

Informe de Vulnerabilidades

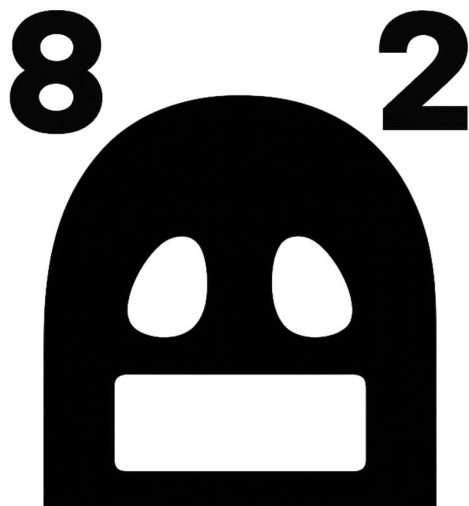
Web- anon.empresa-sec.co

 Fecha del informe: 6 de abril de 2025

 Elaborado por: **Dylan - Analista de Seguridad en 8D2**

 Contacto: 8d2secops@gmail.com

 Sitio analizado: <https://www.anon.empresa-sec.co>



1. Resumen Ejecutivo

Este informe presenta los hallazgos obtenidos durante un análisis de seguridad realizado sobre la aplicación web anon.empresa-sec.co. Se identificaron vulnerabilidades de alta y crítica severidad que comprometen directamente la seguridad del servidor y la confidencialidad de la infraestructura.

El hallazgo más relevante consiste en la ejecución dinámica de código PHP a través de una función `eval()` expuesta, lo que permite la ejecución remota de comandos. Este tipo de vulnerabilidad representa un riesgo crítico, permitiendo a un atacante externo comprometer completamente el sistema.

Se recomienda aplicar de forma urgente las medidas indicadas en este informe para mitigar los riesgos detectados y proteger la infraestructura digital de la organización.

2. Metodología

El análisis fue realizado bajo una metodología **black-box**, simulando el comportamiento de un atacante externo sin acceso al código fuente ni credenciales internas.

Las pruebas incluyeron:

- Manipulación de parámetros GET/POST.
- Análisis de errores del servidor.
- Enumeración de funciones mediante inputs controlados.
- Pruebas de ejecución remota de funciones nativas de PHP.

Se utilizaron herramientas como interceptores de tráfico HTTP (ej: Burp Suite) y scripts personalizados para validar el comportamiento del sistema ante entradas maliciosas.

3. Hallazgos Técnicos

3.1 Evidencia técnica del hallazgo crítico: ejecución dinámica con `eval()`

Ruta vulnerable: `/usuarios/guardar_usuarios.php`

Parámetro afectado: `nombre_registro`

Impacto: Ejecución remota de código (RCE)

Prueba de concepto (PoC) y explotación detallada

1. Prueba inicial de manipulación de entrada

Se modificó la petición original enviada al servidor, reemplazando el valor del parámetro nombre_registro por una expresión con sintaxis de PHP embebida:

```
nombre_registro=${echo("8D2")}
```

La respuesta del servidor incluyó un mensaje de error que reveló el uso de la función eval() en el backend:

```
<b>Fatal error</b>: syntax error, unexpected 'echo' (T_ECHO) in  
<b>/home/anon.empresa-sec/public_html/anon.empresa-  
sec/php/Modulo_Variables_Get.php(41) : eval()'d code</b>
```

Esto indica que el contenido del parámetro está siendo evaluado dinámicamente, lo que representa una vulnerabilidad crítica de ejecución remota de código (RCE).

2. Confirmación del comportamiento con pruebas adicionales

Se intentó ejecutar funciones PHP adicionales mediante la misma técnica. Por ejemplo:

```
nombre_registro=${print_r(scandir('/'))}
```

El error devuelto fue:

```
<b>Fatal error</b>: Uncaught Error: Call to undefined function _r() in ...
```

Esto sugiere que existe una manipulación de nombres de funciones antes de ser evaluadas, posiblemente para aplicar una "filtro" o "saneamiento", aunque claramente insuficiente.

3. Enumeración de funciones definidas por el usuario

Para identificar funciones internas utilizadas en el procesamiento, se ejecutó:

```
nombre_registro=${var_dump(get_defined_functions()["user"])}
```

El resultado reveló la función limpiar_parametro, presumiblemente aplicada antes del eval():

```
array(1) {  
  [0]=>  
    string(17) "limpiar_parametro"
```

```
}
```

4. Ejecución exitosa de código con función `phpinfo()`

Finalmente, se utilizó una construcción dinámica para invocar `phpinfo()` a través de la función `limpiar_parametro`:

```
nombre_registro=${limpiar_parametro(("php"."info"))}
```

La ejecución fue exitosa y desplegó la configuración completa del servidor (`phpinfo()`), confirmando que:

- La función `eval()` está siendo ejecutada directamente sobre la entrada del usuario.
 - Es posible ejecutar funciones arbitrarias del entorno PHP si se elude o satisface el mecanismo de "limpieza".
-

Impacto

Este comportamiento permite la ejecución remota de código en el servidor bajo el contexto del usuario web (probablemente `www-data`). Un atacante podría utilizar esto para:

- Acceder a archivos sensibles del sistema.
 - Filtrar información de la infraestructura.
 - Escalar privilegios mediante explotación lateral.
 - Instalar backdoors o malware.
-

3.2 Riesgo Crítico adicional: exposición del entorno mediante `phpinfo()`

Gracias al acceso no autorizado a la función `phpinfo()`, se pudo visualizar información sensible sobre la configuración del servidor. Aprovechando esto, se realizó una revisión

exhaustiva para identificar configuraciones inseguras y vulnerabilidades presentes. A continuación, se detallan los hallazgos críticos detectados a partir de esta exposición:

Riesgo Críticos

ID	Descripción	OWASP Top 10	Impacto	Solución Recomendada
C- 01	Ejecución remota de código vía parámetro GET (eval())	A03:2021 - Inyección	Compromiso total del servidor, ejecución de comandos	Eliminar eval(), usar funciones seguras, validar y sanear TODA entrada.
C- 02	Exposición pública de phpinfo()	A06:2021 - Configuración defectuosa	Revelación completa del entorno	Eliminar el archivo o restringir su acceso mediante autenticación o IPs autorizadas.

Riesgo Alto

ID	Descripción	OWASP Top 10	Impacto	Solución Recomendada
H- 01	Versión obsoleta de PHP (7.3.33)	A06:2021 - Configuración defectuosa	Exposición a vulnerabilidades sin parches	Actualizar a PHP 8.1 o superior.
H- 02	allow_url_fopen habilitado	A05:2021 - Fallas en control de acceso	Posible inclusión remota de archivos (RFI)	Deshabilitar si no es requerido. Validar fuentes externas.
H- 03	display_errors habilitado en prod.	A06:2021 - Configuración insegura	Revelación de rutas y errores a usuarios maliciosos	Desactivarlo en producción. Usar logging en archivos seguros.

Riesgo Bajo

ID	Descripción	OWASP Top 10	Impacto	Solución Recomendada
L- 01	Módulos innecesarios habilitados (mysqli, curl)	A06:2021 - Configuración defectuosa	Superficie de ataque ampliada	Revisar y desactivar módulos que no se utilicen.
L- 02	Falta de cabeceras de seguridad HTTP	A05:2021 - Seguridad del lado cliente	Riesgo de XSS, clickjacking, etc.	Implementar CSP, X-Frame-Options, X-Content-Type-Options, entre otras.

3.3 Riesgo Crítico adicional: Exposición pública de documentos sensibles indexados por Google

Ruta afectada: Indexación de directorios públicos en Google

Impacto: Fuga de datos personales sensibles accesibles desde buscadores

Descripción:

Durante el proceso de recolección de información, se identificó que múltiples rutas dentro del dominio anon.empresa-sec.co están indexadas por motores de búsqueda como Google. Estas rutas permiten el acceso directo a carpetas que contienen documentos personales de clientes sin ningún tipo de autenticación ni control de acceso.

Ejemplo de búsqueda utilizada:

site:anon.empresa-sec.co intitle:"index of"

Entre los hallazgos se encuentran directorios como:

- /anon.empresa-sec13/clientes/datos_clientes/5854/
- /plantillas/assets/plugins/datatables/extensions/ColReorder/

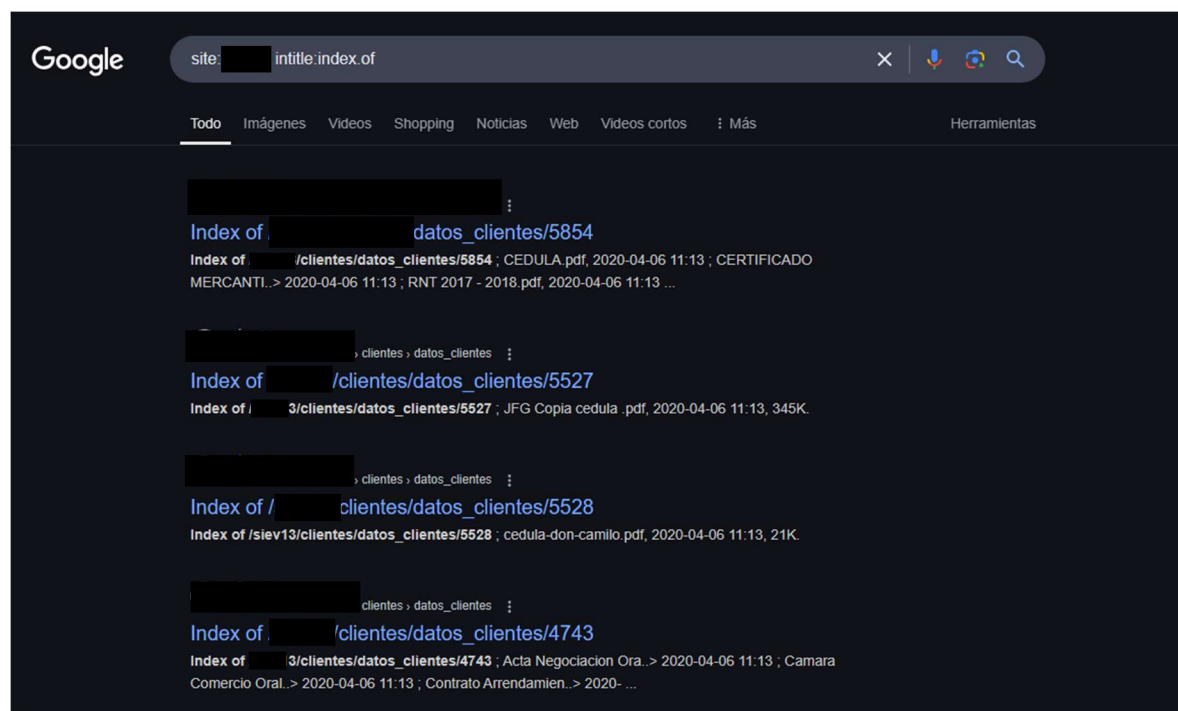
En el primer caso, se expone documentación extremadamente sensible, como:

- CEDULA.pdf
- CERTIFICADO MERCANTIL.pdf
- RNT 2017 - 2018.pdf
- RUT ALEX ORIGINAL.pdf

Estos archivos pueden ser abiertos por cualquier usuario desde Google, sin ningún tipo de autenticación.

Evidencia:

- Captura de resultados en Google:



Impacto:

- Fuga de datos personales protegidos por ley
- Violación directa a normativas de protección de datos (ej. Habeas Data, GDPR)
- Exposición de información legal, tributaria y de identidad
- Alto riesgo de suplantación de identidad y fraudes
- Reputación institucional severamente comprometida

Recomendaciones:

- Solicitar la eliminación inmediata de los directorios indexados mediante Google Search Console
- Deshabilitar el listado de directorios en el servidor (Options -Indexes si usan Apache)

- Proteger con autenticación cualquier carpeta que contenga información de clientes
- Implementar un escaneo de exposición digital periódicamente para detectar filtraciones por motores de búsqueda

4. Recomendaciones Generales


1. **Eliminar completamente el uso de eval():** No debe existir bajo ningún caso código que evalúe dinámicamente datos ingresados por el usuario.
 2. **Actualizar el entorno PHP:** Migrar a una versión actual (8.x) con soporte activo de seguridad.
 3. **Configurar adecuadamente el entorno de producción:**
 - Deshabilitar display_errors
 - Desactivar módulos innecesarios
 - Restringir acceso a funciones sensibles (phpinfo(), scandir, etc.)
 4. **Implementar encabezados de seguridad HTTP:**
 - Content-Security-Policy
 - X-Frame-Options
 - Strict-Transport-Security
 5. **Aplicar control de acceso a rutas internas:** Usar autenticación, whitelist de IPs y separación de entornos (desarrollo vs producción).
-

Firmado:

Dylan

Analista de Seguridad | 8D2

 8d2secops@gmail.com

 6 de abril de 2025

NOTA: este pdf nunca me lo pagaron, piensan que sobrevivo a puro pan y agua haha