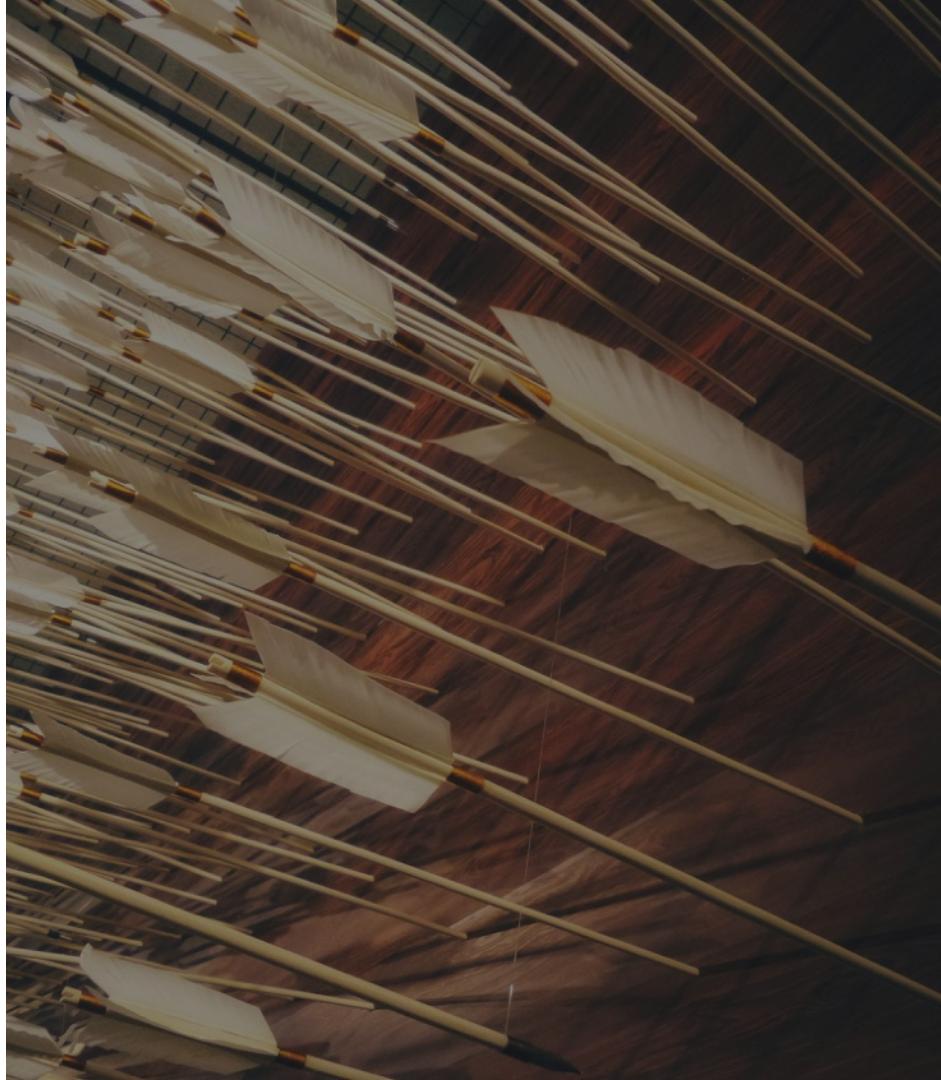
¿CÓMO NOS PODEMOS  
PROTEGER?



# UTILIZAR SOFTWARE FIRMADO

Verificar la integridad de todo el software que utilizamos haciendo comprobaciones de sus firmas. Esto es aplicable a todos los niveles, desde el sistema operativo hasta las librerías que utilizamos en el código.

Prestar especial atención a los procesos de CI/CD.



# CHALLENGE X: SECURITY LOGGING AND MONITORING FAILURES

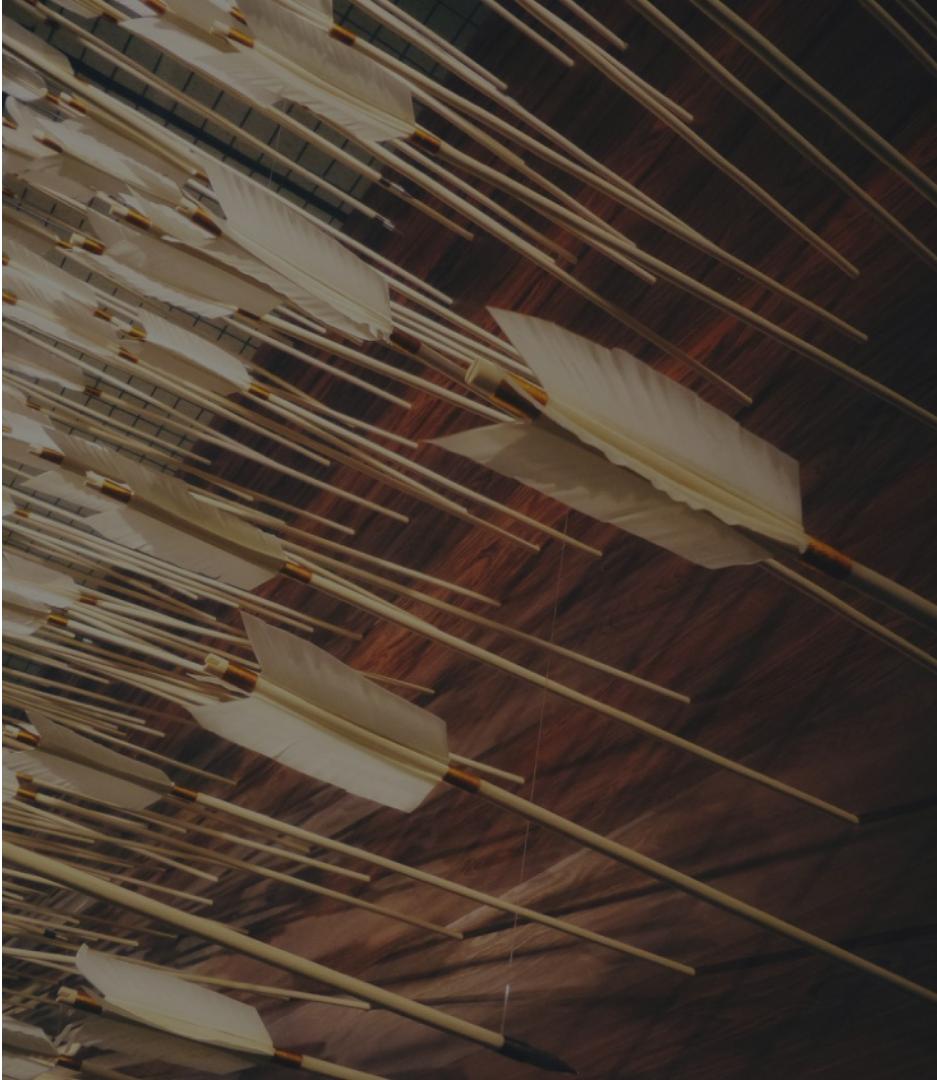
# A09:2021

## SECURITY LOGGING AND MONITORING

Esta categoría pretende ayudar a detectar, escalar y dar respuesta ante incidentes de seguridad.

Sin sistemas de logging y monitorización es imposible detectar incidentes. Ocurre cuando:

- / Los logs de las aplicaciones y APIs no se monitorizan en busca de actividad sospechosa.
- / Las herramientas de pentesting y escaneo de código no lanzan alertas.
- / No se tratan los logs como información sensible.



LET'S HACK!



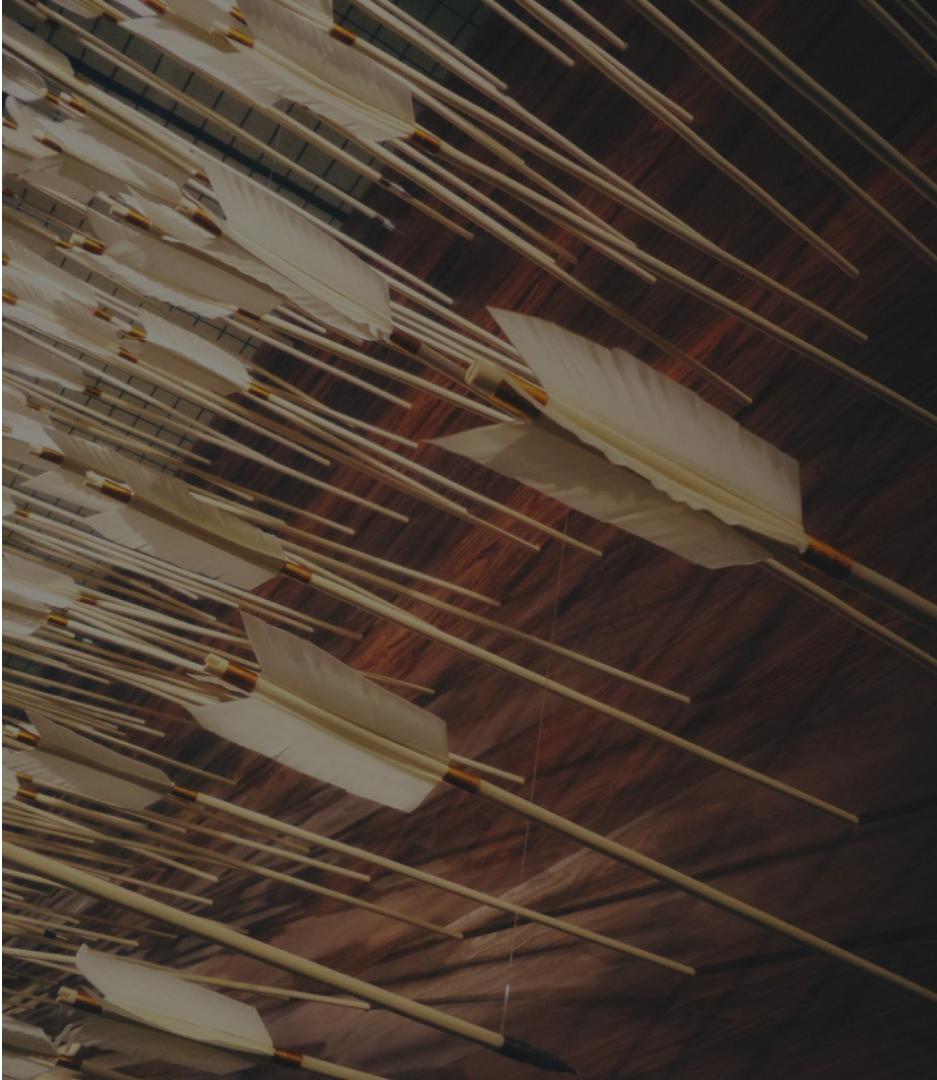
¿CÓMO NOS PODEMOS  
PROTEGER?



# PREVENCIÓN

Es necesario aplicar medidas para detectar actividades sospechosas:

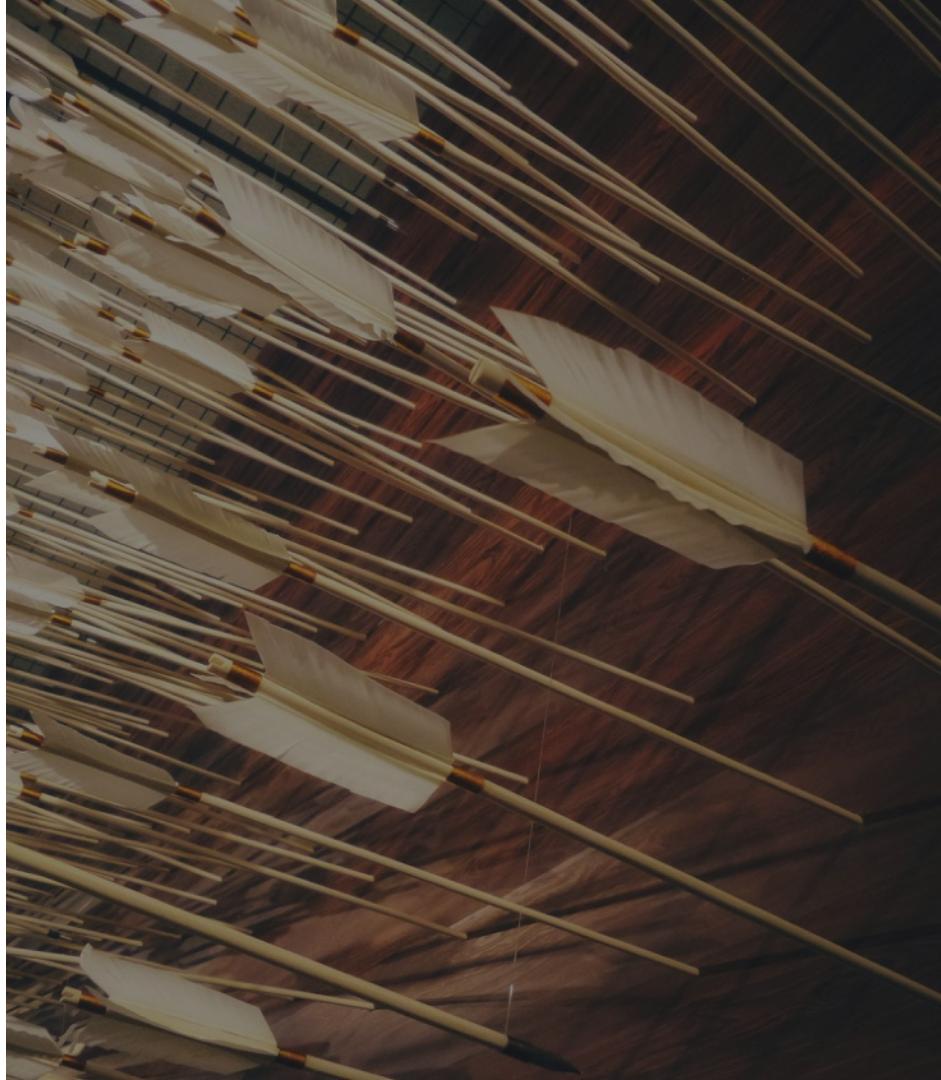
- / Garantizar que cualquier fallo de login, control de acceso o validación se escribe al log con suficiente contexto para poder ser investigado.
- / Utilizar un formato de logs que se pueda consumir fácilmente.
- / Garantizar que los logs están codificados y que no pueden ser alterados.



# PREVENCIÓN

- / Los equipos de DevSecOps tienen que establecer criterios de monitorización efectiva para detectar y responder rápidamente ante posibles ataques.
- / Adoptar planes de respuesta a incidentes y recuperación como los marcados por el estándar 800-61r2 del (NIST).

Hay infinidad de productos que ofrecen este tipo de funcionalidades: Elasticsearch, Logstash, Kibana, Datadog, New Relic, Dynatrace,...

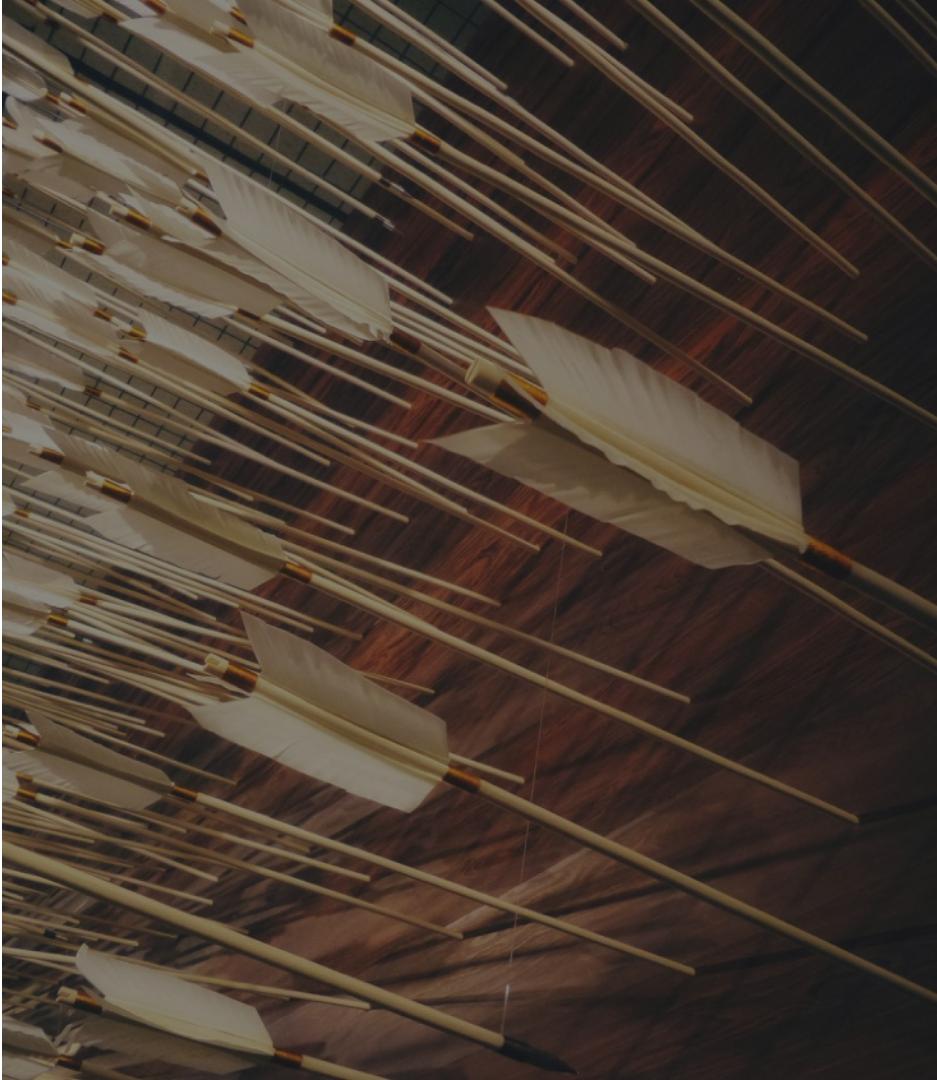


# CALLENCE XI: SERVER SIDE REQUEST FORGERY

# A10:2021 SSRF

SSRF se da cuando una aplicación está accediendo a una URL remota proporcionada por el usuario sin ser validada.

Permite al atacante enviar una petición a un destino no esperado pudiendo provocar fuga de información saltándose firewalls o VPNs.



LET'S HACK!



# ALGUNOS CASOS POPULARES

# PAYPAL

En 2014 hubo un robo de identidades de Apple aprovechando que Paypal no tenía restringidas las URLs a las que redirigía una vez realizado un pago.

El ataque consistía en el envío de un email de phishing con recibos del iTunes Store relativos a la compra de productos caros e invitando a cancelar el pedido si no habían sido ellos los que lo habían hecho.

El link de cancelación parecía legítimo: [www.order.itunes.com/verify/cancel](http://www.order.itunes.com/verify/cancel). Sin embargo, llevaba a la página de PayPal diciendo que se les iban a realizar una serie de preguntas personales y financieras para verificar la pertenencia de la cuenta.

Incident Response, TDR



## PayPal phishing websites spike in 2014, easy vector for attackers

Adam Greenberg May 29, 2014

Phishers have their crosshairs steadied on PayPal now more than ever, according to ["The Internet Threats Trend Report April 2014,"](#) a collaborative effort between cloud-based internet security solutions provider CYREN and network security appliances provider Cyberoam.

In analyzing information security trends in the first quarter of this year and comparing the data to 2013, researchers observed a 73 percent increase in the number of phishing websites related to PayPal, according to the report, which adds the sites go after personal data, including Social Security numbers.

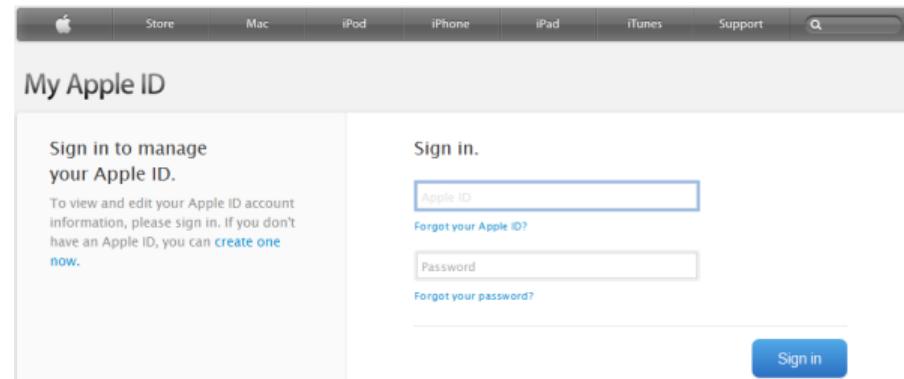
The report indicates that about 18,600 PayPal-related phishing websites were identified in a two-week span, which trumps the runner-up in the same timeframe – about 2,261 Apple phishing websites – by a significant number.

These findings did not surprise Jérôme Segura, senior security researcher with Malwarebytes.

# PAYPAL

La URL que utilizaban para acceder a PayPal era la siguiente: <https://www.paypal-communication.com/r/4V2JION/PPPU5A/GDY6I8I/20PEVD/7ZS7MP/7M/h?a=http://192.185.##.##/~broo23yo/>

Esta URL redirigía inmediatamente a las víctimas a un sitio de la apariencia de Apple.



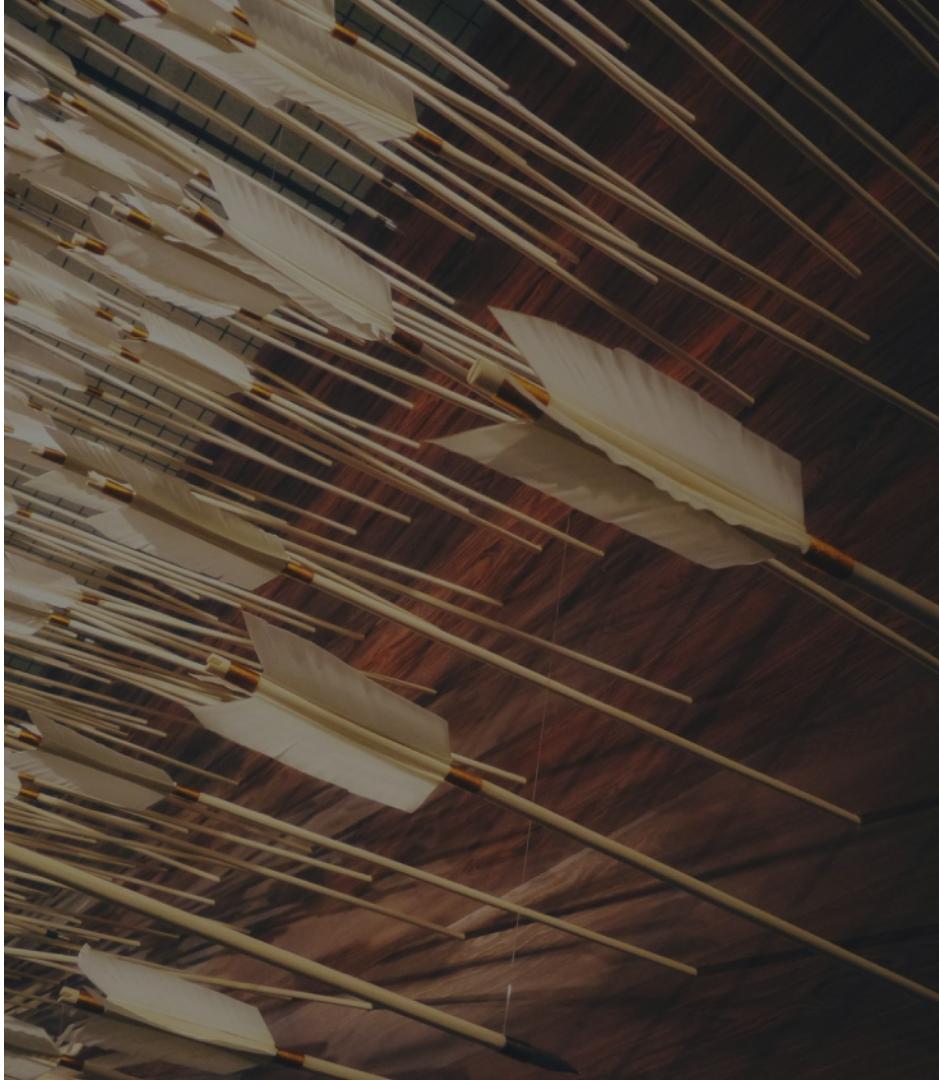
¿CÓMO NOS PODEMOS  
PROTEGER?



# PREVENCIÓN

Desde el punto de vista de la aplicación se pueden tomar las siguientes medidas:

- / Limpiar y validar todo lo que nos llega del cliente.
- / Obligar a que la URL recibida se encuentre en una Allow List.



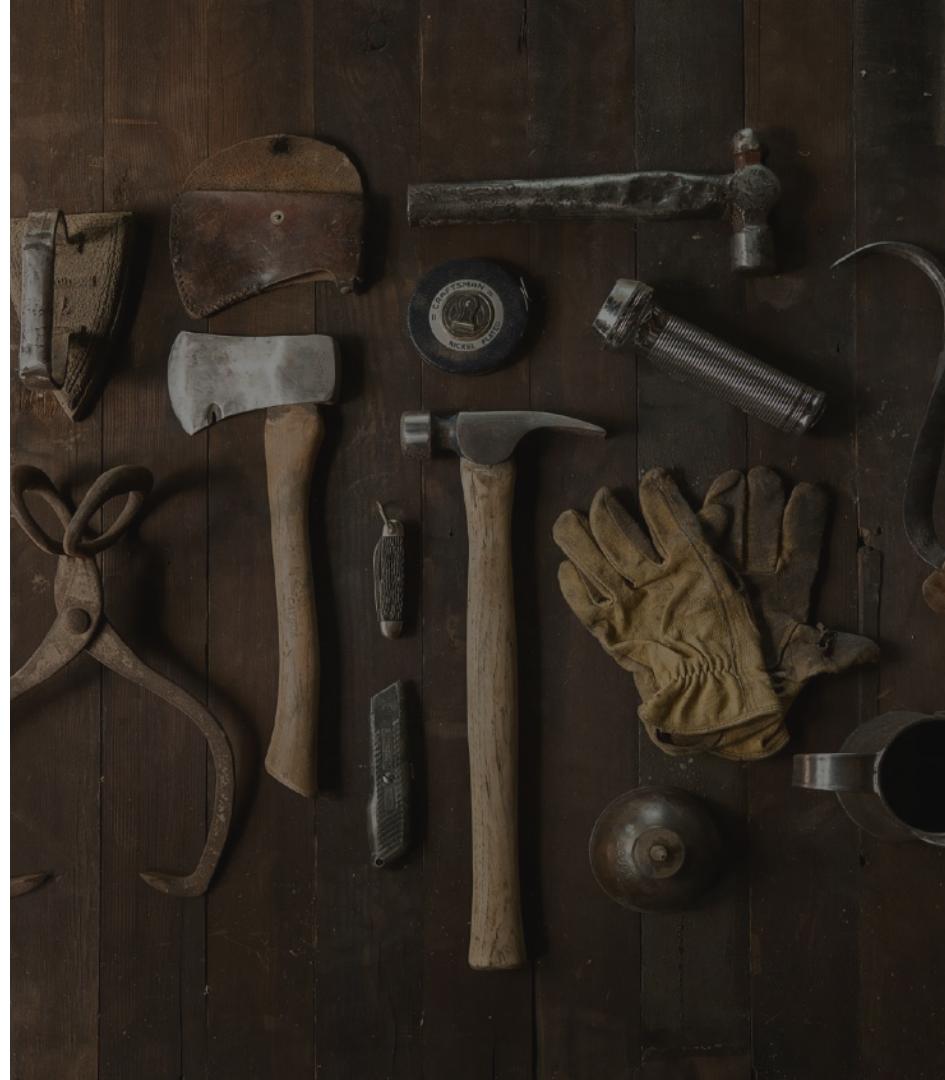
# 04. / HERRAMIENTAS

# HERRAMIENTAS

Más allá de las buenas prácticas que podamos seguir, existen infinidad de herramientas que pueden ayudarnos en nuestro trabajo.

Se pueden identificar distintas categorías de productos aunque cada vez se difumina más la responsabilidad de cada uno:

- / SCA
- / SAST
- / DAST
- / IAST



# SCA

Los SCA (Software Composition Analysis) son herramientas que permiten identificar y gestionar las vulnerabilidades de seguridad en las librerías de terceros utilizadas en las aplicaciones.

El SCA hace una revisión de todas las dependencias con el objetivo de detectar componentes desactualizados, licencias no compatibles y vulnerabilidades conocidas.

Se integran en el ciclo de vida del desarrollo para identificar y mitigar los riesgos en fases tempranas.



WhiteSource

# SAST

Un SAST (Static Application Security Testing) analiza el código fuente de las aplicaciones en búsqueda de potenciales riesgos de seguridad.

Aplicando reglas puede identificar patrones que pueden implicar vulnerabilidades como XSS, SQLI, passwords.

Al analizar el código estático sin ningún contexto, pueden dar falsos positivos ya que puede ocurrir que un bloque de código que pueda incluir un XSS, en la práctica no sea tal porque nunca se le llega a injectar un valor provisto por el usuario.

Suelen integrarse en los repositorios de código fuente para realizar análisis periódicos.



# DAST

Los DAST (Dynamic Application Security Testing) son herramientas que se enfocan en identificar vulnerabilidades en aplicaciones simulando ataques externos y sin conocer nada sobre el código.

Estas herramientas son muy útiles para encontrar problemas de seguridad como configuraciones erróneas del servidor, errores en tiempo de ejecución, vulnerabilidades XSS y SQLI, etc.



# IAST

Los IAST (Interactive Application Security Testing) tienen la particularidad de que monitorizan la aplicación mientras esta se está ejecutando en búsqueda de potenciales vulnerabilidades.

Al tener el contexto de ejecución pueden identificar qué vulnerabilidades son riesgos reales.

No detectan vulnerabilidades que puedan darse en partes de la aplicación que por cualquier motivo no se han ejecutado.





✉ aritz@arima.eu

 <https://www.linkedin.com/in/aritzberasarte/>

 <https://twitter.com/Raskasso>