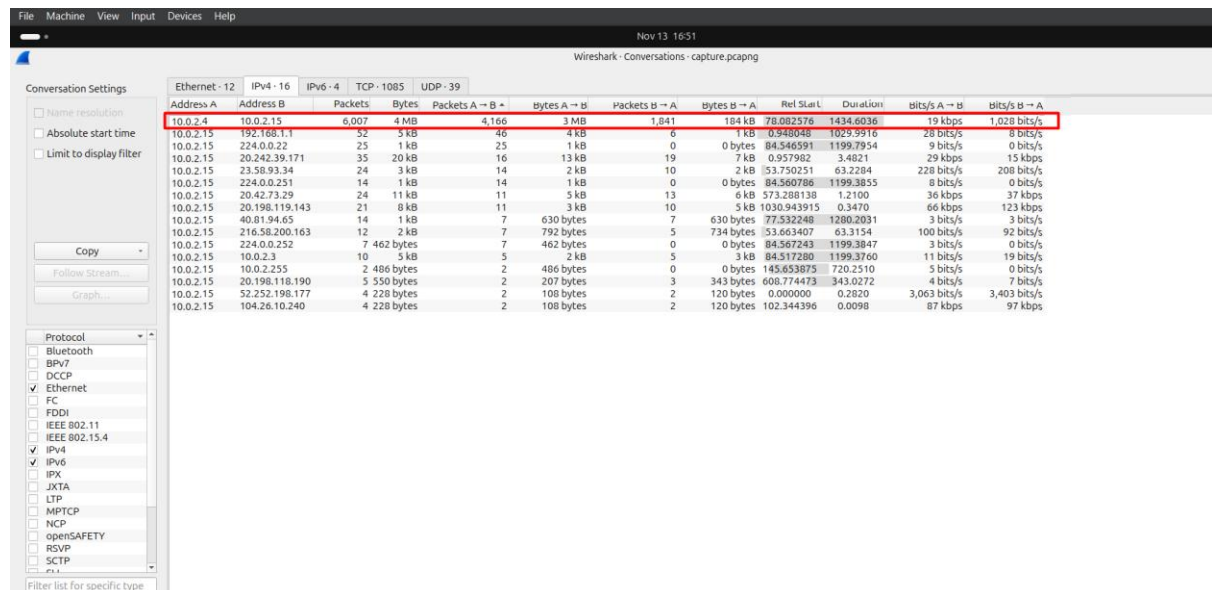


## ASPX reverse shell via SMB

In this pcap file, we can see that ASPX reverse shell was planted via SMB protocol.

First step of investigating the pcap file, we have filtered the IP addresses in Wireshark as:

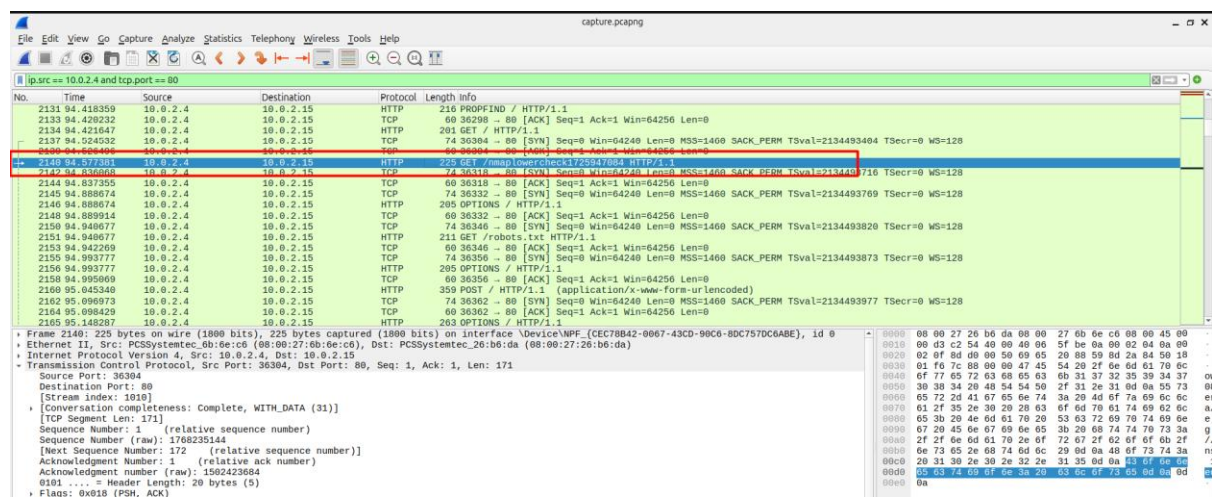


No.	Time	Source	Destination	Protocol	Length	Info
2131	94.418359	10.0.2.4	10.0.2.15	HTTP	216	PROPFIND / HTTP/1.1
2133	94.420232	10.0.2.4	10.0.2.15	TCP	60	36296 → 80 [ACK] Seq=1 Win=64256 Len=0
2134	94.421647	10.0.2.4	10.0.2.15	HTTP	201	GET / HTTP/1.1
2137	94.024532	10.0.2.4	10.0.2.15	TCP	74	36304 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2134493484 TSecr=0 WS=128
2140	94.577381	10.0.2.4	10.0.2.15	HTTP	225	GET /nmapLowercheck1725947684 HTTP/1.1
2142	94.636968	10.0.2.4	10.0.2.15	TCP	74	36316 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2134493716 TSecr=0 WS=128
2144	94.637355	10.0.2.4	10.0.2.15	TCP	60	36316 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0
2145	94.888674	10.0.2.4	10.0.2.15	TCP	74	36332 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2134493769 TSecr=0 WS=128
2146	94.888674	10.0.2.4	10.0.2.15	HTTP	205	OPTIONS / HTTP/1.1
2148	94.889914	10.0.2.4	10.0.2.15	TCP	60	36332 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0
2150	94.940677	10.0.2.4	10.0.2.15	TCP	74	36346 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2134493820 TSecr=0 WS=128
2151	94.940677	10.0.2.4	10.0.2.15	HTTP	211	GET /robots.txt HTTP/1.1
2153	94.942269	10.0.2.4	10.0.2.15	TCP	60	36346 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0
2155	94.993777	10.0.2.4	10.0.2.15	TCP	74	36356 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2134493873 TSecr=0 WS=128
2156	94.993777	10.0.2.4	10.0.2.15	HTTP	205	OPTIONS / HTTP/1.1
2158	94.995909	10.0.2.4	10.0.2.15	TCP	60	36356 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0
2160	95.045348	10.0.2.4	10.0.2.15	HTTP	359	POST / HTTP/1.1 (application/x-www-form-urlencoded)
2162	95.096973	10.0.2.4	10.0.2.15	TCP	74	36362 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2134493977 TSecr=0 WS=128
2164	95.098429	10.0.2.4	10.0.2.15	TCP	60	36362 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0
2165	95.148287	10.0.2.4	10.0.2.15	HTTP	263	OPTIONS / HTTP/1.1

Lots of conversation is happening between these two IP addresses, so let's search our filter.

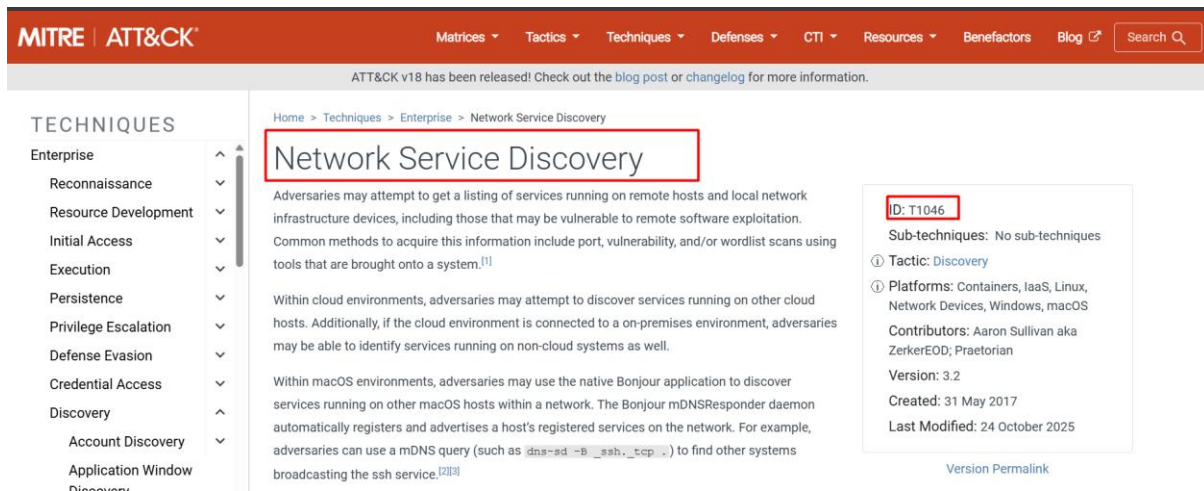
Filter: ip.src == 10.0.2.4 and tcp.port == 80

By applying this filter, we can see nmap, which is popular tool for network discovery and security auditing. By this we can see all the open ports in the target host. It finds the open services like SMP, HTTP or reverse shell paths.



No.	Time	Source	Destination	Protocol	Length	Info
2131	94.418359	10.0.2.4	10.0.2.15	HTTP	216	PROPFIND / HTTP/1.1
2133	94.420232	10.0.2.4	10.0.2.15	TCP	60	36296 → 80 [ACK] Seq=1 Win=64256 Len=0
2134	94.421647	10.0.2.4	10.0.2.15	HTTP	201	GET / HTTP/1.1
2137	94.024532	10.0.2.4	10.0.2.15	TCP	74	36304 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2134493484 TSecr=0 WS=128
2140	94.577381	10.0.2.4	10.0.2.15	HTTP	225	GET /nmapLowercheck1725947684 HTTP/1.1
2142	94.636968	10.0.2.4	10.0.2.15	TCP	74	36316 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2134493716 TSecr=0 WS=128
2144	94.637355	10.0.2.4	10.0.2.15	TCP	60	36316 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0
2145	94.888674	10.0.2.4	10.0.2.15	TCP	74	36332 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2134493769 TSecr=0 WS=128
2146	94.888674	10.0.2.4	10.0.2.15	HTTP	205	OPTIONS / HTTP/1.1
2148	94.889914	10.0.2.4	10.0.2.15	TCP	60	36332 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0
2150	94.940677	10.0.2.4	10.0.2.15	TCP	74	36346 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2134493820 TSecr=0 WS=128
2151	94.940677	10.0.2.4	10.0.2.15	HTTP	211	GET /robots.txt HTTP/1.1
2153	94.942269	10.0.2.4	10.0.2.15	TCP	60	36346 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0
2155	94.993777	10.0.2.4	10.0.2.15	TCP	74	36356 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2134493873 TSecr=0 WS=128
2156	94.993777	10.0.2.4	10.0.2.15	HTTP	205	OPTIONS / HTTP/1.1
2158	94.995909	10.0.2.4	10.0.2.15	TCP	60	36356 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0
2160	95.045348	10.0.2.4	10.0.2.15	HTTP	359	POST / HTTP/1.1 (application/x-www-form-urlencoded)
2162	95.096973	10.0.2.4	10.0.2.15	TCP	74	36362 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2134493977 TSecr=0 WS=128
2164	95.098429	10.0.2.4	10.0.2.15	TCP	60	36362 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0
2165	95.148287	10.0.2.4	10.0.2.15	HTTP	263	OPTIONS / HTTP/1.1

## ASPX reverse shell via SMB



The screenshot shows the MITRE ATT&CK framework page for the technique "Network Service Discovery" (ID: T1046). The page is part of the "Enterprise" category under "Techniques". The description states: "Adversaries may attempt to get a listing of services running on remote hosts and local network infrastructure devices, including those that may be vulnerable to remote software exploitation. Common methods to acquire this information include port, vulnerability, and/or wordlist scans using tools that are brought onto a system.<sup>[1]</sup>" It also mentions that within cloud environments, adversaries may attempt to discover services running on other cloud hosts, and within macOS environments, they may use the native Bonjour application to discover services. The page includes a sidebar with a list of techniques, a search bar, and a right-hand panel with details about the technique, including its sub-techniques, tactic (Discovery), platforms (Containers, IaaS, Linux, Network Devices, Windows, macOS), contributors (Aaron Sullivan aka ZerkerEOD, Praetorian), version (3.2), creation date (31 May 2017), and last modified date (24 October 2025).

So, the source address 10.0.2.4 is scanning the web server 10.0.2.15. This is attacker is requesting <http://10.0.2.15/nmaplowercheck1725947084> and in response the server is giving 404 Not found.



The screenshot shows a Wireshark packet capture of an HTTP 404 response. The packet is labeled "GET /nmaplowercheck1725947084 HTTP/1.1" and the response is "HTTP/1.1 404 Not Found". The response body contains HTML code for a 404 error page, including a title "404 - File or directory not found." and a style block with a background color of #EEEEEE.

Exploring the protocol involved.

## ASPX reverse shell via SMB

Protocol	Percent	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Internet Control Message Protocol v6	0.4	26	0.0	708	3	26	708	3	26
Internet Protocol Version 4	98.4	6260	3.4	125300	662	0	0	0	6260
User Datagram Protocol	1.7	107	0.0	856	4	0	0	0	107
Network Time Protocol	0.2	14	0.0	672	3	14	672	3	14
NetBIOS Name Service	0.1	4	0.0	414	2	4	414	2	4
NetBIOS Datagram Service	0.0	2	0.0	402	2	0	0	0	2
SMB (Server Message Block Protocol)	0.0	2	0.0	238	1	0	0	0	2
SMB MailSlot Protocol	0.0	2	0.0	50	0	0	0	0	2
Microsoft Windows Browser Protocol	0.0	2	0.0	66	0	2	66	0	2
Multicast Domain Name System	0.2	14	0.0	686	3	14	686	3	14
Link-local Multicast Name Resolution	0.1	7	0.0	168	0	7	168	0	7
Dynamic Host Configuration Protocol	0.2	10	0.1	4240	22	10	4240	22	10
Domain Name System	0.8	52	0.1	2472	13	52	2472	13	52
Data	0.1	4	0.0	30	0	4	30	0	4
Transmission Control Protocol	96.3	6125	93.7	3483334	18 k	3811	1147128	6,066	6125
Transport Layer Security	0.4	24	0.9	34744	183	34	34744	183	34
NetBIOS Session Service	0.4	535	28.5	1059890	5,605	34	1660	8	535
SMB2 (Server Message Block Protocol version 2)	7.7	492	28.4	1055378	5,581	486	38916	205	492
Distributed Computing Environment / Remote Procedure Call (DCE/RPC)	0.1	4	0.0	632	3	2	140	0	4
Server Service	0.0	2	0.0	444	2	2	444	2	2
SMB (Server Message Block Protocol)	0.1	8	0.0	735	3	8	735	3	8
Data	0.0	3	27.3	1015287	5,369	3	1015287	5,369	3
Hypertext Transfer Protocol	1.1	70	0.8	30447	161	54	11869	62	70
Line-based text data	0.2	15	0.4	15211	80	15	15211	80	15
HTML Form URL Encoded	0.0	1	0.0	88	0	1	88	0	1
Data	26.3	1675	58.9	2189765	11 k	1675	2189765	11 k	1675
Internet Group Management Protocol	0.4	25	0.0	400	2	25	400	2	25
Internet Control Message Protocol	0.0	3	0.0	124	0	0	0	0	3
Data	0.0	3	0.0	16	0	3	16	0	3
Address Resolution Protocol	0.8	52	0.1	1924	10	52	1924	10	52

In the protocol hierarchy. SMB2 is there so, filtering with SMB2

ip.src == 10.0.2.4 && smb2

No.	Time	Source	Destination	Protocol	Length	Info
2388	137.932590	10.0.2.4	10.0.2.15	SMB2	238	Negotiate Protocol Request
2397	138.041107	10.0.2.4	10.0.2.15	SMB2	238	Negotiate Protocol Request
2618	239.339908	10.0.2.4	10.0.2.15	SMB2	284	Negotiate Protocol Request
2622	240.674813	10.0.2.4	10.0.2.15	SMB2	220	Session Setup Request, NTLMSSP_NEGOTIATE
2626	240.730522	10.0.2.4	10.0.2.15	SMB2	580	Session Setup Request, NTLMSSP_AUTH, User: WORKGROUP\root
2629	240.778552	10.0.2.4	10.0.2.15	SMB2	162	Tree Connect Request Tree: \\10.0.2.15\IPC\$
2631	240.788624	10.0.2.4	10.0.2.15	SMB2	190	Create Request File: srvsvc
2633	240.821279	10.0.2.4	10.0.2.15	DCERPC	250	Bind: call_id: 1, Fragment: Single, 1 context items: SRVSVC V3.0 (32bit NDR)
2636	240.874425	10.0.2.4	10.0.2.15	SRVSVC	270	NetShareEnumAll request
2639	240.882214	10.0.2.4	10.0.2.15	SMB2	146	Close Request File: srvsvc
2641	240.885199	10.0.2.4	10.0.2.15	SMB2	126	Tree Disconnect Request
2666	263.287025	10.0.2.4	10.0.2.15	SMB2	284	Negotiate Protocol Request
2668	263.293534	10.0.2.4	10.0.2.15	SMB2	220	Session Setup Request, NTLMSSP_NEGOTIATE
2678	263.297878	10.0.2.4	10.0.2.15	SMB2	580	Session Setup Request, NTLMSSP_AUTH, User: WORKGROUP\root
2672	263.304629	10.0.2.4	10.0.2.15	SMB2	162	Tree Connect Request Tree: \\10.0.2.15\IPC\$
2674	263.306977	10.0.2.4	10.0.2.15	SMB2	222	Ioctl Request FSCTL_DFS_GET_REFERRALS, File: \\10.0.2.15\Documents
2676	263.309040	10.0.2.4	10.0.2.15	SMB2	146	Tree Disconnect Request
2678	263.312045	10.0.2.4	10.0.2.15	SMB2	172	Tree Connect Request Tree: \\10.0.2.15\Documents
2681	263.440229	10.0.2.4	10.0.2.15	SMB2	146	KeepAlive Request
2684	265.805134	10.0.2.4	10.0.2.15	SMB2	179	Create Request File:
2687	265.807184	10.0.2.4	10.0.2.15	SMB2	156	Find Request File: SMB2_FIND_ID_BOTH_DIRECTORY_INFO Pattern: *
2689	265.810289	10.0.2.4	10.0.2.15	SMB2	156	Find Request File: SMB2_FIND_ID_BOTH_DIRECTORY_INFO Pattern: *
2691	265.815482	10.0.2.4	10.0.2.15	SMB2	146	Close Request File:
2693	265.817348	10.0.2.4	10.0.2.15	SMB2	179	Create Request File:
2695	265.822787	10.0.2.4	10.0.2.15	SMB2	163	GetInfo Request FS: INFO/FileSizeInformation File:
2697	265.826264	10.0.2.4	10.0.2.15	SMB2	146	Close Request File:
2701	267.847858	10.0.2.4	10.0.2.15	SMB2	126	KeepAlive Request
2704	272.391299	10.0.2.4	10.0.2.15	SMB2	179	Create Request File:
2707	272.394643	10.0.2.4	10.0.2.15	SMB2	156	Find Request File: SMB2_FIND_ID_BOTH_DIRECTORY_INFO Pattern: *
2709	272.403711	10.0.2.4	10.0.2.15	SMB2	156	Find Request File: SMB2_FIND_ID_BOTH_DIRECTORY_INFO Pattern: *
2711	272.415666	10.0.2.4	10.0.2.15	SMB2	146	Close Request File:
2714	272.932287	10.0.2.4	10.0.2.15	SMB2	126	KeepAlive Request
2768	327.835747	10.0.2.4	10.0.2.15	SMB2	126	KeepAlive Request
2773	332.894727	10.0.2.4	10.0.2.15	SMB2	126	KeepAlive Request
2778	337.991332	10.0.2.4	10.0.2.15	SMB2	126	KeepAlive Request
2783	342.482279	10.0.2.4	10.0.2.15	SMB2	198	Create Request File: shell.aspx
3505	342.611215	10.0.2.4	10.0.2.15	SMB2	494	Write Request Len:1015024 Off:0 File: shell.aspx
3509	342.614219	10.0.2.4	10.0.2.15	SMB2	146	Close Request File: shell.aspx
3514	342.919228	10.0.2.4	10.0.2.15	SMB2	126	KeepAlive Request
3517	347.884043	10.0.2.4	10.0.2.15	SMB2	126	KeepAlive Request
3523	352.846514	10.0.2.4	10.0.2.15	SMB2	126	KeepAlive Request
3526	357.907498	10.0.2.4	10.0.2.15	SMB2	126	KeepAlive Request
3531	362.872570	10.0.2.4	10.0.2.15	SMB2	126	KeepAlive Request
3534	367.918896	10.0.2.4	10.0.2.15	SMB2	126	KeepAlive Request

Establishment of tree connect request

- When a windows client wants to access a share on a remote machine via SMB, it first sends a tree request to specify the share it want to access.
- \\10.0.2.15\IPC\$ → the **administrative IPC share**, often used for remote management and named pipes.

## ASPX reverse shell via SMB

- `\\10.0.2.15\Documents` → a **regular document share**, where the attacker is trying to store the malicious file.

Along with that, there is a create / write request.

Once connected to the share:

- **Create Request** → tells the SMB server:

“I want to create a new file named `shell.aspx`.”

- **Write Request** → actually **writes the content** of the file (your malicious ASPX shell) into the newly created file on the server.

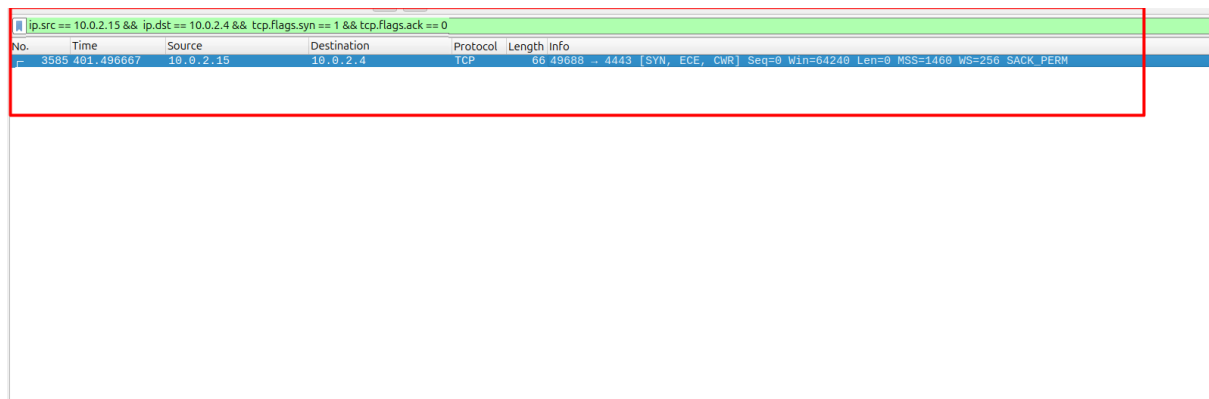
After the write, the malicious file now exists on the server: `Documents\shell.aspx`.

### Port used:

```
ip.src == 10.0.2.15 && ip.dst == 10.0.2.4 && tcp.flags.syn == 1 && tcp.flags.ack == 0
```

SYN flag is set to 1 → this packet is **trying to start a new TCP connection**

ACK flag is 0 → the packet is **not a reply**; it's the first packet in the handshake. This is the very **first message from the server to the attacker** when opening the connection.



The image shows a Wireshark packet capture window. The top packet list pane shows a single packet (No. 3585) at time 481.496667, from source 10.0.2.15 to destination 10.0.2.4, protocol TCP, length 66. The packet details pane shows the TCP header with sequence number 4443, window size 64240, and flags SYN, ECE, CWR. The packet bytes pane is empty. A red rectangle highlights the packet list and details panes.

No.	Time	Source	Destination	Protocol	Length	Info
3585	481.496667	10.0.2.15	10.0.2.4	TCP	66	49688 → 4443 [SYN, ECE, CWR] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM

4443 listening port number is used by attacker for the reverse shell.

### Execution:

Once the malicious `shell.aspx` is on the server:

- The attacker can access it via HTTP on the webserver (or IIS) by navigating to `http://10.0.2.15/Documents/shell.aspx`.
- When visited, the shell executes code on the server —, it sets up a **reverse shell**.

## ASPX reverse shell via SMB

- Then the server initiates an **outbound TCP connection** to the attacker, connecting to the attacker's listening port.

Inside of shell.aspx

[illegible]

Certainly, this is a malicious shell execution code

### Steps Involved:

1. SMB Access: Tree Connect Request → connects to Documents share
2. Planting Shell: Create Request + Write Request → uploads shell.aspx
3. Triggering Shell: HTTP request to shell.aspx → executes malicious code
4. Reverse Shell Connections: Server (10.0.2.15) initiates TCP connection → attacker (10.0.2.4) listening port.