Wireshark IP Packet Analysis: File Upload using html and node.js

Bibek Timilsina

July 13,2024

Introduction:

This report analyzes an IP packet for a file upload captured using Wireshark.

Node.js File upload:

```
const upload = multer({ storage: storage });

// Create uploads folder if it doesn't exist
const fs = require('fs');
const dir = './uploads';

if (!fs.existsSync(dir)){
  fs.mkdirSync(dir);
}
app.get('/', (req, res) => {
    res.sendFile(path.join(__dirname, 'index.html'));
  });
// Handle file upload
app.post('/upload', upload.single('file'), (req, res) => {
    console.log(req)
    console.log(req.file)
  if (!req.file) {
    return res.status(400).send('No file uploaded.');
  }
  res.send(`File uploaded: ${req.file.filename}`);
});

app.listen(port, () => {
  console.log(`Server is running on http://localhost:${port}`);
});
```

Html for file upload:

```
<body>
  <h1>Upload a File</h1>
  <form id="uploadForm">
    <input type="file" id="fileInput" name="file" required>
    <button type="submit">Upload</button>
  </form>
  <div id="response"></div>

  <script>
    document.getElementById('uploadForm').addEventListener('submit', async (event) => {
      event.preventDefault();
      const fileInput = document.getElementById('fileInput');
      const formData = new FormData();
      formData.append('file', fileInput.files[0]);

      try {
        const response = await fetch('http://localhost:3000/upload', {
          method: 'POST',
          body: formData
        });
        const result = await response.text();
        document.getElementById('response').innerText = result;
      } catch (error) {
        console.error('Error uploading file:', error);
      }
    });
  </script>
</body>
</html>
```

Packet capture details:

The packet capture is performed using Wireshark with the following steps:

1. **Open Wireshark** and select the network interface used for local communication.(Adapter for loopback traffic capture in windows)
2. Start the capture.
3. Go to your browser and navigate to http://localhost:3000 and upload a file using the html form.
4. Stop packet capture.
5. Filter to capture only the HTTP POST requests by using http.request.method == "POST"

The capture packet is analyzed in the following sections:

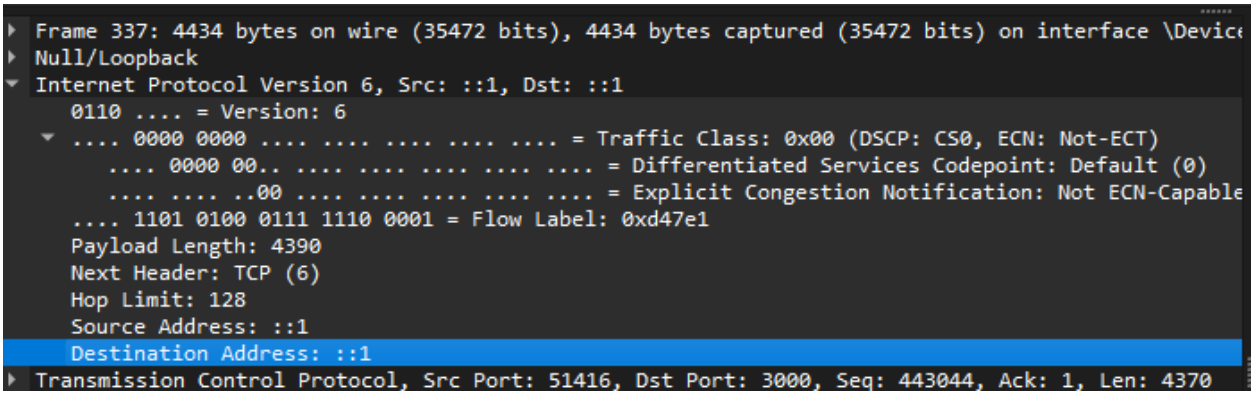

Figure: Headers Fields

Figure: Hex for Ip packet

IP Header Data:

60 0d 47 e1 11 26 06 80 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01

IP Header Analysis:

Field	Value(hex)	Length	Details
Version	6	4	IPv6 packet.
Traffic Class	00	8	Used to classify & prioritize packets. 0 for default traffic class
Flow Level	0d 47 e1	20	for special handling of particular flows of data.
Payload Length	11 62	12	Length of the payload following the IPv6 header (1162 bytes).
Next Header	06	8	Indicates the type of header following the IPv6 header
Hop Limit	80	8	Maximum number of hops the packet can traverse (similar to TTL in IPv4).
Source Address	::1	128	IPv6 address of the packet's source (loopback address).
Destination Address	::1	128	IPv6 address of the packet's destination (loopback address).

TCP Header Data:

Ce 5c 0b b8 c5 a2 19 fd e8 62 c3 97 50 18 27 f6 09 af 00 00

TCP Header details:

field Name	Length (bits)	Value	Explanation
Source Port	16	51416	Port number from which the packet was sent on the source device.
Destination Port	16	3000	Port number to which the packet is

			being sent on the destination device.
Sequence Number	32	443044	Sequence number of the first byte of data in this segment.
Acknowledgment Number	32	1	Next sequence number the sender expects to receive (if ACK flag is set).
Data Offset	4	5	Indicates where the data begins; number of 32-bit words in the TCP header.
Reserved	3	0	Reserved for future use; should be set to zero.
Flags	9	Various	Control flags (e.g., SYN, ACK, FIN, RST).
Window Size	16	4370	Size of the sender's receive window (buffer space available).
Checksum	16	Varies	Used for error-checking the header and data.
Urgent Pointer	16	0	Indicates the offset from the sequence number where the urgent data is (if URG flag is set).