Unicode

Autor de la máquina: MrR3boot

Autor del writeup: CafeciToRoot

19 de Diciembre del 2021

1. Introducción

Unicode es una máquina Linux de dificultad media en la que tenemos que modificar un token JWT para llevar a cabo un secuestro de cuenta, sobrepasar un WAF para poder explotar una vulnerabilidad Local File Inclusion, y decompilar un binario para después entender cómo funciona y ganar acceso como root.

2. Reconocimiento y exploit inicial

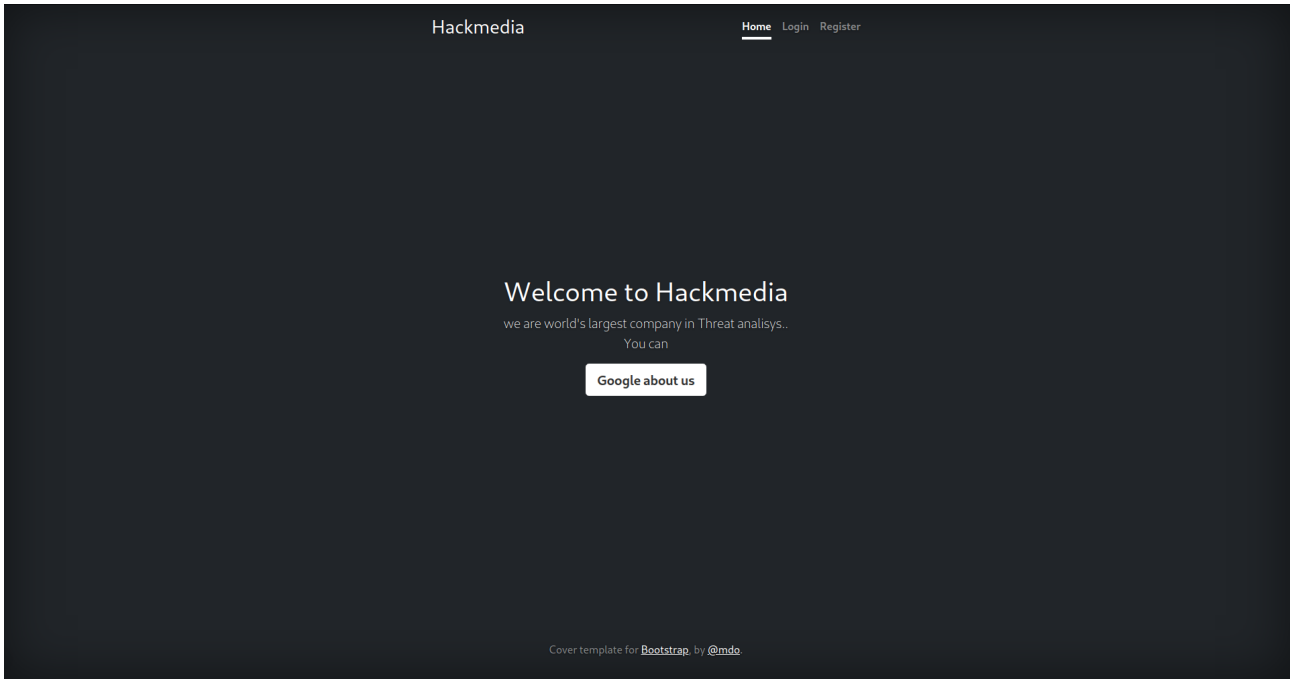
Empezamos escaneando la máquina en busca de puertos TCP abiertos utilizando nmap:

```
ports=$(nmap -p- --min-rate=1000 -T4 10.10.11.126 | grep ^[0-9] | cut -d '/'
    -f 1 | tr '\n' ',' | sed s/,$//)

nmap -p$ports -sV -sC 10.10.11.126
```

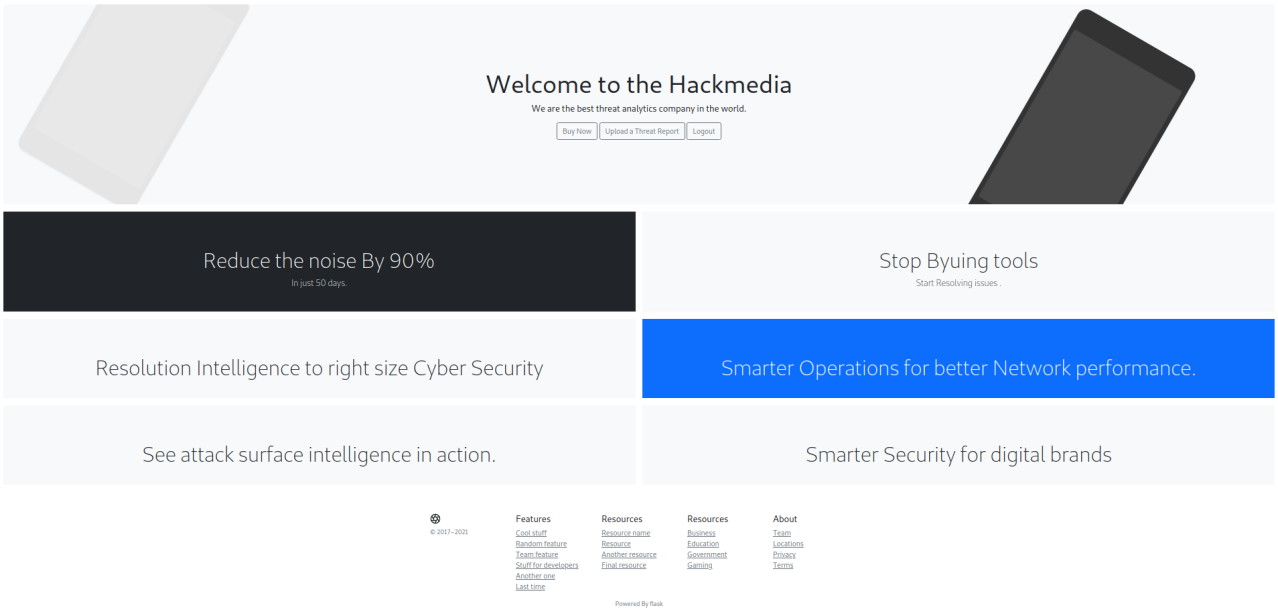
```
PORT   STATE SERVICE REASON  VERSION
22/tcp open  ssh     syn-ack OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 fd:a0:f7:93:9e:d3:cc:bd:c2:3c:7f:92:35:70:d7:77 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQC2tldCNzbxK22LE6gPT7x5wliLZL0gLpI9BHO8qz1I7EIL7ygZ0vLk7tZ5JtDFvkdJiMfsbluCGZT
pnHuFUY620uPgYEdfYu+RcTH97ldEyio6GKNkhGN+MRi8swttVWFr24sGGU4FEjhQTBG8/aivffqn+w0yksEIQCmXbh/y4xo5MBLeh/n0tMm67e/wrjUg
3Y8zDCXXNVpNWzZMtyR8cThY/adlk1F8TatvcHOzG/MC4Xg16B9qjJ1CzJmztbIHpRRe64ow9vdiO6ofyVroiazMkMaE6ltWEl5XC4rKurbD2DySFDUdR
r8QT3aAQpNYNPNU2Q9hJYUN1gKZAUCg0mMUmBIbQXyiKiq/b5JGGLPhUkoD6Pl2WjE60D+3ZnNqW8jabBMzUotwi6KdJ5v4HvJiNxNrzjQRpNCJ6rBhIF
OUqZQHBsDsFiyOSLXEPpYnNhG502TGELiHOFuK15QMh9CqCZn9PvwIiACTyeet9NdUyHtxHT8gklpnHdNSXY8=
|   256 8b:b6:98:2d:fa:00:e5:e2:9c:8f:af:0f:44:99:03:b1 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBNpOPaB/e8Wk54u5TO7EABpkTxMt6Mz10v02RBzyUPJ
yzpXi1eC8X2VvIpCngtg4Uvbv07ZEm72Tb9S6IoJ/8MI=
|   256 c9:89:27:3e:91:cb:51:27:6f:39:89:36:10:41:df:7c (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAID/Us0SuyKOvqEgu/XGps4fVOmhy+iczFKeIrtjRWxUN
80/tcp open  http    syn-ack nginx 1.18.0 (Ubuntu)
|_http-generator: Hugo 0.83.1
|_http-title: 503
|_http-favicon: Unknown favicon MD5: E06EE2ACCCCCD12A0FD09983B44FE9D9
| http-methods:
|_  Supported Methods: GET HEAD OPTIONS
|_http-server-header: nginx/1.18.0 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

La máquina solamente tiene abiertos los puertos TCP 22 (SSH) y 80 (HTTP). Abrimos nuestro navegador e introducimos la dirección IP para ver el servicio HTTP.



Una vez en la página principal, podemos navegar a un servicio de login (/login), uno de registro (/register) y uno llamado "Google about us", que mediante un open redirect nos lleva a google.com (/redirect?url=google.com).

Tras intentar explotar el open redirect para acceder a servicios o rutas internas sin éxito, nos registramos y hacemos login, lo que nos redirige al dashboard (/dashboard).



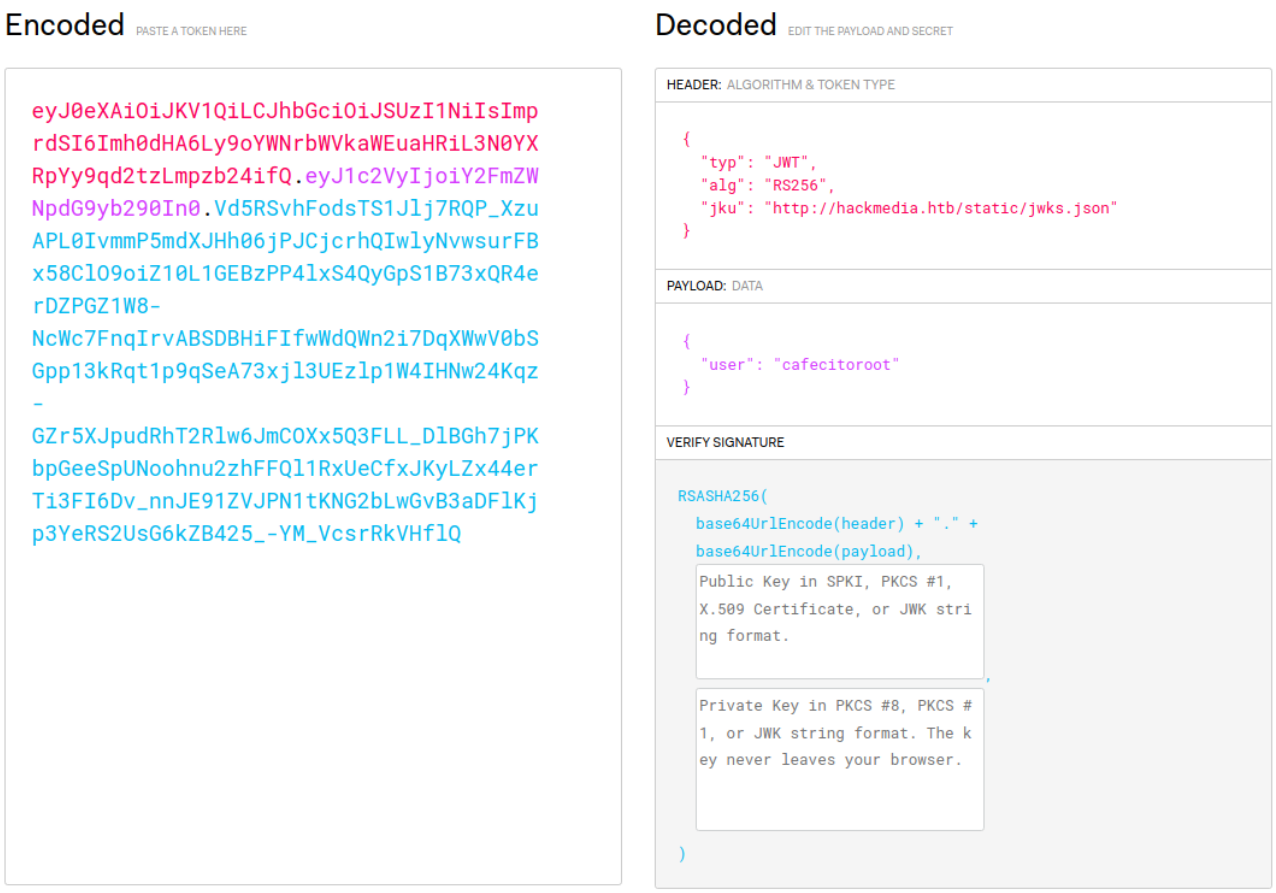
En el dashboard no hay nada interesante, salvo un servicio (/upload) que parece permitir la subida arbitraria de archivos llamado 'Upload a Threat Report'.

Al logearnos nos ha sido entregada una cookie de autenticación llamada 'auth', que contiene un JSON Web Token (JWT). Se trata de un tipo de token muy común, fácilmente reconocible

porque cumple con el formato "xxxxx.yyyyy.zzzzz", o "header,payload,signature". Para más información introducción a JWT.

```
auth=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImprdSI6Imh0dHA6Ly9oYWNrbWVkaWEuaH
RiL3N0YXRpYy9qd2tzLmpzb24ifQ.eyJ1c2VyIjoiY2FmZWNpdG9yb290In0.Vd5RSvhFodsTS1
Jlj7RQP_XzuAPL0IvmmP5mdXJHh06jPJCjcrhQIwlyNvwsurFBx58ClO9oiZ10L1GEBzPP4lxS4
QyGpS1B73xQR4erDZPGZ1W8-NcWc7FnqIrvABSDBHiFIfwWdQWn2i7DqXWwV0bSGpp13kRqt1p9
qSeA73xjl3UEzlp1W4IHNw24Kqz-GZr5XJpudRhT2Rlw6JmCOXx5Q3FLL_DlBGh7jPKbpGeeSpU
Noohnu2zhFFQl1RxUeCfxJKyLZx44erTi3FI6Dv_nnJE91ZVJPN1tKNG2bLwGvB3aDFlKjp3YeR
S2UsG6kZB425_-YM_VcsrRkVHflQ
```

En la página oficial de JWT podemos descifrar el contenido de la cookie:



Al descifrar la cookie vemos que hay un campo con la clave 'jku' y cuyo valor es una URI, buscando un poco de información sobre este campo llegamos a un blog de Shivam Bathla.

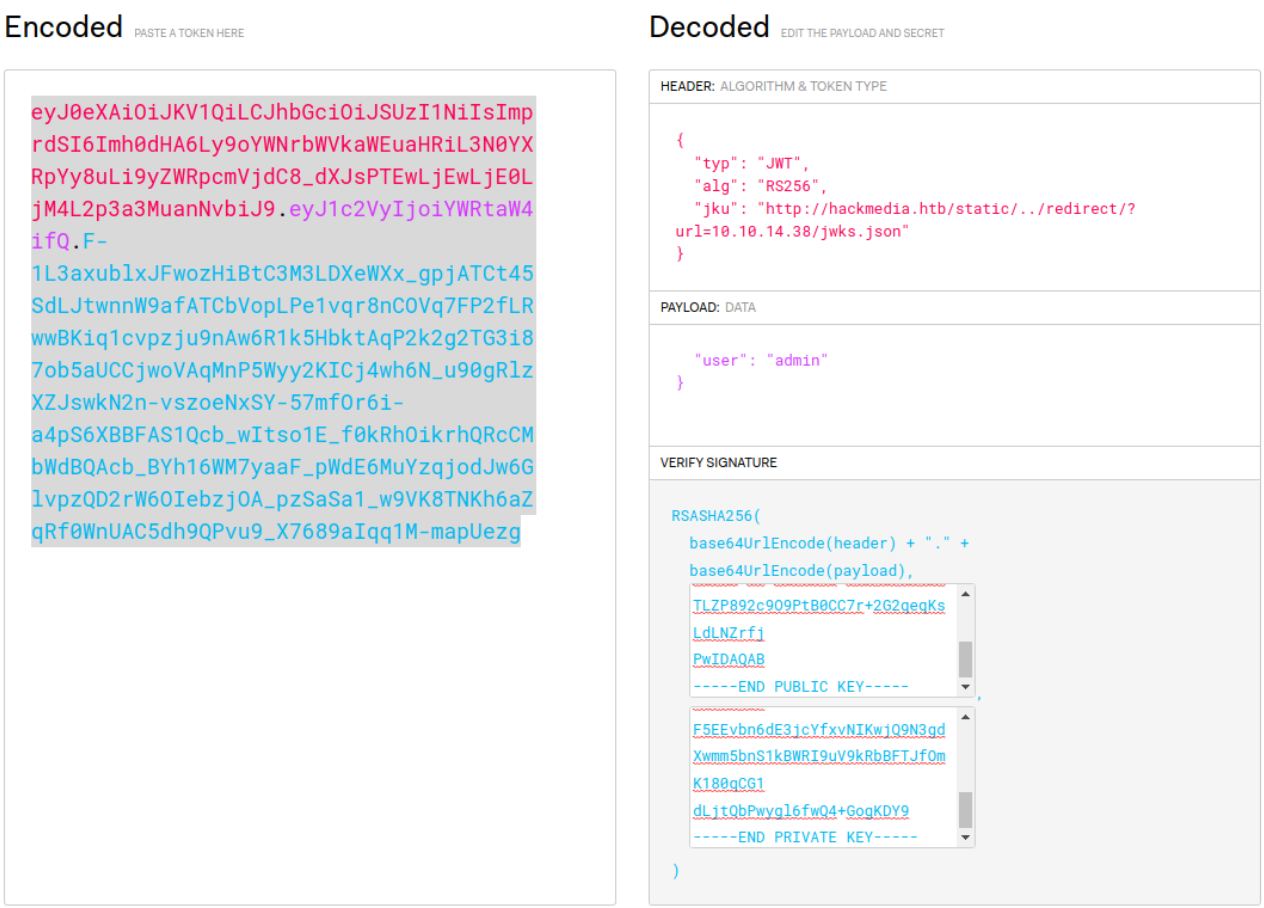
Al parecer, el valor de la clave 'jku' es una URI que contiene una colección de claves públicas en formato JSON, una de las cuales corresponde a la clave con la que se ha firmado el token.

Si logramos controlar o sustituir esta URI, podemos cambiar el token a nuestro placer y el servidor lo seguirá considerando válido.

Añadimos el dominio a nuestro archivo hosts, accedemos a la ruta y encontramos la colección de claves públicas, que en nuestro caso es solo una:

En nuestro caso, la URI tiene que pertenecer al dominio **hackmedia.htb/static/** para que el servidor la acepte, por lo que podemos abusar del open redirect que encontramos anteriormente para crear una URI que podamos controlar con el siguiente formato:
http://hackmedia.htb/static/../redirect/?url=IP /jwks.json

Ahora solo queda crear una par de claves con el mismo formato que la del servidor, algo que podemos hacer en este servicio, y ya podemos modificar el token. Utilizamos nuestras claves para firmar el token y cambiamos el nombre de usuario por 'admin', de modo que quede de la siguiente manera:



Después, nos descargamos el archivo jwks.json del servidor y sustituimos el valor del campo "n"por el de nuestra clave pública, alojamos el nuevo archivo en un servidor web, cambiamos

el valor de nuestra cookie, volvemos a acceder al servicio y ya somos el usuario admin, lo que nos lleva al siguiente dashboard:

3. Exploit y Escalada de privilegios

3.1. User

El dashboard del admin se trata de una página estática, cuyos enlaces tiene la siguiente forma:

http://hackmedia.htb/display/?page=monthly.pdf

Parece ser vulnerable a Local File Inclusion (LFI). Si intentamos explotar la vulnerabilidad, por ejemplo, accediendo al recurso:

http://hackmedia.htb/display/?page=/etc/passwd

Nos encontramos con un error que dice:

We do a lot input filtering you can never bypass our filters. Have a good day.

El servidor está filtrando las peticiones a recursos que no deberían ser accedidos. Ya que el nombre de la máquina es **Unicode**, probamos a codificar el payload con diferentes carácteres unicode, hasta que el U+FE30 funciona como sustituto de **".."**, por lo que esta URI nos lleva al archivo passwd de la máquina:

http://hackmedia.htb/display/?page=%EF%B8%B0/%EF%B8%B0/%EF%B8%B0/%EF%B8%B0/etc/passwd

Como sabemos, el servidor está ejecutando Nginx, por lo que probamos a acceder a los archivos de configuración más comunes de este servicio, hasta que damos con **/etc/nginx/sites-enabled/default**, que contiene información que puede ser de utilidad:

```
limit_req_zone $binary_remote_addr zone=mylimit:10m rate=800r/s; server{ #
   Change the Webroot from /home/code/app/ to /var/www/html/ #change the
   user password from db.yaml listen 80; error_page 503 /rate-limited/;
   location / { limit_req zone=mylimit; proxy_pass http://localhost:8000;
   include /etc/nginx/proxy_params; proxy_redirect off; } location /static/{
    alias /home/code/coder/static/styles/; } }
```

Encontramos algunos directorios de la máquina, y la existencia de un archivo db.yaml donde se almacenan credenciales. Después de probar unas cuantas combinaciones, parece que el arhivo se encuentra en la ruta:

http://hackmedia.htb/display/?page=%EF%B8%B0/%EF%B8%B0/%EF%B8%B0/%EF%B8%B0/code/coder/db.yaml

```
mysql_host: "localhost" mysql_user: "code" mysql_password: "B3stC0d3r2021@@
   !" mysql_db: "user"$
```

Probamos a logearnos con estas credenciales mediante ssh y conseguimos acceso como el usuario **code**, que tiene permiso de lectura para **user.txt**.

3.2. Root

test text asd