

# **BLUE SHIELD TEAM**



## **Informe de auditoría web**

### **Auditores:**

**Dario Maroto López**

**Rodrigo Verdú Fernández**

**Alejandro Navarro de la Torre**

**Raúl Carballo García**

**Mariama Salhi Seidi**

# ÍNDICE

|   |    |
|---|----|
| 1. INTRODUCCIÓN .....   | 5  |
| 2. ALCANCE .....  | 6  |
| 2.1. OBJETIVO .....   | 6  |
| 2.2. ACTIVO .....   | 6  |
| 3. METODOLOGÍA .....  | 6  |
| 4. RESUMEN EJECUTIVO .....  | 7  |
| 4.1. Resumen de vulnerabilidades .....  | 8  |
| 4.2. Recomendaciones generales .....  | 8  |
| 5. DETALLE TÉCNICO VULNERABILIDADES .....   | 9  |
| 5.1. CROSS SITE SCRIPTING - XSS .....   | 9  |
| 5.2. INYECCIÓN HTML .....   | 13 |
| 5.3. AUSENCIA DE LA CABECERA X-FRAME-OPTIONS - VULNERABILIDAD<br>CLICKJACKING ..... | 16 |
| 5.4. AUSENCIA DE CABECERAS DE SEGURIDAD .....                                       | 19 |
| 5.5. USO DE CIFRADOS TLS/SSL INSEGUROS - LUCKY13 .....                              | 23 |
| 5.6. USO DE CIFRADOS TLS/SSL INSEGUROS - BREACH .....                               | 25 |
| 5.7. COOKIES SIN PARÁMETROS DE SEGURIDAD ESTABLECIDOS .....                         | 28 |
| 6. ANEXO I: CVSS: SISTEMA DE PUNTUACIÓN .....                                       | 30 |
| 6.1. Puntuación de Vulnerabilidades .....   | 30 |
| 6.1.1. Métricas de explotabilidad .....   | 30 |
| 6.1.2. Alcance (Scope) .....  | 31 |
| 6.1.3. Métricas de impacto .....  | 32 |
| 6.2. Categorización .....   | 33 |

# ÍNDICE DE FIGURAS

|   |    |
|---|----|
| Figura 1. Parámetro insertado en URL reflejado en HTML.....                     | 10 |
| Figura 2. Inserción de parámetro en variable .....                              | 10 |
| Figura 3. Cierre de la clase <input> .....                                      | 11 |
| Figura 4. Inyección de código JavaScript exitoso .....                          | 11 |
| Figura 5. Introducción del parámetro “buscando” en la URL .....                 | 13 |
| Figura 6. Introducción del segundo parámetro en la URL .....                    | 14 |
| Figura 7. Extracción del valor del atributo "value" .....                       | 14 |
| Figura 8. Inyección de formulario HTML.....                                     | 15 |
| Figura 9. Ausencia de la cabecera X-Frame-Options por ShCheck.....              | 16 |
| Figura 10. Código HTML para prueba de ClicJacking .....                         | 17 |
| Figura 11. Explotación correcta de ClicJacking .....                            | 17 |
| Figura 12. Cabeceras ausentes .....   | 20 |
| Figura 13. Recurso externo cargado dentro del aplicativo.....                   | 21 |
| Figura 14. Servidor potencialmente vulnerable a Lucky13 .....                   | 23 |
| Figura 15. Cifrados CBC usados por el servidor .....                            | 24 |
| Figura 16. Activo vulnerable a BREACH .....                                     | 25 |
| Figura 17. Petición con Accept-Encodig: compress, gzip .....                    | 26 |
| Figura 18. Configuración de BurpSuite para recibir peticiones comprimidas ..... | 26 |
| Figura 19. Respuesta del servidor comprimida .....                              | 27 |
| Figura 20. Cookies sin parámetros de seguridad establecidos.....                | 29 |

# ÍNDICE DE TABLAS

|   |    |
|---|----|
| Tabla 1. Alcance .....                                  | 6  |
| Tabla 2. Número de vulnerabilidades por severidad ..... | 7  |
| Tabla 3. Resumen de vulnerabilidades .....              | 8  |
| Tabla 4. Métricas de explotabilidad .....               | 31 |
| Tabla 5. Alcance .....                                  | 31 |
| Tabla 6. Métricas de impacto .....                      | 32 |
| Tabla 7. Categorización CVSS.....                       | 33 |



## 1. INTRODUCCIÓN

La seguridad de la información en el ámbito empresarial es un desafío muy presente en la actualidad, donde la tecnología desempeña un papel fundamental en las operaciones diarias. Las organizaciones se enfrentan a una amplia gama de amenazas, desde ataques cibernéticos sofisticados llevados a cabo por usuarios malintencionados hasta brechas de seguridad internas que pueden comprometer la integridad de los datos sensibles. En este contexto, el Pentesting emerge como una estrategia fundamental para evaluar y fortalecer la postura de seguridad de una empresa.

El Pentesting, también conocido como pruebas de penetración, ofrece una solución efectiva para identificar y mitigar vulnerabilidades. Esta técnica implica la simulación de ataques cibernéticos contra los sistemas de una organización utilizando las mismas herramientas y técnicas que utilizarían los ciberdelincuentes con el objetivo de descubrir y explotar posibles brechas de seguridad antes de que puedan ser aprovechadas por actores externos.

La distinción clave del Pentesting radica en su enfoque completo para evaluar la seguridad empresarial. A diferencia de las herramientas automatizadas de escaneo, que pueden detectar vulnerabilidades superficiales, el Pentesting manual profundiza, examinando el modelo de seguridad en su conjunto. Los Pentesters, especialistas en seguridad informática, no se limitan a identificar vulnerabilidades técnicas; también analizan la eficacia de los controles y políticas de seguridad.

Una de las principales ventajas del Pentesting es su capacidad para proporcionar una evaluación realista del riesgo. Al simular ataques reales, es posible identificar las vulnerabilidades más críticas y proporcionar recomendaciones específicas para su mitigación, permitiendo así a las organizaciones priorizar sus recursos en abordar los problemas de seguridad más graves.

Además, el Pentesting manual ofrece una profundidad de análisis que las herramientas automatizadas no pueden igualar. Los Pentesters no solo identifican vulnerabilidades técnicas, sino que también evalúan la lógica de negocio de las aplicaciones y sistemas, buscando posibles puntos de explotación. Esto es especialmente importante en entornos empresariales complejos, donde las vulnerabilidades pueden surgir de interacciones inesperadas entre diferentes sistemas y procesos.

## 2. ALCANCE

### 2.1. OBJETIVO

El objetivo de un análisis de vulnerabilidades web es examinar exhaustivamente un aplicativo web en busca de posibles fallos de seguridad que podrían ser explotados por actores malintencionados. Este proceso implica identificar y evaluar diversas vulnerabilidades potenciales que podrían comprometer la seguridad del sitio, como problemas de configuración, deficiencias en el control de acceso, fallos de validación de datos y otras vulnerabilidades comunes. El análisis de vulnerabilidades web tiene como propósito proporcionar una visión clara de los riesgos de seguridad asociados con el sitio web y recomendar medidas correctivas para mitigar dichos riesgos y fortalecer la seguridad en línea.

### 2.2. ACTIVO

El ámbito de un análisis web, se determina por los activos que la organización desea examinar, y se comunica al equipo auditor.

A continuación, se especifica el alcance acordado:

| ACTIVO  | TIPO DE PRUEBA | ENTORNO    |
|---|----------------|------------|
|  | Caja Negra     | Producción |

Tabla 1. Alcance

En el ámbito de la ciberseguridad, una prueba de tipo caja negra permite evaluar la seguridad de un sistema sin tener ningún conocimiento previo sobre su el mismo con la intención de simular un ataque real hacia el activo.

La principal ventaja de este enfoque es que proporciona una perspectiva más realista de cómo un atacante externo podría intentar explotar las vulnerabilidades de un sistema.

## 3. METODOLOGÍA

Para el análisis realizado se ha llevado a cabo la Metodología OWASP (*Open Web Application Security Project*) es un proyecto de código abierto dedicado a determinar y combatir las causas que hacen que el software o las páginas webs sean inseguros. La Fundación OWASP es un organismo sin ánimo de lucro que apoya y gestiona los proyectos e infraestructura de OWASP. La comunidad OWASP está formada por empresas, organizaciones educativas y particulares de todo el mundo. Juntos constituyen una comunidad de seguridad informática que trabaja para crear artículos, metodologías, documentación, herramientas y tecnologías que se liberan y pueden ser usadas gratuitamente por cualquiera.

Si bien no existe una única metodología, OWASP propone un enfoque por fases que puede adaptarse a las necesidades específicas de cada proyecto. En general, estas fases pueden incluir:

- Identificación de amenazas: Se utilizan diversas técnicas para identificar posibles vulnerabilidades en la aplicación, como el análisis de código estático y dinámico, pruebas de caja negra y blanca, y revisiones de configuración.
- Explotación de vulnerabilidades: Se intenta explotar las vulnerabilidades identificadas para comprender su gravedad y potencial impacto.
- Informe y recomendaciones: Se documenta el proceso de evaluación, las vulnerabilidades encontradas y se brindan recomendaciones para corregirlas.

## 4. RESUMEN EJECUTIVO

Tras el análisis realizado al activo web [REDACTED] sobre la empresa [REDACTED] se indica que el nivel de seguridad es mejorable ya que se han encontrado vulnerabilidades que afectan a la confidencialidad e integridad del activo.

El nivel de riesgo es considerado **medio**, ya que se han encontrado vulnerabilidades de severidad media que han permitido la inyección de código JavaScript que pueden llegar a implicar el robo de información a usuarios así como la inyección de código HTML pudiéndose llegar a emplearse en suplantaciones de identidad.

De igual forma, se reportan diversas vulnerabilidades de criticidad baja entre las cuales se observa, la ausencia de cabeceras de seguridad en las respuestas del servidor que pueden derivar en diversos ataques hacia el aplicativo como por el ejemplo suplantación de identidad. Por otro lado, se han encontrado varios problemas de seguridad en los comunicaciones debido al uso de cifrados considerados obsoletos que pueden llegar a implicar el descifrado de las comunicaciones seguras, así como la implementación incorrecta del cifrado que puede llevar a vulnerabilidades de tipo BREACH. Finalmente, se han hallado problemas en la configuración de las cookies que puede llegar a implicar el robo de las mismas a través de diversos ataques.

En la siguiente tabla se clasifican las 7 vulnerabilidades encontradas en función de su severidad:

| MEDIA | BAJA | TOTAL |
|-------|------|-------|
| 2     | 5    | 7     |

Tabla 2. Número de vulnerabilidades por severidad

#### 4.1. Resumen de vulnerabilidades

| Vulnerabilidad   | Riesgo | Ocurrencias |
|--|--------|-------------|
| Cross Site Scripting (XSS)   | Media  | 1           |
| Inyección HTML   | Media  | 1           |
| Ausencia de la cabecera X-Frame-Options – Vulnerabilidad ClicJacking | Baja   | 1           |
| Ausencia de cabeceras de seguridad.                                  | Baja   | 1           |
| Uso de cifrados TLS/SSL inseguros - Lucky 13.                        | Baja   | 1           |
| Uso de cifrados TLS/SSL inseguros Vulnerabilidad Breach              | Baja   | 1           |
| Cookies sin atributos de seguridad establecidos.                     | Baja   | 1           |

Tabla 3. Resumen de vulnerabilidades

#### 4.2. Recomendaciones generales

Finalizadas las pruebas, se indican una serie de recomendaciones generales ordenadas según la prioridad de mitigación para mejorar la seguridad del aplicativo web.

- **A corto plazo:**
  - Validar de forma correcta los campos de entrada de datos del aplicativo, para evitar la inyección de código JavaScript y HTML.
  - Implementar las cabeceras de seguridad HTTP en las respuestas del servidor.
- **A medio plazo:**
  - Evitar el uso de cifrados CBC obsoletos.
  - Implementar mecanismos de cifrado correctos para evitar los ataques de tipo BREACH.
  - Establecer los parámetros de seguridad en las cookies.
- **A largo plazo:**
  - Realizar de manera periódica pruebas de seguridad.



## 5. DETALLE TÉCNICO VULNERABILIDADES

En el presente apartado se recogen los detalles técnicos de cada una de las vulnerabilidades encontradas

### 5.1. CROSS SITE SCRIPTING - XSS

|                     |                                     |                 |     |           |       |
|---------------------|-------------------------------------|-----------------|-----|-----------|-------|
| ID                  |                                     | Puntuación CVSS | 4.2 | Severidad | Media |
| Vector CVSS         | AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:L/A:N |                 |     |           |       |
| Clasificación OWASP | A03:2021-Injection                  |                 |     |           |       |
| Sistema afectado    |                                     |                 |     |           |       |

#### 5.1.1. Descripción

La vulnerabilidad de Cross Site Scripting (XSS) permite a un atacante la inyección de código JavaScript/HTML malicioso en páginas web. Esta vulnerabilidad se manifiesta cuando el sitio no filtra adecuadamente las entradas del usuario, lo que permite la ejecución de scripts no autorizados en el navegador de los usuarios que visitan determinadas páginas, potencialmente comprometiendo la integridad y seguridad de la información.

#### 5.1.2. Impacto

Esta vulnerabilidad puede representar un impacto significativo en la seguridad del aplicativo web y sus usuarios. Los ataques XSS pueden permitir a un atacante robar sesiones de usuario, redirigir a los usuarios a sitios web maliciosos, modificar el contenido de la página web, o incluso robar información confidencial como contraseñas o cookies de sesión.

#### 5.1.3. Descripción extendida

Para la explotación de la vulnerabilidad se ha seguido el siguiente procedimiento.

Se ha descubierto que, si se declara una variable a continuación de la URL, esta es reflejada dentro del atributo HTML `"data-search-name"` dentro a su vez de la clase denominada `"SEOElement"`.



*Figura 1. Parámetro insertado en URL reflejado en HTML*

A continuación, si se añade un valor a la variable declarada anteriormente, en este caso denominado “encontrado”, se verá reflejado dentro del atributo “value” de la clase “SEOEElement”.

Resultando de la siguiente forma: “/?buscando=encontrado”.



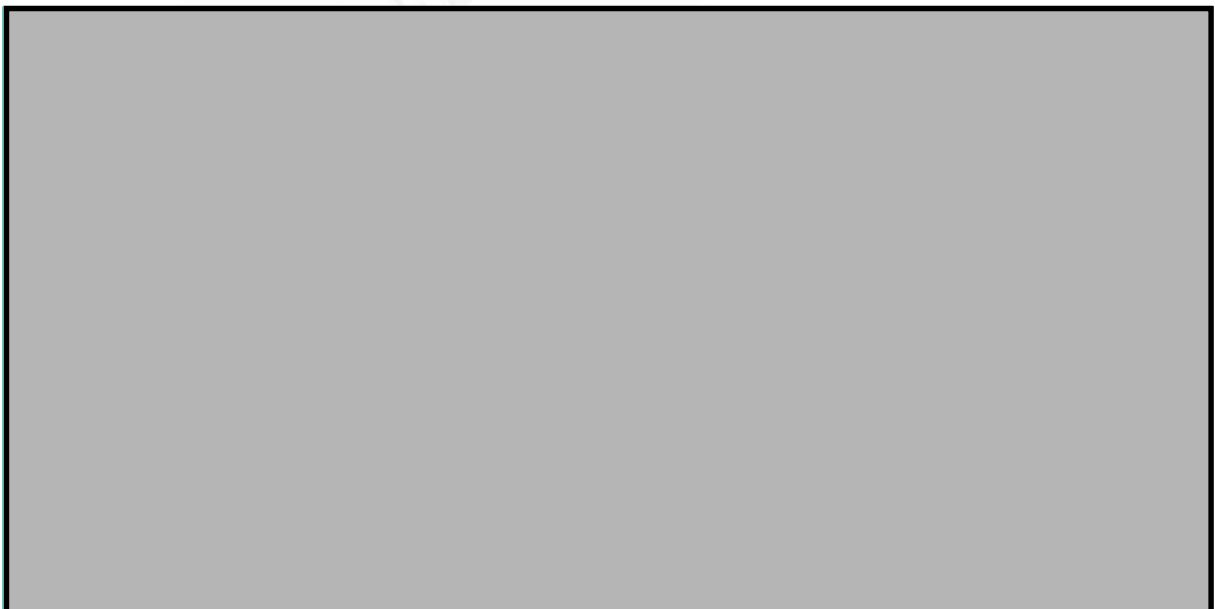
*Figura 2. Inserción de parámetro en variable*

Manteniéndose en la URL, si se modifican los parámetros introducidos añadiendo una comilla (") y un símbolo de mayor qué (>) a continuación del símbolo igual (=), se permite cerrar la clase `<input>`, extrayendo el código de la propia clase.



*Figura 3. Cierre de la clase `<input>`*

Una vez realizado esta comprobación, se procede a inyectar en el parámetro el código JavaScript `<script>alert(document.domain)</script>`. Esto permite que al cargar la página con la URL alterada el código introducido en la misma se ejecute, mostrando en este caso una alerta con el nombre del dominio como contenido.



*Figura 4. Inyección de código JavaScript exitoso*

#### **5.1.4. Recomendaciones**

Validar correctamente los datos de entrada en el servidor y hacer uso de una codificación de caracteres adecuada para evitar la inyección de cualquier tipo de código incluyendo JavaScript.

#### **5.1.5. Referencias**

- [https://owasp.org/Top10/A03\\_2021-Injection/](https://owasp.org/Top10/A03_2021-Injection/)

## 5.2. INYECCIÓN HTML

|                     |                                     |                 |     |           |       |
|---------------------|-------------------------------------|-----------------|-----|-----------|-------|
| ID                  |                                     | Puntuación CVSS | 4.2 | Severidad | Media |
| Vector CVSS         | AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:L/A:N |                 |     |           |       |
| Clasificación OWASP | A03:2021-Injection                  |                 |     |           |       |
| Sistema afectado    |                                     |                 |     |           |       |

### 5.2.1. Descripción

La inyección HTML es una técnica maliciosa en la que un atacante inserta código HTML no deseado en una página web

### 5.2.2. Impacto

Una inyección HTML puede permitir a los atacantes robar datos sensibles como contraseñas o información financiera, manipular el contenido de la página para difundir información falsa o engañosa, redirigir a los usuarios a sitios maliciosos para llevar a cabo ataques adicionales, y dañar la reputación de la organización al comprometer la seguridad y la confianza de los usuarios en el sitio web afectado.

### 5.2.3. Descripción extendida

Para la explotación de la vulnerabilidad se han seguido los siguientes pasos:

Se ha descubierto que, si se declara una variable a continuación de la URL, esta es reflejada dentro del atributo “data-search-name” de la clase denominada “SEOELEMENT” del código HTML.

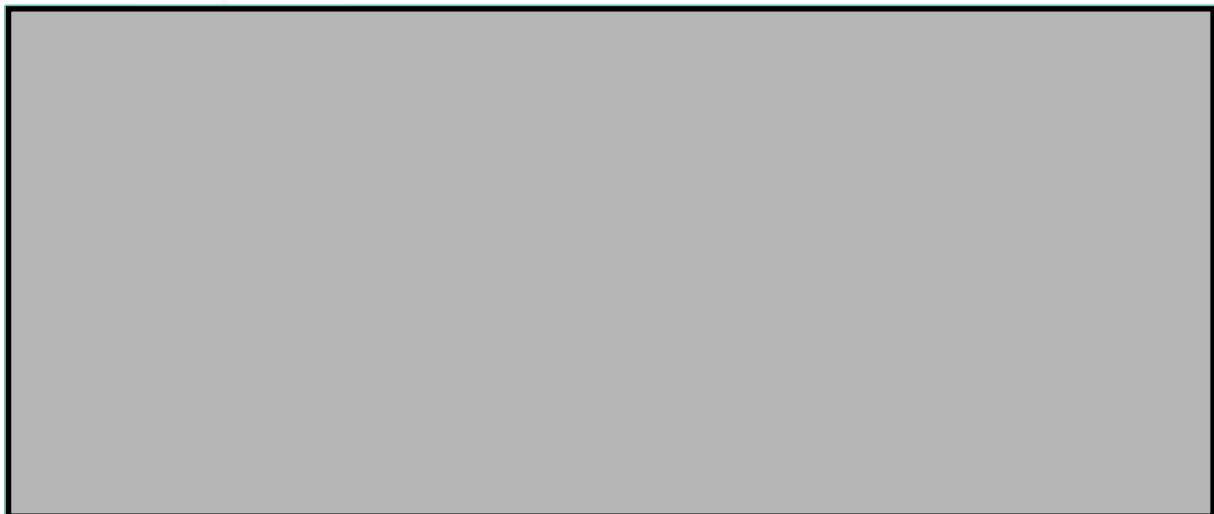


Figura 5. Introducción del parámetro “buscando” en la URL

A continuación, si se añade un valor a la variable declarada anteriormente, en este caso denominado “encontrado”, se verá reflejado dentro del atributo “*value*” de la clase “*SEElement*”. Resultando de la siguiente forma: “/?buscando=encontrado”.



*Figura 6. Introducción del segundo parámetro en la URL*

Manteniéndose en la URL, si se modifican los parámetros introducidos añadiendo una comilla (") y un símbolo de mayor que (>) a continuación del símbolo igual (=), se puede extraer de la entrada de usuario “<input>” el valor de su atributo “*value*”, facilitando la inserción de un nuevo código ejecutable.



*Figura 7. Extracción del valor del atributo "value"*

Una vez realizada la extracción, sustituyendo el parámetro “encontrado” por cualquier código HTML, en este caso un formulario “<form></form>”, permite que, al cargar la página con la URL alterada, el código introducido en la misma se ejecute, mostrando un formulario malicioso para obtener datos del usuario de manera ilícita.



*Figura 8. Inyección de formulario HTML*

#### **5.2.4. Recomendaciones**

Validar correctamente los datos de entrada en el servidor así como utilizar una codificación de caracteres adecuada para evitar la inyección de código HTML.

#### **5.2.5. Referencias**

- [https://owasp.org/Top10/A03\\_2021-Injection/](https://owasp.org/Top10/A03_2021-Injection/)

### 5.3. AUSENCIA DE LA CABECERA X-FRAME-OPTIONS - VULNERABILIDAD CLICKJACKING

|                     |   |                 |     |           |      |
|---------------------|---|-----------------|-----|-----------|------|
| ID                  |   | Puntuación CVSS | 3.1 | Severidad | Baja |
| Vector CVSS         | AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:N/A:N             |                 |     |           |      |
| Clasificación OWASP | A05: 2021-Configuración de Seguridad Incorrecta |                 |     |           |      |
| Sistema afectado    |   |                 |     |           |      |

#### 5.3.1. Descripción

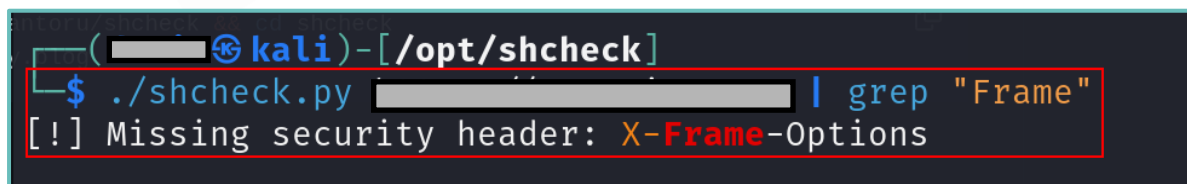
El clickjacking es una técnica maliciosa en la que los atacantes engañan a los usuarios para que hagan clic en elementos de una página web sin saberlo. Esto se logra superponiendo contenido malicioso sobre una página legítima, lo que puede llevar a acciones no deseadas como compras no autorizadas o divulgación de información confidencial.

#### 5.3.2. Impacto

Esta vulnerabilidad puede tener un impacto significativo al permitir que los atacantes realicen acciones no autorizadas en nombre del usuario, como realizar compras, divulgar información confidencial o incluso tomar el control total de la cuenta del usuario. Esto puede resultar en pérdidas financieras, robo de identidad y daños a la reputación del usuario o de la empresa afectada.

#### 5.3.3. Descripción extendida:

Haciendo uso de la herramienta "shcheck.py", se ha encontrado que la cabecera X-Frame-Options se encuentra ausente en el aplicativo.



```
(kali)-[/opt/shcheck]
$ ./shcheck.py [redacted] | grep "Frame"
[!] Missing security header: X-Frame-Options
```

Figura 9. Ausencia de la cabecera X-Frame-Options por ShCheck



Como prueba de concepto, se ha creado un código HTML en el que se ha insertado en una etiqueta “<iframe></iframe>” la URL de la web, junto con el demás código necesario.

```
3 <head>
6 <style>
8  iframe {
14  margin-top:100px;
15  }
16
17  form {
18  text-align:center;
19  margin:30px auto;
20  }
21 </style>
22 </head>
23
24 <body>
25
26  <form>
27    <label>Email:</label>
28    <input>
29    <br>
30    <label>Password:</label>
31    <input>
32    <br>
33    <button type="submit">CÓDIGO HTML MALIGNO</button>
34  </form>
35  <iframe src=" " "></iframe>
36 </body>
37 </html>
38
```

Figura 10. Código HTML para prueba de ClicJacking

Finalmente, tras ejecutar el fichero HTML en el navegador se aprecia como tiene lugar la explotación, insertándose el sitio web vulnerable a continuación del código malicioso diseñado:

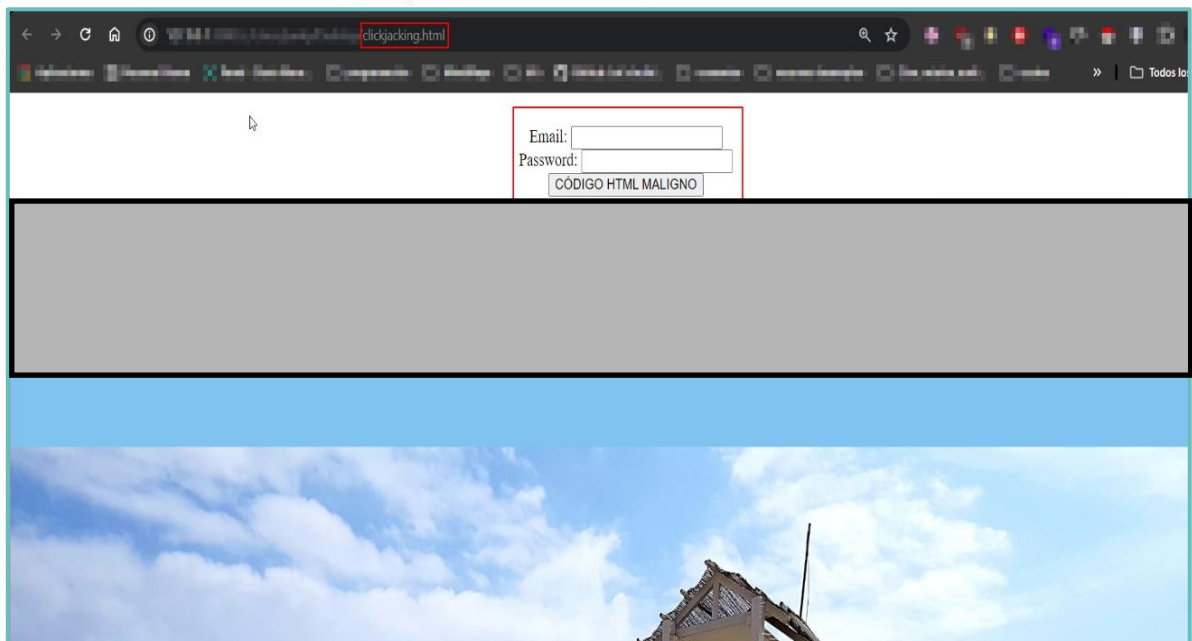


Figura 11. Explotación correcta de ClicJacking

#### 5.3.4. Recomendaciones

Añadir la cabecera X-Frame-Options a todas las respuestas HTTP de tu sitio web para indicar a los navegadores si deben permitir o no que tu contenido se incruste en iframes. Pueden establecerse los siguientes valores:

- X-Frame-Options: DENY: Impide que el contenido se incruste en iframes en otros sitios web.
- X-Frame-Options: SAMEORIGIN: Solo permite que el contenido se incruste en iframes en el mismo origen (dominio) que el sitio web.
- X-Frame-Options: ALLOW-FROM uri: Solo permite que el contenido se incruste en iframes en los sitios web especificados por la URI.

#### 5.3.5. Referencias

- [https://owasp.org/Top10/es/A05\\_2021-Security\\_Misconfiguration/](https://owasp.org/Top10/es/A05_2021-Security_Misconfiguration/)

## 5.4. AUSENCIA DE CABECERAS DE SEGURIDAD

|                     |   |                 |     |           |      |
|---------------------|---|-----------------|-----|-----------|------|
| ID                  |   | Puntuación CVSS | 3.1 | Severidad | Baja |
| Vector CVSS         | AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:N/A:N             |                 |     |           |      |
| Clasificación OWASP | A05: 2021-Configuración de Seguridad Incorrecta |                 |     |           |      |
| Sistema afectado    |   |                 |     |           |      |

### 5.4.1. Descripción

La ausencia de cabeceras de seguridad en una página web puede considerarse una vulnerabilidad significativa ya que se consideran componentes importantes de la configuración de seguridad de un sitio web y ayudan a proteger contra una gran variedad de amenazas y ataques potenciales.

### 5.4.2. Impacto

La falta de implementación de cabeceras de seguridad puede dejar a un sitio web vulnerable a una diversidad de ataques, como el robo de sesiones, la inyección de contenido malicioso o la suplantación de identidad. Esto puede resultar en la pérdida de datos sensibles, la comprometida integridad del sitio web y la confianza del usuario, así como posibles sanciones legales y daños a la reputación de la empresa.

### 5.4.3. Descripción extendida:

Haciendo uso de la herramienta "shcheck.py" se obtienen las cabeceras ausentes en la respuesta del servidor, las cuales son las siguientes:

- X-Frame-Options: Esta cabecera permite a un sitio web controlar si sus páginas pueden ser cargadas dentro de un marco (iframe) en otro sitio web. Ayuda a prevenir ataques de clickjacking al evitar que el contenido de un sitio web sea incrustado en un marco malicioso sin el conocimiento del usuario.
- X-Content-Type-Options: Esta cabecera se utiliza para controlar cómo los navegadores web interpretan el contenido de una página en términos de tipo MIME. Al establecer esta cabecera en "nosniff", se indica a los navegadores que no intenten adivinar o modificar el tipo de contenido del recurso, lo que ayuda a prevenir ataques de tipo MIME sniffing.
- Strict-Transport-Security (HSTS): Esta cabecera indica a los navegadores que solo se deben cargar recursos a través de conexiones seguras (HTTPS) y ayuda a prevenir ataques de tipo man-in-the-middle. Una vez que un navegador ha recibido esta cabecera, solo accederá al sitio web a través de HTTPS durante un período de tiempo especificado, incluso si el usuario intenta acceder a través de HTTP.

- Content-Security-Policy (CSP): Esta cabecera permite a los propietarios de sitios web especificar qué recursos pueden ser cargados y desde dónde, ayudando a prevenir ataques de inyección de código, como XSS (Cross-Site Scripting). CSP define una política de seguridad que restringe la ejecución de scripts, el uso de iframes, la carga de recursos externos y otras actividades potencialmente peligrosas.
- Referrer-Policy: Esta cabecera controla cómo se envía el encabezado Referer en las solicitudes HTTP, que indica al servidor de origen desde qué página se originó la solicitud. Esto ayuda a proteger la privacidad del usuario al limitar la cantidad de información de referencia que se comparte con otros sitios web.
- Permissions-Policy: Esta cabecera permite a los sitios web controlar y restringir el acceso a ciertas API del navegador, como la cámara, el micrófono y la geolocalización, ayudando a proteger la privacidad y la seguridad del usuario al limitar qué sitios pueden acceder a estos recursos.
- Cross-Origin-Embedder-Policy: Esta cabecera permite a los desarrolladores controlar cómo se comporta un recurso incrustado en un contexto de navegación cruzada (cross-origin). Ayuda a prevenir ataques de navegación cruzada bloqueando la incrustación de recursos de origen mixto en páginas HTTP.
- Cross-Origin-Resource-Policy: Esta cabecera permite a los sitios web controlar cómo se pueden compartir recursos entre diferentes orígenes (cross-origin). Ayuda a mitigar riesgos de seguridad al limitar el acceso de otros sitios a los recursos de tu sitio web.
- Cross-Origin-Opener-Policy: Esta cabecera permite a los sitios web controlar cómo se comporta una ventana o un marco al navegar entre orígenes diferentes (cross-origin). Ayuda a prevenir ataques de navegación cruzada y proteger la privacidad y la seguridad del usuario al limitar cómo se pueden compartir datos entre diferentes orígenes.

La siguiente evidencia muestra la ausencia de estas cabeceras en el activo:

```
C:\[redacted] python shcheck.py [redacted]

> shcheck.py - santoru .....

Simple tool to check security headers on a webserver

[*] Analyzing headers of [redacted]
[*] Effective URL: [redacted]
[!] Missing security header: X-Frame-Options
[!] Missing security header: X-Content-Type-Options
[!] Missing security header: Strict-Transport-Security
[!] Missing security header: Content-Security-Policy
[!] Missing security header: Referrer-Policy
[!] Missing security header: Permissions-Policy
[!] Missing security header: Cross-Origin-Embedder-Policy
[!] Missing security header: Cross-Origin-Resource-Policy
[!] Missing security header: Cross-Origin-Opener-Policy

[!] Headers analyzed for [redacted]
[+] There are 0 security headers
[-] There are not 9 security headers
```

Figura 12. Cabeceras ausentes

Como prueba de concepto para la ausencia de la cabecera Content-Security-Policy se ha y aprovechando la vulnerabilidad de inyección de código HTML, se ha ejecutado el siguiente código, que carga un recurso externo, en concreto una imagen, de un servidor externo, dentro de la propia web vulnerable.

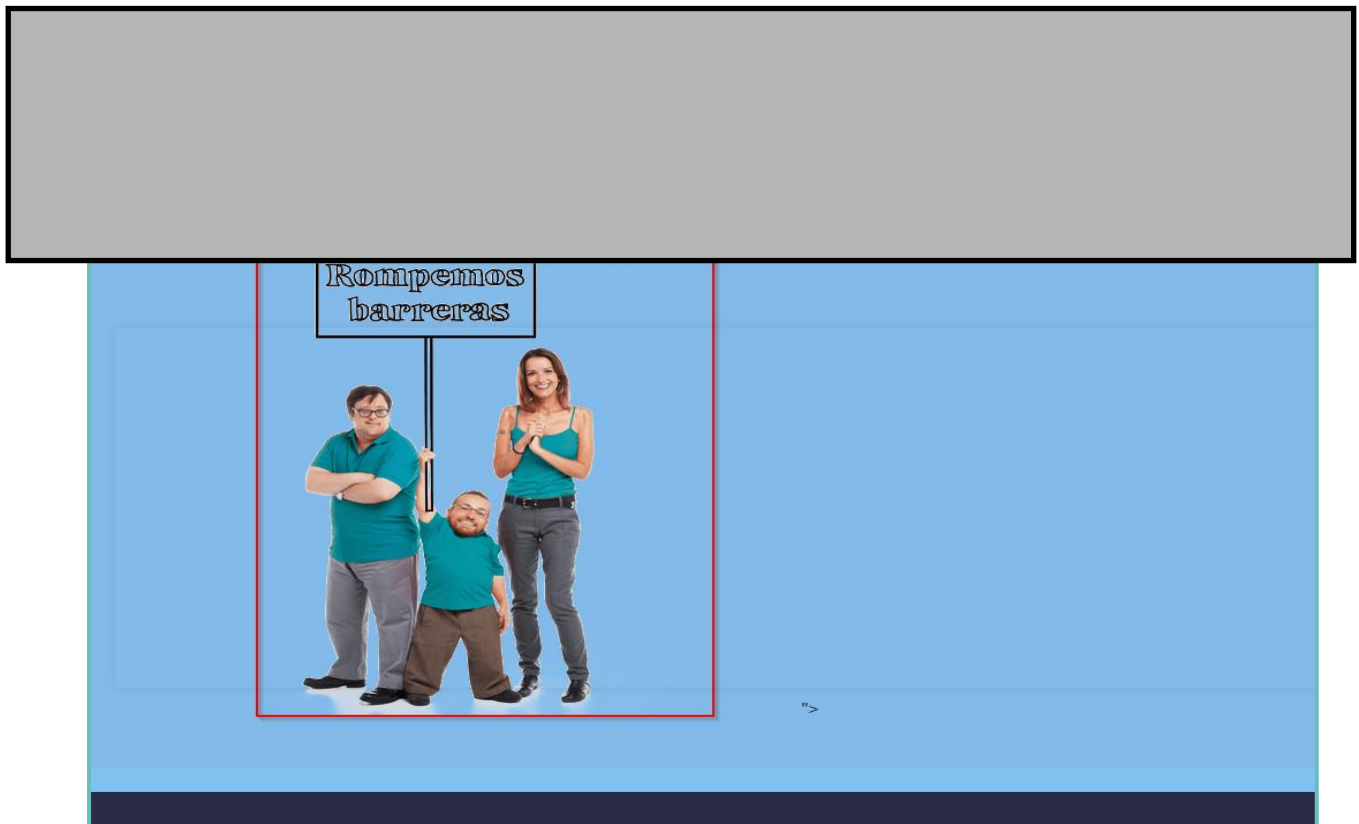


Figura 13. Recurso externo cargado dentro del aplicativo

#### 5.4.4. Recomendaciones

Se recomienda implementar las cabeceras de seguridad ausentes con los siguientes parámetros:

- **X-Frame-Options:** Se recomienda configurarla con el valor "DENY" para evitar que el contenido de tu sitio web sea incrustado en un marco en otro sitio.
- **X-Content-Type-Options:** Establecer su valor en "nosniff" para prevenir que el navegador adivine incorrectamente el tipo de contenido, lo que podría conducir a ataques de tipo MIME sniffing.
- **Strict-Transport-Security (HSTS):** Habilitarla con un período de tiempo largo, como "max-age=31536000", para asegurar que las comunicaciones solo se realicen a través de HTTPS y prevenir ataques de intermediarios.
- **Content-Security-Policy (CSP):** Definir una política de seguridad adecuada que especifique qué recursos pueden ser cargados y desde dónde, ayudando a prevenir ataques de inyección de código como XSS.
- **Referrer-Policy:** Se sugiere configurarla en "strict-origin-when-cross-origin" para controlar cómo se envía el encabezado Referer en las solicitudes HTTP, protegiendo así la privacidad del usuario.

- **Permissions-Policy:** Definir las políticas de permisos según las necesidades de tu sitio web, restringiendo el acceso a ciertas API del navegador y protegiendo la privacidad y seguridad del usuario.
- **Cross-Origin-Embedder-Policy:** Configurar con el valor "require-corp" para controlar cómo se comportan los recursos incrustados en un contexto de navegación cruzada, previniendo así ataques de navegación cruzada.
- **Cross-Origin-Resource-Policy:** Establecer en "same-origin" para controlar cómo se pueden compartir recursos entre diferentes orígenes y mitigar riesgos de seguridad.
- **Cross-Origin-Opener-Policy:** Definir con el valor "same-origin" para controlar cómo se comportan las ventanas o marcos al navegar entre diferentes orígenes, protegiendo así la privacidad y seguridad del usuario.

#### 5.4.5. Referencias

- [https://owasp.org/Top10/es/A05\\_2021-Security\\_Misconfiguration/](https://owasp.org/Top10/es/A05_2021-Security_Misconfiguration/)

## 5.5. USO DE CIFRADOS TLS/SSL INSEGUROS - LUCKY13

|                     |                                     |                 |     |           |      |
|---------------------|-------------------------------------|-----------------|-----|-----------|------|
| ID                  |                                     | Puntuación CVSS | 3.1 | Severidad | Baja |
| Vector CVSS         | AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:N/A:N |                 |     |           |      |
| Clasificación OWASP | A02:2021-Fallos criptográficos      |                 |     |           |      |
| Sistema afectado    |                                     |                 |     |           |      |

### 5.5.1. Descripción

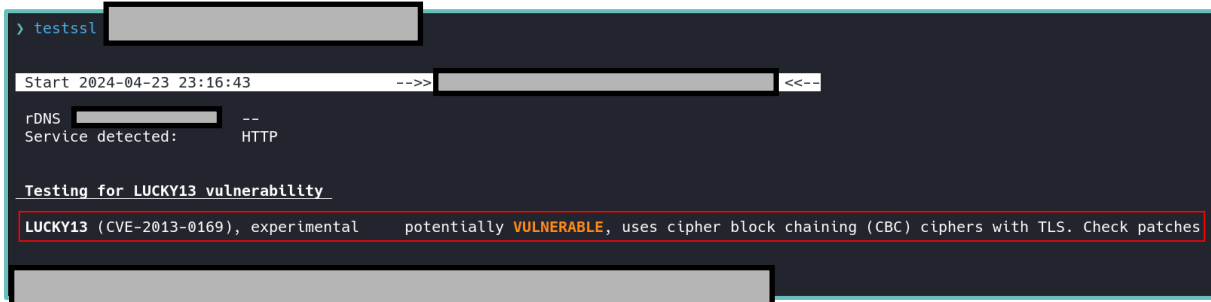
La vulnerabilidad "Lucky13" afecta a los cifrados inseguros que utilizan el modo de cifrado de bloque (CBC), como SSL/TLS. Se aprovecha de una debilidad en la implementación del "padding" utilizado en CBC, permitiendo a un atacante realizar ataques de canal lateral para deducir información sensible, como claves de cifrado, a través del tiempo que toma el servidor en responder a las solicitudes.

### 5.5.2. Impacto

La vulnerabilidad Lucky13 puede tener un impacto significativo en la seguridad de los sistemas que utilizan cifrados inseguros CBC en protocolos como SSL/TLS. Los posibles impactos incluyen la revelación de datos confidenciales, como claves de cifrado, la manipulación de la comunicación segura y la posibilidad de realizar ataques de canal lateral para deducir información sensible. Esto puede resultar en la comprometida confidencialidad e integridad de los datos transmitidos, lo que afecta tanto a la privacidad como a la seguridad de los sistemas y la información protegida.

### 5.5.3. Descripción extendida

Haciendo uso de la herramienta "testssl" ha sido posible enumerar los cifrados CBC en uso lo que hace que el servidor web que almacena el aplicativo sea potencialmente vulnerable a Lucky13



```

> testssl [redacted]
Start 2024-04-23 23:16:43 --> [redacted] <--
rDNS [redacted] --
Service detected: HTTP

Testing for LUCKY13 vulnerability
LUCKY13 (CVE-2013-0169), experimental potentially VULNERABLE, uses cipher block chaining (CBC) ciphers with TLS. Check patches
  
```

Figura 14. Servidor potencialmente vulnerable a Lucky13

La siguiente evidencia muestra los cifrados CBC en uso lo conlleva a que pueden ser vulnerables a posibles descifrados de información.

```
> testssl --lucky13 --single-cipher CBC [redacted]

Start 2024-04-23 23:18:05 --> [redacted] <<--

rDNS ([redacted]): --
Service detected: HTTP

Testing ciphers with matching number pattern "CBC"

Hexcode Cipher Suite Name (OpenSSL) KeyExch. Encryption Bits Cipher Suite Name (IANA/RFC)
-----
xc028 ECDHE-RSA-AES256-SHA384 ECDH 256 AES 256 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
xc014 ECDHE-RSA-AES256-SHA ECDH 256 AES 256 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
x6b DHE-RSA-AES256-SHA256 DH 2048 AES 256 TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
x39 DHE-RSA-AES256-SHA DH 2048 AES 256 TLS_DHE_RSA_WITH_AES_256_CBC_SHA
x3d AES256-SHA256 RSA AES 256 TLS_RSA_WITH_AES_256_CBC_SHA256
x35 AES256-SHA RSA AES 256 TLS_RSA_WITH_AES_256_CBC_SHA
xc027 ECDHE-RSA-AES128-SHA256 ECDH 256 AES 128 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
xc013 ECDHE-RSA-AES128-SHA ECDH 256 AES 128 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
x67 DHE-RSA-AES128-SHA256 DH 2048 AES 128 TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
x33 DHE-RSA-AES128-SHA DH 2048 AES 128 TLS_DHE_RSA_WITH_AES_128_CBC_SHA
x3c AES128-SHA256 RSA AES 128 TLS_RSA_WITH_AES_128_CBC_SHA256
x2f AES128-SHA RSA AES 128 TLS_RSA_WITH_AES_128_CBC_SHA

Done 2024-04-23 23:18:29 [ 33s ] --> [redacted] <<--
```

Figura 15. Cifrados CBC usados por el servidor

#### 5.5.4. Recomendaciones

Deshabilitar el uso de cifrados de tipo CBC.

#### 5.5.5. Referencias

- [https://owasp.org/www-project-web-security-testing-guide/v41/4-Web\\_Application\\_Security\\_Testing/09-Testing\\_for\\_Weak\\_Cryptography/01-Testing\\_for\\_Weak\\_SSL\\_TLS\\_Ciphers\\_Insufficient\\_Transport\\_Layer\\_Protection](https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/09-Testing_for_Weak_Cryptography/01-Testing_for_Weak_SSL_TLS_Ciphers_Insufficient_Transport_Layer_Protection)



## 5.6. USO DE CIFRADOS TLS/SSL INSEGUROS - BREACH

|                     |                                     |                 |     |           |      |
|---------------------|-------------------------------------|-----------------|-----|-----------|------|
| ID                  |                                     | Puntuación CVSS | 3.1 | Severidad | Baja |
| Vector CVSS         | AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:N/A:N |                 |     |           |      |
| Clasificación OWASP | A02:2021-Fallos criptográficos      |                 |     |           |      |
| Sistema afectado    |                                     |                 |     |           |      |

### 5.6.1. Descripción

La vulnerabilidad Breach es un ataque de canal lateral que explota la implementación de almacenamiento de datos en el navegador, como cookies de sesión, para deducir información confidencial a través del tiempo de carga de recursos de terceros. La compresión Gzip puede aumentar el riesgo al reducir el tiempo de carga de los recursos, facilitando la medición del tiempo por parte de los atacantes para deducir información sensible.

### 5.6.2. Impacto

Esta vulnerabilidad puede comprometer la privacidad del usuario al exponer información confidencial, como identidad o datos de sesión. Puede suponer un impacto significativo, permitiendo a los atacantes acceder a datos sensibles y potencialmente causar daños financieros o pérdida de reputación para la víctima o la empresa afectada. La compresión Gzip puede amplificar este impacto al acelerar el proceso de deducción de información sensible, aumentando así el riesgo para los usuarios y las organizaciones.

### 5.6.3. Descripción extendida

Haciendo uso de la herramienta “testssl” se observa que el servidor es potencialmente vulnerable a “BREACH ya que detecta la compresión “gzip” por HTTP.

```

Start 2024-04-25 20:05:00 --> <---
rDNS (213.13.156.134): --
Service detected: HTTP

Testing for BREACH (HTTP compression) vulnerability

BREACH (CVE-2013-3587) potentially NOT ok, "gzip" HTTP compression detected. - only supplied "/" tested
Can be ignored for static pages or if no secrets in the page

Done 2024-04-25 20:05:09 [ 13s] --> <---
```

Figura 16. Activo vulnerable a BREACH

Para verificar que el servidor acepta la compresión Gzip, se ha empelado la herramienta BurpSuite, y realizando una petición con la cabecera “*Accept-Encoding: compress, gzip*” se indica al servidor que aceptamos la respuesta en formato comprimido.



Figura 17. Petición con *Accept-Encoding: compress, gzip*

Para facilitar la recepción de esta petición se deshabilita la configuración “*unpack compressed responses*”.

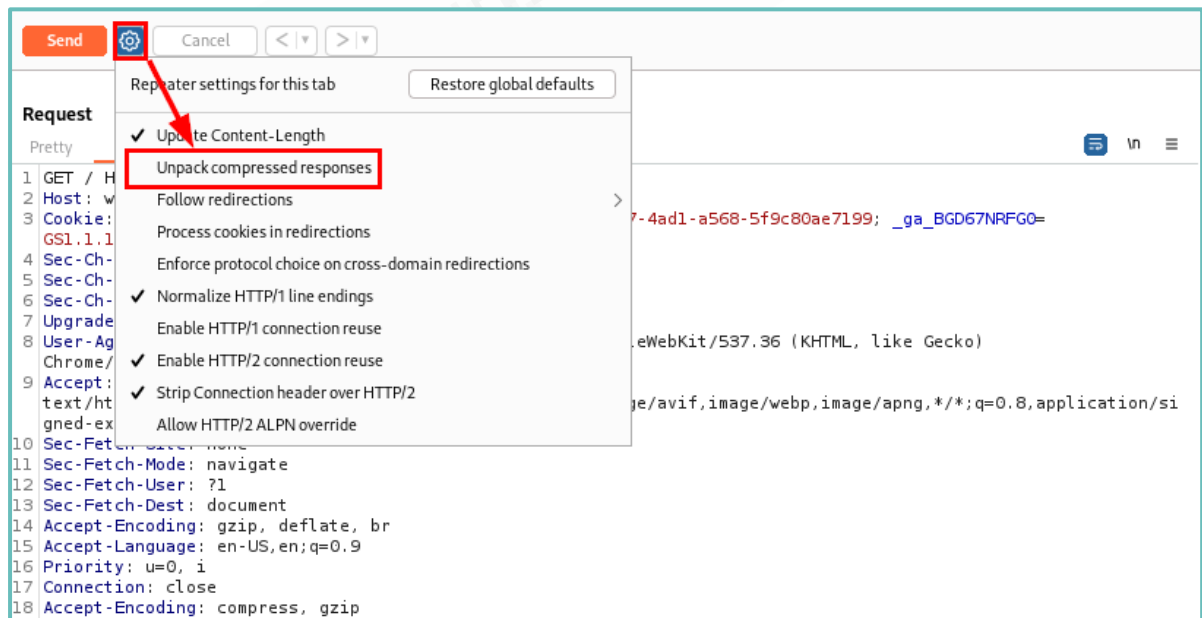


Figura 18. Configuración de BurpSuite para recibir peticiones comprimidas

Enviando la petición se observa cómo la respuesta del servidor se obtiene en formato comprimido, por lo que se verifica que el servidor acepta la compresión.

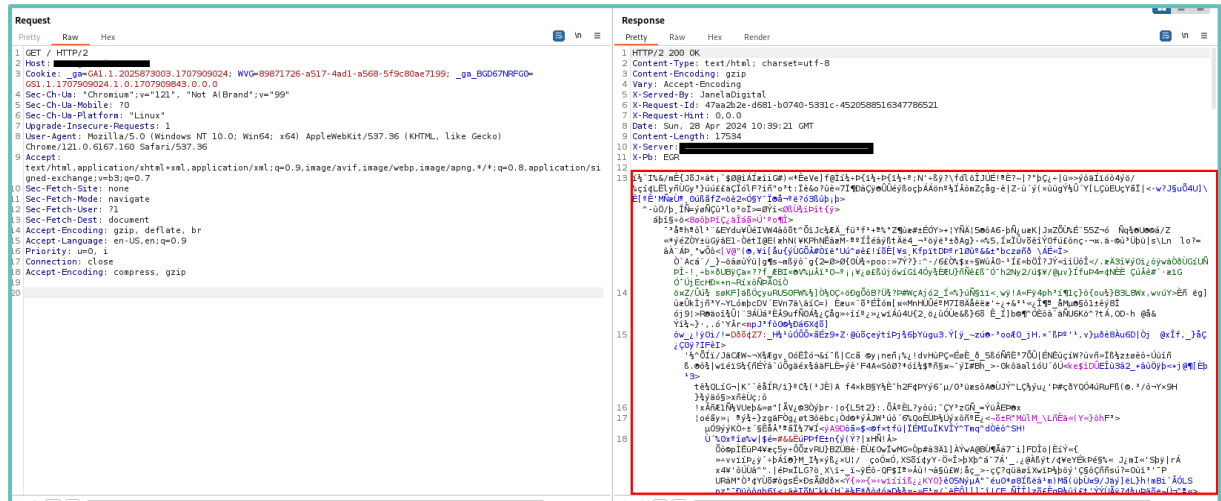


Figura 19. Respuesta del servidor comprimida



### 5.6.4. Recomendaciones

Deshabilitar la compresión Gzip en las opciones del servidor.

### 5.6.5. Referencias

- <https://breachattack.com/>
- [https://owasp.org/www-project-web-security-testing-guide/v41/4-Web\\_Application\\_Security\\_Testing/09-Testing\\_for\\_Weak\\_Cryptography/01-Testing\\_for\\_Weak\\_SSL\\_TLS\\_Ciphers\\_Insufficient\\_Transport\\_Layer\\_Protecti](https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/09-Testing_for_Weak_Cryptography/01-Testing_for_Weak_SSL_TLS_Ciphers_Insufficient_Transport_Layer_Protecti) on

## 5.7. COOKIES SIN PARÁMETROS DE SEGURIDAD ESTABLECIDOS

|                     |  |                 |     |           |      |
|---------------------|--|-----------------|-----|-----------|------|
| ID                  |   | Puntuación CVSS | 3.1 | Severidad | Baja |
| Vector CVSS         | AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:N/A:N  |                 |     |           |      |
| Clasificación OWASP | A05: 2021-Configuración de Seguridad Incorrecta                                    |                 |     |           |      |
| Sistema afectado    |  |                 |     |           |      |

### 5.7.1. Descripción

La ausencia de parámetros de seguridad en las cookies como Secure, HttpOnly o SameSite, puede permitir a los atacantes el acceso a las mismas o a su manipulación, lo que podría llevar a ataques como el robo de sesiones, la suplantación de identidad o la divulgación de información confidencial.

### 5.7.2. Impacto

La falta de parámetros de seguridad en las cookies puede resultar en el robo de sesiones, suplantación de identidad y divulgación de información confidencial, lo que pone en riesgo la privacidad y seguridad de los usuarios y puede tener repercusiones financieras y de reputación para las empresas afectadas. Cada parámetro de seguridad mal configurado puede implicar un riesgo, entre ellos:

- **HttpOnly:** Cuando esta configuración está en False, las cookies pueden ser accedidas a través de scripts del lado del cliente (como JavaScript). Esto aumenta el riesgo de que las cookies sean robadas o manipuladas por atacantes maliciosos, ya que pueden ser accedidas fácilmente a través de vulnerabilidades de XSS (Cross-Site Scripting).
- **Secure:** Al establecer Secure en False, las cookies pueden ser enviadas a través de conexiones no seguras (HTTP), lo que las hace vulnerables a ataques de tipo man-in-the-middle, donde un atacante puede interceptar y manipular el tráfico de red para robar o modificar las cookies.
- **SameSite:** Al establecer SameSite en None, las cookies no están restringidas en su envío desde un sitio web a otro, lo que aumenta el riesgo de ataques de CSRF (Cross-Site Request Forgery), donde un atacante puede aprovechar las cookies del usuario para realizar acciones no autorizadas en su nombre en otros sitios web.

### 5.7.3. Descripción Extendida

La siguiente evidencia recoge las cookies generadas por la web, que no implementa ninguno de los parámetros de seguridad.

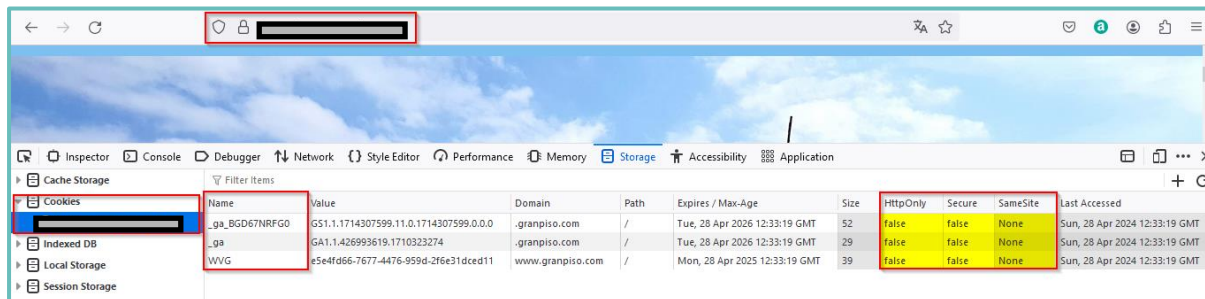


Figura 20. Cookies sin parámetros de seguridad establecidos

### 5.7.4. Recomendación:

Se recomienda establecer los siguientes parámetros:

- **Establecer HttpOnly:** Configura todas las cookies como HttpOnly para evitar que sean accedidas a través de scripts del lado del cliente (como JavaScript). Esto ayuda a prevenir ataques de XSS (Cross-Site Scripting) al limitar el acceso a las cookies solo desde el lado del servidor.
- **Habilitar Secure:** Configura todas las cookies como Secure para garantizar que solo se envíen a través de conexiones seguras (HTTPS). Esto ayuda a proteger las cookies contra ataques de tipo man-in-the-middle al garantizar que se transmitan de forma segura entre el navegador del usuario y el servidor.
- **Configurar SameSite:** Establece el atributo SameSite en "Strict" o "Lax" para todas las cookies según sea necesario. Esto ayuda a mitigar el riesgo de ataques de CSRF (Cross-Site Request Forgery) al restringir el envío de cookies a otros sitios web. Utiliza "Strict" para cookies sensibles que no deben ser enviadas en solicitudes de origen cruzado y "Lax" para cookies que pueden ser enviadas en solicitudes de origen cruzado, como aquellas utilizadas para seguimiento de análisis.
- **Revisar las políticas de cookies:** Realiza una revisión exhaustiva de todas las cookies utilizadas en tu sitio web y asegúrate de que estén configuradas correctamente en cuanto a HttpOnly, Secure y SameSite. Identifica y elimina cualquier cookie que no sea necesaria para el funcionamiento del sitio o que represente un riesgo de seguridad.

### 5.7.5. Referencias

- [https://owasp.org/www-chapter-london/assets/slides/OWASPLondon20171130\\_Cookie\\_Security\\_Myths\\_Misc\\_onceptions\\_David\\_Johansson.pdf](https://owasp.org/www-chapter-london/assets/slides/OWASPLondon20171130_Cookie_Security_Myths_Misc_onceptions_David_Johansson.pdf)

## 6. ANEXO I: CVSS: SISTEMA DE PUNTUACIÓN

A lo largo del informe se presentan diversas puntuaciones numéricas que pretenden cuantificar distintos elementos como la gravedad de las vulnerabilidades, el nivel de seguridad de cada una de las áreas analizadas o el nivel de seguridad global.

### 6.1. Puntuación de Vulnerabilidades

La gravedad de cada vulnerabilidad se determina utilizando la puntuación base del estándar CVSSv3 (Common Vulnerability Score System). Este estándar ofrece una manera de capturar las características clave de una vulnerabilidad y asignarle una puntuación numérica que refleje su severidad, junto con una descripción textual. Esta puntuación numérica puede luego traducirse en una representación cualitativa (como baja, media, alta y crítica), lo que ayuda a las organizaciones a evaluar y priorizar sus procesos de gestión de vulnerabilidades. La puntuación considera métricas específicas de la vulnerabilidad, independientemente del momento o el entorno en que se encuentre.

Estas métricas están clasificadas en los siguientes tres grupos:

#### 6.1.1. Métricas de explotabilidad

Las métricas de explotabilidad determinan las probabilidades de que una vulnerabilidad sea explotada. A continuación, se presentan las diversas métricas que componen este grupo, junto con sus posibles valores:

| Código | Descripción  |
|--------|--|
| AV     | <b>Vector de ataque</b><br>Esta métrica determina cómo puede ser explotada esta vulnerabilidad, y mide los requisitos de accesibilidad tanto físicos como lógicos que se requieren para la explotación satisfactoria de la vulnerabilidad. Los valores de esta métrica son: <ul style="list-style-type: none"><li>• Network (N): Requiere visibilidad a nivel de red (capa 3).</li><li>• Adjacent (A): Requiere visibilidad a nivel de enlace (capa 2).</li><li>• Local (L): Requiere acceso previo no privilegiado.</li><li>• Physical (P): Requiere acceso físico.</li></ul> |
|        | <b>Complejidad del Ataque</b><br>Esta métrica determina la complejidad de ataque requerida para hacer uso de la vulnerabilidad. Los valores de esta métrica son: <ul style="list-style-type: none"><li>• Low (L): No se requieren condiciones o circunstancias especiales para explotar la vulnerabilidad.</li><li>• High (H): El éxito de un ataque depende de que se cumplan ciertas condiciones o circunstancias. Por ejemplo, una vulnerabilidad que requiere el conocimiento previo de cierta información o configuraciones del sistema.</li></ul>                        |

| Código | Descripción   |
|--------|---|
| PR     | <b>Privilegios requeridos</b><br>Esta métrica determina el nivel de privilegios que un atacante debe tener antes poder explotar se forma satisfactoria una vulnerabilidad. Los valores de esta métrica son: <ul style="list-style-type: none"> <li>• None (N): El atacante no requiere de ningún tipo de privilegio para explotar la vulnerabilidad de forma satisfactoria.</li> <li>• Low (L): El atacante requiere de privilegios mínimos (por ejemplo, acceso con un usuario básico) para explotar la vulnerabilidad de forma satisfactoria.</li> <li>• High (H): El atacante requiere de privilegios elevados (por ejemplo, acceso con un usuario administrador) para explotar la vulnerabilidad de forma satisfactoria.</li> </ul> |
|        | <b>Interacción del usuario</b><br>Esta métrica determina si es necesaria la intervención del usuario para la explotación satisfactoria de la vulnerabilidad. Los niveles de esta métrica son: <ul style="list-style-type: none"> <li>• None (N): La vulnerabilidad puede explotarse si ser necesaria la iteración por parte del usuario.</li> <li>• Required (R): La explotación de la vulnerabilidad requiere que el usuario lleve a cabo determinadas acciones.</li> </ul>  |

Tabla 4. Métricas de explotabilidad

### 6.1.2. Alcance (Scope)

El alcance (Scope) de una vulnerabilidad determina si su explotación puede afectar a otros recursos o componentes más allá del sistema o la aplicación que sufre la vulnerabilidad.

| Código | Descripción   |
|--------|---|
| S      | <b>Scope</b><br>Esta métrica determina si la explotación satisfactoria de la vulnerabilidad puede afectar indirectamente a otros componentes fuera del alcance del sistema o aplicación con la vulnerabilidad. Los valores de esta métrica son los siguientes: <ul style="list-style-type: none"> <li>• Unchanged (U): El componente vulnerable y el componente afectado por la explotación de la vulnerabilidad es el mismo.</li> <li>• Changed (C): El componente vulnerable y el componente afectado por la explotación de la vulnerabilidad son distintos.</li> </ul> |
|        |   |

Tabla 5. Alcance

### 6.1.3. Métricas de impacto

Las métricas de impacto evalúan las consecuencias de explotar una vulnerabilidad. A continuación, se describen las diversas métricas que componen este grupo, junto con sus posibles valores.

| Código   | Descripción   |
|----------|---|
| <b>C</b> | <b>Impacto en la confidencialidad</b><br>La confidencialidad es la propiedad de un documento, mensaje o dato que únicamente está autorizado para ser leído o entendido por algunas personas o entidades. Los valores de esta métrica son los siguientes: <ul style="list-style-type: none"> <li>High (H): El compromiso de información confidencial (passwords, claves de cifrado, etc.) es total.</li> <li>Low (L): El compromiso de información confidencial es parcial (se obtiene cierta información, pero sin que el atacante tenga control sobre qué información puede obtener).</li> <li>None (N): No hay pérdida de confidencialidad.</li> </ul>                            |
| <b>I</b> | <b>Impacto en la integridad</b><br>La integridad es la propiedad de un documento, mensaje o dato que garantiza la veracidad de la información. Los valores de esta métrica son los siguientes: <ul style="list-style-type: none"> <li>High (H): Hay una pérdida total de integridad. Por ejemplo, un atacante puede modificar ficheros o información valiosa.</li> <li>Low (L): Hay una pérdida parcial de integridad. Por ejemplo, un atacante puede modificar ficheros o información, pero sin tener el control sobre qué información puede modificar.</li> <li>None (N): No hay pérdida de integridad.</li> </ul>  |
| <b>A</b> | <b>Impacto en la disponibilidad</b><br>La disponibilidad es la propiedad de un sistema, servicio o aplicación que es accesible sin impedimentos. Los valores de esta métrica son los siguientes: <ul style="list-style-type: none"> <li>High (H): Hay una pérdida total de disponibilidad. Por ejemplo, un atacante puede denegar totalmente el acceso a un recurso por parte de usuarios legítimos.</li> <li>Low (L): Hay una pérdida parcial de disponibilidad. Por ejemplo, un atacante puede degradar el servicio, pero no llega a denegar totalmente el acceso al recurso por parte de los usuarios legítimos.</li> <li>None (N): No hay pérdida de disponibilidad.</li> </ul> |

Tabla 6. Métricas de impacto



## 6.2. Categorización

Todas las vulnerabilidades están acompañadas de su vector CVSSv3, que contiene las cifras base calculadas. El vector de cálculo utilizado en CVSSv3 tiene el siguiente formato:

**(AV:[P,L,A,N]/AC:[H,L]/PR:[H,L,N]/UI(N,R)/S:[U,C]/C:[N,L,H]/I:[N,L,H]/A:[N,L,H]).**

Cada vulnerabilidad y recomendación se clasifican utilizando los valores base CVSS v3 derivados de las diferentes métricas. A continuación, se presentan los niveles de criticidad para diferentes rangos de valores base, lo que permite a la organización desarrollar un plan de acción efectivo y abordar primero las vulnerabilidades más críticas.

| Nivel de Criticidad | Descripción   |
|---------------------|---|
| <b>Crítico</b>      | <p><b>Vulnerabilidades con un CVSS superior a 9.0</b></p> <p>Este tipo de vulnerabilidades suelen permitir un compromiso total en la confidencialidad, integridad o la disponibilidad de los datos soportados por la aplicación y su explotación requiere de un nivel de esfuerzo o de privilegio bajo, por lo que es importante establecer medidas correctivas de inmediato.</p>                       |
| <b>Alto</b>         | <p><b>Vulnerabilidades cuyo CVSS se encuentre entre 7.0 y 8.9</b></p> <p>Este tipo de vulnerabilidades suelen suponer un riesgo elevado para la confidencialidad, integridad o la disponibilidad de los datos soportados por la aplicación y su explotación puede requerir un nivel de esfuerzo o de privilegio bajo o medio, por lo que es importante establecer medidas correctivas de inmediato.</p> |
| <b>Medio</b>        | <p><b>Vulnerabilidades cuyo CVSS se encuentre entre 4.0 y 6.9</b></p> <p>Este tipo de vulnerabilidades suelen suponer un riesgo para confidencialidad, integridad o la disponibilidad de los datos soportados por la aplicación y su explotación puede requerir un nivel de esfuerzo o de privilegio medio o alto, por lo que es importante establecer medidas correctivas.</p>                         |
| <b>Bajo</b>         | <p><b>Vulnerabilidades cuyo CVSS se encuentre entre 0.1 y 3.9</b></p> <p>Este tipo de vulnerabilidades suelen suponer, en situaciones determinadas, un riesgo para la confidencialidad, integridad o la disponibilidad de los datos soportados por la aplicación y su explotación puede requerir un nivel de esfuerzo o de privilegio alto.</p>   |
| <b>Informativo</b>  | <p><b>Vulnerabilidades con un CVSS igual a 0</b></p> <p>Este tipo de vulnerabilidades no suelen suponer un riesgo. En muchos casos se trata de posibles mejoras para la seguridad, pero no es estrictamente necesaria su aplicación.</p>  |

Tabla 7. Categorización CVSS

# **BLUE SHIELD TEAM**



**Instalación y  
configuración de un  
SIEM**

# ÍNDICE

|      |  |   |
|------|--|---|
| 1.   | Instalación de las herramientas de Administración de directivas de grupo ..... | 3 |
| 1.1. | ¿Por qué instalar una GPO? .....   | 4 |
| 1.2. | ¿En qué consiste esta GPO? .....   | 4 |
| 2.   | Necesidad de un Backup .....   | 5 |
| 2.1. | ¿Por qué es necesaria una copia de seguridad? .....                            | 5 |
| 2.2. | Metodología 3-2-1 .....  | 6 |
| 3.   | Instalación de SIEM en Windows Server .....                                    | 6 |
| 3.1. | ¿Por qué se hace necesario el uso de un SIEM? .....                            | 7 |
| 3.2. | Instalación de Splunk .....  | 8 |

# ÍNDICE DE FIGURAS

Figura 1. Selección de roles..... 3

Figura 2. Configuración de GPO ..... 4

Figura 3. Comienzo de instalación de Splunk ..... 8

Figura 4. Proceso de instalación de Splunk ..... 8

Figura 5. Panel principal de Splunk..... 9

Figura 6. Panel de administración de Splunk ..... 9

Figura 7. Consola de monitorización Splunk ..... 10

## 1. Instalación de las herramientas de Administración de directivas de grupo

Para crear y configurar objetos de directiva de grupo (GPO), debe instalar las herramientas de Administración de directivas de grupo. Estas herramientas se pueden instalar como una característica en Windows Server. Para obtener más información sobre cómo instalar las herramientas administrativas en un cliente de Windows, consulte cómo se instalan las herramientas de administración remota del servidor (RSAT).

1. Inicie sesión en la máquina virtual de administración. Para los pasos sobre cómo conectarse usando el centro de administración de Microsoft Entra, consulte Conectarse a una máquina virtual de Windows Server.
2. Administrador del servidor debería abrirse de forma predeterminada al iniciar sesión en la máquina virtual. Si no es así, en el menú Inicio, seleccione Administrador del servidor.
3. En el Panel de información de la ventana Administrador del servidor, seleccione Agregar roles y características.
4. En la página Antes de comenzar del Asistente para agregar roles y características, seleccione Siguiente.
5. En Tipo de instalación, deje activada la opción Instalación basada en características o en roles y seleccione Siguiente.
6. En la página Selección de servidor, elija la máquina virtual actual del grupo de servidores, por ejemplo mivm.aaddscontoso.com, y seleccione Siguiente.
7. En la página Roles de servidor, haga clic en Siguiente. En la página Características, seleccione la Administración de directivas de grupo.

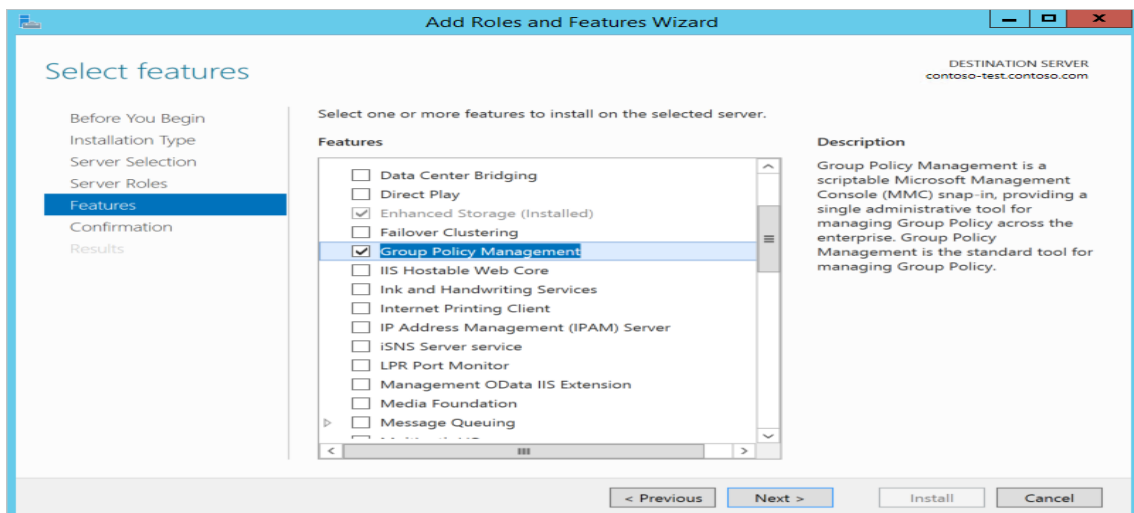


Figura 1. Selección de roles

8. En la página Confirmación, seleccione Instalar. Las herramientas de Administración de directivas de grupo pueden tardar un minuto o dos en instalarse.
9. Cuando finalice la instalación de la característica, haga clic en Cerrar para salir del Asistente para Agregar roles y características.

## 1.1. ¿Por qué instalar una GPO?

Hemos instalado una GPO para denegar el acceso al Panel de control en Active Directory es importante para reforzar la seguridad, cumplir con políticas corporativas, mantener la consistencia en la configuración de dispositivos y prevenir errores accidentales. Esto asegura que solo los usuarios con permisos de administración puedan realizar cambios críticos en el sistema, reduciendo así los riesgos de seguridad y manteniendo la integridad del entorno de red.

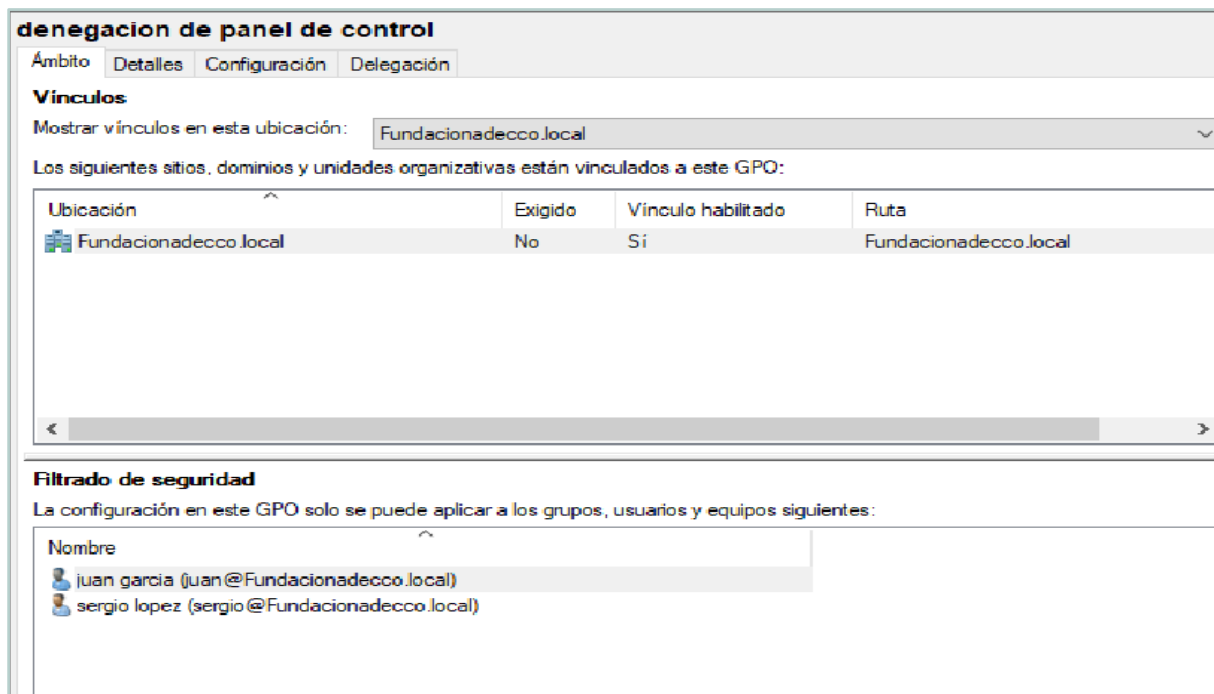


Figura 2. Configuración de GPO

## 1.2. ¿En qué consiste esta GPO?

La configuración de esta GPO permite las siguientes características:

- **Restricción de acceso:** Esta política permite a los administradores bloquear el acceso de los usuarios al Panel de Control por completo o restringir el acceso solo a ciertos elementos del Panel de Control.
- **Prevención de cambios no autorizados:** Al restringir el acceso al Panel de Control, los administradores pueden evitar que los usuarios realicen cambios no autorizados en la configuración del sistema.
- **Mayor seguridad:** Al limitar el acceso al Panel de Control, se puede mejorar la seguridad del sistema al prevenir cambios que podrían comprometer la integridad del sistema.
- **Control centralizado:** La política permite a los administradores controlar el acceso de forma centralizada para todos los usuarios del dominio o unidades organizativas específicas.

## INSTALACIÓN Y CONFIGURACIÓN DE UN SIEM EN WINDOWS SERVER

La política de denegación del Panel de Control es una herramienta importante para los administradores de sistemas que desean limitar el acceso de los usuarios a configuraciones sensibles del sistema. Esto puede mejorar la seguridad y la estabilidad de los sistemas en entornos de red administrados.

De igual forma se configuraron las siguientes GPOs

1. Control de acceso a direcciones IP privadas: Esta política permite definir qué intervalos de direcciones IP privadas son accesibles para aplicaciones o dispositivos en la red.
2. Segmentación de la red: Al establecer intervalos de direcciones IP privadas permitidas, los administradores pueden segmentar la red en subredes para mejorar la organización y la seguridad.
3. Seguridad: La política de intervalos de red privada puede ayudar a evitar conexiones no autorizadas entre segmentos de la red y proteger los recursos.
4. Políticas de red: Los intervalos de direcciones IP privadas también pueden estar relacionados con otras políticas de red, como políticas de acceso remoto, enrutamiento y filtrado de tráfico.
5. Configuración de aplicaciones: Algunas aplicaciones pueden requerir acceso a direcciones IP específicas dentro de la red. Esta política puede permitir o restringir el acceso a esas direcciones.

La política de intervalo de red privada para aplicaciones es una herramienta importante para gestionar y proteger el acceso a direcciones IP privadas dentro de una red, asegurando que las aplicaciones solo tengan acceso a los intervalos permitidos y manteniendo la integridad y seguridad de la red.

## **2. Necesidad de un Backup**

Un backup es una copia de seguridad a mayor o menor escala. Puede ser una versión reciente de la información contenida en todos los equipos de nuestra compañía, o puede tratarse de servidores completos con ingentes cantidades de datos.

Gracias a las copias de seguridad, conseguimos tener un plan de acción en caso de que se produzca un problema con los sistemas de la empresa. Así, en caso de que perdamos parte o toda la información, un servicio o ciertos sistemas que permiten operar, podremos recuperarnos rápidamente. Con esto reducimos el tiempo de respuesta ante la incidencia, y tendremos capacidad de maniobra en cualquier circunstancia adversa.

Los Backus permiten, por tanto, que en la compañía tengamos la tranquilidad de saber que la información siempre se guarda en una copia. Puede tratarse de una copia creada de forma automática cada cierto tiempo, o de un procedimiento que llevemos a cabo de forma manual. En todo caso, el objetivo es el mismo: mejorar la seguridad de la empresa y reducir al mínimo el tiempo de reacción frente a un problema.

### **2.1. ¿Por qué es necesaria una copia de seguridad?**

Todo el mundo debería tener un plan para mantener a salvo sus datos, ya sean archivos personales como fotos familiares, o documentos empresariales como hojas de cálculo y bases de datos.

Es especialmente vital para las empresas tener una estrategia de copia de seguridad. No hay nada peor que verse totalmente sorprendido por la pérdida de datos críticos debido a la caída de los servidores, el robo de la computadora portátil o un incidente similar de pérdida de datos.

La pérdida irreversible de datos no siempre es un desastre total, pero es algo que se puede evitar con una sólida estrategia de copias de seguridad. Las copias de seguridad te ofrecen protección contra los ciberataques y el ransomware, errores humanos como el borrado accidental de archivos, desastres naturales como inundaciones y fallas de hardware.

Al crear múltiples copias de tus archivos, te aseguras de que tu empresa pueda volver a funcionar rápidamente si algo sale mal.

## **2.2. Metodología 3-2-1**

La regla 3-2-1 es un método para realizar copias de seguridad que persigue conseguir el acceso a una copia de seguridad para poder ser respaldada siempre que sea necesario

La regla del 3-2-1 dice lo siguiente:

Siempre se deben realizar y mantener tres copias de seguridad de los datos a respaldar. Se utilizarán al menos dos soportes distintos para realizar estas copias y uno de ellos tiene que estar siempre fuera de la empresa (en el entorno actual de trabajo, en la nube). Para aplicar la regla de 3-2-1 para backups en una empresa hay que seguir sus tres principios básicos.

Realizar tres copias de los datos. No basta con hacer una sola copia de seguridad, el proceso de backup debe realizar tres copias diferentes.

Guardar al menos dos copias en soportes distintos.

Una de las copias se debe mantener fuera de la empresa.

## **3. Instalación de SIEM en Windows Server**

Un SIEM, o Security Information and Event Management, es una herramienta integral que ayuda a las empresas a gestionar la seguridad de su red y sistemas informáticos. Recopila y analiza datos de diversos recursos, como registros de eventos, registros de seguridad, tráfico de red y más, para detectar amenazas y proporcionar una visión completa de la postura de seguridad de la empresa.



### **3.1. ¿Por qué se hace necesario el uso de un SIEM?**

Las pequeñas empresas no están exentas de ataques cibernéticos y los ciberdelincuentes pueden considerarlas como objetivos más fáciles pues se suele percibir que son blancos más fáciles por su falta de recursos y menor inversión en ciberseguridad.

Para Fundación Adecco un SIEM ofrece unos beneficios muy importantes:

1. Visibilidad en tiempo real de los eventos de seguridad y potenciales amenazas.
2. Cumplimiento de la normativa puesto que algunos ciberataques se llevan a cabo con la intención de robar datos sensibles y en los últimos años se están poniendo en práctica regulaciones muy estrictas sobre la seguridad de esa información. El incumplirlas puede acarrear fuertes multas.
3. Identificar y responder a posibles amenazas incluso sin contar con un equipo de ciberseguridad dedicado.
4. Optimización de recursos: un SIEM ayuda a automatizar tareas de monitoreo y análisis permitiendo a los empleados enfocarse en otras tareas.

En nuestro caso hemos elegido instalar Splunk porque:

- Splunk es conocido por su capacidad para recopilar, indexar y analizar grandes volúmenes de datos de diversas fuentes.
- Puede escalar fácilmente para manejar un mayor volumen de datos a medida que la empresa crece, lo que lo convierte en una solución a largo plazo viable.
- Splunk ofrece una interfaz de usuario intuitiva y fácil de usar que facilita la búsqueda, el análisis y la visualización de datos de seguridad.
- Splunk ofrece una amplia gama de aplicaciones y complementos que permiten a las empresas personalizar y ampliar las capacidades de su SIEM según sus necesidades específicas.
- Splunk cuenta con un sólido soporte técnico y una comunidad activa de usuarios que pueden proporcionar orientación y asistencia en la implementación y el uso de la plataforma.

### 3.2. Instalación de Splunk

Comienzo de instalación:

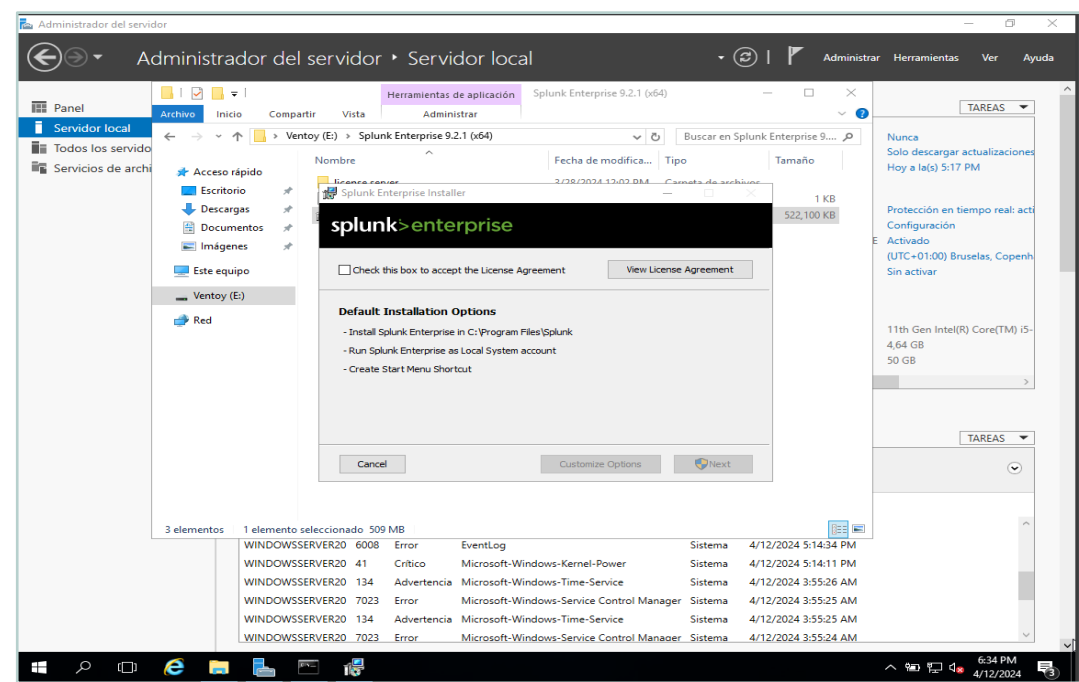


Figura 3. Comienzo de instalación de Splunk

Vista de Splunk después de la instalación:

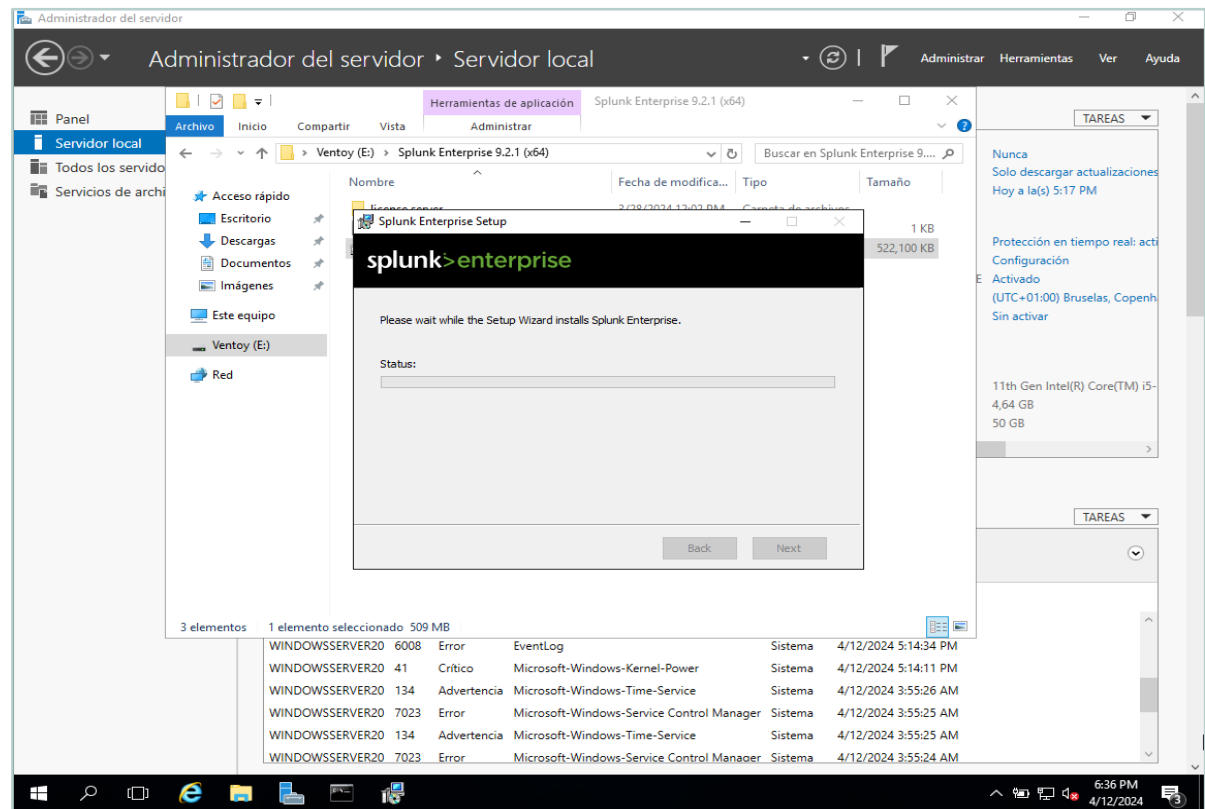


Figura 4. Proceso de instalación de Splunk

# INSTALACIÓN Y CONFIGURACIÓN DE UN SIEM EN WINDOWS SERVER

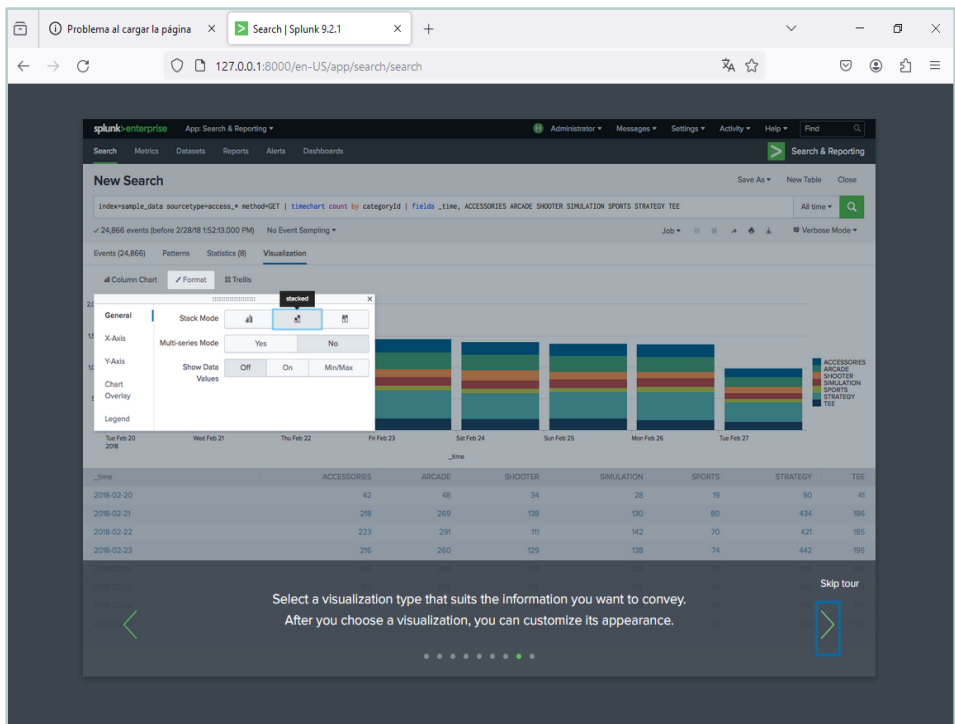


Figura 5. Panel principal de Splunk

## Página administración de Splunk:

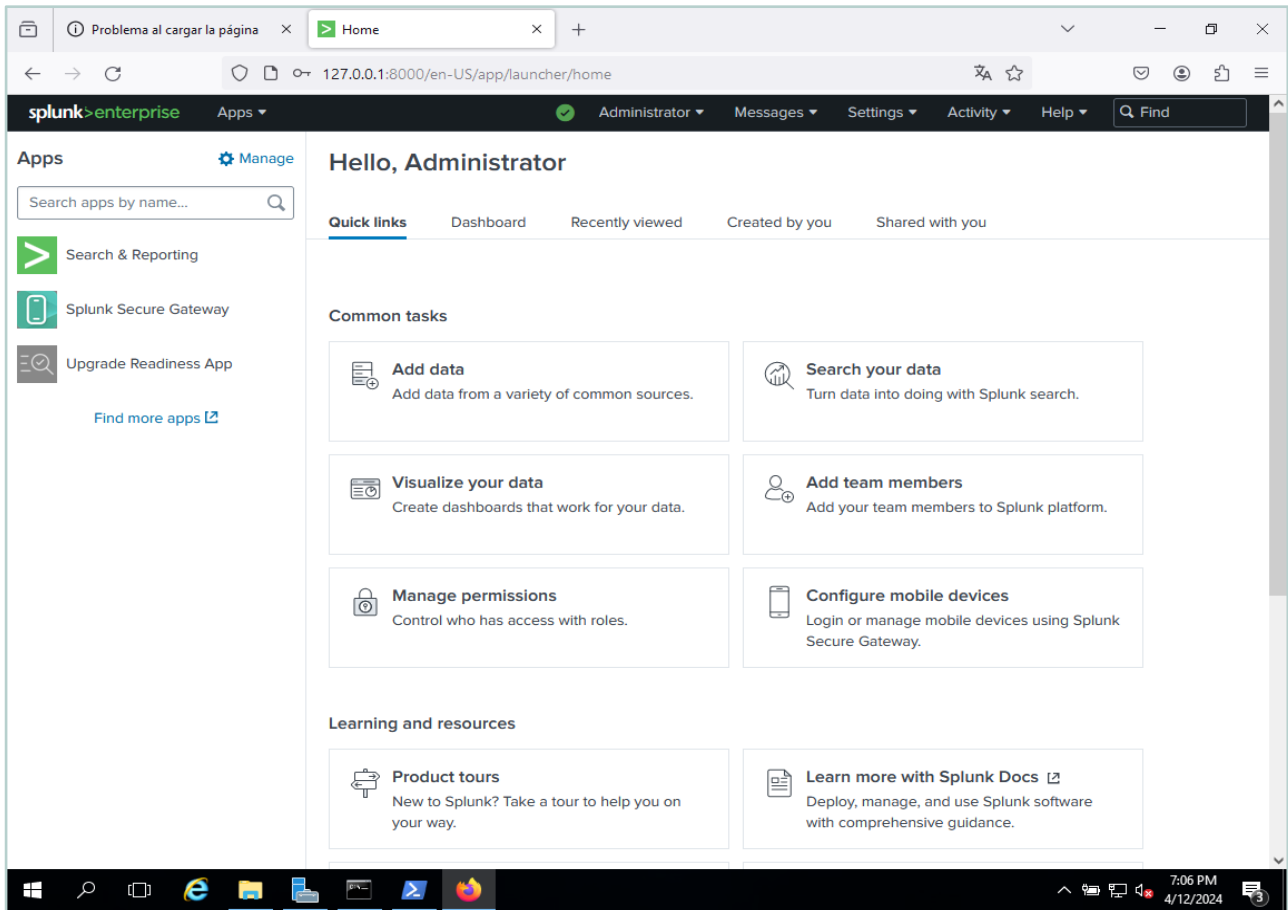


Figura 6. Panel de administración de Splunk

# INSTALACIÓN Y CONFIGURACIÓN DE UN SIEM EN WINDOWS SERVER

Consola de monitorización de Splunk:

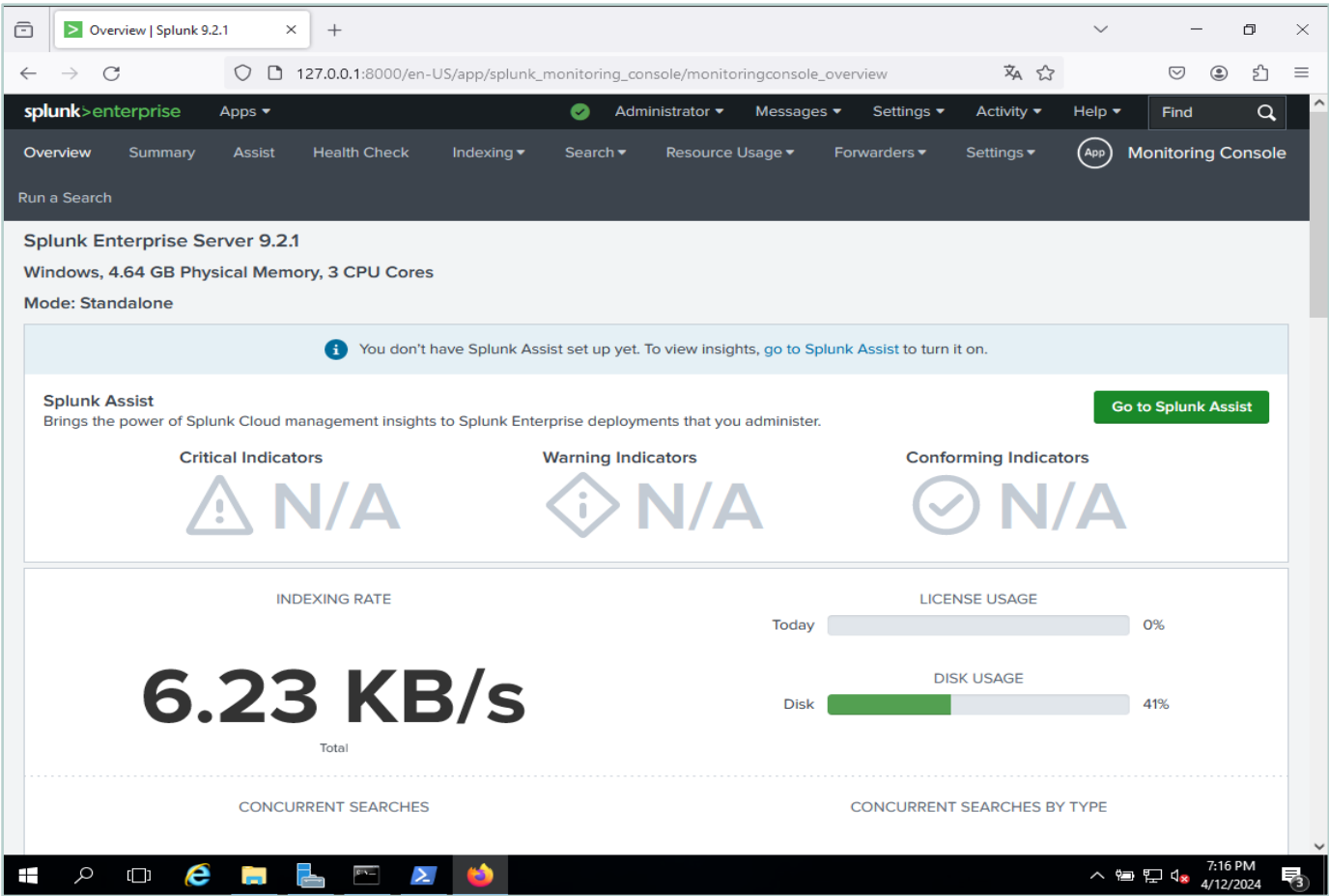


Figura 7. Consola de monitorización Splunk