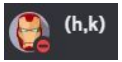
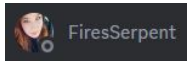
Fireside Fridays

Packet Sniffing Tools
Week 8

Thanks to our sponsors!

Special Thanks to…

- Hermon 

  - Who will have a new job as a Cyber Threat Hunter in April :-)
- Emily 

- Both gave up many late nights to help with QA and development of this content
- Very much appreciate their efforts!
- Please give them a warm "thank you"

Lab requirements for this section

- Labs this week!
- You will need:
  - Install Wireshark
  - tshark in your path
  - Install ngrep (sudo apt -y install ngrep)
  - The decode1 pcap file
  - The fiesta-c2 pcap file
  - Optional - tcpdump

Packet sniffing

- Permits you to see raw packets on the wire
- Can also read/write "pcap" files
- Different tools will display the data slightly differently
- This gives each their strengths and weaknesses
- GUI - Good for detailed analysis, bad for large captures
- CMD - Good for large captures, less auto decoding

Wireshark

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Sequence Number	Acknowledg	Protocol	Total Length	Info
1	0.000000	10.0.2.15	68.183.138.51	0	0	TCP	52	49884 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
2	0.047450	68.183.138.51	10.0.2.15	0	1	TCP	44	80 → 49884 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
3	0.047500	10.0.2.15	68.183.138.51	1	1	TCP	40	49884 → 80 [ACK] Seq=1 Ack=1 Win=65535 Len=0
4	0.047745	10.0.2.15	68.183.138.51	1	1	HTTP	586	GET /include/template/isx.php HTTP/1.1
5	0.048050	68.183.138.51	10.0.2.15	1	547	TCP	40	80 → 49884 [ACK] Seq=1 Ack=547 Win=65535 Len=0
6	0.108535	68.183.138.51	10.0.2.15	1	547	TCP	254	80 → 49884 [PSH, ACK] Seq=1 Ack=547 Win=65535 Len=214 [TCP segment of a reassembled PDU]
7	0.108568	10.0.2.15	68.183.138.51	547	215	TCP	40	49884 → 80 [ACK] Seq=547 Ack=215 Win=65535 Len=0
8	0.109136	68.183.138.51	10.0.2.15	215	547	HTTP	190	HTTP/1.1 200 OK (text/html)

```
> Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
> Ethernet II, Src: PCSSystemtec_af:09:1e (08:00:27:af:09:1e), Dst: 52:54:00:12:35:02 (52:54:00:12:35:02)
> Internet Protocol Version 4, Src: 10.0.2.15, Dst: 68.183.138.51
v Transmission Control Protocol, Src Port: 49884, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 49884
    Destination Port: 80
    [Stream index: 0]
  > [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 3925567981
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x002 (SYN)
    Window: 65535
    [Calculated window size: 65535]
    Checksum: 0xdb1f [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted
  > [Timestamps]
```

```
0000  52 54 00 12 35 02 08 00  27 af 09 1e 08 00 45 00   RT··5··· '·····E·
0010  00 34 f3 4e 40 00 80 06  00 00 0a 00 02 0f 44 b7   ·4·N@··· ------D·
0020  8a 33 c2 dc 00 50 e9 fb  69 ed 00 00 00 00 80 02   ·3···P·· i-------
0030  ff ff db 1f 00 00 02 04  05 b4 01 03 03 08 01 01   -------- ------··
0040  04 02                                              ··
```

Follow → TCP Stream

Wireshark · Follow TCP Stream (tcp.stream eq 0) · decode1.pcap

```
.............jGz........-(...(u~.,..g....,.......+./.,.0............/.5.
...q..................wss-primary.slack.com.....#...
.................................uP........
.
...............
....A...=....s.......n..B.Esf..
.m....h...D..0.....................#................0...0............:f!tt8M6U..G.0
.        *.H..
.....0M1.0        ..U....US1.0...U.
..DigiCert Inc1'0%..U....DigiCert SHA2 Secure Server CA0..
180208000000Z.
210212120000Z0i1.0         ..U....US1.0      ..U....CA1.0...U...
San Francisco1!0...U.
..Slack Technologies, Inc.1.0...U...         slack.com0.."0
.        *.H..
..........0..
.......
.T....P.OSu[..l..dd......T.u..k...:.TiO....P...j..Z........V.N.Z-Q..$.
i..6A.C.....e~^.._gkEE..B....T.pB.X.Yn.T.6.D.kN..9...v...J?.....B.........q*...........C,r|!.W.....(0.%\C1........./.5.(.x...*
....nt.8..U.n...L..BUu.....O.V....{.z....{j..
}.?.C............0...0...U.#..0.....a..1a./(..F8.,....0...U.......Z..<\B........l        ..-0!..U....0..   slack.com..*.slack.
com0...U...........0...U.%..0...+........+.......0k..U...d0b0/.-.+.)http://crl3.digicert.com/ssca-sha2-g6.crl0/.-.+.)http://c
rl4.digicert.com/ssca-sha2-g6.crl0L..U. .E0C07.      `.H...l..0*0(..+........https://www.digicert.com/CPS0...g.....0|..+....
....p0n0$..+.....0...http://ocsp.digicert.com0F..+.....0..:http://cacerts.digicert.com/DigiCertSHA2SecureServerCA.crt0...U....
...0.0....
+.....y..............v...  ...X......gp
<5.......w...
.....awd=......G0E.!..k.r....(.......6I..4..+%.....c}. P..6..^.:.....m..."........A!....v..u..Y|..C._..n.V.GV6.J.`....^......a
wd=......G0E. H.......I.......o.R"....S.....t..!.."......v.........     ..].....w.7....v.......q...#...{G8W.
.R....d6.......awd> .....G0E.!.........a...4....;>?\.D.....r./L. .....Z.b...g..p.....K^sW...Y..q..v.oSv.1.1.....Q..w.......)..
...7.....awd?......G0E.!..-.R....S{d<.F....Q.>.|...... ... .Y[....&..........p.<.v8....V...0
.        *.H..
```

Packet 5. 1,882 client pkts, 2,057 server pkts, 3,759 turns. Click to select.

Entire conversation (396 kB) | Show as ASCII | No delta times | Stream 0

Find: | Case sensitive | Find Next

Filter Out This Stream | Print | Save as... | Back | Close | Help

Wireshark strengths and weaknesses

- The good
  - Easy to view any header field
  - Re-assemble payloads in a stream
  - Expert notes and color coding
  - Lots of helpful built in tools
- The bad
  - Cumbersome to use on large pcaps
  - More likely to miss packets when sniffing

tcpdump

```
cbrenton@cb-lab:~/pcaps$ tcpdump -nn -r decode1.pcap | head
reading from file decode1.pcap, link-type EN10MB (Ethernet)
23:53:31.286257 IP 10.0.0.204.55121 > 34.194.201.2.443: Flags [S], seq 4267111507, win 8192, options [mss 1460,no
p,wscale 2,nop,nop,sackOK], length 0
23:53:31.330253 IP 34.194.201.2.443 > 10.0.0.204.55121: Flags [S.], seq 3093938554, ack 4267111508, win 26883, op
tions [mss 1460,nop,nop,sackOK,nop,wscale 12], length 0
23:53:31.330299 IP 10.0.0.204.55121 > 34.194.201.2.443: Flags [.], ack 1, win 16425, length 0
23:53:31.330664 IP 10.0.0.204.55121 > 34.194.201.2.443: Flags [P.], seq 1:192, ack 1, win 16425, length 191
23:53:31.376824 IP 34.194.201.2.443 > 10.0.0.204.55121: Flags [.], seq 1:1461, ack 192, win 7, length 1460
23:53:31.382083 IP 34.194.201.2.443 > 10.0.0.204.55121: Flags [.], seq 1461:2921, ack 192, win 7, length 1460
23:53:31.382083 IP 34.194.201.2.443 > 10.0.0.204.55121: Flags [P.], seq 2921:3433, ack 192, win 7, length 512
23:53:31.382120 IP 10.0.0.204.55121 > 34.194.201.2.443: Flags [.], ack 3433, win 16425, length 0
23:53:31.384517 IP 10.0.0.204.55121 > 34.194.201.2.443: Flags [P.], seq 192:318, ack 3433, win 16425, length 126
23:53:31.438655 IP 34.194.201.2.443 > 10.0.0.204.55121: Flags [P.], seq 3433:3691, ack 318, win 7, length 258
```

tcpdump -X

```
cbrenton@cb-lab:~/pcaps$ tcpdump -Xnn -r decode1.pcap | head
reading from file decode1.pcap, link-type EN10MB (Ethernet)
23:53:31.286257 IP 10.0.0.204.55121 > 34.194.201.2.443: Flags [S], seq 4267111507, win 8192, options [mss 1460,no
p,wscale 2,nop,nop,sackOK], length 0
        0x0000:  4500 0034 1b7b 4000 8006 e8b8 0a00 00cc  E..4.{@.........
        0x0010:  22c2 c902 d751 01bb fe56 f453 0000 0000  "....Q...V.S....
        0x0020:  8002 2000 8cce 0000 0204 05b4 0103 0302  ................
        0x0030:  0101 0402                                ....
23:53:31.330253 IP 34.194.201.2.443 > 10.0.0.204.55121: Flags [S.], seq 3093938554, ack 4267111508, win 26883, op
tions [mss 1460,nop,nop,sackOK,nop,wscale 12], length 0
        0x0000:  4500 0034 0000 4000 2a06 5a34 22c2 c902  E..4..@.*.Z4"...
        0x0010:  0a00 00cc 01bb d751 b869 c17a fe56 f454  .......Q.i.z.V.T
        0x0020:  8012 6903 c9cb 0000 0204 05b4 0101 0402  ..i.............
        0x0030:  0103 030c                                ....
```

tcpdump strengths and weaknesses

- The good
  - Runs on everything and usually it's just there
  - Nice abbreviated summary
  - Full Hex output so you can decode yourself
  - Easy to use rudimentary filtering
- The bad
  - Does not decode all fields
  - Little customization of display options

tshark

```
cbrenton@cb-lab:~/pcaps$ tshark -n -r decode1.pcap | head
    1   0.000000   10.0.0.204 → 34.194.201.2 TCP 66 55121 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PER
M=1
    2   0.043996 34.194.201.2 → 10.0.0.204   TCP 66 443 → 55121 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=1460 S
ACK_PERM=1 WS=4096
    3   0.044042   10.0.0.204 → 34.194.201.2 TCP 54 55121 → 443 [ACK] Seq=1 Ack=1 Win=65700 Len=0
    4   0.044407   10.0.0.204 → 34.194.201.2 TLSv1 245 Client Hello
    5   0.090567 34.194.201.2 → 10.0.0.204   TLSv1.2 1514 Server Hello
    6   0.095826 34.194.201.2 → 10.0.0.204   TCP 1514 443 → 55121 [ACK] Seq=1461 Ack=192 Win=28672 Len=1460 [TCP
segment of a reassembled PDU]
    7   0.095826 34.194.201.2 → 10.0.0.204   TLSv1.2 566 Certificate, Server Key Exchange, Server Hello Done
    8   0.095863   10.0.0.204 → 34.194.201.2 TCP 54 55121 → 443 [ACK] Seq=192 Ack=3433 Win=65700 Len=0
    9   0.098260   10.0.0.204 → 34.194.201.2 TLSv1.2 180 Client Key Exchange, Change Cipher Spec, Encrypted Hands
hake Message
   10   0.152398 34.194.201.2 → 10.0.0.204   TLSv1.2 312 New Session Ticket, Change Cipher Spec, Encrypted Handsh
ake Message
```

tcpdump Vs tshark

```
cbrenton@rita-v5:~/lab$ tcpdump -nn -r decode1.pcap | head
reading from file decode1.pcap, link-type EN10MB (Ethernet), snapshot length 65535
23:53:31.286257 IP 10.0.0.204.55121 > 34.194.201.2.443: Flags [S], seq 4267111507, win 8192, options [mss 1460,nop,wscale 2,nop,no
p,sackOK], length 0
23:53:31.330253 IP 34.194.201.2.443 > 10.0.0.204.55121: Flags [S.], seq 3093938554, ack 4267111508, win 26883, options [mss 1460,n
op,nop,sackOK,nop,wscale 12], length 0
23:53:31.330299 IP 10.0.0.204.55121 > 34.194.201.2.443: Flags [.], ack 1, win 16425, length 0
23:53:31.330664 IP 10.0.0.204.55121 > 34.194.201.2.443: Flags [P.], seq 1:192, ack 1, win 16425, length 191
23:53:31.376824 IP 34.194.201.2.443 > 10.0.0.204.55121: Flags [.], seq 1:1461, ack 192, win 7, length 1460
23:53:31.382083 IP 34.194.201.2.443 > 10.0.0.204.55121: Flags [.], seq 1461:2921, ack 192, win 7, length 1460
23:53:31.382083 IP 34.194.201.2.443 > 10.0.0.204.55121: Flags [P.], seq 2921:3433, ack 192, win 7, length 512
23:53:31.382120 IP 10.0.0.204.55121 > 34.194.201.2.443: Flags [.], ack 3433, win 16425, length 0
23:53:31.384517 IP 10.0.0.204.55121 > 34.194.201.2.443: Flags [P.], seq 192:318, ack 3433, win 16425, length 126
23:53:31.438655 IP 34.194.201.2.443 > 10.0.0.204.55121: Flags [P.], seq 3433:3691, ack 318, win 7, length 258
```

```
cbrenton@rita-v5:~/lab$ tshark -n -r decode1.pcap | head
    1   0.000000   10.0.0.204 → 34.194.201.2 TCP 66 55121 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM
    2   0.043996 34.194.201.2 → 10.0.0.204   TCP 66 443 → 55121 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=1460 SACK_PERM WS=4096
    3   0.044042   10.0.0.204 → 34.194.201.2 TCP 54 55121 → 443 [ACK] Seq=1 Ack=1 Win=65700 Len=0
    4   0.044407   10.0.0.204 → 34.194.201.2 TLSv1 245 Client Hello (SNI=wss-primary.slack.com)
    5   0.090567 34.194.201.2 → 10.0.0.204   TLSv1.2 1514 Server Hello
    6   0.095826 34.194.201.2 → 10.0.0.204   TCP 1514 443 → 55121 [ACK] Seq=1461 Ack=192 Win=28672 Len=1460 [TCP segment of a reas
sembled PDU]
    7   0.095826 34.194.201.2 → 10.0.0.204   TLSv1.2 566 Certificate, Server Key Exchange, Server Hello Done
    8   0.095863   10.0.0.204 → 34.194.201.2 TCP 54 55121 → 443 [ACK] Seq=192 Ack=3433 Win=65700 Len=0
    9   0.098260   10.0.0.204 → 34.194.201.2 TLSv1.2 180 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
   10   0.152398 34.194.201.2 → 10.0.0.204   TLSv1.2 312 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
```

tshark "fields"

```
$ tshark -r thunt-lab.pcapng -T fields -e dns.qry.name
udp.port==53 | head -10

6dde0175375169c68f.dnsc.r-1x.com
6dde0175375169c68f.dnsc.r-1x.com
0b320175375169c68f.dnsc.r-1x.com
0b320175375169c68f.dnsc.r-1x.com
344b0175375169c68f.dnsc.r-1x.com
344b0175375169c68f.dnsc.r-1x.com
0f370175375169c68f.dnsc.r-1x.com
0f370175375169c68f.dnsc.r-1x.com
251e0175375169c68f.dnsc.r-1x.com
251e0175375169c68f.dnsc.r-1x.com
```

Try this command

```
tshark -n -r fiesta-c2.pcap -T fields -e ip.src -e tcp.srcport -e ip.dst -e
tcp.dstport -e http.request.uri -e http.content_type -e http.user_agent |
less
```

```
cbrenton@rita-v5:~/lab$ tshark -n -r fiesta-c2.pcap -T fields -e ip.src -e tcp.srcport -e ip.dst -e tcp.dstport -e http.request.ur
i -e http.content_type -e http.user_agent | head
10.0.2.15       49884   68.183.138.51   80
68.183.138.51   80      10.0.2.15       49884
10.0.2.15       49884   68.183.138.51   80
10.0.2.15       49884   68.183.138.51   80      /include/template/isx.php               Mozilla/5.0 (Windows; U; MSIE 7.0; Windows
 NT 5.2) Java/1.5.0_08
68.183.138.51   80      10.0.2.15       49884
68.183.138.51   80      10.0.2.15       49884
10.0.2.15       49884   68.183.138.51   80
68.183.138.51   80      10.0.2.15       49884           text/html
68.183.138.51   80      10.0.2.15       49884
10.0.2.15       49884   68.183.138.51   80
cbrenton@rita-v5:~/lab$
```

What I love about tshark

```
tshark -n -r fiesta-c2.pcap -T fields -e ip.dst -e http.user_agent http | sort | uniq
| tr -s ' ' | cut -f 2 | sort | uniq -c | sort -rn
```

```
cbrenton@rita-v5:~/lab$ tshark -n -r fiesta-c2.pcap -T fields -e ip.dst -e http.user_agent http | sort | uniq |
 tr -s ' ' | cut -f 2 | sort | uniq -c | sort -rn
     15 Microsoft-CryptoAPI/10.0
     12 Microsoft-WNS/10.0
      1 Mozilla/5.0 (Windows; U; MSIE 7.0; Windows NT 5.2) Java/1.5.0_08
      1
cbrenton@rita-v5:~/lab$ _
```

You can do some pretty complex pre-processing with tshark.
This would be really challenging to do with Wireshark.
Can be fully automated, unlike Wireshark.

The above command reports the HTTP user agent strings being used, and the count of external IPs against which each user agent string was used. This could easily be wrapped in an automated script that fires on a regular basis.

Finding display fields

- Need to know the header where field is located
- 204,000+ fields - can be challenging to finding right one
- Luckily grep can help
- Some examples:

```
tshark -G | grep "[[:space:]]ip\." | less -S -x40
tshark -G | grep "[[:space:]]tcp\." | less -S -x40
tshark -G | grep "[[:space:]]dns\." | less -S -x40
tshark -G | grep "[[:space:]]http\." | less -S -x40
```

tshark strengths and weaknesses

- The good
  - Display and capture filters same as Wireshark
  - Control over the fields that get displayed
  - Some prefer default output to tcpdump
- The bad
  - Slower than tcpdump, noticeable on big pcaps
  - Usually not installed by default

ngrep

- Pattern match on passing packets
- Like "grep" for network traffic
- Useful for quick checks
  - NIDS with signature better choice for long term
- Useful switches
  - "-q" = Don't print "#" for non-matches
  - "-I" = Read a pcap file
  - "-i" = Case insensitive search

https://github.com/jpr5/ngrep
sudo apt install ngrep

ngrep examples

```
cbrenton@cb-lab:~/pcaps$ ngrep -q -I weird-icmp.pcap SSH tcp and port 22
input: weird-icmp.pcap
filter: ( tcp and port 22 ) and ((ip || ip6) || (vlan && (ip || ip6)))
match: SSH

T 167.71.123.148:22 -> 218.92.0.207:54001 [AP] #100
  SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8..

T 218.92.0.207:54001 -> 167.71.123.148:22 [AP] #107
  SSH-2.0-PUTTY..
cbrenton@cb-lab:~/pcaps$ _
```

```
cbrenton@cb-lab:~/pcaps$ ngrep -I weird-icmp.pcap -q 'tasklist|DIR' | tail -12

I 67.7.80.242 -> 167.71.123.148 8:0 #212
  ..../2016  06:55 PM    <DIR>          steghide-0.5.1-win32..03/31/20

I 67.7.80.242 -> 167.71.123.148 8:0 #225
  ....AM               814 todo.txt..07/21/2016  06:55 PM    <DIR>

I 167.71.123.148 -> 67.7.80.242 0:0 #504
  ...Ntasklist.

I 67.7.80.242 -> 167.71.123.148 8:0 #506
  ...Otasklist.....
```

Try this command

```
ngrep -iq -I fiesta-c2.pcap google.com | head -20
```

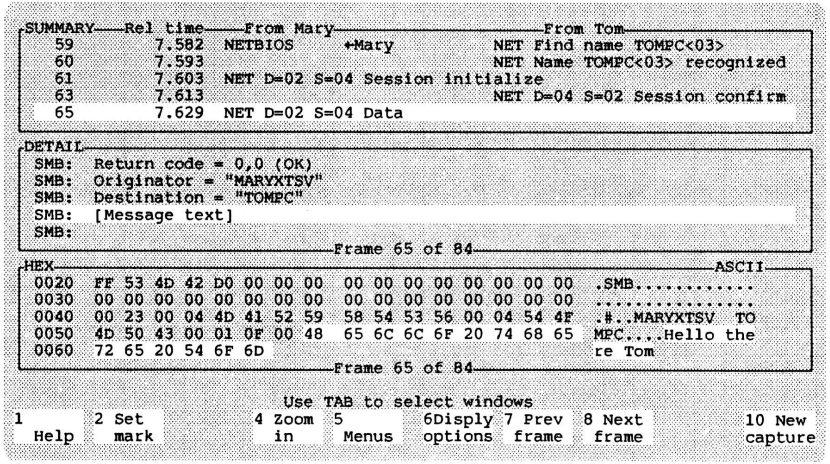
```
cbrenton@rita-v5:~/lab$ ngrep -iq -I fiesta-c2.pcap google.com | head -20
input: fiesta-c2.pcap
filter: ((ip || ip6) || (vlan && (ip || ip6)))
match (JIT): google.com

T 10.0.2.15:49884 -> 68.183.138.51:80 [AP] #4
  GET /include/template/isx.php HTTP/1.1..Referer: http://www.google.com..Accept: text/xml,application/xml,a
  pplication/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5..Accept-Language: en-us,en;q=0.5
  ..Cookie: Pi11P0gI1NbuS4D7R5H3aGjOVGXx6/fxyth9vhdNpT59SrcmZ4ZQGbJVa52nyEPXJ9We7wrHEIgPAg/CBpdpq9djwMoWVgfd
  OkOJjj8ZLjgQohM/r7wiehTBysuY5vwzAr/OnRItWTFP5t54Db4+qguyvIFH8hHc8zGPtqE1Xr0=..User-Agent: Mozilla/5.0 (Win
  dows; U; MSIE 7.0; Windows NT 5.2) Java/1.5.0_08..Host: 68.183.138.51..Connection: Keep-Alive..Cache-Contr
  ol: no-cache....

T 10.0.2.15:49885 -> 68.183.138.51:80 [AP] #25
  GET /include/template/isx.php HTTP/1.1..Referer: http://www.google.com..Accept: text/xml,application/xml,a
  pplication/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5..Accept-Language: en-us,en;q=0.5
  ..Cookie: Pi11P0gI1NbuS4D7R5H3aGjOVGXx6/fxyth9vhdNpT59SrcmZ4ZQGbJVa52nyEPXJ9We7wrHEIgPAg/CBpdpq9djwMoWVgfd
  OkOJjj8ZLjgQohM/r7wiehTBysuY5vwzAr/OnRItWTFP5t54Db4+qguyvIFH8hHc8zGPtqE1Xr0=..User-Agent: Mozilla/5.0 (Win
  dows; U; MSIE 7.0; Windows NT 5.2) Java/1.5.0_08..Host: 68.183.138.51..Connection: Keep-Alive..Cache-Contr
  ol: no-cache....

cbrenton@rita-v5:~/lab$ _
```

So many tools, so little time

- There are lots of other packet sniffing tools
- Please share your favorites and why you like them in the Discord channel :-)

"The sniffer" that @ErikG and I first used. :-D

Who recognizes it/will admit they are THAT old?



Next Week on Fireside Fridays!

- I have jury duty so…
- The illustrious Bill Stearns will be presenting!
- ACM's "Master of $h1&&y Little Shell Scripts"
- A man that's so clever, sometimes he doesn't understand a single word he's saying
- Linux networking? Hold my milk…
- Bill will be presenting "IPv6 for IPv4 Users"
- No tools or preloads needed for this preso

Wrap up

- Thank you for attending!
- Certs will go out by Monday
- Video should be posted within 24 hours
- If you have any lingering questions, drop me an email at chris@activecountermeasures.com