

## Robando sesiones y bypassando 2FA con EvilnoVNC

Proyecto **EvilnoVNC**.

Se trata de una pequeña herramienta, que nos permitirá robar sesiones y *bypassear* 2FA a través de una conexión **noVNC** maliciosa. Todo esto, se realizará a través de Docker de forma totalmente automática, gracias a un sencillo script escrito en bash.



Uno de los ejercicios más habituales en todas las grandes compañías (incluso en aquellas que no hacen ejercicios de Red Team), son las campañas de *phishing*. Gracias a estas, se **consigue** pretender concienciar a los usuarios, para que no accedan a sitios maliciosos, que **pueden terminar comprometiendo a los mismos, o incluso, a toda la empresa**.

Desde el punto de vista ofensivo, existen diferentes formas de abordar esta ardua tarea. Por poner algunos ejemplos bastante habituales:

- Podemos hacer un clon idéntico de la página que queremos suplantar
- Podemos crear una página nueva para que el usuario se registre
- Podemos utilizar un «proxy inverso» para hacer un MiTM

Cada una de estas técnicas, tiene sus ventajas e inconvenientes, siendo algunas más transparentes para la víctima y otras más evidentes, ante el ojo experto o el usuario desconfiado. **En algunos casos, obtendremos credenciales en texto plano y en otras, solo obtendremos las cookies de sesión**, permitiéndonos acceder a la cuenta del usuario durante un breve espacio de tiempo.

En cualquier caso, existen muchos artículos en internet con información al respecto y muchísimas herramientas, tan buenas como [Gophish](#) o [evilginx2](#) por nombrar algunas.

En el artículo de hoy, hablaremos de una técnica diferente (que podría ser tan válida como cualquier otra). Evidentemente, **esta no escapa de tener sus pros y sus contras**, así que dejo en vuestras manos decidir si será vuestra próxima herramienta favorita, o, por el contrario, pensáis que no merece la pena utilizarla en un entorno real.

Antes de comenzar, es obligatorio mencionar que esta idea no es mía, y que [podéis encontrar toda la información en el artículo](#) que escribió [@mrd0x](#) al respecto.

Básicamente, el concepto es hacer que un usuario acceda a un enlace malicioso, en el que se mostrará la página original que queremos suplantar. Esta página **se cargará en un navegador que el atacante puede controlar**, a través de una conexión [noVNC](#) (un VNC a través del navegador) de forma totalmente transparente para la víctima.

A simple vista, puede parecer algo complejo, pero se trata de algo completamente trivial. Realmente, la única dificultad, es realizar el proceso de instalación y **modificar algunos componentes de noVNC para que no se muestren controles en pantalla**, pantallas de carga, ni ninguna otra cosa que pueda hacer sospechar al usuario.

Así que una vez más, me dispuse a crear mi propia prueba de concepto, a la que llamaría **EvilnoVNC**. A continuación, os dejo el enlace del proyecto en GitHub:

<https://github.com/JoelGMSec/EvilnoVNC>

Ahora que ya conocemos la teoría, pasemos a la parte práctica



Por ahora, esta herramienta ha sido concebida para ser utilizada en Linux a través de bash, siendo necesario tener Docker y Chromium instalado para poder hacerla funcionar. En mi caso, todas las pruebas han sido realizadas sobre la última versión de Kali Linux.

Como suele ser habitual, clonaremos el repositorio y ejecutaremos los siguientes comandos para instalar el contenedor en nuestro sistema:

```
git clone https://github.com/JoelGMSec/EvilnoVNC  
cd EvilnoVNC ; sudo chown -R 103 Downloads  
sudo docker build -t joelgmsec/evilnovnc .
```

The terminal window shows the following output:

```
Terminal  
Archivo Editar Ver Terminal Pestañas Ayuda  
Step 12/19 : COPY Files/cookies.py /home/user/  
--> ec8aa6e9f21e  
Step 13/19 : COPY Files/vnc_lite.html /home/user/noVNC/  
--> 4bcc0590092b  
Step 14/19 : COPY Files/cursor.js /home/user/noVNC/core/util/  
--> aa7a0428e7a0  
Step 15/19 : COPY Files/rfb.js /home/user/noVNC/core/  
--> 6eedc4ac9230  
Step 16/19 : COPY Files/ui.js /home/user/noVNC/app/  
--> c810270bae15  
Step 17/19 : ENTRYPOINT ["/bin/bash", "-c", "startVNC () { ./startVNC.  
sh \"$@\"; }; \"$@\"", "foo"]  
--> Running in 4f71fb7f8ef6  
Removing intermediate container 4f71fb7f8ef6  
--> 6e4020b9e360  
Step 18/19 : EXPOSE 5980  
--> Running in dfbf587e66eb  
Removing intermediate container dfbf587e66eb  
--> 38f9f9bb4ef8  
Step 19/19 : CMD ["startVNC"]  
--> Running in 2ffbe4775c05  
Removing intermediate container 2ffbe4775c05  
--> ffc92c78a974  
Successfully built ffc92c78a974  
Successfully tagged joelgmsec/evilnovnc:latest
```

The terminal window has a title bar labeled "Terminal". The status bar at the bottom shows "JoelGMSec > .../Tools/EvilnoVNC" and "Session 1". A watermark "darkbyte.net" is visible in the bottom right corner of the terminal window.

Una vez realizada la instalación, ejecutaremos el siguiente comando para ver la ayuda del programa:

```
./start.sh -h
```

The screenshot shows a terminal window titled "Terminal". The command entered is `./start.sh -h`. The output is as follows:

```
----- by @JoelGMSec -----  
Usage: ./start.sh $resolution $url  
Examples:  
1280x720 16bits: ./start.sh 1280x720x16 http://example.com  
1280x720 24bits: ./start.sh 1280x720x24 http://example.com  
1920x1080 16bits: ./start.sh 1920x1080x16 http://example.com  
1920x1080 24bits: ./start.sh 1920x1080x24 http://example.com
```

Session 1 darkbyte.net

Como podéis comprobar, la ejecución del script no tiene ningún misterio. Lo único que tendremos que hacer para que funcione, será indicar la resolución de la víctima y la página que queremos suplantar.

Es muy importante tener en cuenta el tema de la resolución, ya que actualmente, **esta técnica no soporta utilizar una resolución dinámica**, siendo necesario introducirla antes de cada ejecución. Esto hará que el escritorio virtual tenga las dimensiones que nosotros le indiquemos, **siendo muy evidente para la víctima si nos equivocamos** al introducir un valor demasiado bajo o demasiado alto.

Por suerte, la gran mayoría de usuarios utiliza una resolución de 1920×1080 en sus equipos portátiles y de sobremesa. En cualquier caso, **sería posible obtener esta información a través de javascript** y adaptar la resolución a los datos obtenidos.

Ahora que ya sabemos cómo funciona, hagamos una simple prueba con una página real. Para este ejercicio, utilizaré el login de Google como ejemplo:

```
./start.sh 1920x1080x24 https://accounts.google.com
```

The screenshot shows a terminal window titled "Terminal". The menu bar includes "Archivo", "Editar", "Ver", "Terminal", "Pestañas", and "Ayuda". The command line shows the path "JoelGMSec > ..../Tools/EvilnoVNC > ./start.sh 1920x1080x24 https://accounts.google.com". The output of the script is displayed:  
----- by @JoelGMSec -----  
[>] EvilnoVNC Server is running..  
[+] URL: http://localhost:5980/index.html?autoconnect=true&password=false  
[!] Press Ctrl+C at any time to close!  
[+] Cookies will updated every 30 seconds..

Session 1

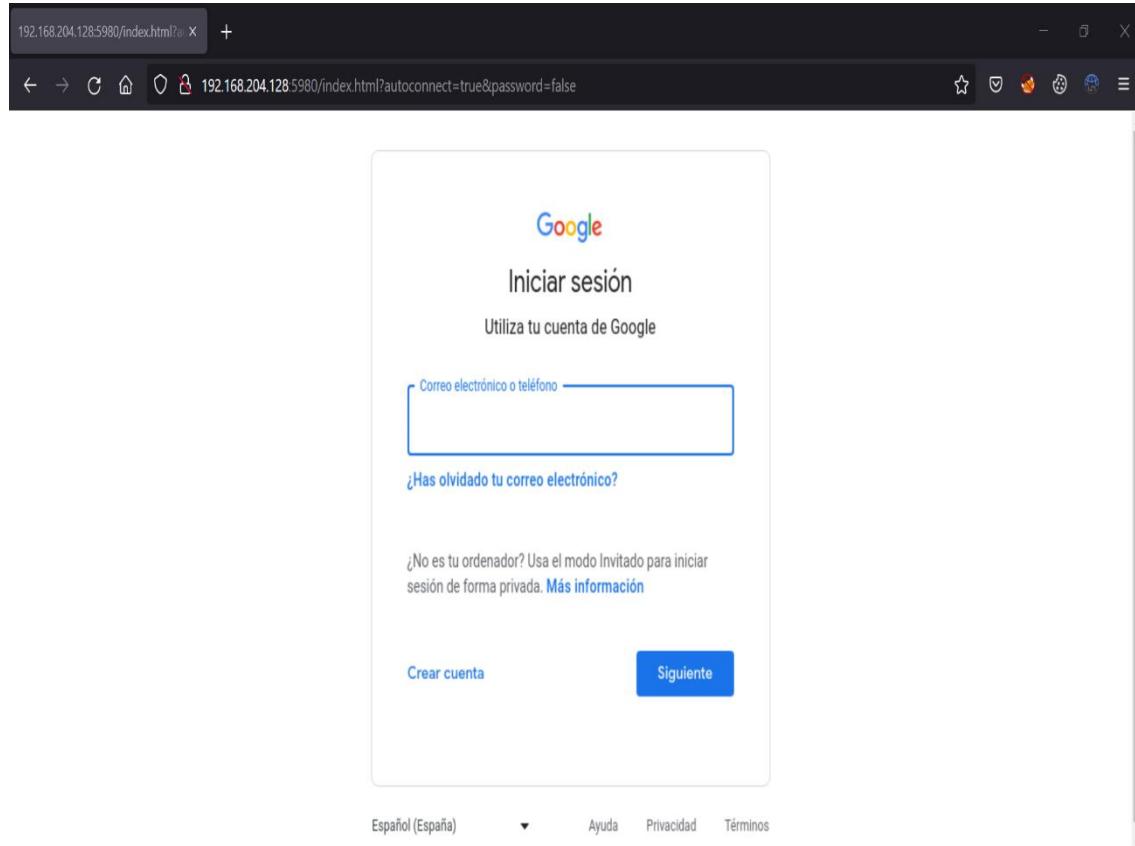
darkbyte.net

Una vez lanzada la herramienta, esta nos devolverá una URL con los datos de acceso, que será la que utilizaremos para ver en tiempo real lo que está sucediendo. Por otra parte, el auditor será quién tendrá que comprar un dominio y redireccionar el acceso (a través de NAT, o lanzando el script desde un VPS) para que la víctima pueda llegar hasta él.

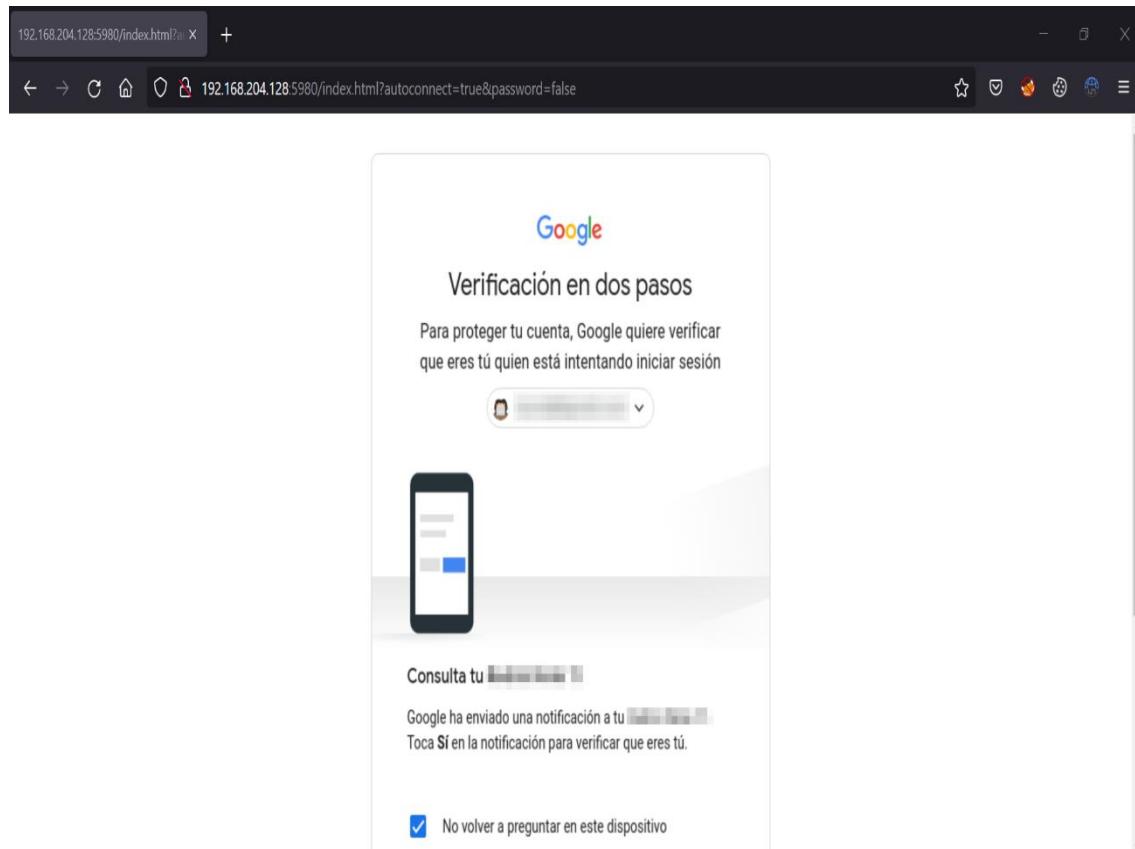
A partir de aquí, enviaremos el enlace malicioso y esperaremos para ver lo que sucede. Os recomiendo encarecidamente abrir dos navegadores la primera vez que ejecutéis el programa, para ver como en ambos se refleja exactamente lo mismo. De hecho, podríamos escribir o mover el scroll y la víctima lo vería en tiempo real, así que tened cuidado con esto.

Como puede verse en la siguiente imagen, el acceso a la página de Google es idéntico al original (de hecho, es original, pero

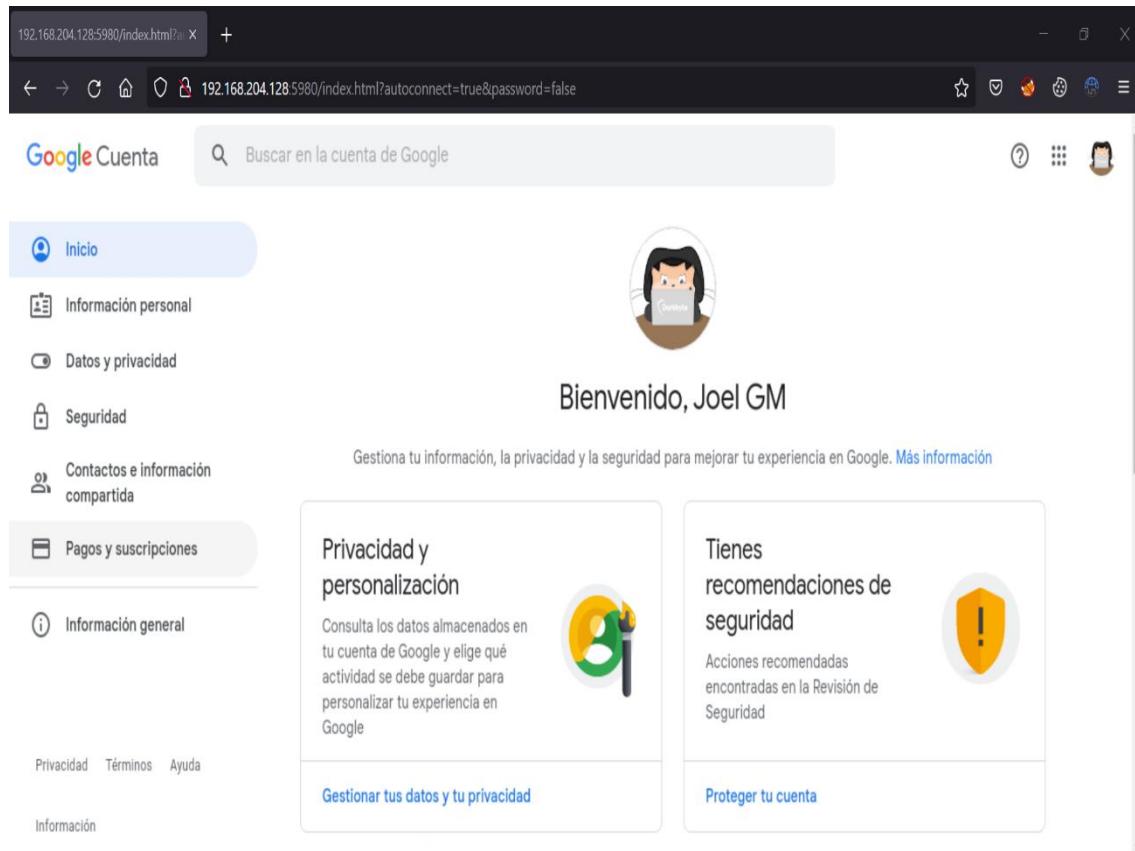
está cargando en nuestro navegador virtual):



Una vez la víctima introduzca sus credenciales, aparecerá el diálogo de 2FA si lo tiene habilitado:



Y por último, la víctima llegará a su cuenta y realizará las acciones que tenía previstas por hacer:



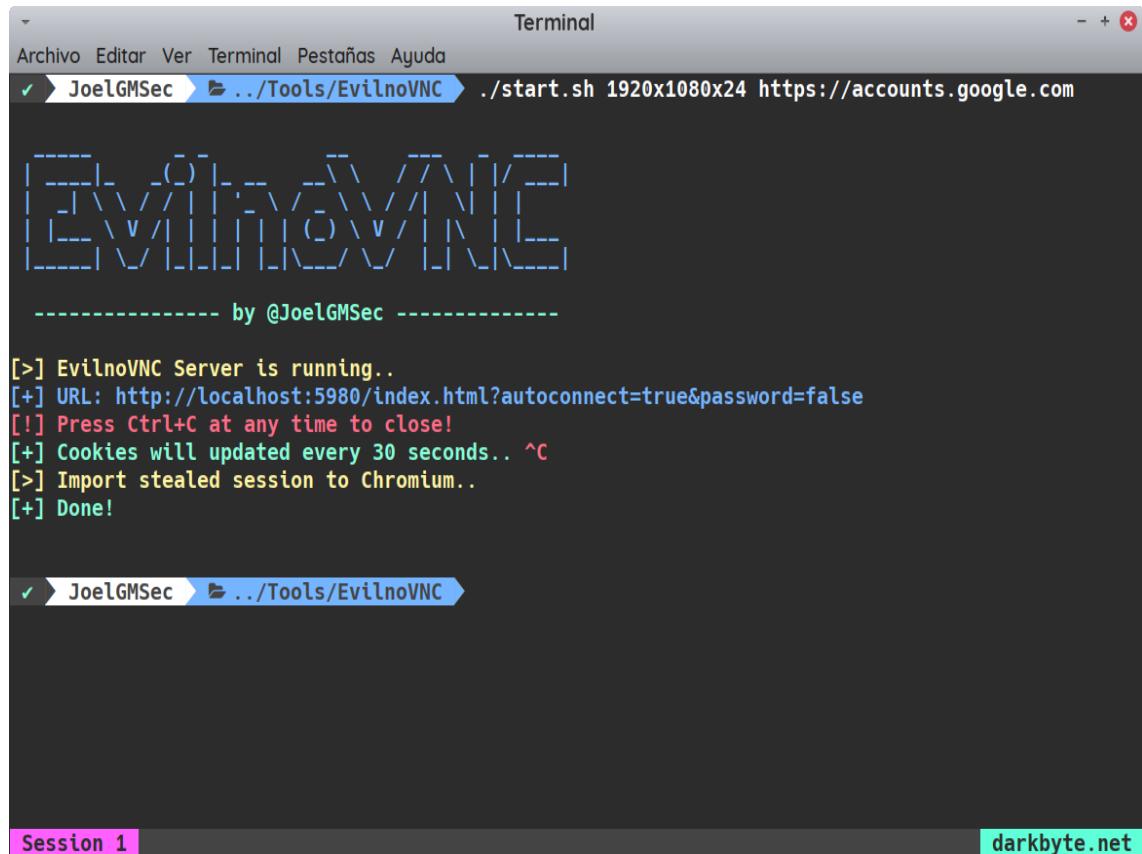
The screenshot shows a web browser window with the URL `192.168.204.128:5980/index.html?autoconnect=true&password=false`. The page is a Google Account settings page. On the left, there's a sidebar with links: Inicio (highlighted), Información personal, Datos y privacidad, Seguridad, Contactos e información compartida, Pagos y suscripciones, and Información general. At the bottom of the sidebar are links for Privacidad, Términos, Ayuda, and Información. The main content area features a profile picture of a cartoon character and the greeting "Bienvenido, Joel GM". Below this, there are two main sections: "Privacidad y personalización" (with a pencil icon) and "Tienes recomendaciones de seguridad" (with a shield icon). Both sections have a "Gestionar tus datos y tu privacidad" link at the bottom.

A partir de aquí, como atacantes, tenemos varios accesos a toda esta información:

- En primer lugar, aunque la víctima cierre el navegador, **nosotros continuaremos teniendo acceso al mismo**, con la cuenta iniciada correctamente.
- Adicionalmente, **todos los datos del usuario, se guardarán** en la carpeta «Downloads» así como las cookies de sesión, en el fichero «Cookies.txt».
- Y por si esto no fuera suficiente, **todos los ficheros que descargue el usuario** se guardarán en nuestra carpeta, **así como todas las contraseñas** que se guarden en el navegador.

Si en cualquier momento terminamos la ejecución del script pulsando «Ctrl+C», **la herramienta copiará el perfil completo de la víctima** en nuestro sistema, abriendo tras de sí un Chromium

con toda esta información:



The screenshot shows a terminal window titled "Terminal". The command executed is `./start.sh 1920x1080x24 https://accounts.google.com`. The output includes a decorative ASCII art banner, the author's handle (@JoelGMSec), and several informational messages about the server's status, URL, and update frequency. The terminal window has a dark background with light-colored text. The title bar and some UI elements are visible at the top and bottom.

```
Terminal
Archivo Editar Ver Terminal Pestañas Ayuda
✓ > JoelGMSec > ..../Tools/EvilnoVNC > ./start.sh 1920x1080x24 https://accounts.google.com

----- by @JoelGMSec -----

[>] EvilnoVNC Server is running..
[+] URL: http://localhost:5980/index.html?autoconnect=true&password=false
[!] Press Ctrl+C at any time to close!
[+] Cookies will updated every 30 seconds.. ^C
[>] Import stealed session to Chromium..
[+] Done!

✓ > JoelGMSec > ..../Tools/EvilnoVNC >
```

Session 1 darkbyte.net

Pasados unos segundos, se abrirá el navegador y podremos consultar todo lo que ha hecho el usuario, teniendo acceso a

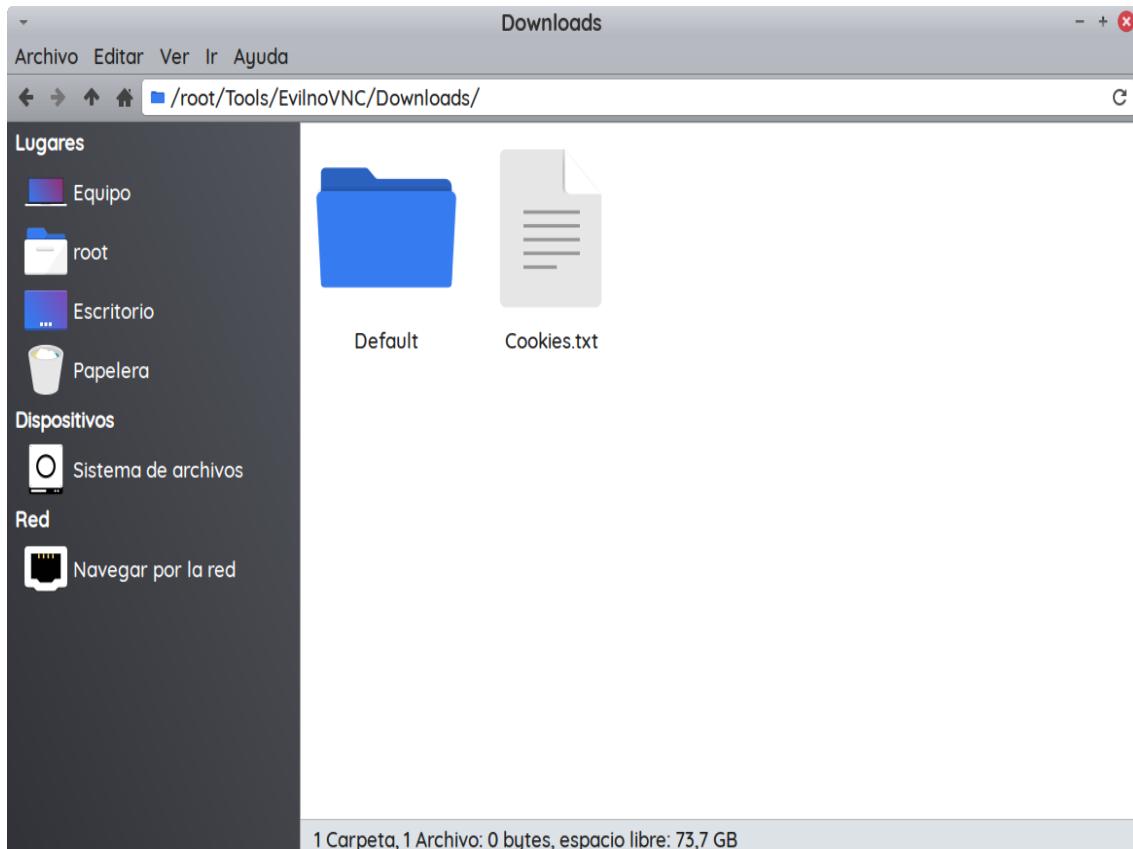
**todas sus cookies y contraseñas almacenadas:**

The screenshot shows the Chromium browser's history page. At the top, there are tabs for 'Google' and 'History'. The main area is titled 'History' and includes a search bar labeled 'Search history'. Below the search bar are two navigation buttons: 'List' and 'Journeys'. The date 'Today - Sunday, September 4, 2022' is displayed. A list of recent visits is shown, each with a delete icon, timestamp, Google logo, activity name, and URL. The entries are:

Time	Activity	URL
12:28 PM	Crea una cuenta de Google	accounts.google.com
12:28 PM	Crea una cuenta de Google	accounts.google.com
12:26 PM	Inicia sesión: Cuentas de Google	accounts.google.com
12:26 PM	Google	www.google.com

Como comentaba anteriormente, en la carpeta «Downloads» tendremos las cookies, el perfil del navegador y cualquier

fichero que se haya descargado la víctima:



Durante la ejecución del script, todas las cookies del navegador alargarán su vida el máximo posible automáticamente. De esta forma, una vez importado el perfil de la víctima, podremos acceder a sus sesiones durante el máximo tiempo que nos permita el servidor.

Lo único que tendremos que tener en cuenta, es que **algunas páginas (como Google o Microsoft)**, no permiten mantener la sesión a través de las cookies si las importamos a otro navegador, por lo que para poder realizar nuestras **maldades** auditorías, tendremos que hacerlo sin terminar la ejecución del script en ningún momento.

Antes de terminar este artículo, me gustaría decir que el mismo no hace justicia a la herramienta, ya que se entiende mucho mejor al verla en funcionamiento. Si tienes la oportunidad, pruébala y **verás que es mucho más impactante ver los movimientos de la víctima en tiempo real**.

Por último, la herramienta se encuentra en fase «beta» y es probable que se produzcan fallos en algunos escenarios. Si encuentras alguno, no dudes en hacérmelo saber 😊

Espero que os haya gustado y os resulte útil en vuestras próximas auditorías.

Nos vemos en la próxima!

Darkbyte © 2018 - 2025

Contenido protegido por [Creative Commons 4.0](#)