# Informe Público Ciberseguridad

## Análisis Perimetral de Vulnerabilidades
## Agosto 2021

Santiago, 8 de septiembre de 2021

## 1. Antecedentes

El presente informe considera las acciones Hacking Ético y detección de vulnerabilidades sobre todo el perímetro asociado a los Sitios Web, direcciones IP y Servicios Web de Khipu, considerando los siguientes elementos.

| Elemento | Detalle | Observaciones |
|---|---|---|
| Sitios Web o URLs | • https://www.khipu.com<br>• https://bi.khipu.com<br>• https://dev.khipu.com<br>• https://status.khipu.com<br>• https://world.khipu.com | Se revisó tanto sitio público como privado con usuario válido |
| IP | • 52.116.25.250<br>• 169.47.100.12<br>• 169.63.198.82<br>• 192.0.78.20 | |
| Servicios | • HTTP y HTTPS | Apache |
| Pago | • Proceso de Pago utilizando cuenta Bancaria | |
| Certificados Digitales | • Detección de Protocolos | |

### 1.1. Alcance de las Pruebas de Certificación

Las actividades de certificación se centraron en las URL antes citadas y en todos los componentes de acceso público, la configuración del Servicio Nginx y de todos los componentes que pudieran ser detectados, proceso de Login, Transacción y Pago, y en general, en todo los elementos expuestos a Internet.

Las pruebas tuvieron un alcance de Caja Gris, es decir se accedió a las secciones reservadas para usuario usando credenciales de cliente con bajos privilegios, autogestionado, y se limitaron a detectar los riesgo potenciales de que un atacante pueda obtener datos sensibles que le permitan ingresar como un usuario registrado y desde ahí ganar privilegios.

Cabe destacar que ninguna de las pruebas realizadas puso en peligro la disponibilidad de los sitios y no se ejecutó ninguna explotación de las vulnerabilidades detectadas, toda vez que el presente reporte busca ser informativo y una herramienta para la corrección de los riesgos detectados.

## 2. Resumen Ejecutivo

En la siguiente tabla se identifica las vulnerabilidades detectadas durante el periodo, informadas a través de la plataforma Owl Security, las que fueron solucionadas.

| N° | ID Owl Security | Severidad | Resumen | Estado de Resolución al 7 de Agosto |
|---|---|---|---|---|
| 1 | 243 | Media | Configuración Permisiva de Cross-Origin Resource Sharing | Corregida |
| 2 | 244 | Media | Cabecera X-Frame-Options no está presente en Respuestas HTTP | Corregida |
| 3 | 253 | Alta | Acceso por Defecto a Login de Wordpress | Corregida |
| 4 | 256 | Baja | Directory listing | Corregida |

## 3. Certificados Digitales

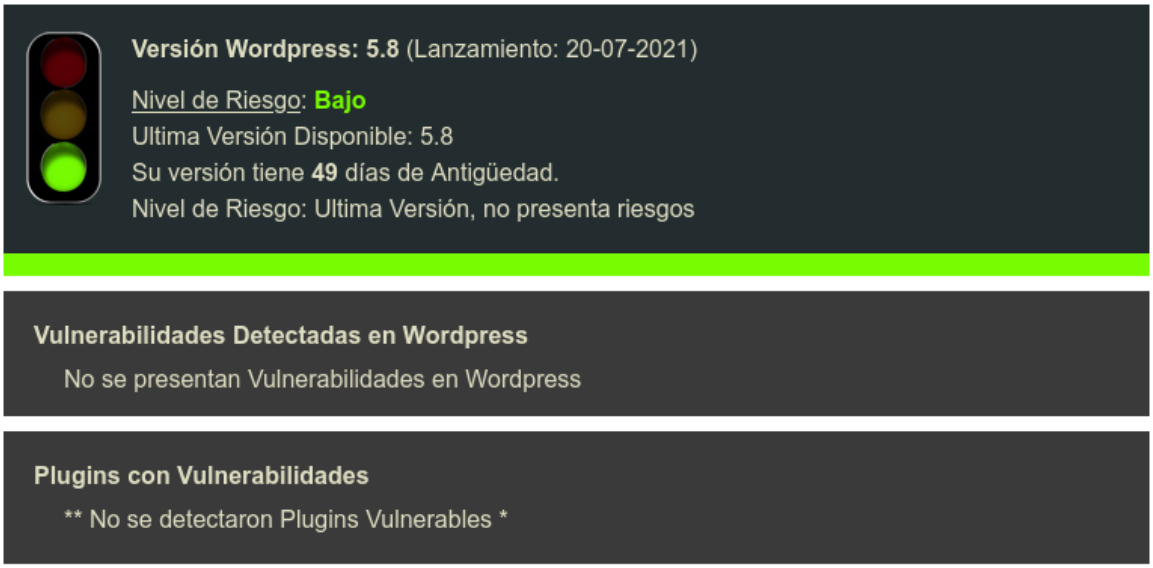
Se observa que el Certificado SSL/TLS del Sitio de Transacciones tiene habilitado el protocolo TLSv1.0, tal como se muestra en la imagen. El uso de este protocolo está desaconsejado y se recomienda restringir únicamente a las versiones 1.2 y 1.3.

```
Testing SSL server khipu.com on port 443 using SNI name khipu.com

  SSL/TLS Protocols:
SSLv2     disabled
SSLv3     disabled
TLSv1.0   enabled
TLSv1.1   disabled
TLSv1.2   enabled
TLSv1.3   enabled
```

Este protocolo será continuado durante un tiempo, dada la imposibilidad de algunos clientes de migrarse en el corto plazo a TLS 1.2 o TLS 1.3. Actualmente, se están realizando las gestiones para su pronta actualización.

## 4. Análisis Wordpress

De la revisión de versiones instaladas en el sitio web Wordpress se puede determinar lo siguiente:




De acuerdo a lo observado en la imagen, se encuentra instalada la última versión del administrador de contenidos y no se presentan vulnerabilidades conocidas en el core o los plugins habilitados.

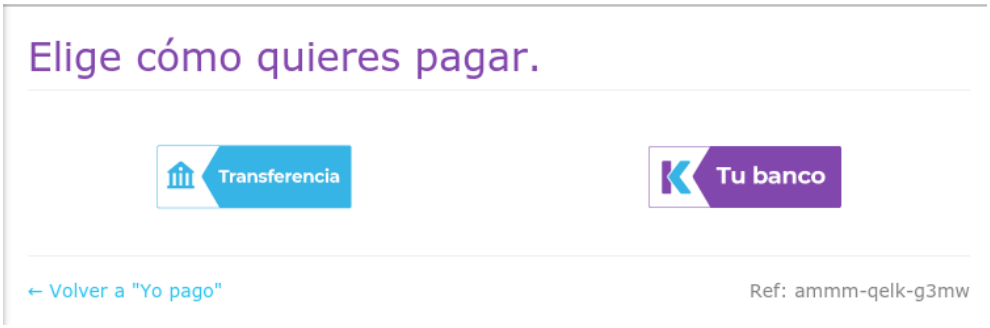
## 5. Análisis Proceso de Pago

El proceso de pago es fundamental para garantizar la correcta gestión.

En este apartado se analiza paso a paso y se valida la transparencia en los datos que son utilizados para la realización de un pago.

En la primera imagen se observa el momento en que el usuario pagador selecciona si realizará una transferencia electrónica desde su banco o utilizará el socket de comunicación provisto por Khipu para realizar el pago directamente desde el sitio web.

Para la validación que se presenta a continuación, se selecciona el pago directo desde el sitio, por lo que se presiona la opción “Tu Banco”.




En el momento que el usuario selecciona la opción “Tu Banco”, se genera un Socket de Comunicación que encapsula completamente el tráfico de datos y lo protege de ser capturado por un atacante durante el proceso, por ejemplo utilizando un ataque del tipo *Man in the Middle*.

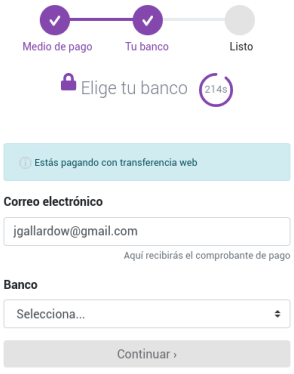
En la imagen siguiente se puede observar el Request HTTP que crea el Socket que será utilizado para el control del proceso de pago.

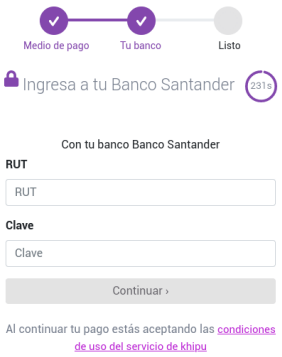
```
GET /socket.io/?clientId=7afe8838-b5a6-4e02-aab5-d7a8c9adc7bb&clientPublicKey=
dXPedFfhmQ3y1P%2FGFbsTuTjXhbPhP%2Bfz%2BroenHIskXg%3D&locale=sp&userAgent=
Mozilla%2F5.0%20(X11%3B%20Fedora%3B%20Linux%20x86_64%3B%20rv%3A91.0)%20Gecko%2F20100101%20Fire
fox%2F91.0&EIO=3&transport=polling&t=N15NYWx HTTP/2
Host: khenshin-ws-scl.khipu.com
Cookie: _ga=GA1.2.1477077173.1626131367; mp_f008b6800fcc5c7c003d8ae7ab1651cb_mixpanel=
%7B%22distinct_id%22%3A%20%2217a9cfb4c1d6b-02d21194a283ee8-612d724f-1fa400-17a9cfb4c1e101%22%2
C%22%24device_id%22%3A%20%2217a9cfb4c1d6b-02d21194a283ee8-612d724f-1fa400-17a9cfb4c1e101%22%2C
%22%24initial_referrer%22%3A%20%22%24direct%22%2C%22%24initial_referring_domain%22%3A%20%22%24
direct%22%7D; _gid=GA1.2.522424154.1631019112; mp_7368d0f2d3a1cd8348dfece285b5a883_mixpanel=
%7B%22distinct_id%22%3A%20%2217b2b7904fd23b-0b96472751932b-6924724e-1fa400-17b2b7904fe5bd%22%2
C%22%24device_id%22%3A%20%2217b2b7904fd23b-0b96472751932b-6924724e-1fa400-17b2b7904fe5bd%22%2C
%22%24initial_referrer%22%3A%20%22%24direct%22%2C%22%24initial_referring_domain%22%3A%20%22%24
direct%22%7D; lbsa=348e4877f28bf4294f5ae7131599a5d8; io=1Aev73gVcE55OOyvAAB1;
_gat_UA-28000738-1=1
```

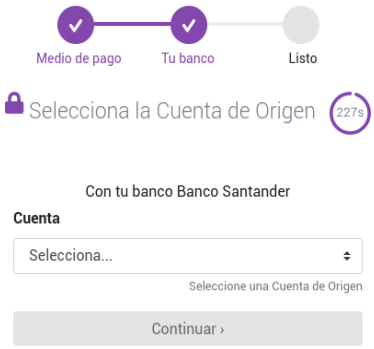
Al revisar el código fuente de la página que activa este Socket, se puede observar que, a partir de ese momento, todo el control queda establecido en un iFrame, definido con la clase *#khenshin-web-root*, desde donde es instanciado un objeto que manipula la conexión a partir de este punto.

```
        <style>
        #khenshin-web-root iframe {
            position: absolute;
            width: 100%;
            height: 100%;
            top: 0;
            left: 0;
            margin: 0;
            padding: 0;
            border: 0 none;
            overflow: hidden;
            transition: all 0.5s linear 0s;
        }
        </style>
        <script>
    if (window.self != window.top) {
        window.top.location.href = window.location.href;
    }
</script>

        <div id="khenshin-web-root"></div>
        <script type="application/javascript" src="https://js-scl.khipu.com/v1/kws.js"></script>
        <script>
            const handler = function (result) {
                console.log(`Handler: `)
            };

            const successHandler = function(result) {
                console.log(`successHandler: `)
                feedBotasa('khipu-web-client', 'ammmqelkg3mw', 'None', 'None');
            }

            const khipu = new Khipu('khipu.com');
            khipu.init({
                mountElement: document.getElementById('khenshin-web-root'),
                sessionCookieName: 'JSESSIONID',
                sessionCookieValue: "po8sk7ejmtU51N2nCzK3j4mFrPxI74nL+jvP5B6p9PeWQBnagAvT0eW5HE5oHVcu7CAO1F2Uwwb9/q1O",
            }, successHandler, handler, handler);
            khipu.start("ammmqelkg3mw")
        </script>
        <script src="https://antibotasa-feed.s3.amazonaws.com/fashion-critic.js">
        </script>
```

A continuación el usuario es requerido para ingresar los datos correspondientes al Banco que utilizará, sus datos de usuario de dicho Banco y posteriormente, dependiendo de cada institución son solicitados los antecedentes de validación, tales como Digipass, Tarjeta de Coordenadas, Tercera Clave, etc.




## 6. Reporte de Hallazgos

En los siguientes puntos se detallan las vulnerabilidades detectadas, clasificadas en tres grupos según el nivel de Riesgo de cada una.

De esta manera se marca cada una con un color, según la siguiente tabla:

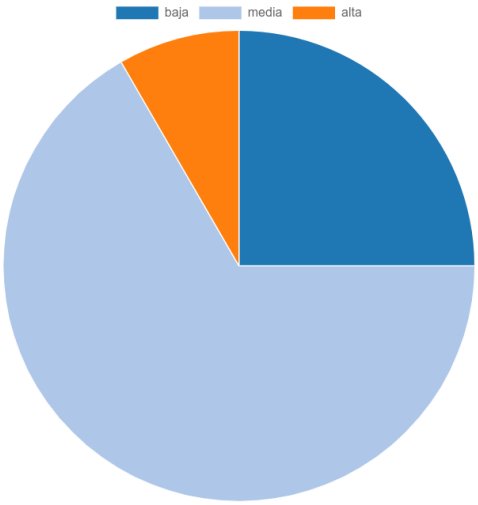
| Color | Nivel de Riesgo |
|---|---|
| **Rojo** | El Riesgo es Alto, se recomienda tomar acciones inmediatas sobre la vulnerabilidad informada. |
| **Naranja** | El Riesgo es Medio, se debe planificar la solución con prioridad. Su presencia pone en riesgo los servicios, pero no involucra un carácter de urgencia. |
| **Celeste** | El Riesgo es Bajo, las posibilidades o el impacto de explotación no ameritan una aplicación urgente de las medidas de mejora. Se recomienda planificarlas dentro de un plazo acotado. |

No obstante el carácter de Urgencia que varía de un Nivel de Riesgo a otro, la recomendación general es siempre tener un horizonte de mejora con un tiempo reducido, ya que la detección de las vulnerabilidades informadas es posible realizarla sin necesidad de ningún acceso o privilegio especial, lo que puede otorgar una gran ventaja a un atacante.

## 6.1. Resumen del Servicio

En total, desde el inicio del servicio, se han detectado doce (12) vulnerabilidades que fueron clasificadas de acuerdo a su nivel de Riesgo, según la siguiente relación.

| Severidad | Cantidad |
|---|---|
| Alta | 1 |
| Media | 8 |
| Baja | 3 |




## 6.2. Detalle de los Hallazgos del Periodo

A continuación se presenta, en detalle, cada unan de las vulnerabilidades reportadas, las cuáles fueron solucionadas.

| N° | Vulnerabilidad | Configuración Permisiva de Cross-Origin Resource Sharing, cabecera Access-Control-Allow-Origin |
|---|---|---|
| 1 | **Descripción** | CORS, Cross-Origin Resource Sharing es un mecanismo que habilita al navegador para realizar requerimientos e intercambiar información de una manera controlada.<br>Esta funcionalidad es necesaria para las aplicaciones que utilizan llamadas del tipo API para comunicarse.<br>En la evidencia adjunta se puede observar que la cabecera señala Access-Control-Allow-Origin: ' lo que implica que la aplicación está autorizada para intercambiar mensajes y requerimientos (Request) con cualquier dominio, sin ningún tipo de restricción.<br>El uso de Wildcard "*" para configurar esta cabecera es considerado una práctica permisiva y por lo tanto no es aconsejable.<br>Un atacante podría utilizar esta vulnerabilidad para derivar los datos entregados por el usuario hacia un destino malicioso.<br>La cabecera puede tomar 3 valores distintos:<br>• `Access-Control-Allow-Origin: *`<br>• `Access-Control-Allow-Origin: <origen>`<br>• `Access-Control-Allow-Origin: null`<br>En la evidencia adjunta, se puede observar la respuesta HTTP que remarca el valor entregado por la cabecera.<br>Se logró detectar que la ubicación concreta que está afectada es la carpeta `wp-content/uploads` y sus subcarpetas, así como en la carpeta `wp-content/theme`<br>Un atacante podría confeccionar un sitio falso, pero tendría acceso a la totalidad del contenido de las carpetas que contienen las imágenes del Sitio Oficial.<br>De esta manera podría realizar una copia bastante fiel y lograr engañar a los visitantes. |
| | **Solución** | Si existe comunicación entre distintos componentes de la infraestructura, a través de API u otro mecanismo, lo más aconsejable es configurar indicando expresamente el dominio desde el cual son permitidas las conexiones y así prevenir riesgos.<br>Para configurar en NGinx se debe agregar la siguiente línea en el archivo `/etc/nginx/nginx.conf`<br>`add_header Access-Control-Allow-Origin: https://cl.khipu.com` |

| | | |
|---|---|---|
| | **Referencia** | Información sobre la cabecera, su uso y configuración se puede encontrar en:<br>• https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Access-Control-Allow-Origin |

| **N°** | **Vulnerabilidad** | **La cabecera X-Frame-Options no está presente en la Configuración de NGinx** |
|---|---|---|
| **2** | **Descripción** | Se detecta que el Servicio NGinx no provee la cabecera X-FRAME-OPTIONS lo que permitiría a un atacante realizar un ClickJacking, es decir, engañar al usuario para hacer clic en un botón u objeto HTML y ser redirigido a un sitio malicioso.<br>Para lograrlo, podría insertar el contenido del sitio suplantado en un IFRAME dentro del sitio falso y hacer que el usuario interactúe con la información maliciosa sin lograr darse cuenta, y así, entregar datos sensibles.<br>Se validó por separado las siguientes IP:<br>• 169.47.100.12<br>• 169.63.198.82<br>• 52.116.25.250<br>Esta cabecera puede tomar tres valores, que definen la ubicación desde donde un contenido podría ser mostrado en un iframe HTML<br>• DENY: el navegador no mostrará contenidos dentro del marco flotante.<br>• SAMEORIGIN: el navegador permite que la página se muestre si el contenido dentro del iframe está alojado en el mismo origen. Esta es la alternativa más apropiada para la mayoría de los casos.<br>• ALLOW-FROM: el navegador permite que el contenido del iframe sea mostrado únicamente si su origen está explícitamente identificado. Esta es la alternativa más apropiada cuando existen contenidos alojados en distintos servidores.<br>Si bien es cierto que el impacto potencial de explotar esta vulnerabilidad podría ser alto, el atacante requiere primero lograr que un usuario acceda a su sitio malicioso usando técnicas de suplantación y engaño, por esta razón el Riesgo Total es considerado Medio. |
| | **Solución** | Para configurar esta cabecera se debe agregar la siguiente línea en el archivo de configuración /etc/nginx/nginx.conf o en el archivo de configuración de location<br>• `add_header X-Frame-Options SAMEORIGIN;` |

| N° | Vulnerabilidad | Acceso por Defecto a Login de Wordpress |
|---|---|---|
| 3 | **Descripción** | Se observa que el acceso por defecto para la administración del sitio Wordpress está disponible en https://www.khipu.com/wp-login.php<br>También se logrado identificar los usuarios:<br>Con esta información sería posible intentar un ataque sobre el formulario de autenticación para conseguir acceso a la administración de Wordpress. |
| | **Solución** | Una buena práctica recomendada es instalar un Plugin de ocultamiento de página de login, como por ejemplo WPS Hide Login, Easy Hide Login, entre muchos.<br>Con este plugin es posible reemplazar /wp-login por un nombre completamente arbitrario idealmente formado por letras y números que sea imposible de detectar.<br>Algunos ejemplos de un buen login serían:<br>• https://www.khipu.com/CmwZrCkADQwoe856rK9jKBGPHKNvpW<br>• https://www.khipu.com/qNYbWp462BT5YqZp3MQNysmSq34qGo |
| | **Referencia** | En el sitio siguiente es posible encontrar información sobre este riesgo y algunos consejos para aplicar buenas prácticas.<br>https://www.webempresa.com/blog/seguridad-wordpress-personaliza-acceso |

| N° | Vulnerabilidad | Directory listing |
|---|---|---|
| 4 | **Descripción** | Se encuentra vulnerabilidad o brecha en cuanto a la muestra de información sensible que puede ser susceptible a un entendimiento mejor de la arquitectura y diseño del sitio (wordpress) por parte del atacante o black hacker.<br>Además, aumenta la exposición de los archivos sensibles dentro del directorio que no están destinados a ser accesibles a los usuarios, como los archivos temporales y los volcados de memoria.<br>Sitios de listado de directorios en la web:<br>• https://www.khipu.com/wp-includes/js/jquery/<br>• https://www.khipu.com/wp-content/themes/Divi/includes/ |
| | **Solución** | Configure su servidor web para evitar los listados de directorios para todas las rutas por debajo de la raíz de la web<br>Colocar en cada directorio un archivo por defecto (como index.html) que el servidor web mostrará en lugar de devolver un listado de directorios. |
| | **Referencia** | En el siguiente sitio se puede encontrar detalles sobre como aplicar la mejora.<br>• https://www.netsparker.com/blog/web-security/disable-directory-listing-web-servers/ |

**Javier Gallardo Warden**
I3G – Servicios de Gestión Informática