

## **Tcpdump Lab**

### **Introduction:**

**tcpdump** - is a command-line utility used primarily for packet analysis. It captures packet data from network interfaces and can use the Berkley Packet Filter (BPF) syntax. The key point to remember here is that it is a lightweight packet analysis tool and it is a common suite used by many organizations to conduct packet captures (PCAPs).

### **Some of the most common parameters used with tcpdump:**

- #: Display line/packet number
- c count: Number of packets to capture before tcpdump automatically exits.
- D: Display interfaces
- e: Display Ethernet header data.
- expression: Specify a Berkeley Packet Filter (BPF) statement to filter traffic.
- i interface: Specify from which network interface you would like tcpdump to sniff. This generally requires administrative permissions.
- n: Don't resolve hostnames or well-known port numbers to their service.
- r file: Specify an existing pcap file to read from instead of a network interface.
- s snaplen: Snapshot length, or the number of bytes to capture per packet. Default is 262,144 bytes but this may vary across platforms.
- w file: Specify a new pcap file to place filtered packets in.
- X: Show packet contents in hexadecimal and ASCII.
- v : Display verbose output

### **Task 1: Examining Packet Headers**

1.1) We will start out by using tcpdump to examine packet headers. For example, if we wanted to display headers and line numbers of the first 20 packets, we would type the following command

**Command:** `tcpdump -n -r investigate.pcap -c 20 -#`

```
sec401@slingshot: /sec401/labs/1.1
File Edit View Search Terminal Help
sec401@slingshot: /sec401/labs/1.1$ tcpdump -n -r investigate.pcap -c 20 -#
reading from file investigate.pcap, link-type EN10MB (Ethernet), snapshot length 262144
 1 21:23:57.196268 IP 10.130.8.94.57810 > 10.130.8.2.53: 44934+ [1au] PTR? 94.8.130.10.in-addr.arpa. (53)
 2 21:23:57.197180 IP 10.130.8.2.53 > 10.130.8.94.57810: 44934 1/0/1 PTR ip-10-130-8-94.us-east-2.compute.internal. (108)
 3 06:01:58.456061 IP 10.130.8.94.33878 > 10.130.8.2.53: 53095+ [1au] SRV? _http._tcp.us-east-2.ec2.archive.ubuntu.com. (72)
 4 06:01:58.456899 IP 10.130.8.2.53 > 10.130.8.94.33878: 53095 NXDomain 0/1/1 (133)
 5 06:01:58.456931 IP 10.130.8.94.33878 > 10.130.8.2.53: 53095+ SRV? _http._tcp.us-east-2.ec2.archive.ubuntu.com. (61)
 6 06:01:58.457520 IP 10.130.8.2.53 > 10.130.8.94.33878: 53095 NXDomain 0/1/0 (122)
 7 15:25:36.329777 IP 135.125.217.54.44366 > 10.130.8.94.80: Flags [S], seq 250837459, win 29200, options [mss 1460,sackOK,TS val 1851076389,
 8 15:25:36.329777 IP 10.130.8.94.80 > 135.125.217.54.44366: Flags [S.], seq 788644517, ack 250837460, win 62643, options [mss 1460,sackOK,TS val 1851076389,
 9 15:25:36.432967 IP 135.125.217.54.44366 > 10.130.8.94.80: Flags [.], ack 1, win 229, options [nop,nop,TS val 1851076270 ecr 221043610]
10 15:25:36.434871 IP 135.125.217.54.44366 > 10.130.8.94.80: Flags [P.], seq 1:231, ack 1, win 229, options [nop,nop,TS val 1851076272 ecr 221043610]
11 15:25:36.434930 IP 10.130.8.94.80 > 135.125.217.54.44366: Flags [.], ack 231, win 488, options [nop,nop,TS val 221043610 ecr 1851076389]
12 15:25:36.435167 IP 10.130.8.94.80 > 135.125.217.54.44366: Flags [P.], seq 1:492, ack 231, win 488, options [nop,nop,TS val 221043610 ecr 1851076389]
13 15:25:36.538373 IP 135.125.217.54.44366 > 10.130.8.94.80: Flags [.], ack 492, win 237, options [nop,nop,TS val 1851076376 ecr 221043610]
14 15:25:36.551502 IP 135.125.217.54.44366 > 10.130.8.94.80: Flags [F.], seq 231, ack 492, win 237, options [nop,nop,TS val 1851076389 ecr 221043610]
15 15:25:36.551627 IP 10.130.8.94.80 > 135.125.217.54.44366: Flags [F.], seq 492, ack 232, win 488, options [nop,nop,TS val 221043727 ecr 1851076389]
16 15:25:36.654979 IP 135.125.217.54.44366 > 10.130.8.94.80: Flags [.], ack 493, win 237, options [nop,nop,TS val 1851076492 ecr 221043610]
17 18:46:03.070623 IP 20.106.124.93.44366 > 10.130.8.94.80: Flags [S], seq 1789817114, win 64240, options [mss 1440,sackOK,TS val 191477177 ecr 221043610]
18 18:46:03.070663 IP 10.130.8.94.80 > 20.106.124.93.44366: Flags [S.], seq 902963796, ack 1789817115, win 62643, options [mss 1460,sackOK,TS val 1851076389,
19 18:46:03.128923 IP 20.106.124.93.44366 > 10.130.8.94.80: Flags [.], ack 1, win 502, options [nop,nop,TS val 191477177 ecr 221917516]
20 18:46:03.128987 IP 20.106.124.93.44366 > 10.130.8.94.80: Flags [P.], seq 1:315, ack 1, win 502, options [nop,nop,TS val 191477177 ecr 221917516]
sec401@slingshot: /sec401/labs/1.1$
```

**Notes:** (Part of the image is cut off, but for the sake of this example, we will focus on the command input that gave us the output above.)

- a.) The “tcpdump” command above tells the system that we want to use tcpdump
- b.) The “-n” command tells the system that we do NOT want hostnames or well-known port numbers resolved to their services.
- c.) “-r investigate.pcap” tells the system we want to open this file. When you type “-r” remember to specify the file you want to open immediately after.
- d.) -c 20” Tell the system that we want to specify the number of packets to display. In this example, we chose to display 20 packets.
- e.) “-#” is used to number the packets neatly and clearly so that they are easier to read.

1.2) In the output of the previous step, each line represents a single packet and shows the details associated with it. The following chart will help you understand how to read the output of these packets.

Field Output	Description
1-10	Line number/Packet number
21:23:57.196268	Timestamp
IP	(Layer 3) Protocol being used

10.130.8.94	Source IP address
57810	Source port
>	Data flow indicator
10.130.8.2	Destination IP address
53	Destination port
44934+ [1au] PTR? 94.8.130.10.in-addr.arpa.	DNS information

Reference:

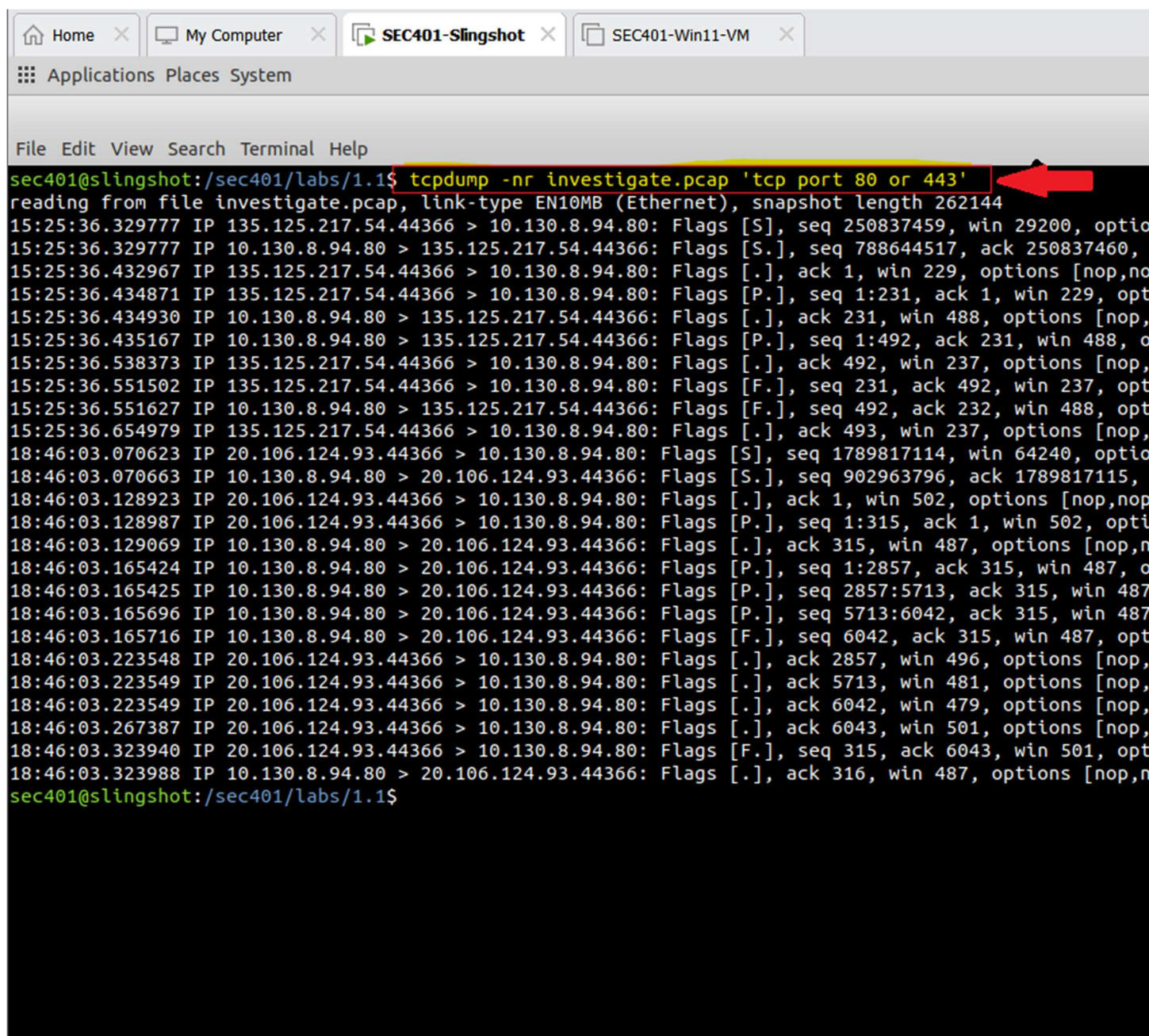
- Tcpcdump. “man page” <https://www.tcpdump.org/manpages/tcpdump.1.html>

## **Task 2: Filtering Traffic from Packet Captures (pcaps)**

2.1) To filter packets and achieve more in-depth granular control, we will use the Berkeley Packet Filter (BPF) method to identify specific hosts and/or protocols to which traffic is going to or coming from. The BPF is essentially a form of a network “tap” that filters traffic/packets based on its parameters between layers 2-4. It can work in conjunction with tcpdump to provide extreme filtration and precision. It is a great tool for intrusion detection analysis.

a.) Let's say that we want TCP traffic from ports 80 or 443. We would type the following command

**Command:** `tcpdump -n -r investigate.pcap 'tcp port 80 or 443'`

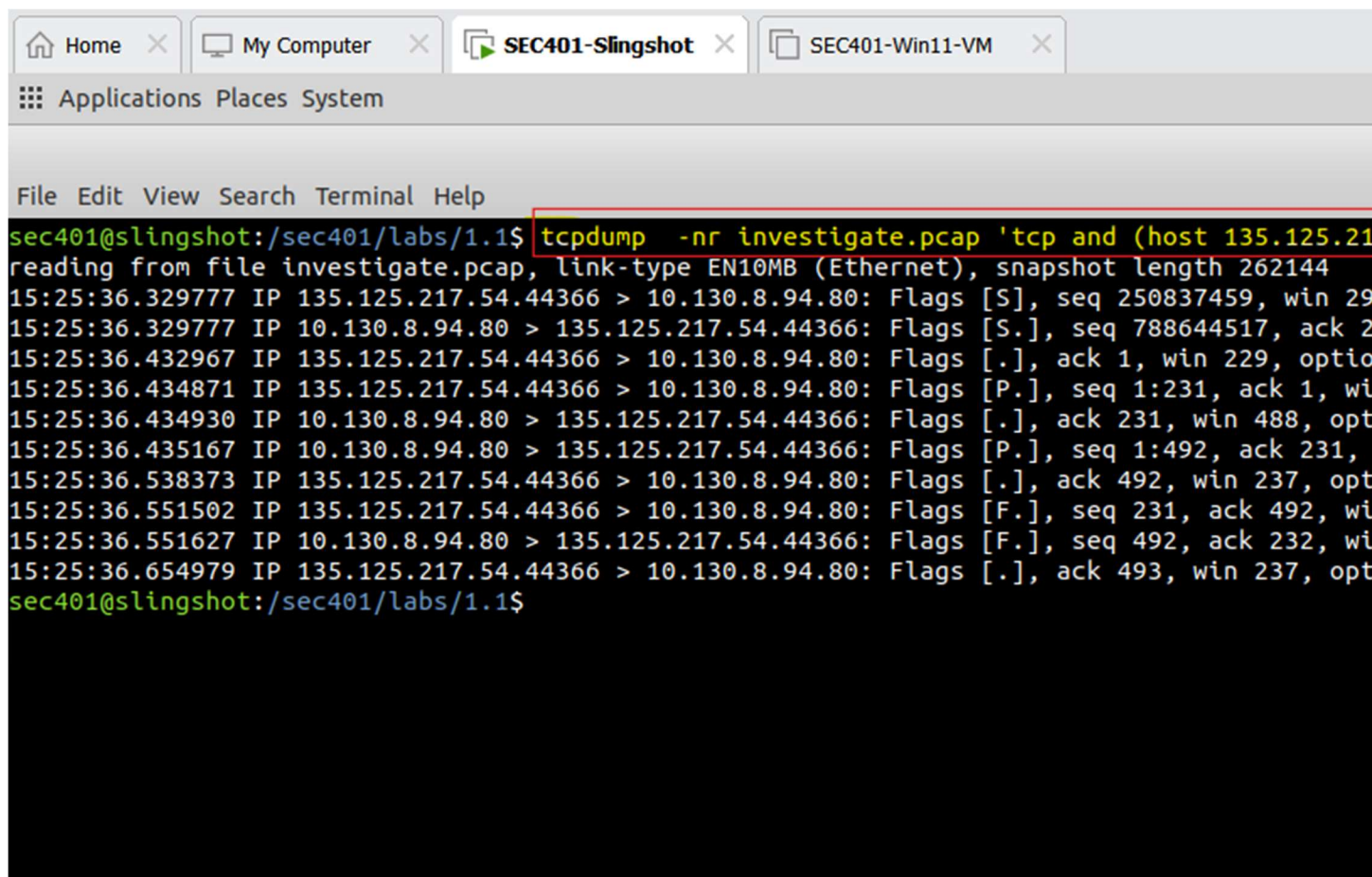


```
sec401@slingshot:/sec401/labs/1.1$ tcpdump -nr investigate.pcap 'tcp port 80 or 443'
reading from file investigate.pcap, link-type EN10MB (Ethernet), snapshot length 262144
15:25:36.329777 IP 135.125.217.54.44366 > 10.130.8.94.80: Flags [S], seq 250837459, win 29200, optio
15:25:36.329777 IP 10.130.8.94.80 > 135.125.217.54.44366: Flags [S.], seq 788644517, ack 250837460,
15:25:36.432967 IP 135.125.217.54.44366 > 10.130.8.94.80: Flags [.], ack 1, win 229, options [nop,no
15:25:36.434871 IP 135.125.217.54.44366 > 10.130.8.94.80: Flags [P.], seq 1:231, ack 1, win 229, opt
15:25:36.434930 IP 10.130.8.94.80 > 135.125.217.54.44366: Flags [.], ack 231, win 488, options [nop,
15:25:36.435167 IP 10.130.8.94.80 > 135.125.217.54.44366: Flags [P.], seq 1:492, ack 231, win 488, o
15:25:36.538373 IP 135.125.217.54.44366 > 10.130.8.94.80: Flags [.], ack 492, win 237, options [nop,
15:25:36.551502 IP 135.125.217.54.44366 > 10.130.8.94.80: Flags [F.], seq 231, ack 492, win 237, opt
15:25:36.551627 IP 10.130.8.94.80 > 135.125.217.54.44366: Flags [F.], seq 492, ack 232, win 488, opt
15:25:36.654979 IP 135.125.217.54.44366 > 10.130.8.94.80: Flags [.], ack 493, win 237, options [nop,
18:46:03.070623 IP 20.106.124.93.44366 > 10.130.8.94.80: Flags [S], seq 1789817114, win 64240, optio
18:46:03.070663 IP 10.130.8.94.80 > 20.106.124.93.44366: Flags [S.], seq 902963796, ack 1789817115,
18:46:03.128923 IP 20.106.124.93.44366 > 10.130.8.94.80: Flags [.], ack 1, win 502, options [nop,nop
18:46:03.128987 IP 20.106.124.93.44366 > 10.130.8.94.80: Flags [P.], seq 1:315, ack 1, win 502, opti
18:46:03.129069 IP 10.130.8.94.80 > 20.106.124.93.44366: Flags [.], ack 315, win 487, options [nop,n
18:46:03.165424 IP 10.130.8.94.80 > 20.106.124.93.44366: Flags [P.], seq 1:2857, ack 315, win 487, o
18:46:03.165425 IP 10.130.8.94.80 > 20.106.124.93.44366: Flags [P.], seq 2857:5713, ack 315, win 487
18:46:03.165696 IP 10.130.8.94.80 > 20.106.124.93.44366: Flags [P.], seq 5713:6042, ack 315, win 487
18:46:03.165716 IP 10.130.8.94.80 > 20.106.124.93.44366: Flags [F.], seq 6042, ack 315, win 487, opt
18:46:03.223548 IP 20.106.124.93.44366 > 10.130.8.94.80: Flags [.], ack 2857, win 496, options [nop,
18:46:03.223549 IP 20.106.124.93.44366 > 10.130.8.94.80: Flags [.], ack 5713, win 481, options [nop,
18:46:03.223549 IP 20.106.124.93.44366 > 10.130.8.94.80: Flags [.], ack 6042, win 479, options [nop,
18:46:03.267387 IP 20.106.124.93.44366 > 10.130.8.94.80: Flags [.], ack 6043, win 501, options [nop,
18:46:03.323940 IP 20.106.124.93.44366 > 10.130.8.94.80: Flags [F.], seq 315, ack 6043, win 501, opt
18:46:03.323988 IP 10.130.8.94.80 > 20.106.124.93.44366: Flags [.], ack 316, win 487, options [nop,n
sec401@slingshot:/sec401/labs/1.1$
```

b.) Now, let's say that we want TCP traffic from ports 44366 and 80 that is between hosts 135.125.217.54 and 10.130.8.94. We would type the following command

**Command:** `tcpdump -n -r investigate.pcap 'tcp and (host 135.125.217.54 and host 10.130.8.94) and (port 44366 and port 80)'`





```
sec401@slingshot:/sec401/labs/1.1$ tcpdump -nr investigate.pcap 'tcp and (host 135.125.217.54 and host 10.130.8.94.80)'
reading from file investigate.pcap, link-type EN10MB (Ethernet), snapshot length 262144
15:25:36.329777 IP 135.125.217.54.44366 > 10.130.8.94.80: Flags [S], seq 250837459, win 29
15:25:36.329777 IP 10.130.8.94.80 > 135.125.217.54.44366: Flags [S.], seq 788644517, ack 2
15:25:36.432967 IP 135.125.217.54.44366 > 10.130.8.94.80: Flags [.], ack 1, win 229, optio
15:25:36.434871 IP 135.125.217.54.44366 > 10.130.8.94.80: Flags [P.], seq 1:231, ack 1, wi
15:25:36.434930 IP 10.130.8.94.80 > 135.125.217.54.44366: Flags [.], ack 231, win 488, opt
15:25:36.435167 IP 10.130.8.94.80 > 135.125.217.54.44366: Flags [P.], seq 1:492, ack 231,
15:25:36.538373 IP 135.125.217.54.44366 > 10.130.8.94.80: Flags [.], ack 492, win 237, opt
15:25:36.551502 IP 135.125.217.54.44366 > 10.130.8.94.80: Flags [F.], seq 231, ack 492, wi
15:25:36.551627 IP 10.130.8.94.80 > 135.125.217.54.44366: Flags [F.], seq 492, ack 232, wi
15:25:36.654979 IP 135.125.217.54.44366 > 10.130.8.94.80: Flags [.], ack 493, win 237, opt
sec401@slingshot:/sec401/labs/1.1$
```

2.2) In the above examples, we used tcpdump in conjunction with the BPF syntax to specify the information we want.

In the first image, we specified a packet capture file and then went on to specify that we ONLY wanted tcp traffic on ports 80 (HTTP) and 443 (HTTPS).

In second image, we dug a little deeper. Not only did we specify that we wanted tcp traffic, but this time we specified on what hosts we wanted that traffic coming from and going to AND we also specified what ports we wanted it on.

As you can see, we can get very particular with these tools. This is a way to help you drill down in your detection and analysis to find the exact information you're looking for.

#### References:

- <https://www.ibm.com/docs/en/qsip/7.4?topic=queries-berkeley-packet-filters>
- <https://docs.securityonion.net/en/2.4/bpf.html>

### **Task 3: Examine Raw Packet Content**

3.1) tcpdump is also able to display raw packet information in various formats (hexdump etc). Let's try examining the first 5 packets in hexdump format from another pcap file.

**Command:** *tcpdump -nr session.pcap -Xvc 5*

```
Home x My Computer x SEC401-Slingshot x SEC401-Win11-VM x
Applications Places System

File Edit View Search Terminal Help
sec401@slingshot:/sec401/labs/1.1$ tcpdump -nr session.cap -Xvc 5
reading from file session.cap, link-type EN10MB (Ethernet), snapshot length 262144
18:46:03.070623 IP (tos 0x0, ttl 44, id 33452, offset 0, flags [DF], proto TCP (6), length 60)
  20.106.124.93.44366 > 10.130.8.94.80: Flags [S], cksum 0x5f2c (correct), seq 1789817114, win 64240, options [mss 1440,s
    0x0000: 4500 003c 82ac 4000 2c06 2869 146a 7c5d E..<..@.,.(i.j]]
    0x0010: 0a82 085e ad4e 0050 6aae 711a 0000 0000 ...^..N.Pj.q.....
    0x0020: a002 faf0 5f2c 0000 0204 05a0 0402 080a ....._,.....
    0x0030: 0b69 b57f 0000 0000 0103 0307 .i.....
18:46:03.070663 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 60)
  10.130.8.94.80 > 20.106.124.93.44366: Flags [S.], cksum 0x9bdb (correct), seq 902963796, ack 1789817115, win 62643, opt
    0x0000: 4500 003c 0000 4000 4006 9715 0a82 085e E..<..@.@.....^
    0x0010: 146a 7c5d 0050 ad4e 35d2 2254 6aae 711b .j]].P.N5."Tj.q.
    0x0020: a012 f4b3 9bdb 0000 0204 05b4 0402 080a .....
    0x0030: 8445 ecfc 0b69 b57f 0103 0307 .E...i.....
18:46:03.128923 IP (tos 0x0, ttl 44, id 33453, offset 0, flags [DF], proto TCP (6), length 52)
  20.106.124.93.44366 > 10.130.8.94.80: Flags [.], cksum 0xbd2b (correct), ack 1, win 502, options [nop,nop,TS val 191477
    0x0000: 4500 0034 82ad 4000 2c06 2870 146a 7c5d E..4..@.,.(p.j]]
    0x0010: 0a82 085e ad4e 0050 6aae 711b 35d2 2255 ...^..N.Pj.q.5."U
    0x0020: 8010 01f6 bd2b 0000 0101 080a 0b69 b5b9 .....+.....i..
    0x0030: 8445 ecfc .E..
18:46:03.128987 IP (tos 0x0, ttl 44, id 33454, offset 0, flags [DF], proto TCP (6), length 366)
  20.106.124.93.44366 > 10.130.8.94.80: Flags [P.], cksum 0x3059 (correct), seq 1:315, ack 1, win 502, options [nop,nop,T
    POST /wp-login.php HTTP/1.0
    Host: www.alphainc.ca
    User-Agent: Mozilla/5.0 (Hydra)
    Content-Length: 106
    Content-Type: application/x-www-form-urlencoded
    Cookie: wordpress_test_cookie=WP%20Cookie%20check

    log=admin&pwd=garment&wp-submit=Log+In&redirect_to=http%3A%2F%2Fwww.alphainc.ca%2Fwp-admin%2F&testcookie=1 [|http]
    0x0000: 4500 016e 82ae 4000 2c06 2735 146a 7c5d E..n..@.,.'5.j]]
    0x0010: 0a82 085e ad4e 0050 6aae 711b 35d2 2255 ...^..N.Pj.q.5."U
    0x0020: 8018 01f6 3059 0000 0101 080a 0b69 b5b9 ....0Y.....i..
    0x0030: 8445 ecfc 504f 5354 202f 7770 2d6c 6f67 .E..POST./wp-log
    0x0040: 696e 2e70 6870 2048 5454 502f 312e 300d in.php.HTTP/1.0.
    0x0050: 0a48 6f73 743a 2077 7777 2e61 6c70 6861 .Host:.www.alpha
    0x0060: 696e 632e 6361 0d0a 5573 6572 2d41 6765 inc.ca..User-Age
    0x0070: 6e74 3a20 4d6f 7a69 6c6c 612f 352e 3020 nt:.Mozilla/5.0.
    0x0080: 2848 7964 7261 290d 0a43 6f6e 7465 6e74 (Hydra)..Content
    0x0090: 2d4c 656e 6774 683a 2031 3036 0d0a 436f -Length:..106..Co
    0x00a0: 6e74 656e 742d 5479 7065 3a20 6170 706c ntent-Type:.appl
    0x00b0: 6963 6174 696f 6e2f 782d 7777 772d 666f ication/x-www-fo
    0x00c0: 726d 2d75 726c 656e 636f 6465 640d 0a43 rm-urlencoded..C
    0x00d0: 6f6f 6b69 653a 2077 6f72 6470 7265 7373 ookie:.wordpress
    0x00e0: 5f74 6573 745f 636f 6f6b 6965 3d57 5025 _test_cookie=WP%
    0x00f0: 3230 436f 6f6b 6965 2532 3063 6865 636b 20Cookie%20check
    0x0100: 0d0a 0d0a 6c6f 673d 6164 6d69 6e26 7077 ....log=admin&pw
    0x0110: 643d 6761 726d 656e 7426 7770 2d73 7562 d=garment&wp-sub
    0x0120: 6d69 743d 4c6f 672b 496e 2672 6564 6972 mit=Log+In&redir
    0x0130: 6563 745f 746f 3d68 7474 7025 3341 2532 ect_to=http%3A%2
    0x0140: 4625 3246 7777 772e 616c 7068 6169 6e63 F%2Fwww.alphainc
    0x0150: 2e63 6125 3246 7770 2d61 646d 696e 2532 .ca%2Fwp-admin%2
    0x0160: 4626 7465 7374 636f 6f6b 6965 3d31 F&testcookie=1
18:46:03.129069 IP (tos 0x0, ttl 64, id 60567, offset 0, flags [DF], proto TCP (6), length 52)
  10.130.8.94.80 > 20.106.124.93.44366: Flags [.], cksum 0xbbc5 (correct), ack 315, win 487, options [nop,nop,TS val 2219
    0x0000: 4500 0034 ec97 4000 4006 aa85 0a82 085e E..4..@.@.....^
    0x0010: 146a 7c5d 0050 ad4e 35d2 2255 6aae 7255 .j]].P.N5."Uj.rU
    0x0020: 8010 01e7 bbc5 0000 0101 080a 8445 ed37 .....E.7
    0x0030: 0b69 b5b9 .i..
sec401@slingshot:/sec401/labs/1.1$
```

If you've noticed, I've been shortening/chaining together commands to save time. For example, instead of typing (*tcpdump -n -r session.pcap -X -v -c 5*), we can just combine the majority of the commands like so (*tcpdump -nr session.pcap -Xvc 5*) and get the same results. As long as the commands make sense to the operating system, you will be good to go!

That is the power and flexibility of Linux! You can become rather creative the more you play around with it. Find commands, or shortcuts, that work for you and help enhance your efficiency!

## **Conclusion:**

That brings our brief tutorial on tcpdump to a close. Please feel free to give any comments or feedback if this lesson provided you with any value, and stay tuned for more labs in the future. Thank you!