80 TO 0 IN UNDER 5 SECONDS: FALSIFYING A MEDICAL PATIENT'S VITALS

DOUGLAS MCKEE

WHO AM I?

- Douglas McKee
- Senior Security Researcher for McAfee's Advanced Threat Research team
- 8+ years experience in vulnerability research, penetration testing and forensics
- @fulmetalpackets
- Not a medical doctor

McAfee™
Together is power.

OUTLINE

- Why these medical devices?
- Overview of systems
- RWHAT?
- Reversing the protocol
- Replay attacks
- Demo
- Impact Scenarios
- Mitigations

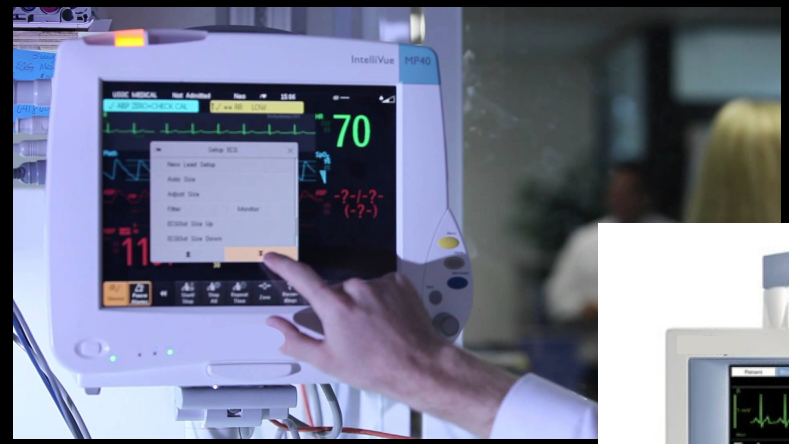
WHY?

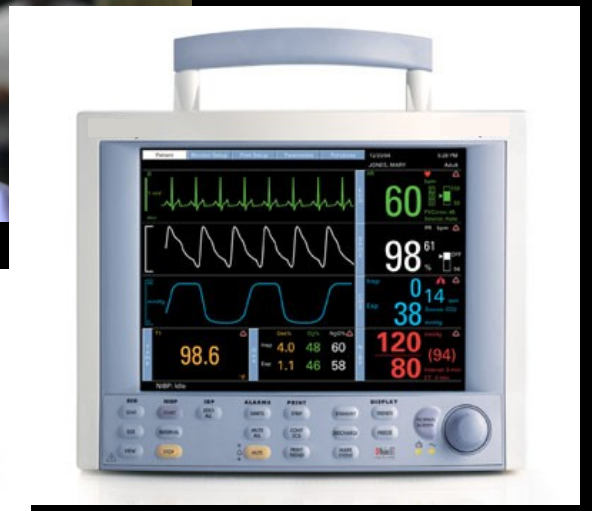
- Potential impact of medical devices
- Patient monitoring will always be a necessary
  - “Vital signs are integral to clinical decision making” – Dr. S. Nordeck
- Reliance on technology in medical field
  - "We have an implicit trust in these types of technologies," Tully says. "We don’t ever get any cybersecurity training in medical school. It’s not something that ever comes up in our literature." Dr. Jeff Tully*

*https://www.hpe.com/us/en/insights/articles/medical-device-security-hacking-prevention-measures-1806.html

PATIENT MONITOR (PM)

- BEDSIDE MONITOR
- MONITORS PATIENT'S VITALS – HEARTRATE, BLOOD PRESSURE, O2 LEVELS, ETC
- HAS WIRED AND WIRELESS (OPTIONAL) NETWORKING
- CONTAINS PERSONAL IDENTIFIABLE INFORMATION (PII)
- *GENERAL PICTURES, NOT TESTED MODEL





CENTRAL MONITORING STATION (CMS)

- Runs Embedded Windows on CF Card
  - Windows XP or Windows 7
- Has two Ethernet ports for two networks
- Runs main application in limited user mode
- A central station that receives data from Patient Monitors
- Thanks Ebay!

POTENTIAL ATTACK VECTORS

- OS
- APPLICATION

POTENTIAL ATTACK VECTORS

- OS
- APPLICATION
- FIRMWARE/HARDWARE
- **NETWORK**
  - **MODIFY PATIENT INFORMATION?**

RWHAT?

RWHAT packets

All monitoring devices on the [redacted] Network periodically broadcast information about themselves in "RWHAT" packets. Among other things, RWHAT packets contain IP address, port number, name, and offered services information about each device.

All monitoring devices listen for RWHAT packets, and maintain a database of information about other devices on the network. When devices need to communicate, the appropriate IP address information is obtained from the database, [redacted] Network-protocol messages are created, and operating system services are used to transmit the message on the network.

For example, when a [redacted] computer communicates with a telemetry device, the telemetry device's IP address is retrieved from the [redacted] computer RWHAT database, the [redacted] Network messages are created, and the [redacted] Windows operating system sends the messages to the telemetry device.

RWHAT?

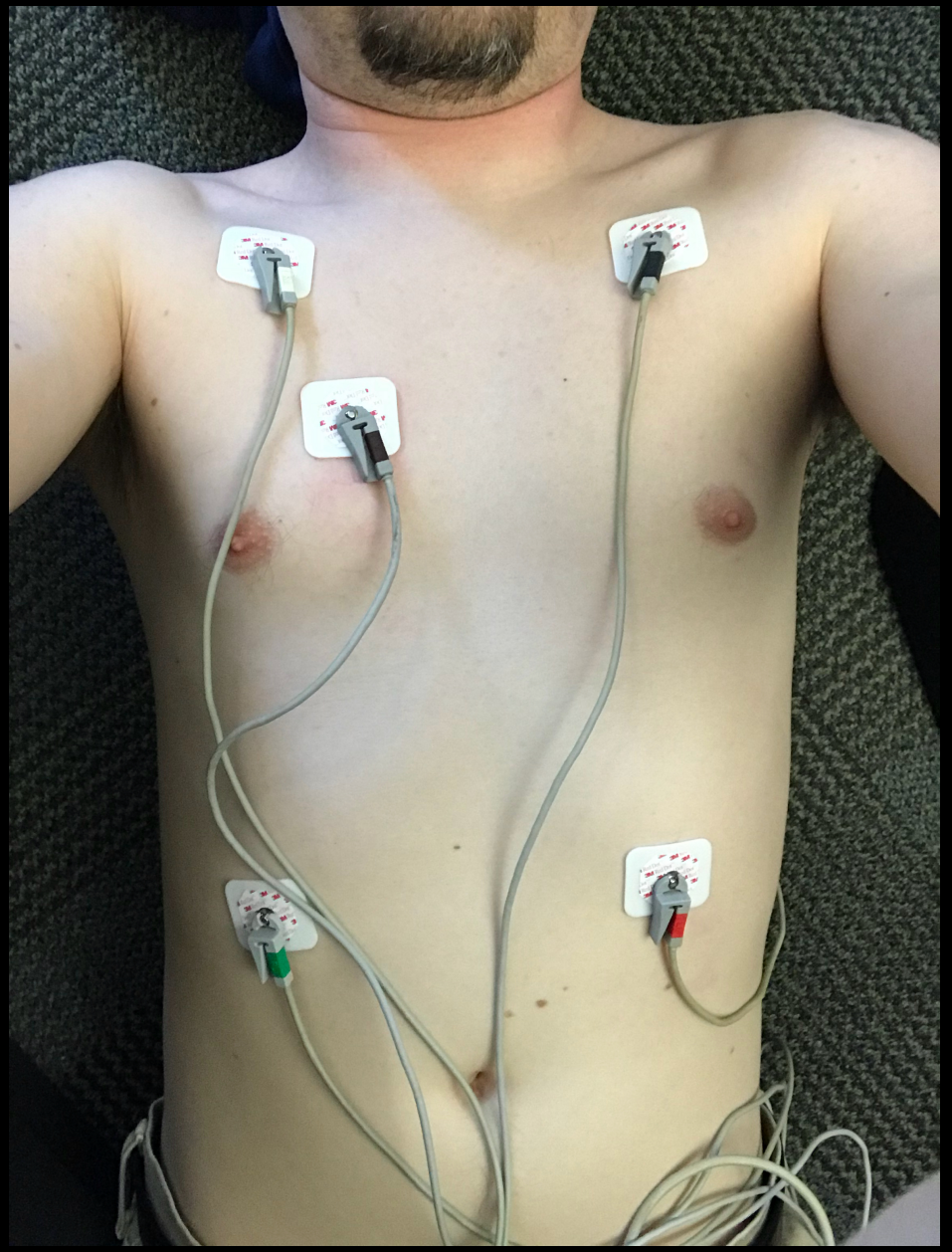
RWHAT packets

All monitoring devices on the [redacted] Network periodically broadcast information about themselves in "RWHAT" packets. Among other things, RWHAT packets contain IP address, port number, name, and offered services information about each device.

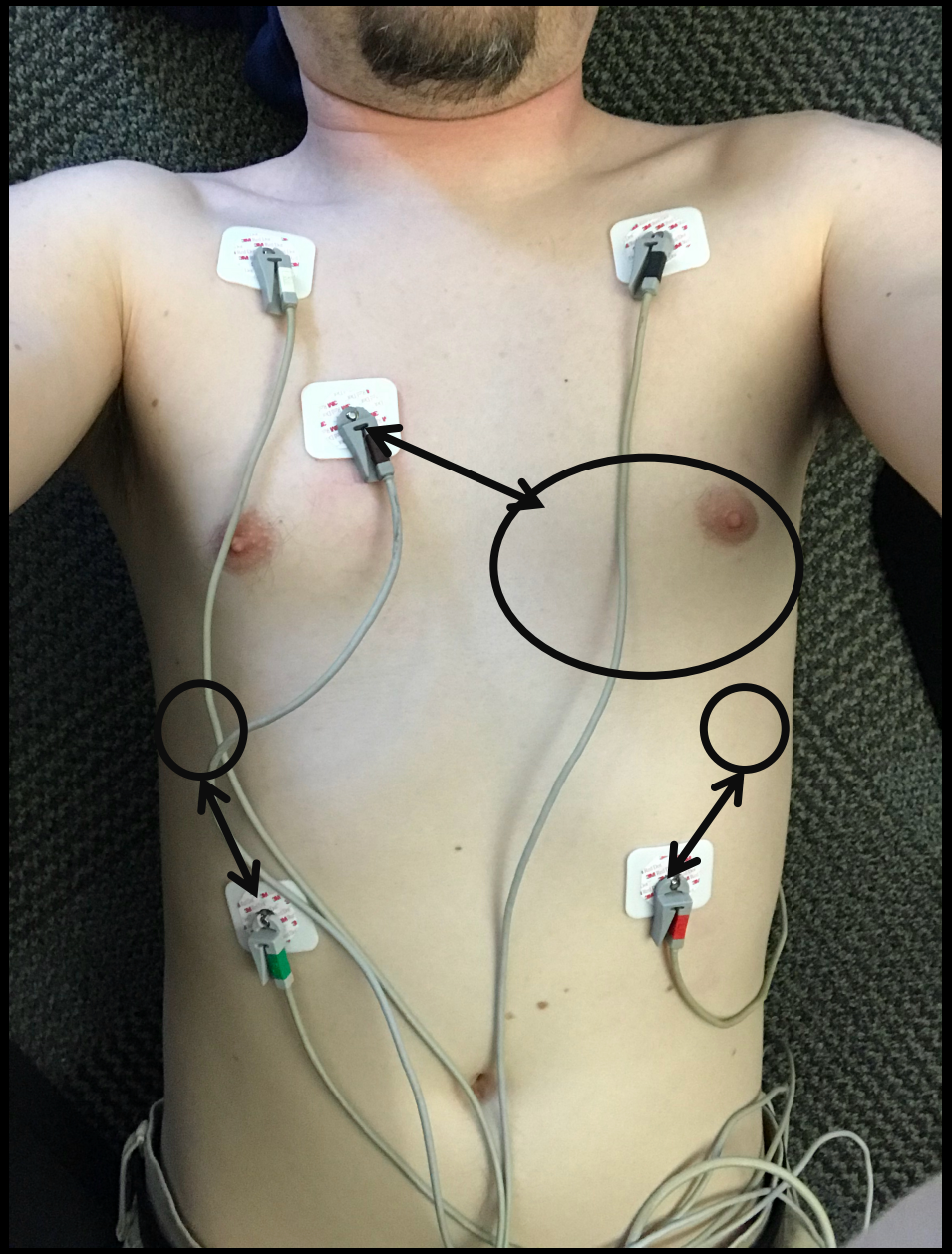
For example, when a [redacted] computer communicates with a telemetry device, the telemetry device's IP address is retrieved from the [redacted] computer RWHAT database, the [redacted] Network messages are created, and the [redacted] Windows operating system sends the messages to the telemetry device.

TESTING SETUP

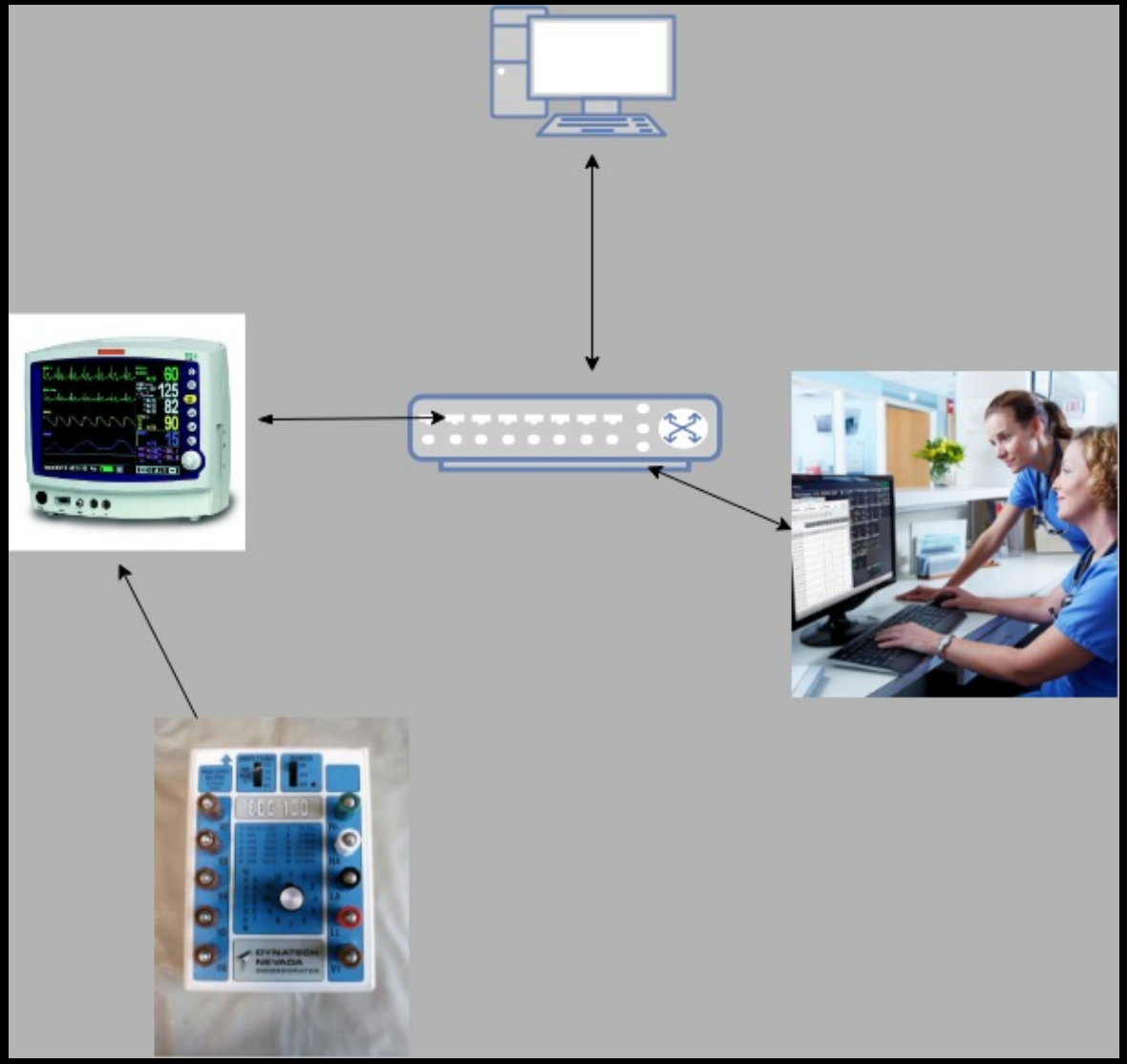
TESTING SETUP



TESTING SETUP



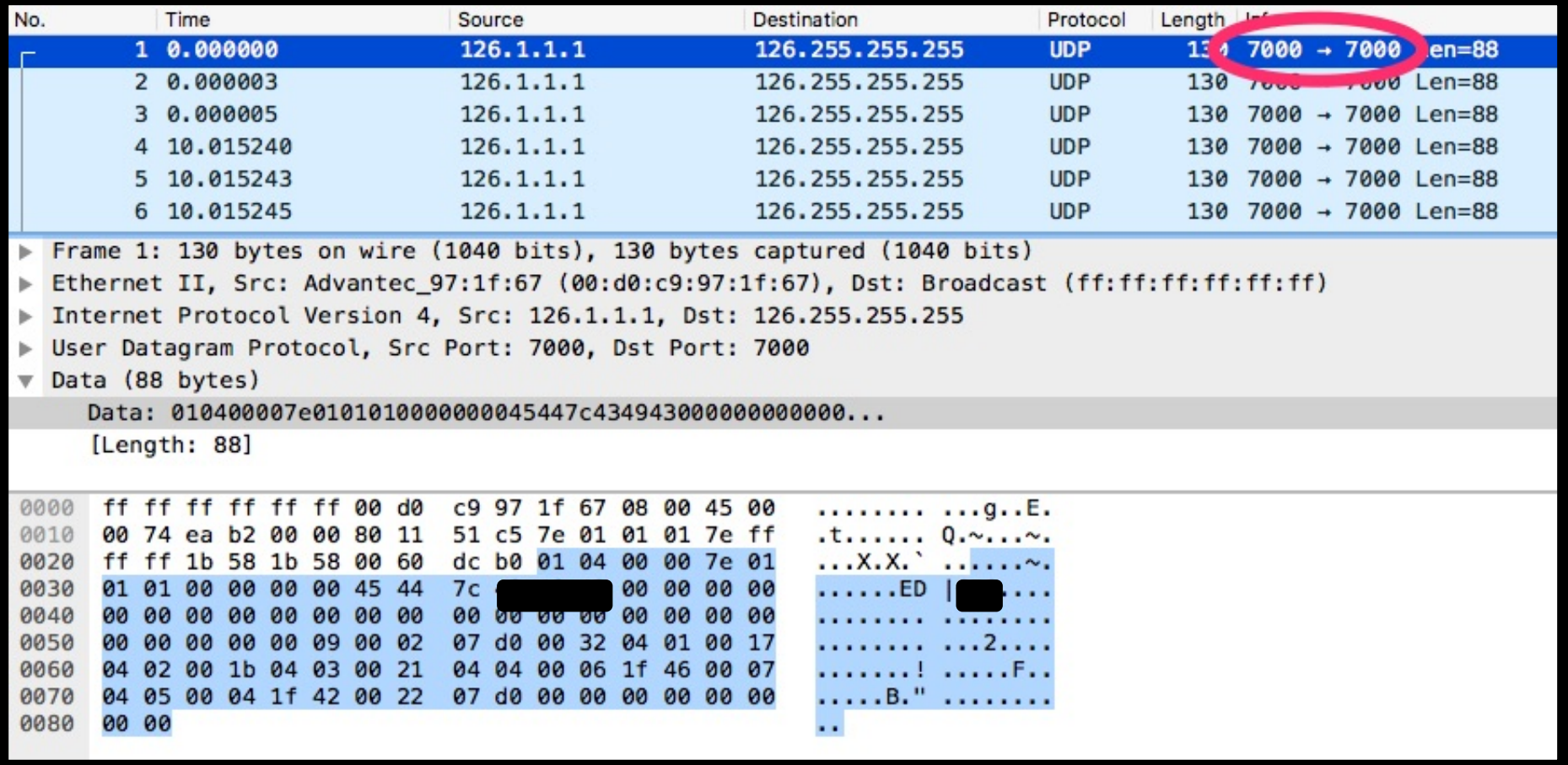
IMPROVED TESTING SETUP



CMS IDLE PACKETS

CMS IDLE PACKETS

- CMS BROADCAST PORT



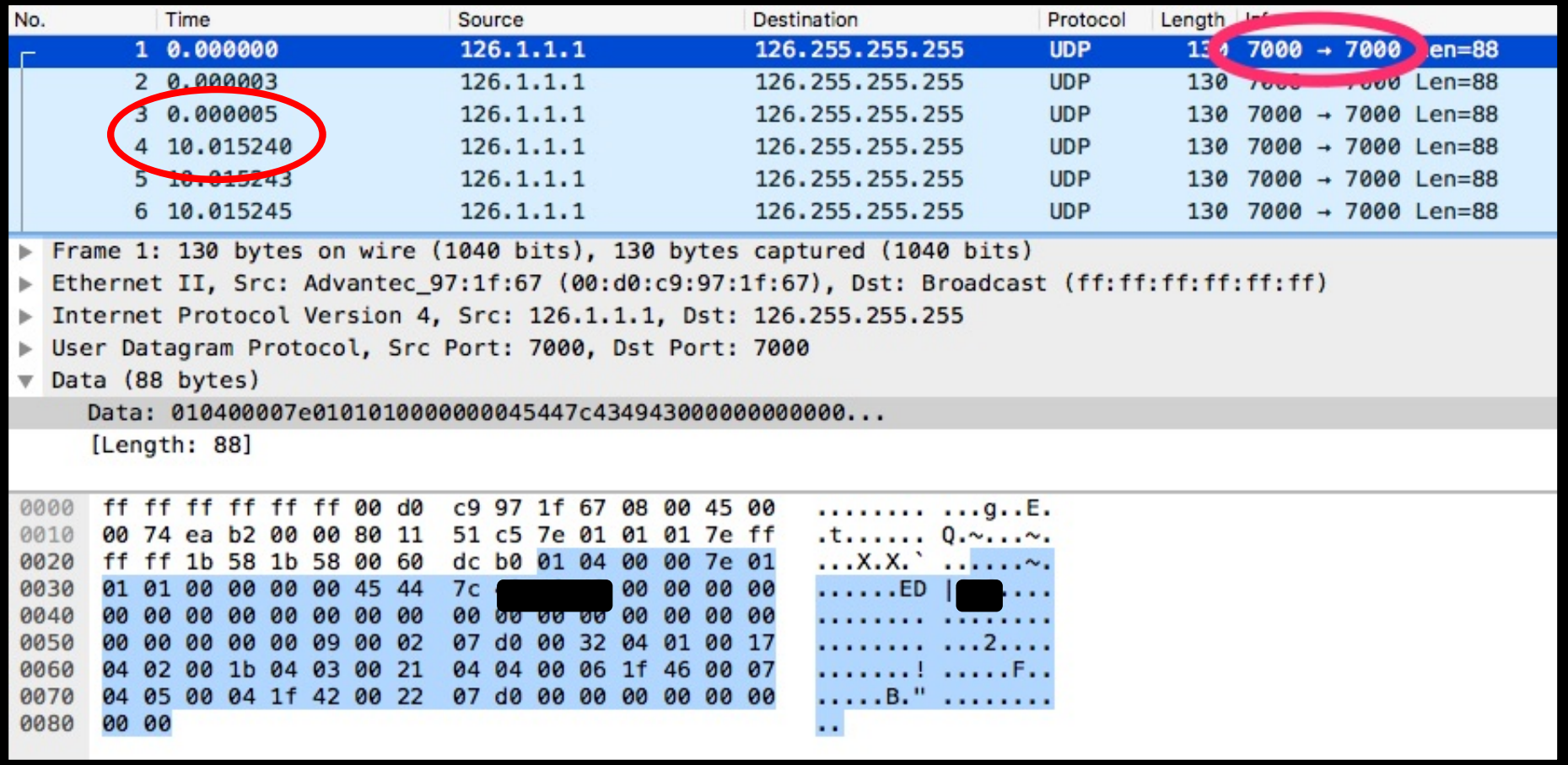
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	126.1.1.1	126.255.255.255	UDP	13[illegible]	7000 → 7000 [illegible]en=88
2	0.000003	126.1.1.1	126.255.255.255	UDP	130	[illegible] Len=88
3	0.000005	126.1.1.1	126.255.255.255	UDP	130	7000 → 7000 Len=88
4	10.015240	126.1.1.1	126.255.255.255	UDP	130	7000 → 7000 Len=88
5	10.015243	126.1.1.1	126.255.255.255	UDP	130	7000 → 7000 Len=88
6	10.015245	126.1.1.1	126.255.255.255	UDP	130	7000 → 7000 Len=88

```
▶ Frame 1: 130 bytes on wire (1040 bits), 130 bytes captured (1040 bits)
▶ Ethernet II, Src: Advantec_97:1f:67 (00:d0:c9:97:1f:67), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▶ Internet Protocol Version 4, Src: 126.1.1.1, Dst: 126.255.255.255
▶ User Datagram Protocol, Src Port: 7000, Dst Port: 7000
▼ Data (88 bytes)
    Data: 010400007e010101000000045447c434943000000000000...
    [Length: 88]
```

```
0000  ff ff ff ff ff ff 00 d0  c9 97 1f 67 08 00 45 00   ........ ...g..E.
0010  00 74 ea b2 00 00 80 11  51 c5 7e 01 01 01 7e ff   .t...... Q.~...~.
0020  ff ff 1b 58 1b 58 00 60  dc b0 01 04 00 00 7e 01   ...X.X.` ......~.
0030  01 01 00 00 00 00 45 44  7c [redacted] 00 00 00 00   ......ED |[redacted]...
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0050  00 00 00 00 00 09 00 02  07 d0 00 32 04 01 00 17   ........ ...2....
0060  04 02 00 1b 04 03 00 21  04 04 00 06 1f 46 00 07   .......! .....F..
0070  04 05 00 04 1f 42 00 22  07 d0 00 00 00 00 00 00   .....B." ........
0080  00 00                                             ..
```

CMS IDLE PACKETS

- CMS BROADCAST PORT
- EVERY 10 SECONDS



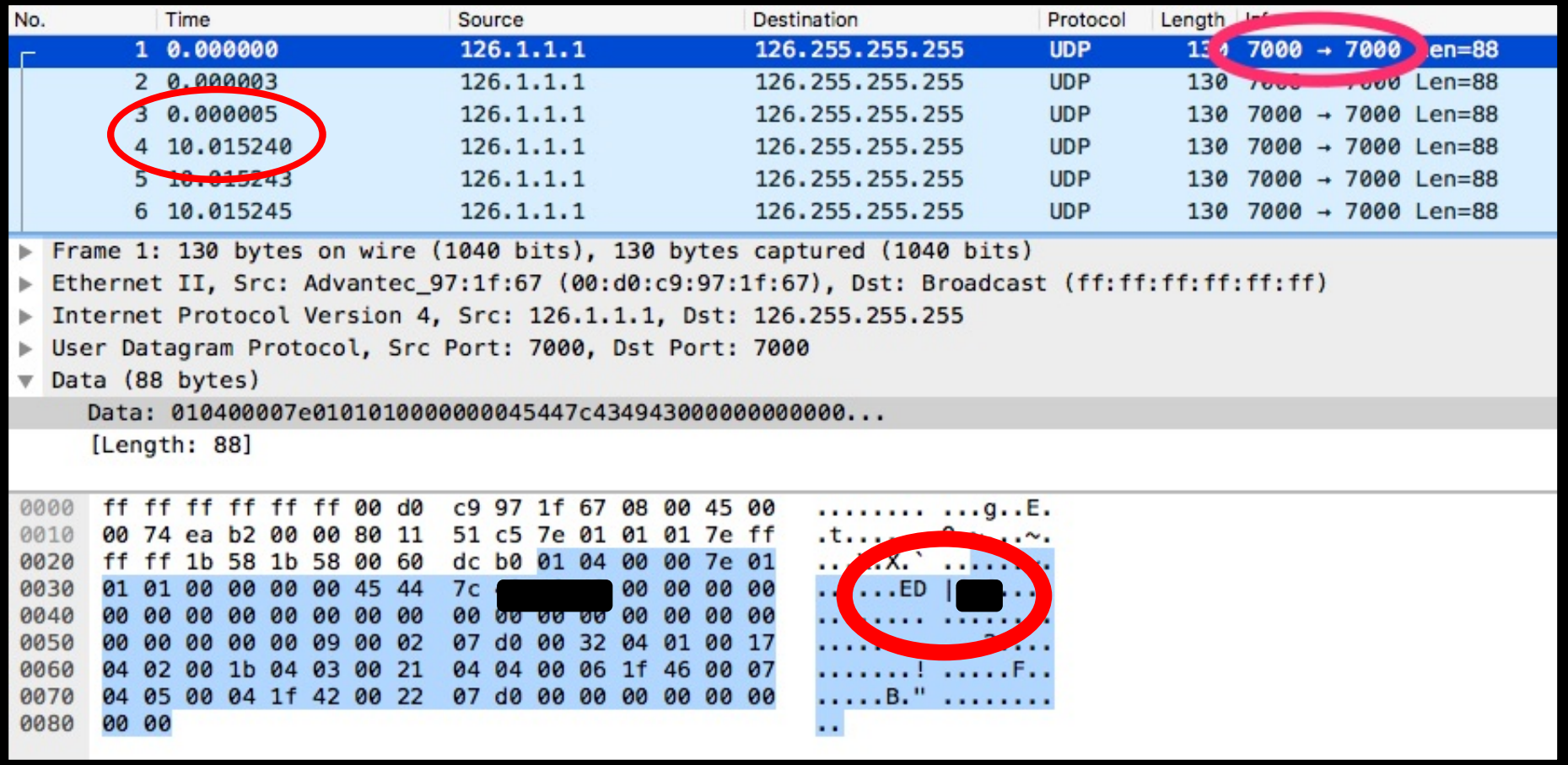
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	126.1.1.1	126.255.255.255	UDP	13[illegible]	7000 → 7000 [illegible]en=88
2	0.000003	126.1.1.1	126.255.255.255	UDP	130	[illegible] Len=88
3	0.000005	126.1.1.1	126.255.255.255	UDP	130	7000 → 7000 Len=88
4	10.015240	126.1.1.1	126.255.255.255	UDP	130	7000 → 7000 Len=88
5	10.015243	126.1.1.1	126.255.255.255	UDP	130	7000 → 7000 Len=88
6	10.015245	126.1.1.1	126.255.255.255	UDP	130	7000 → 7000 Len=88

```
▶ Frame 1: 130 bytes on wire (1040 bits), 130 bytes captured (1040 bits)
▶ Ethernet II, Src: Advantec_97:1f:67 (00:d0:c9:97:1f:67), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▶ Internet Protocol Version 4, Src: 126.1.1.1, Dst: 126.255.255.255
▶ User Datagram Protocol, Src Port: 7000, Dst Port: 7000
▼ Data (88 bytes)
    Data: 010400007e010101000000045447c434943000000000000...
    [Length: 88]
```

```
0000  ff ff ff ff ff ff 00 d0  c9 97 1f 67 08 00 45 00   ........ ...g..E.
0010  00 74 ea b2 00 00 80 11  51 c5 7e 01 01 01 7e ff   .t...... Q.~...~.
0020  ff ff 1b 58 1b 58 00 60  dc b0 01 04 00 00 7e 01   ...X.X.` ......~.
0030  01 01 00 00 00 00 45 44  7c [redacted] 00 00 00 00   ......ED |[redacted]...
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0050  00 00 00 00 00 09 00 02  07 d0 00 32 04 01 00 17   ........ ...2....
0060  04 02 00 1b 04 03 00 21  04 04 00 06 1f 46 00 07   .......! .....F..
0070  04 05 00 04 1f 42 00 22  07 d0 00 00 00 00 00 00   .....B." ........
0080  00 00                                             ..
```

CMS IDLE PACKETS

- CMS BROADCAST PORT
- EVERY 10 SECONDS
- IDENTIFIES NAME



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	126.1.1.1	126.255.255.255	UDP	13[illegible]	7000 → 7000 [illegible]en=88
2	0.000003	126.1.1.1	126.255.255.255	UDP	130	[illegible] Len=88
3	0.000005	126.1.1.1	126.255.255.255	UDP	130	7000 → 7000 Len=88
4	10.015240	126.1.1.1	126.255.255.255	UDP	130	7000 → 7000 Len=88
5	10.015243	126.1.1.1	126.255.255.255	UDP	130	7000 → 7000 Len=88
6	10.015245	126.1.1.1	126.255.255.255	UDP	130	7000 → 7000 Len=88

```
▶ Frame 1: 130 bytes on wire (1040 bits), 130 bytes captured (1040 bits)
▶ Ethernet II, Src: Advantec_97:1f:67 (00:d0:c9:97:1f:67), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▶ Internet Protocol Version 4, Src: 126.1.1.1, Dst: 126.255.255.255
▶ User Datagram Protocol, Src Port: 7000, Dst Port: 7000
▼ Data (88 bytes)
    Data: 010400007e0101010000000045447c434943000000000000...
    [Length: 88]
```

```
0000  ff ff ff ff ff ff 00 d0  c9 97 1f 67 08 00 45 00   ........ ...g..E.
0010  00 74 ea b2 00 00 80 11  51 c5 7e 01 01 01 7e ff   .t...... Q.~...~.
0020  ff ff 1b 58 1b 58 00 60  dc b0 01 04 00 00 7e 01   ...X.X.` ......~.
0030  01 01 00 00 00 00 45 44  7c [redacted] 00 00 00 00   ......ED |[redacted]....
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0050  00 00 00 00 00 09 00 02  07 d0 00 32 04 01 00 17   ........ ...2....
0060  04 02 00 1b 04 03 00 21  04 04 00 06 1f 46 00 07   .......! .....F..
0070  04 05 00 04 1f 42 00 22  07 d0 00 00 00 00 00 00   .....B." ........
0080  00 00                                             ..
```

CMS IDLE PACKETS

```
20    while ( GetExitCodeThread(hThread, &ExitCode) != 1 || ExitCode == 259 )
21    {
22      error = BroadcastRwhat(socket, destPortNum);
23      if ( error == -1 || !error )
24      {
25        winErrorCode = (const wchar_t *)WSAGetLastError();
26        swprintf(&String, (size_t)L"BroadcastRwhat() failed (Err = %d).", winErrorCode);
27        wprintf(L"%s\n", &String);
28        HIDWORD(v5) = &String;
29        LODWORD(v5) = 0;
30        log(v5);
31        break;
32      }
33      SecondsToWait = GetRwhatPeriod();
34      wprintf(L"*** RWHAT PERIOD = %d seconds. ***\n", SecondsToWait);
35      age_rwhat(3 * SecondsToWait);
36      Sleep(1000 * SecondsToWait);
37      if ( dword_405198 > 0 )
38        break;
39    }
```

CMS IDLE PACKETS

```
20    while ( GetExitCodeThread(hThread, &ExitCode) != 1 || ExitCode == 259 )
21    {
22      error = BroadcastRwhat(socket, destPortNum);
23      if ( error == -1 || !error )
24      {
25        winErrorCode = (const wchar_t *)WSAGetLastError();
26        swprintf(&String, (size_t)L"BroadcastRwhat() failed (Err = %d).", winErrorCode);
27        wprintf(L"%s\n", &String);
28        HIDWORD(v5) = &String;
29        LODWORD(v5) = 0;
30        log(v5);
31        break;
32      }
33      SecondsToWait = GetRwhatPeriod();
34      wprintf(L"*** RWHAT PERIOD = %d seconds. ***\n", SecondsToWait);
35      age_rwhat(3 * SecondsToWait);
36      Sleep(1000 * SecondsToWait);
37      if ( dword_405198 > 0 )
38        break;
39    }
```

CMS IDLE PACKETS

```
20    while ( G
21    {
22      error =
23      if ( er
24      {
25        winEr
26        swpri
27        wprin
28        HIDWO
29        LODWO
30        log(v
31        break
32      }
33      SecondsToWait = GetRwhatPeriod();
34      wprintf(L"*** RWHAT PERIOD = %d seconds. ***\n", SecondsToWait);
35      age_rwhat(3 * SecondsToWait);
36      Sleep(1000 * SecondsToWait);
37      if ( dword_405198 > 0 )
38        break;
39    }
```

```
1  int __usercall GetRwhatPeriod@<eax>(wchar_t *a1@<ebp>)
2  {
3    signed int TimeConfigValue; // esi
4
5    WaitForSingleObjectCIC(a1, dword_67A7893C, 0xFFFFFFFF, "rwhatutil.cpp", 419);
6    TimeConfigValue = *(_DWORD *)Mapped_RWHT_DB;
7    ReleaseMutex(dword_67A7893C);
8    if ( TimeConfigValue < 100 )
9      return 10;
10   if ( TimeConfigValue < 150 )
11     return 15;
12   if ( TimeConfigValue >= 200 )
13     return TimeConfigValue >= 250 ? 28 : 25;
14   return 20;
15 }
```

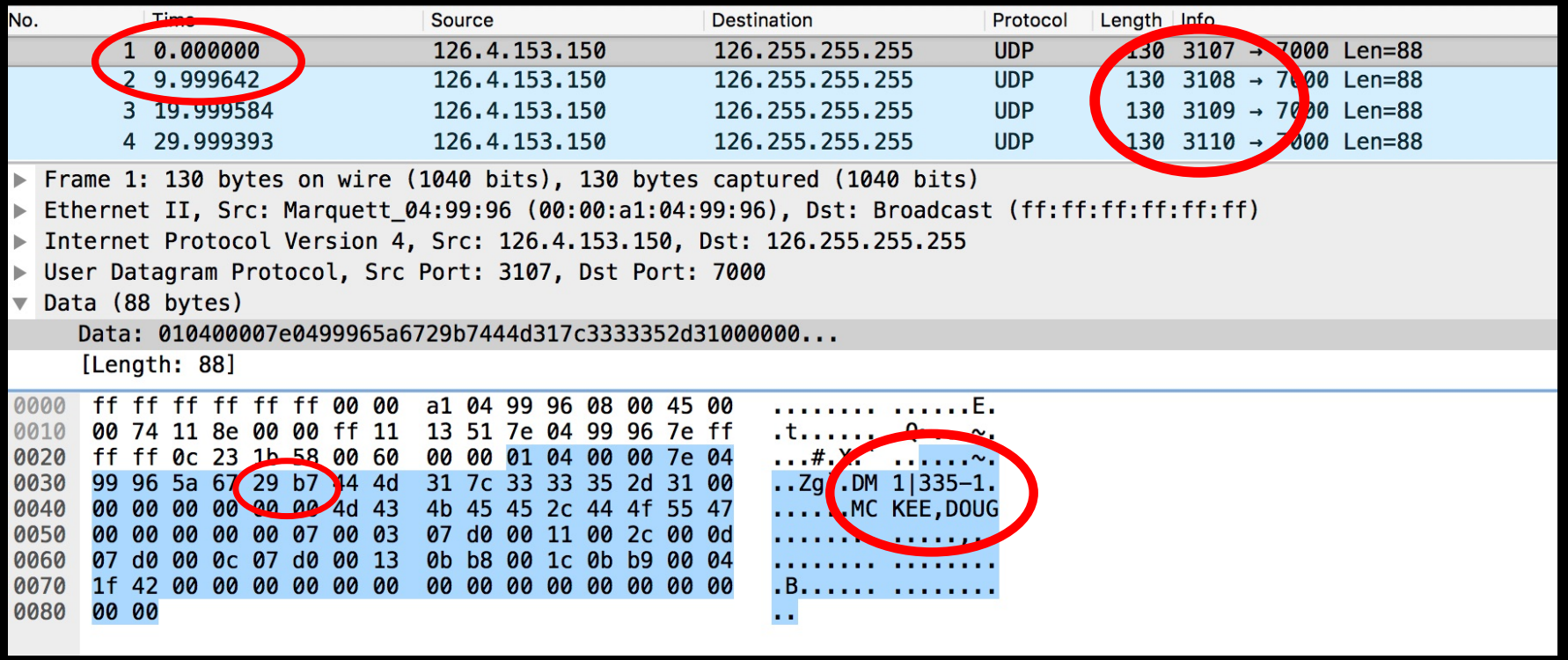
CMS IDLE PACKETS

```
int __usercall GetRwhatPeriod@<eax>(wchar_t *a1@<ebp>)
{
  signed int TimeConfigValue; // esi

  WaitForSingleObjectCIC(a1, dword_67A7893C, 0xFFFFFFFF, "rwhatutil.cpp", 419);
  TimeConfigValue = *(_DWORD *)Mapped_RWHT_DB;
  ReleaseMutex(dword_67A7893C);
  if ( TimeConfigValue < 100 )
    return 10;
  if ( TimeConfigValue < 150 )
    return 15;
  if ( TimeConfigValue >= 200 )
    return TimeConfigValue >= 250 ? 28 : 25;
  return 20;
}
```

```
    while ( G
    {
      error =
      if ( er
      {
        winEr
        swpri
        wprin
        HIDWO
        LODWO
        log(v
        break
      }
      SecondsToWait = GetRwhatPeriod();
      wprintf(L"*** RWHAT PERIOD = %d seconds. ***\n", SecondsToWait);
      age_rwhat(3 * SecondsToWait);
      Sleep(1000 * SecondsToWait);
      if ( dword_405198 > 0 )
        break;
    }
```

PM IDLE/BROADCAST PACKETS

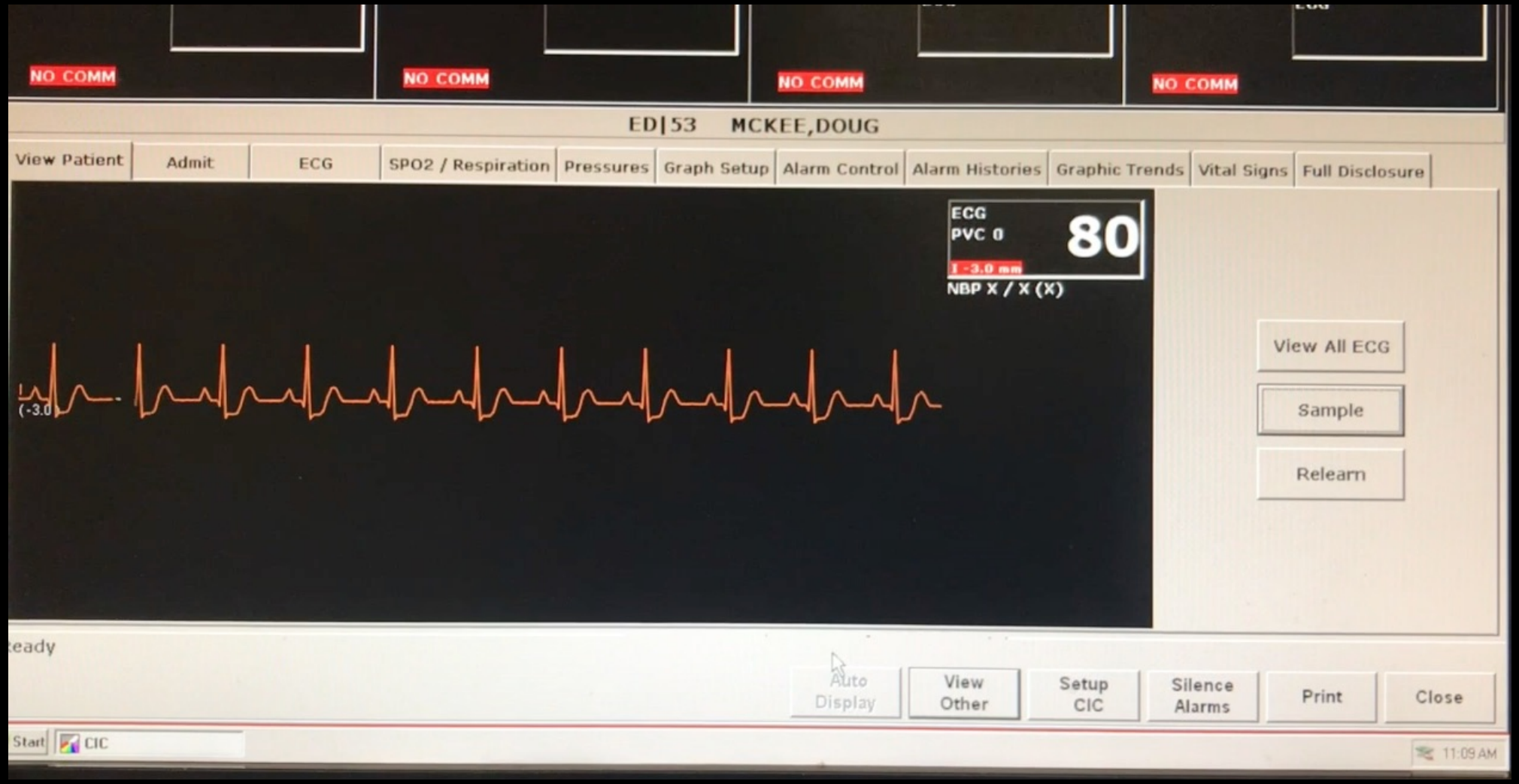
- Every 10 seconds
- Src Ports increment
- Counter by 0xA
- Patient Info



PM IDLE/BROADCAST PACKETS

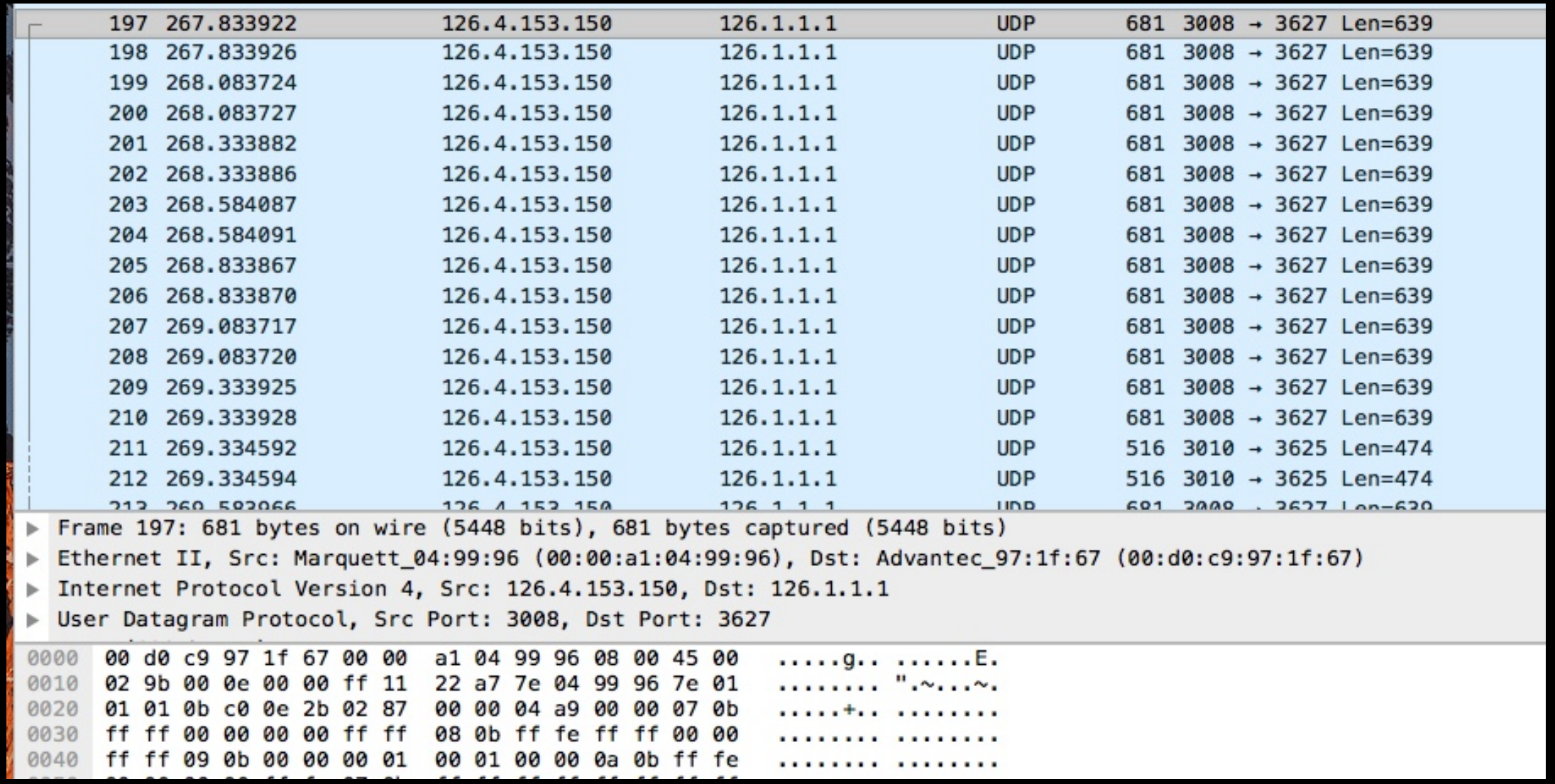
```
35  PatientNameAndBedName = (unsigned __int16 *)((char *)payload + 12);
36  v8 = (unsigned __int16 *)v6;
37  v9 = wcslen((const wchar_t *)payload + 6);
38  if ( v9 < 32 )
39    fill((unsigned __int16 *)payload + v9 + 6, 32 - v9, 0);
40  parse_logical_name(PatientNameAndBedName, 0x20u, &PatientName, &UnitName);
41  a1 = wcscpy;                                  // nonsense
42  if ( !PatientName )
43  {
44    wcscpy(&PatientName, noname);
45    v19 = 0;
46    if ( UnitName )
47      goto LABEL_10;
48    goto LABEL_9;
49  }
50  if ( !UnitName )
51  {
52LABEL_9:
53    wcscpy(&UnitName, nounit);
54    v17 = 0;
55LABEL_10:
56    logicalName = (wchar_t *(__cdecl *)(wchar_t *, const wchar_t *))make_logical_name(&UnitName, &PatientName);
```

CMS – NORMAL OPERATION WITH ECG SIMULATOR



CMS – NORMAL OPERATION WITH ECG SIMULATOR

NORMAL TRAFFIC



NORMAL TRAFFIC

PM

	267.833922	126.4.153.150	126.1.1.1	UDP	681 3008 → 3627 Len=639
	833926	126.4.153.150	126.1.1.1	UDP	681 3008 → 3627 Len=639
	268.083724	126.4.153.150	126.1.1.1	UDP	681 3008 → 3627 Len=639
	268.083727	126.4.153.150	126.1.1.1	UDP	681 3008 → 3627 Len=639
201	268.333882	126.4.153.150	126.1.1.1	UDP	681 3008 → 3627 Len=639
202	268.333886	126.4.153.150	126.1.1.1	UDP	681 3008 → 3627 Len=639
203	268.584087	126.4.153.150	126.1.1.1	UDP	681 3008 → 3627 Len=639
204	268.584091	126.4.153.150	126.1.1.1	UDP	681 3008 → 3627 Len=639
205	268.833867	126.4.153.150	126.1.1.1	UDP	681 3008 → 3627 Len=639
206	268.833870	126.4.153.150	126.1.1.1	UDP	681 3008 → 3627 Len=639
207	269.083717	126.4.153.150	126.1.1.1	UDP	681 3008 → 3627 Len=639
208	269.083720	126.4.153.150	126.1.1.1	UDP	681 3008 → 3627 Len=639
209	269.333925	126.4.153.150	126.1.1.1	UDP	681 3008 → 3627 Len=639
210	269.333928	126.4.153.150	126.1.1.1	UDP	681 3008 → 3627 Len=639
211	269.334592	126.4.153.150	126.1.1.1	UDP	516 3010 → 3625 Len=474
212	269.334594	126.4.153.150	126.1.1.1	UDP	516 3010 → 3625 Len=474
213	269.583966	126.4.153.150	126.1.1.1	UDP	681 3008 → 3627 Len=639

```
▶ Frame 197: 681 bytes on wire (5448 bits), 681 bytes captured (5448 bits)
▶ Ethernet II, Src: Marquett_04:99:96 (00:00:a1:04:99:96), Dst: Advantec_97:1f:67 (00:d0:c9:97:1f:67)
▶ Internet Protocol Version 4, Src: 126.4.153.150, Dst: 126.1.1.1
▶ User Datagram Protocol, Src Port: 3008, Dst Port: 3627
```

```
0000  00 d0 c9 97 1f 67 00 00  a1 04 99 96 08 00 45 00   .....g.. ......E.
0010  02 9b 00 0e 00 00 ff 11  22 a7 7e 04 99 96 7e 01   ........ ".~...~.
0020  01 01 0b c0 0e 2b 02 87  00 00 04 a9 00 00 07 0b   .....+.. ........
0030  ff ff 00 00 00 00 ff ff  08 0b ff fe ff ff 00 00   ........ ........
0040  ff ff 09 0b 00 00 00 01  00 01 00 00 0a 0b ff fe   ........ ........
```

NORMAL TRAFFIC

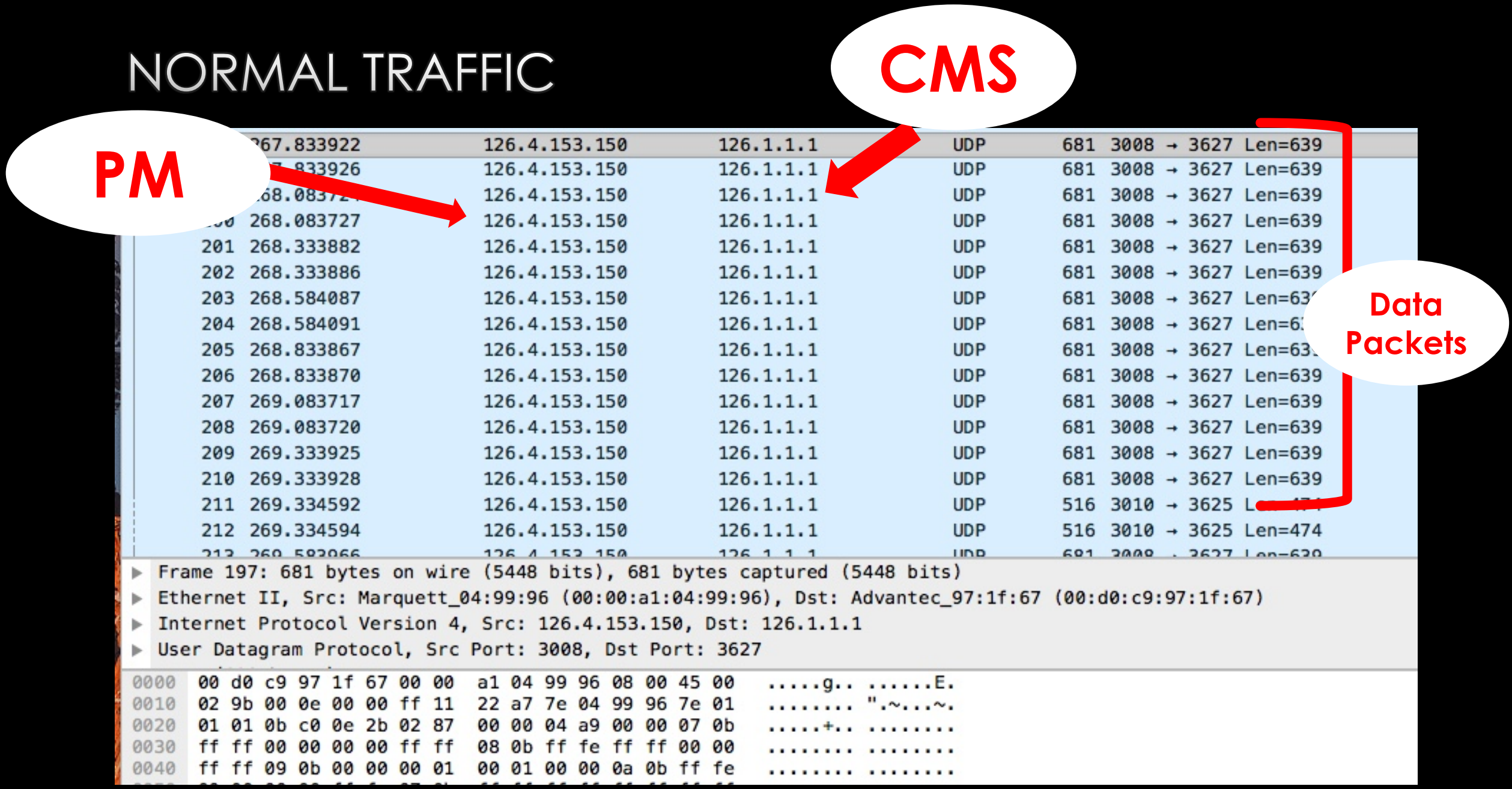
CMS

PM

	267.833922	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639
	833926	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639
	268.08372	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639
	268.083727	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639
201	268.333882	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639
202	268.333886	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639
203	268.584087	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639
204	268.584091	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639
205	268.833867	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639
206	268.833870	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639
207	269.083717	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639
208	269.083720	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639
209	269.333925	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639
210	269.333928	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639
211	269.334592	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
212	269.334594	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
213	269.583966	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639

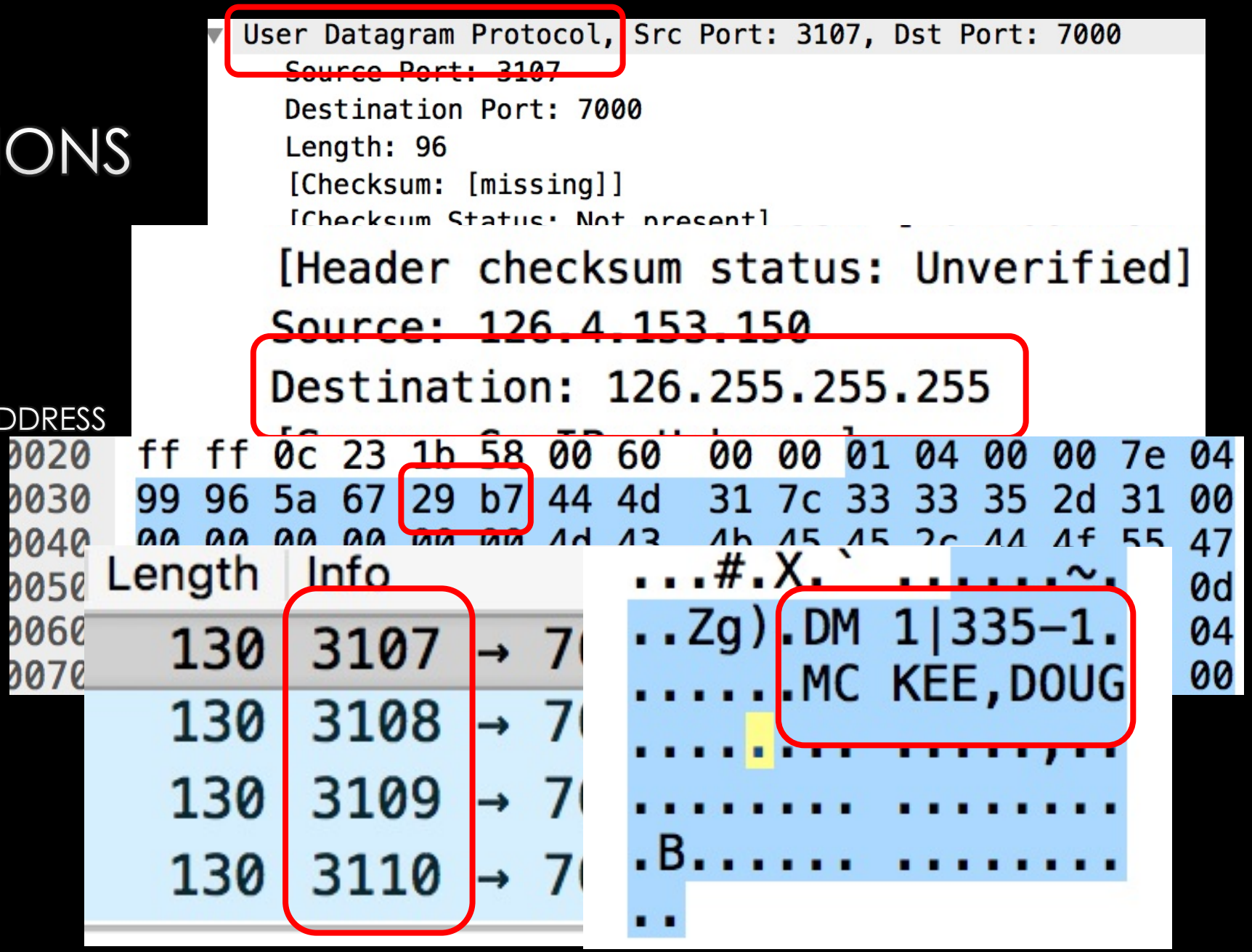
```
▶ Frame 197: 681 bytes on wire (5448 bits), 681 bytes captured (5448 bits)
▶ Ethernet II, Src: Marquett_04:99:96 (00:00:a1:04:99:96), Dst: Advantec_97:1f:67 (00:d0:c9:97:1f:67)
▶ Internet Protocol Version 4, Src: 126.4.153.150, Dst: 126.1.1.1
▶ User Datagram Protocol, Src Port: 3008, Dst Port: 3627

0000  00 d0 c9 97 1f 67 00 00  a1 04 99 96 08 00 45 00   .....g.. ......E.
0010  02 9b 00 0e 00 00 ff 11  22 a7 7e 04 99 96 7e 01   ........ ".~...~.
0020  01 01 0b c0 0e 2b 02 87  00 00 04 a9 00 00 07 0b   .....+.. ........
0030  ff ff 00 00 00 00 ff ff  08 0b ff fe ff ff 00 00   ........ ........
0040  ff ff 09 0b 00 00 00 01  00 01 00 00 0a 0b ff fe   ........ ........
```

NORMAL TRAFFIC



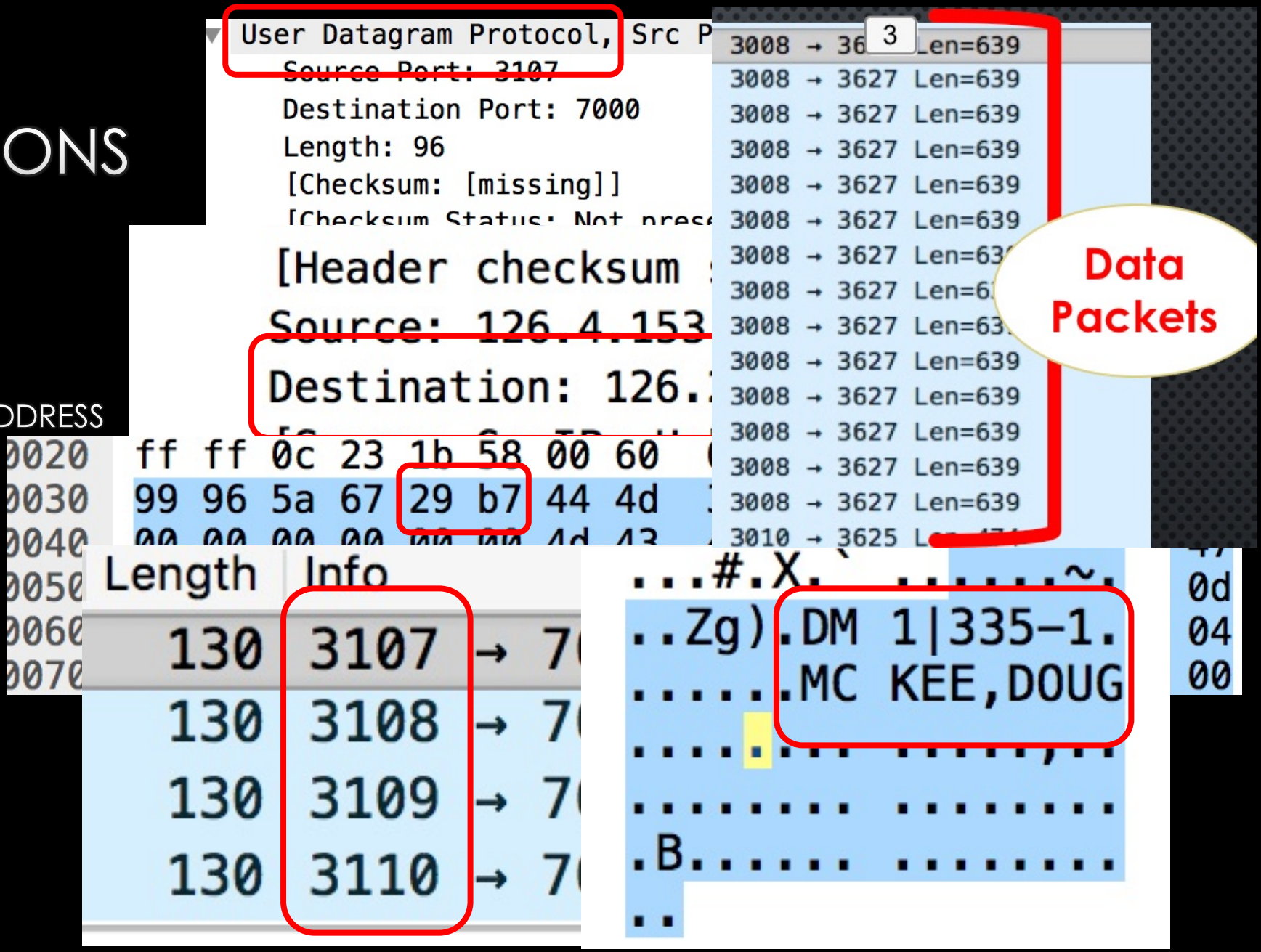
BASIC OBSERVATIONS

- UDP Packets
- Utilization of Broadcast address
- Counters
- Varying Port numbers
- Not encrypted



BASIC OBSERVATIONS

- UDP Packets
- Utilization of Broadcast address
- Counters
- Varying Port numbers
- Not encrypted
- Data Packets



BASIC OBSERVATIONS

- UDP Packets
- Utilization of Broadcast address
- Counters
- Varying Port numbers
- Not encrypted
- Data Packets

- Replay attack ?

SIMPLE REPLAY!!!!

SIMPLE REPLAY!!!!

UNDERSTANDING THE PACKETS: HANDSHAKE

UNDERSTANDING THE PACKETS: HANDSHAKE

173	260.261351	126.1.1.1	126.255.255.255	UDP	130	7000 → 7000 Len=88
174	260.261353	126.1.1.1	126.255.255.255	UDP	130	7000 → 7000 Len=88
183	263.893005	126.1.1.1	126.4.153.150	UDP	104	3625 → 2000 Len=62
184	263.893007	126.1.1.1	126.4.153.150	UDP	104	3625 → 2000 Len=62
185	265.318585	126.4.153.150	126.255.255.255	UDP	130	3015 → 7000 Len=88
186	265.318588	126.4.153.150	126.255.255.255	UDP	130	3015 → 7000 Len=88
187	265.318590	126.4.153.150	126.255.255.255	UDP	130	3015 → 7000 Len=88
188	265.318592	126.4.153.150	126.255.255.255	UDP	130	3015 → 7000 Len=88
189	265.334402	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
190	265.334405	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
191	265.381835	126.1.1.1	126.4.153.150	UDP	104	3626 → 2000 Len=62
192	265.381838	126.1.1.1	126.4.153.150	UDP	104	3626 → 2000 Len=62
193	267.334357	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
194	267.334360	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
195	267.421811	126.1.1.1	126.4.153.150	UDP	104	3627 → 2000 Len=62
196	267.421813	126.1.1.1	126.4.153.150	UDP	104	3627 → 2000 Len=62
197	267.833922	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639
198	267.833926	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639

UNDERSTANDING THE PACKETS: HANDSHAKE

173	260.261351	126.1.1.1	126.255.255.255	UDP	130	7000 → 7000 Len=88
174	260.261353	126.1.1.1	126.255.255.255	UDP	130	7000 → 7000 Len=88
183	263.893005	126.1.1.1	126.4.153.150	UDP	104	3625 → 2000 Len=62
184	263.893007	126.1.1.1	126.4.153.150	UDP	104	3625 → 2000 Len=62
185	265.318585	126.4.153.150	126.255.255.255	UDP	130	3015 → 7000 Len=88
186	265.318588	126.4.153.150	126.255.255.255	UDP	130	3015 → 7000 Len=88
187	265.318590	126.4.153.150	126.255.255.255	UDP	130	3015 → 7000 Len=88
188	265.318592	126.4.153.150	126.255.255.255	UDP	130	3015 → 7000 Len=88
189	265.334402	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
190	265.334405	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
191	265.381835	126.1.1.1	126.4.153.150	UDP	104	3626 → 2000 Len=62
192	265.381838	126.1.1.1	126.4.153.150	UDP	104	3626 → 2000 Len=62
193	267.334357	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
194	267.334360	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
195	267.421811	126.1.1.1	126.4.153.150	UDP	104	3627 → 2000 Len=62
196	267.421813	126.1.1.1	126.4.153.150	UDP	104	3627 → 2000 Len=62
197	267.833922	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639
198	267.833926	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639

CMS Opens Channel "SYN"

UNDERSTANDING THE PACKETS: HANDSHAKE

173	260.261351	126.1.1.1	126.255.255.255	UDP	130	7000 → 7000 Len=88
174	260.261353	126.1.1.1	126.255.255.255	UDP	130	7000 → 7000 Len=88
183	263.893005	126.1.1.1	126.4.153.150	UDP	104	3625 → 2000 Len=62
184	263.893007	126.1.1.1	126.4.153.150	UDP	104	3625 → 2000 Len=62
185	265.318585	126.4.153.150	126.255.255.255	UDP	130	3015 → 7000 Len=88
186	265.318588	126.4.153.150	126.255.255.255	UDP	130	3015 → 7000 Len=88
187	265.318590	126.4.153.150	126.255.255.255	UDP	130	3015 → 7000 Len=88
188	265.318592	126.4.153.150	126.255.255.255	UDP	130	3015 → 7000 Len=88
189	265.334402	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
190	265.334405	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
191	265.381835	126.1.1.1	126.4.153.150	UDP	104	3626 → 2000 Len=62
192	265.381838	126.1.1.1	126.4.153.150	UDP	104	3626 → 2000 Len=62
193	267.334357	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
194	267.334360	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
195	267.421811	126.1.1.1	126.4.153.150	UDP	104	3627 → 2000 Len=62
196	267.421813	126.1.1.1	126.4.153.150	UDP	104	3627 → 2000 Len=62
197	267.833922	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639
198	267.833926	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639

CMS Opens Channel "SYN"

PM "SYN, ACK"

UNDERSTANDING THE PACKETS: HANDSHAKE

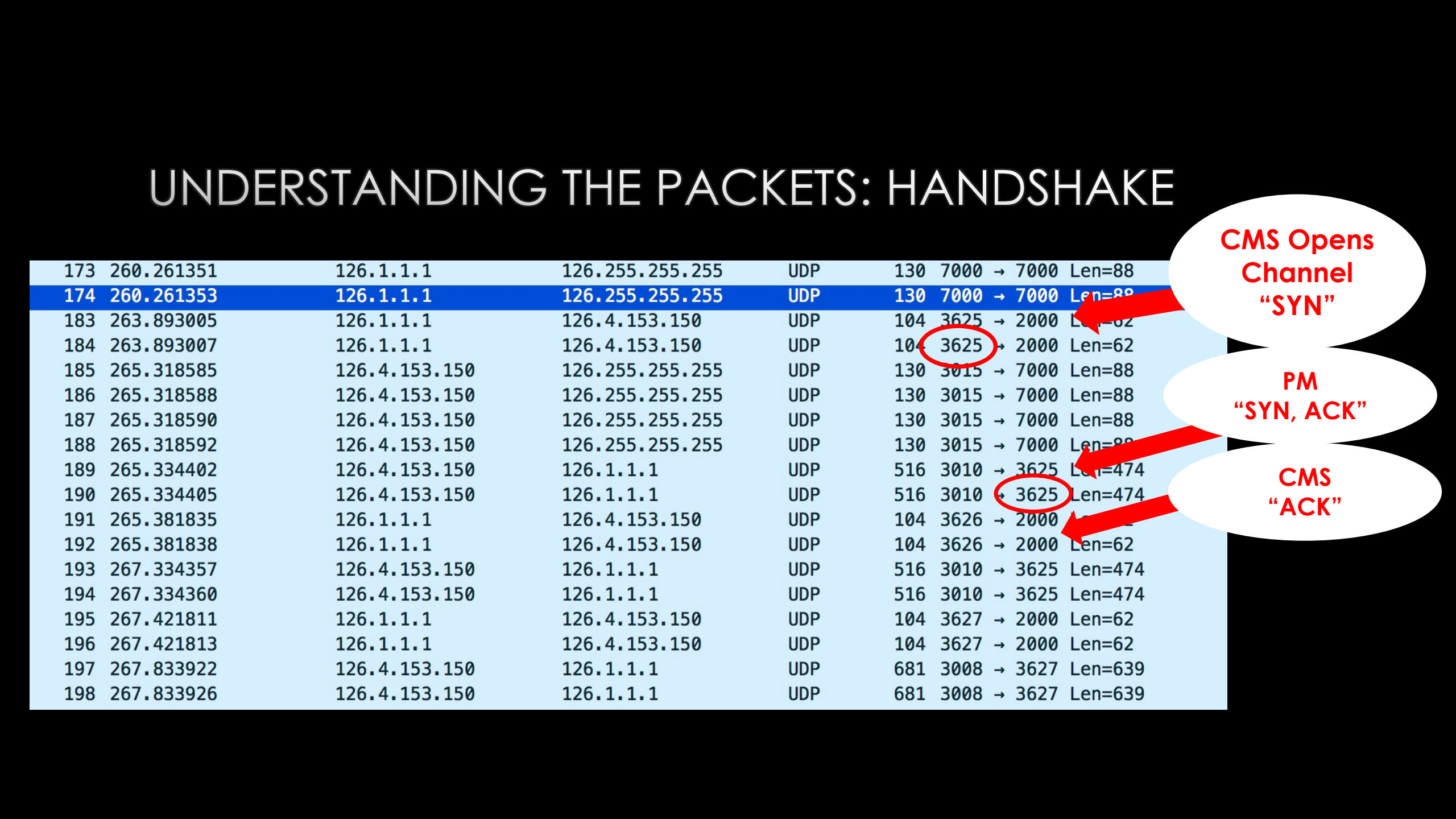
173	260.261351	126.1.1.1	126.255.255.255	UDP	130	7000 → 7000 Len=88
174	260.261353	126.1.1.1	126.255.255.255	UDP	130	7000 → 7000 Len=88
183	263.893005	126.1.1.1	126.4.153.150	UDP	104	3625 → 2000 Len=62
184	263.893007	126.1.1.1	126.4.153.150	UDP	104	3625 → 2000 Len=62
185	265.318585	126.4.153.150	126.255.255.255	UDP	130	3015 → 7000 Len=88
186	265.318588	126.4.153.150	126.255.255.255	UDP	130	3015 → 7000 Len=88
187	265.318590	126.4.153.150	126.255.255.255	UDP	130	3015 → 7000 Len=88
188	265.318592	126.4.153.150	126.255.255.255	UDP	130	3015 → 7000 Len=88
189	265.334402	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
190	265.334405	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
191	265.381835	126.1.1.1	126.4.153.150	UDP	104	3626 → 2000 Len=62
192	265.381838	126.1.1.1	126.4.153.150	UDP	104	3626 → 2000 Len=62
193	267.334357	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
194	267.334360	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
195	267.421811	126.1.1.1	126.4.153.150	UDP	104	3627 → 2000 Len=62
196	267.421813	126.1.1.1	126.4.153.150	UDP	104	3627 → 2000 Len=62
197	267.833922	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639
198	267.833926	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639

CMS Opens Channel "SYN"

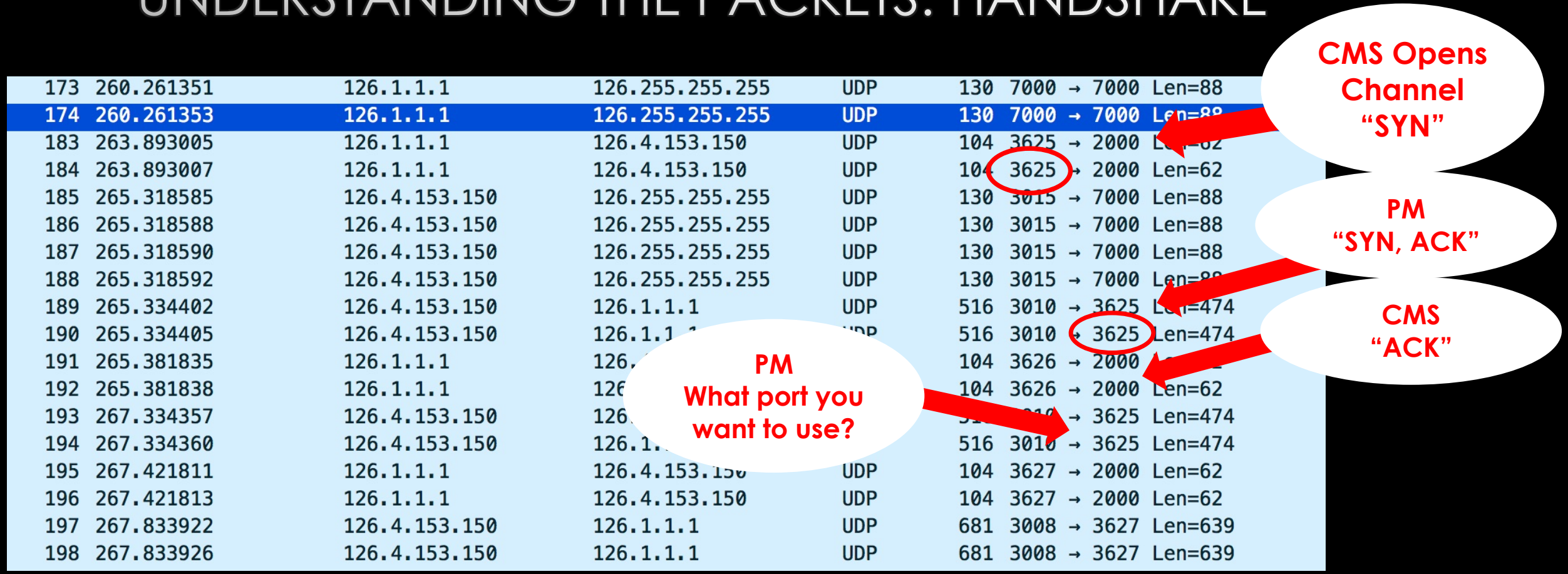
PM "SYN, ACK"

UNDERSTANDING THE PACKETS: HANDSHAKE

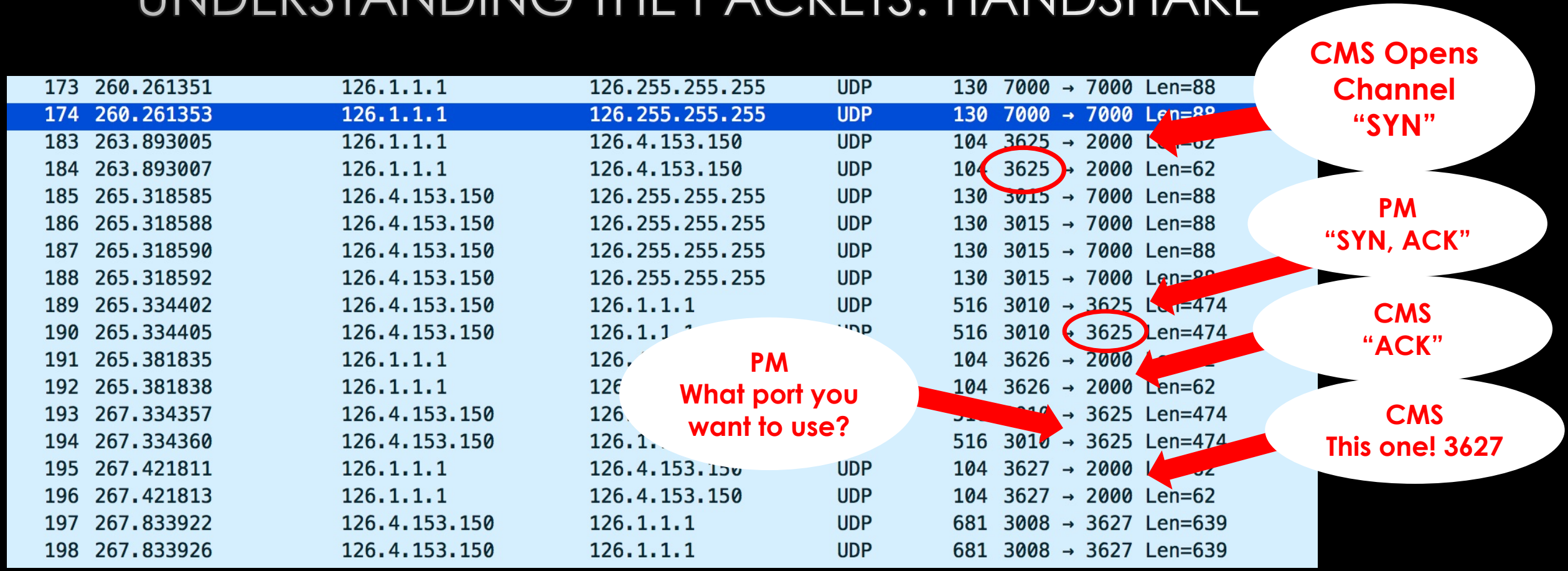
173	260.261351	126.1.1.1	126.255.255.255	UDP	130	7000 → 7000 Len=88
174	260.261353	126.1.1.1	126.255.255.255	UDP	130	7000 → 7000 Len=88
183	263.893005	126.1.1.1	126.4.153.150	UDP	104	3625 → 2000 Len=62
184	263.893007	126.1.1.1	126.4.153.150	UDP	104	3625 → 2000 Len=62
185	265.318585	126.4.153.150	126.255.255.255	UDP	130	3015 → 7000 Len=88
186	265.318588	126.4.153.150	126.255.255.255	UDP	130	3015 → 7000 Len=88
187	265.318590	126.4.153.150	126.255.255.255	UDP	130	3015 → 7000 Len=88
188	265.318592	126.4.153.150	126.255.255.255	UDP	130	3015 → 7000 Len=88
189	265.334402	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
190	265.334405	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
191	265.381835	126.1.1.1	126.4.153.150	UDP	104	3626 → 2000 Len=62
192	265.381838	126.1.1.1	126.4.153.150	UDP	104	3626 → 2000 Len=62
193	267.334357	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
194	267.334360	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
195	267.421811	126.1.1.1	126.4.153.150	UDP	104	3627 → 2000 Len=62
196	267.421813	126.1.1.1	126.4.153.150	UDP	104	3627 → 2000 Len=62
197	267.833922	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639
198	267.833926	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639



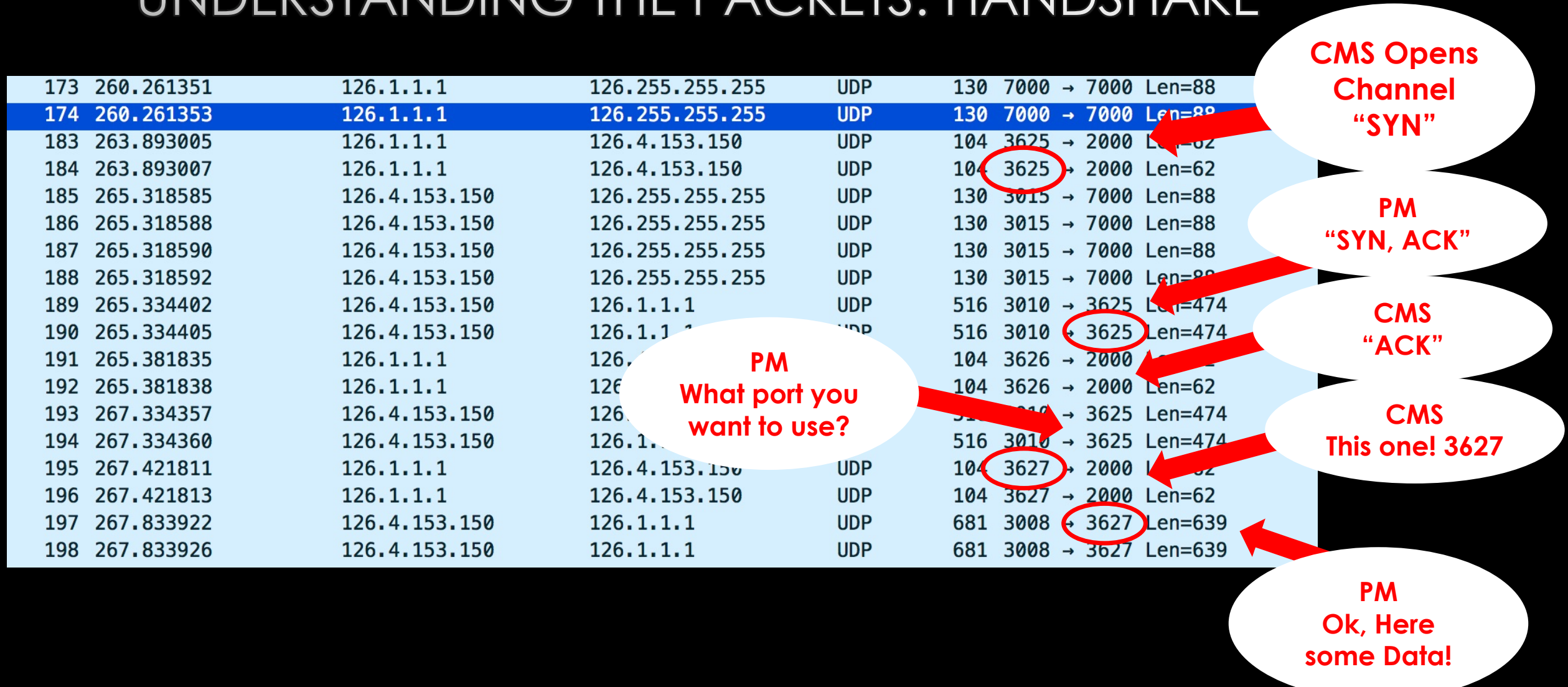
UNDERSTANDING THE PACKETS: HANDSHAKE



UNDERSTANDING THE PACKETS: HANDSHAKE



UNDERSTANDING THE PACKETS: HANDSHAKE



HANDSHAKE HACKED!!

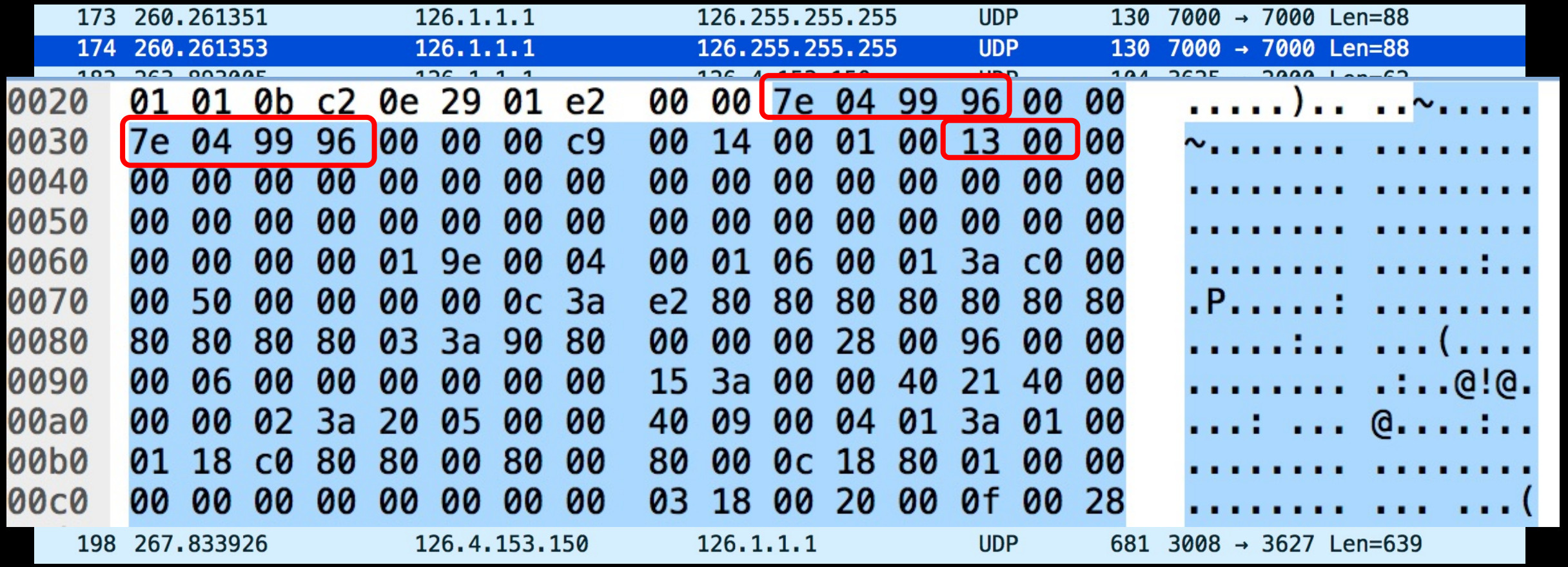
HANDSHAKE HACKED!!

CLOSER LOOK AT “SYN,ACK” PACKETS

CLOSER LOOK AT "SYN,ACK" PACKETS

173	260.261351	126.1.1.1	126.255.255.255	UDP	130	7000 → 7000 Len=88
174	260.261353	126.1.1.1	126.255.255.255	UDP	130	7000 → 7000 Len=88
183	263.893005	126.1.1.1	126.4.153.150	UDP	104	3625 → 2000 Len=62
184	263.893007	126.1.1.1	126.4.153.150	UDP	104	3625 → 2000 Len=62
185	265.318585	126.4.153.150	126.255.255.255	UDP	130	3015 → 7000 Len=88
186	265.318588	126.4.153.150	126.255.255.255	UDP	130	3015 → 7000 Len=88
187	265.318590	126.4.153.150	126.255.255.255	UDP	130	3015 → 7000 Len=88
188	265.318592	126.4.153.150	126.255.255.255	UDP	130	3015 → 7000 Len=88
189	265.334402	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
190	265.334405	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
191	265.381835	126.1.1.1	126.4.153.150	UDP	104	3626 → 2000 Len=62
192	265.381838	126.1.1.1	126.4.153.150	UDP	104	3626 → 2000 Len=62
193	267.334357	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
194	267.334360	126.4.153.150	126.1.1.1	UDP	516	3010 → 3625 Len=474
195	267.421811	126.1.1.1	126.4.153.150	UDP	104	3627 → 2000 Len=62
196	267.421813	126.1.1.1	126.4.153.150	UDP	104	3627 → 2000 Len=62
197	267.833922	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639
198	267.833926	126.4.153.150	126.1.1.1	UDP	681	3008 → 3627 Len=639

CLOSER LOOK AT "SYN,ACK" PACKETS



EMULATION OR REPLAY ATTACK STEPS

1. Send broadcast packets to allow CMS to detect device
2. Perform handshake
    1. Accounting for port changes and counters
3. Send captured heartbeat packets

EMULATION DEMO

IMPACT SCENARIO

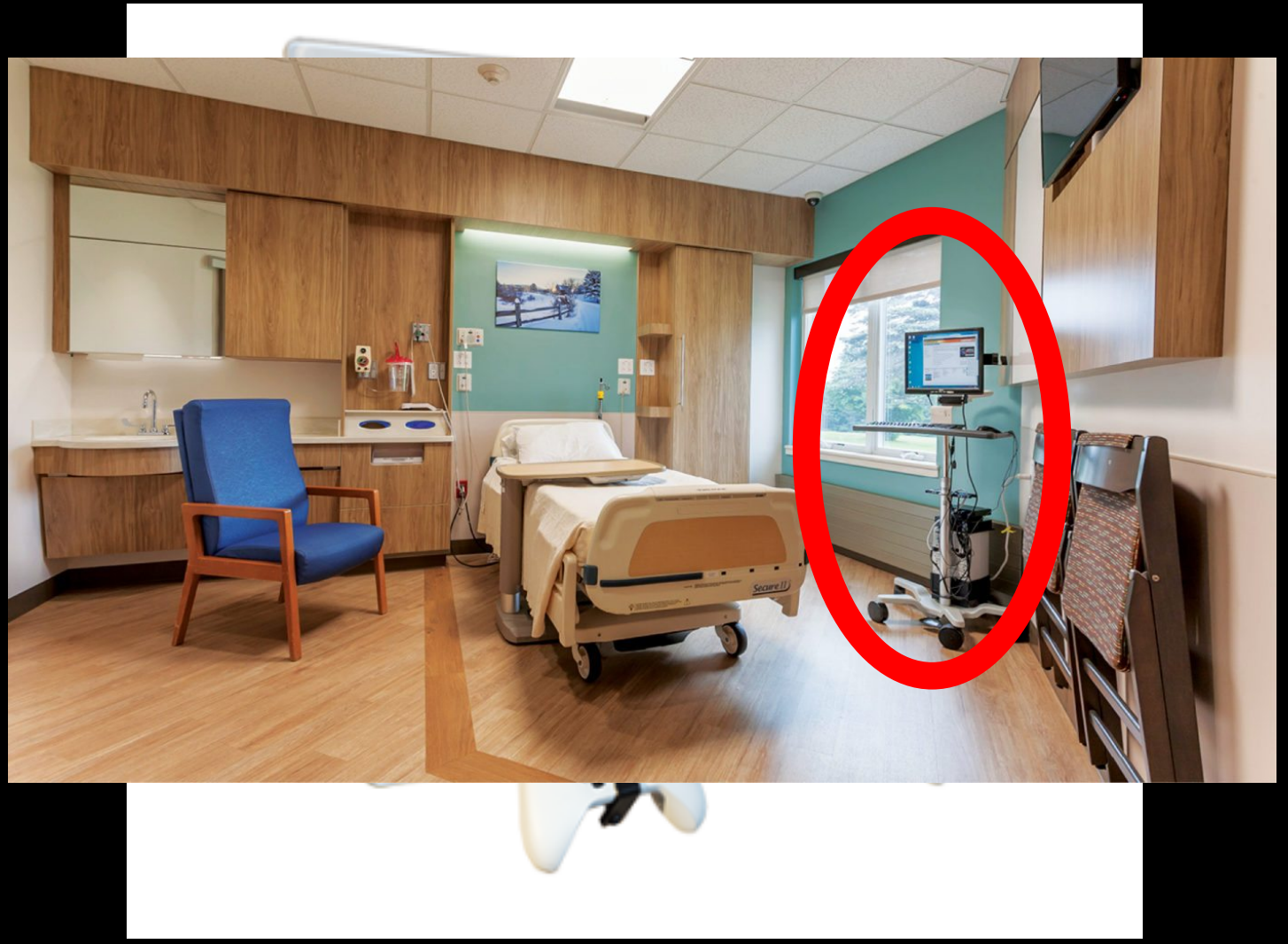
- Loaded onto raspberry pi
- Plug PI into Ethernet jack
- Can harm patient unnoticed

IMPACT SCENARIO

- Loaded onto raspberry pi
- Plug PI into Ethernet jack
- Can harm patient unnoticed

IMPACT SCENARIO

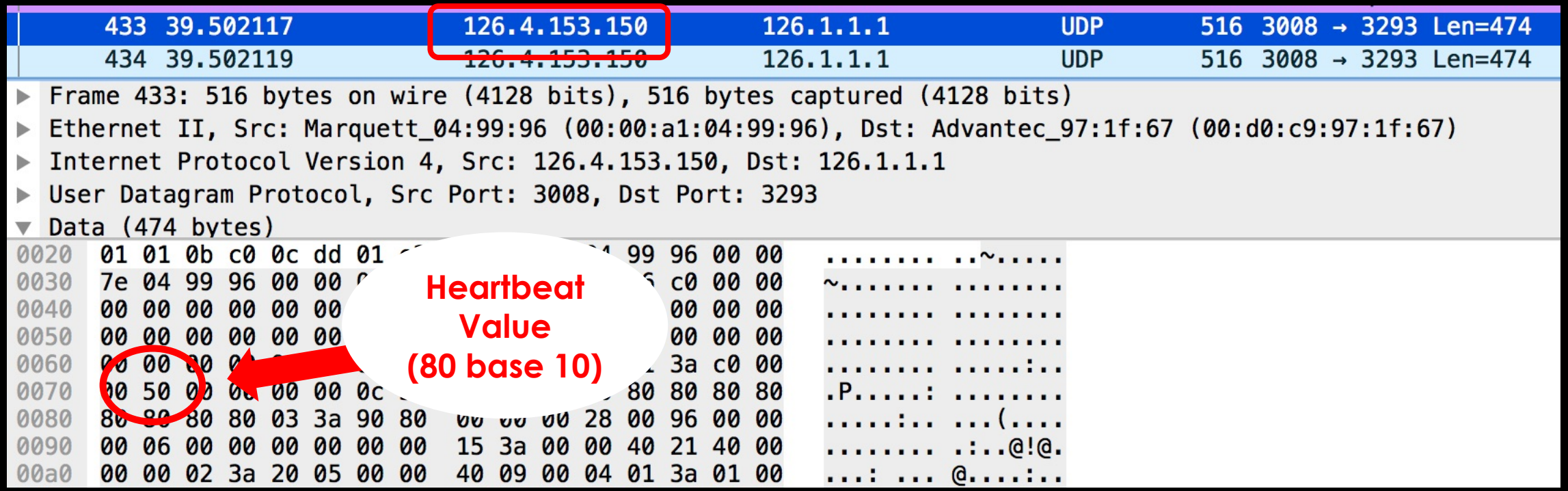
- Loaded onto raspberry pi
- Plug PI into Ethernet jack
- Can harm patient unnoticed



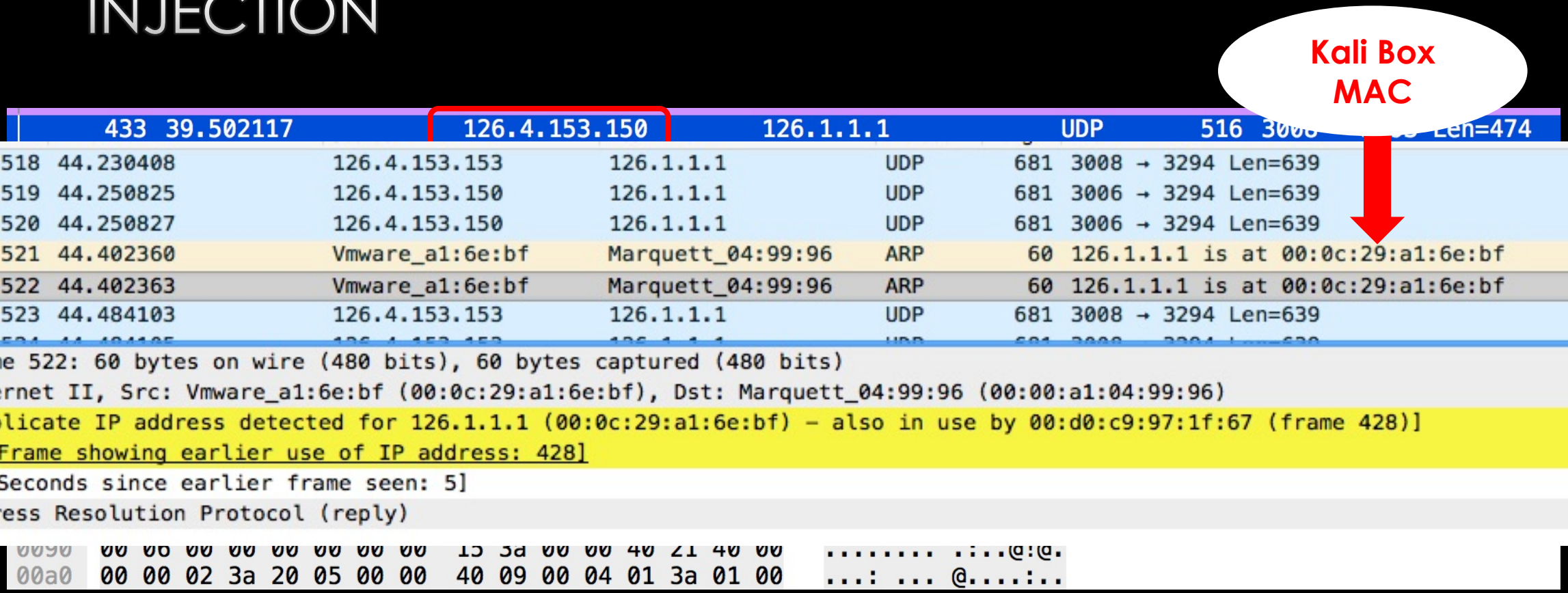
WHAT ABOUT MODIFYING EXISTING DATA?

- HANDSHAKE IS ALREADY COMPLETED
- DATA PORT? RESPOND PACKET PORT?
  - ARP SPOOF
- SEND NEW DATA ON SAME PORT

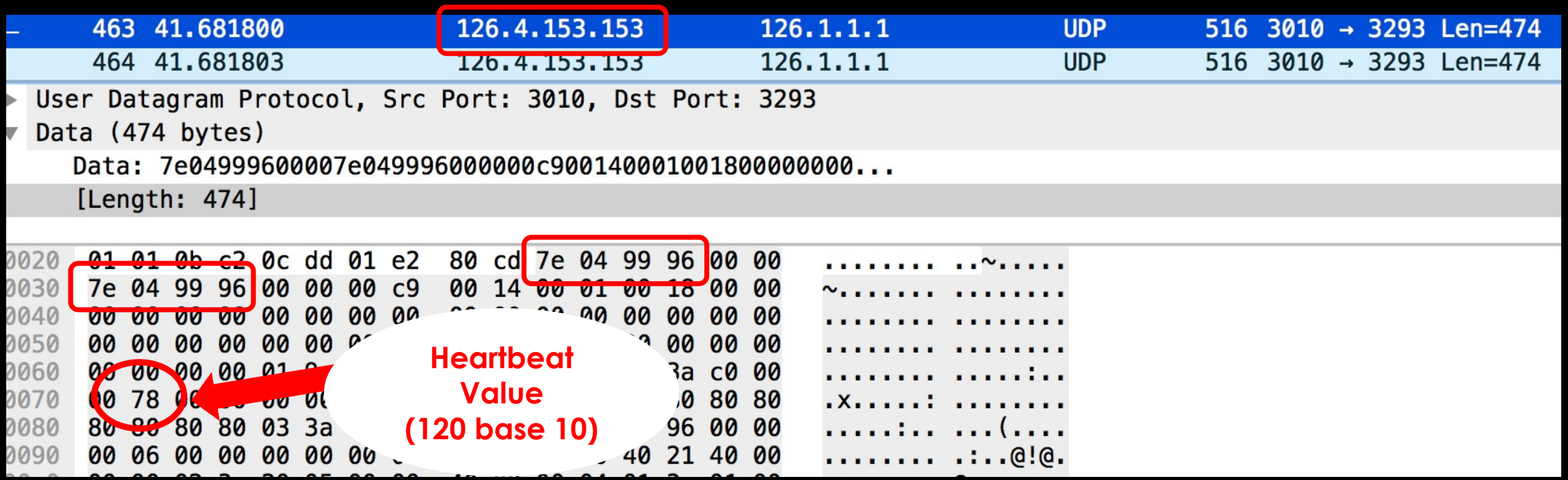
INJECTION


433 39.502117 126.4.153.150 126.1.1.1 UDP 516 3008 → 3293 Len=474
434 39.502119 126.4.153.150 126.1.1.1 UDP 516 3008 → 3293 Len=474
Frame 433: 516 bytes on wire (4128 bits), 516 bytes captured (4128 bits)
Ethernet II, Src: Marquett_04:99:96 (00:00:a1:04:99:96), Dst: Advantec_97:1f:67 (00:d0:c9:97:1f:67)
Internet Protocol Version 4, Src: 126.4.153.150, Dst: 126.1.1.1
User Datagram Protocol, Src Port: 3008, Dst Port: 3293
Data (474 bytes)
0020 01 01 0b c0 0c dd 01 99 96 00 00
0030 7e 04 99 96 00 00 c0 00 00
0040 00 00 00 00 00 00 00 00 00
0050 00 00 00 00 00 00 00 00 00
0060 3a c0 00
0070 .P.....: ....
0080 80 80 80 80 03 3a 90 80 00 00 00 28 00 96 00 00 .....:.. ...(....
0090 00 06 00 00 00 00 00 00 15 3a 00 00 40 21 40 00 ....:..@!@.
00a0 00 00 02 3a 20 05 00 00 40 09 00 04 01 3a 01 00 ...: ... @....:..
Heartbeat Value (80 base 10)

INJECTION


Kali Box
MAC
433 39.502117 126.4.153.150 126.1.1.1 UDP 516 3008 → 3294 Len=474
518 44.230408 126.4.153.153 126.1.1.1 UDP 681 3008 → 3294 Len=639
519 44.250825 126.4.153.150 126.1.1.1 UDP 681 3006 → 3294 Len=639
520 44.250827 126.4.153.150 126.1.1.1 UDP 681 3006 → 3294 Len=639
521 44.402360 Vmware_a1:6e:bf Marquett_04:99:96 ARP 60 126.1.1.1 is at 00:0c:29:a1:6e:bf
522 44.402363 Vmware_a1:6e:bf Marquett_04:99:96 ARP 60 126.1.1.1 is at 00:0c:29:a1:6e:bf
523 44.484103 126.4.153.153 126.1.1.1 UDP 681 3008 → 3294 Len=639
Frame 522: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
Ethernet II, Src: Vmware_a1:6e:bf (00:0c:29:a1:6e:bf), Dst: Marquett_04:99:96 (00:00:a1:04:99:96)
[Duplicate IP address detected for 126.1.1.1 (00:0c:29:a1:6e:bf) - also in use by 00:d0:c9:97:1f:67 (frame 428)]
[Frame showing earlier use of IP address: 428]
[Seconds since earlier frame seen: 5]
Address Resolution Protocol (reply)
00a0 00 00 02 3a 20 05 00 00 40 09 00 04 01 3a 01 00

INJECTION


463 41.681800 126.4.153.153 126.1.1.1 UDP 516 3010 → 3293 Len=474
464 41.681803 126.4.153.153 126.1.1.1 UDP 516 3010 → 3293 Len=474
User Datagram Protocol, Src Port: 3010, Dst Port: 3293
Data (474 bytes)
Data: 7e04999600007e049996000000c900140001001800000000...
[Length: 474]
0020 01 01 0b c2 0c dd 01 e2 80 cd 7e 04 99 96 00 00
0030 7e 04 99 96 00 00 00 c9 00 14 00 01 00 18 00 00
Heartbeat Value (120 base 10)

DEMO PACKET INJECTION

DEMO PACKET INJECTION 2

IMPACT SCENARIO

- Power of misinformation
- Unconscious patient
- Reliance on technology
- Smaller changes, larger impact

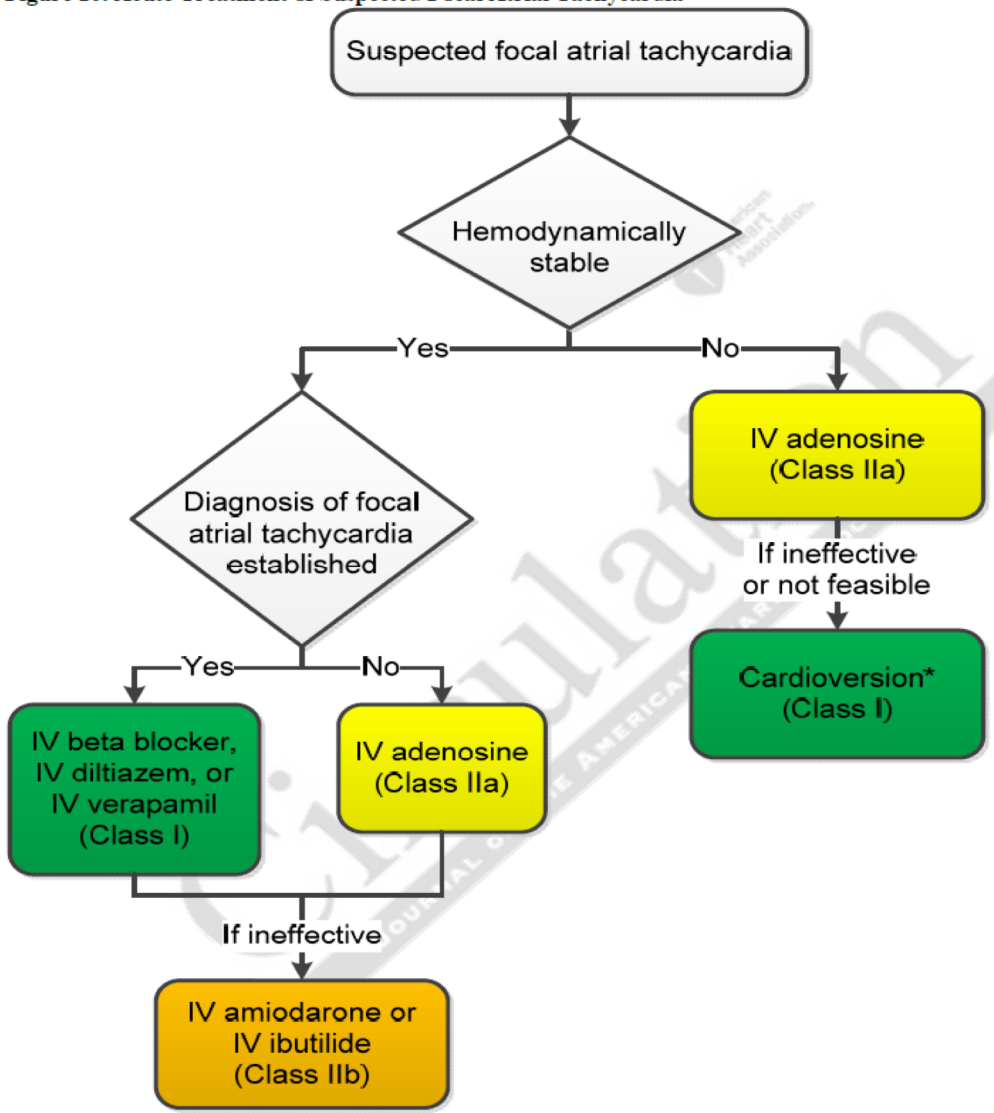
"Fictitious cardiac rhythms, even intermittent, could lead to extended hospitalization, additional testing, and side effects from medications to control heart rhythm and/or prevent clots. The hospital could also suffer resource consumption."

- Dr. S. Nordeck

AMER. HEART ASSOC. & AMER. COLLEGE OF CARDIOLOGY – ARRHYTHMIA MANAGEMENT GUIDELINES

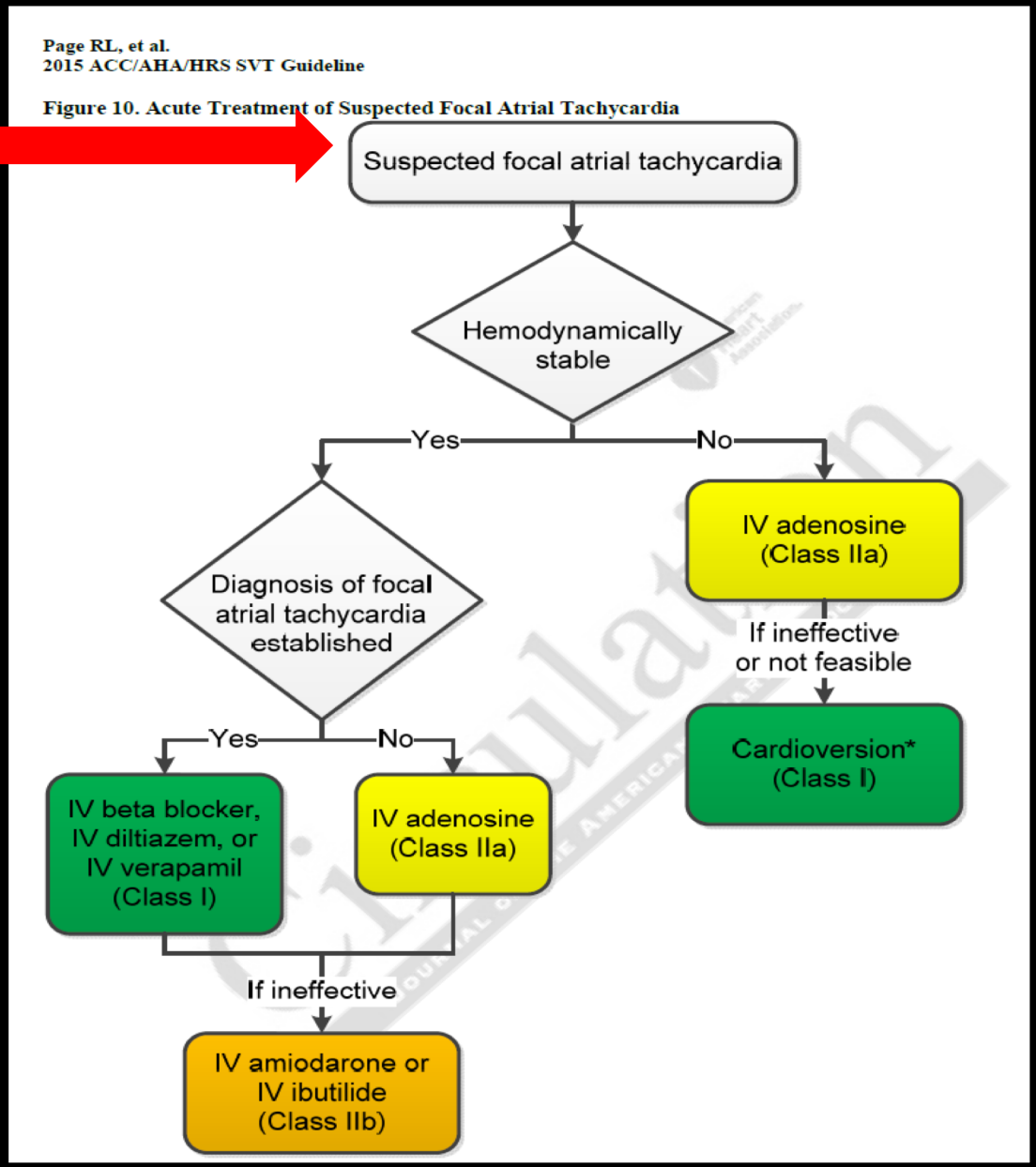
Page RL, et al.
2015 ACC/AHA/HRS SVT Guideline

Figure 10. Acute Treatment of Suspected Focal Atrial Tachycardia



AMER. HEART ASSOC. & AMER. COLLEGE OF CARDIOLOGY – ARRHYTHMIA MANAGEMENT GUIDELINES

Fictitious intermittent cardiac rhythms



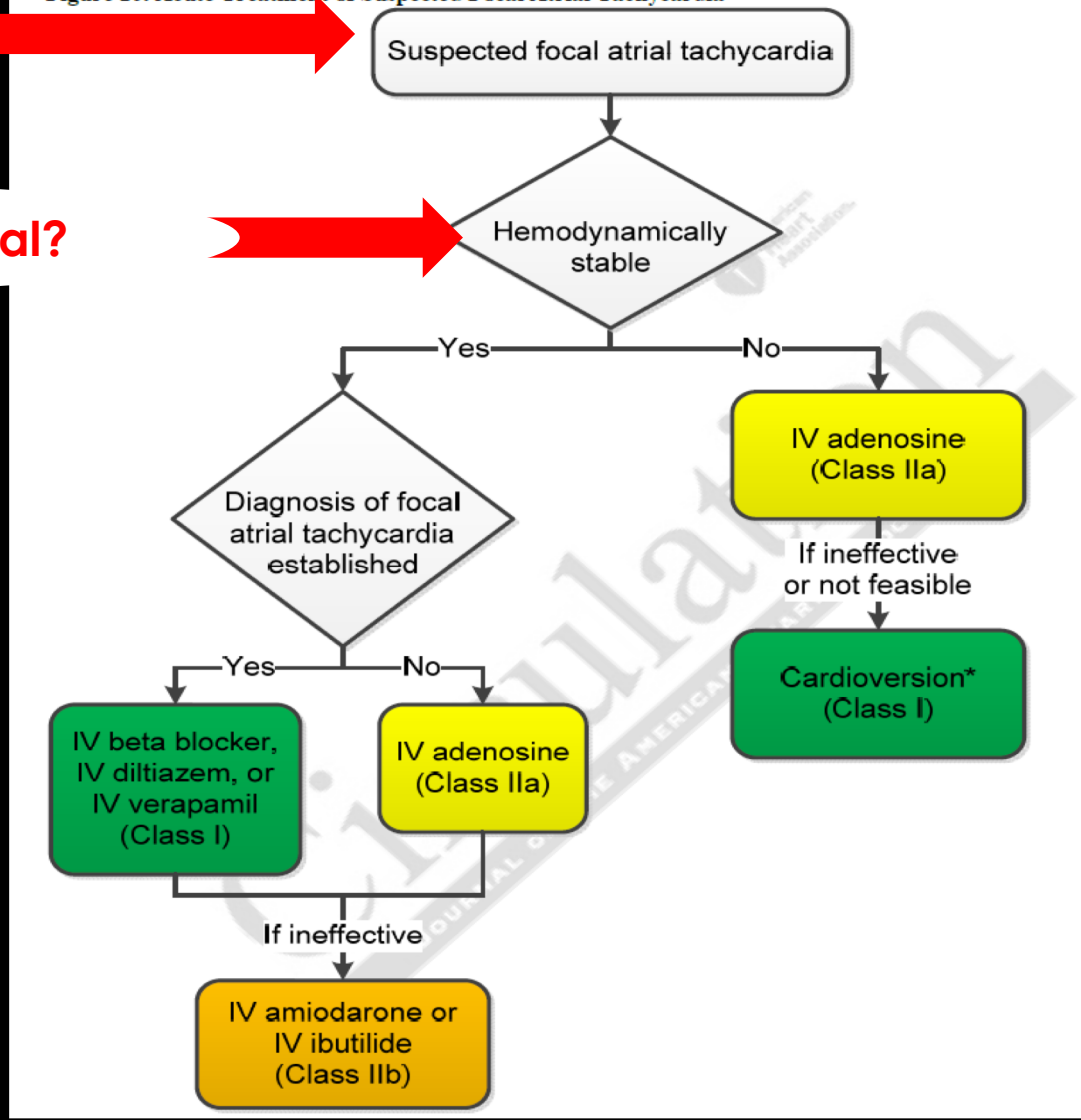
AMER. HEART ASSOC. & AMER. COLLEGE OF CARDIOLOGY – ARRHYTHMIA MANAGEMENT GUIDELINES

Fictitious intermittent cardiac rhythms

Blood Pressure Normal?

Page RL, et al.
2015 ACC/AHA/HRS SVT Guideline

Figure 10. Acute Treatment of Suspected Focal Atrial Tachycardia



AMER. HEART ASSOC. & AMER. COLLEGE OF CARDIOLOGY – ARRHYTHMIA MANAGEMENT GUIDELINES

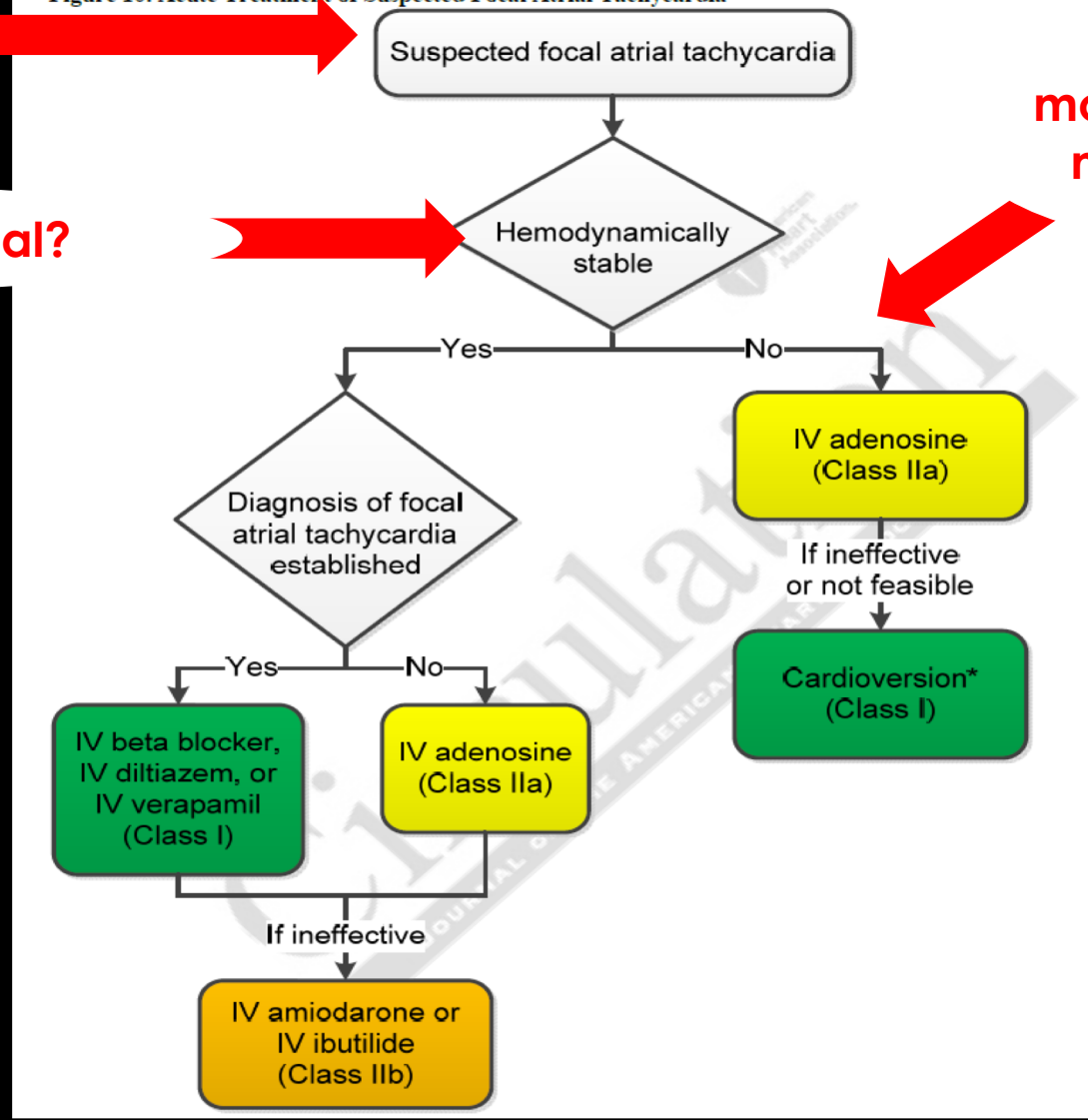
Fictitious intermittent cardiac rhythms

Blood Pressure Normal?

If PM monitor was modified

Page RL, et al.
2015 ACC/AHA/HRS SVT Guideline

Figure 10. Acute Treatment of Suspected Focal Atrial Tachycardia

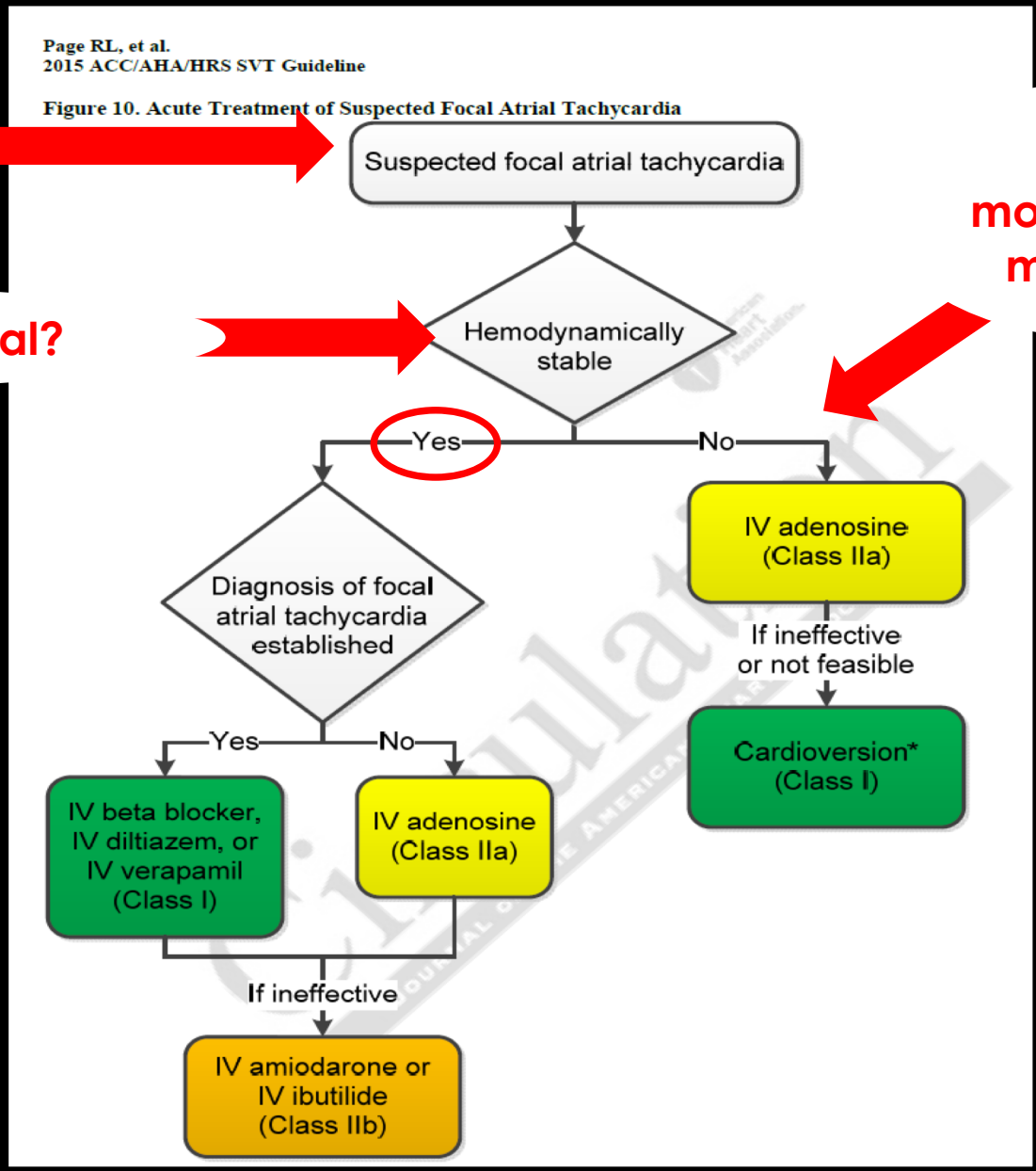


AMER. HEART ASSOC. & AMER. COLLEGE OF CARDIOLOGY – ARRHYTHMIA MANAGEMENT GUIDELINES

Fictitious intermittent cardiac rhythms

Blood Pressure Normal?

If PM monitor was modified



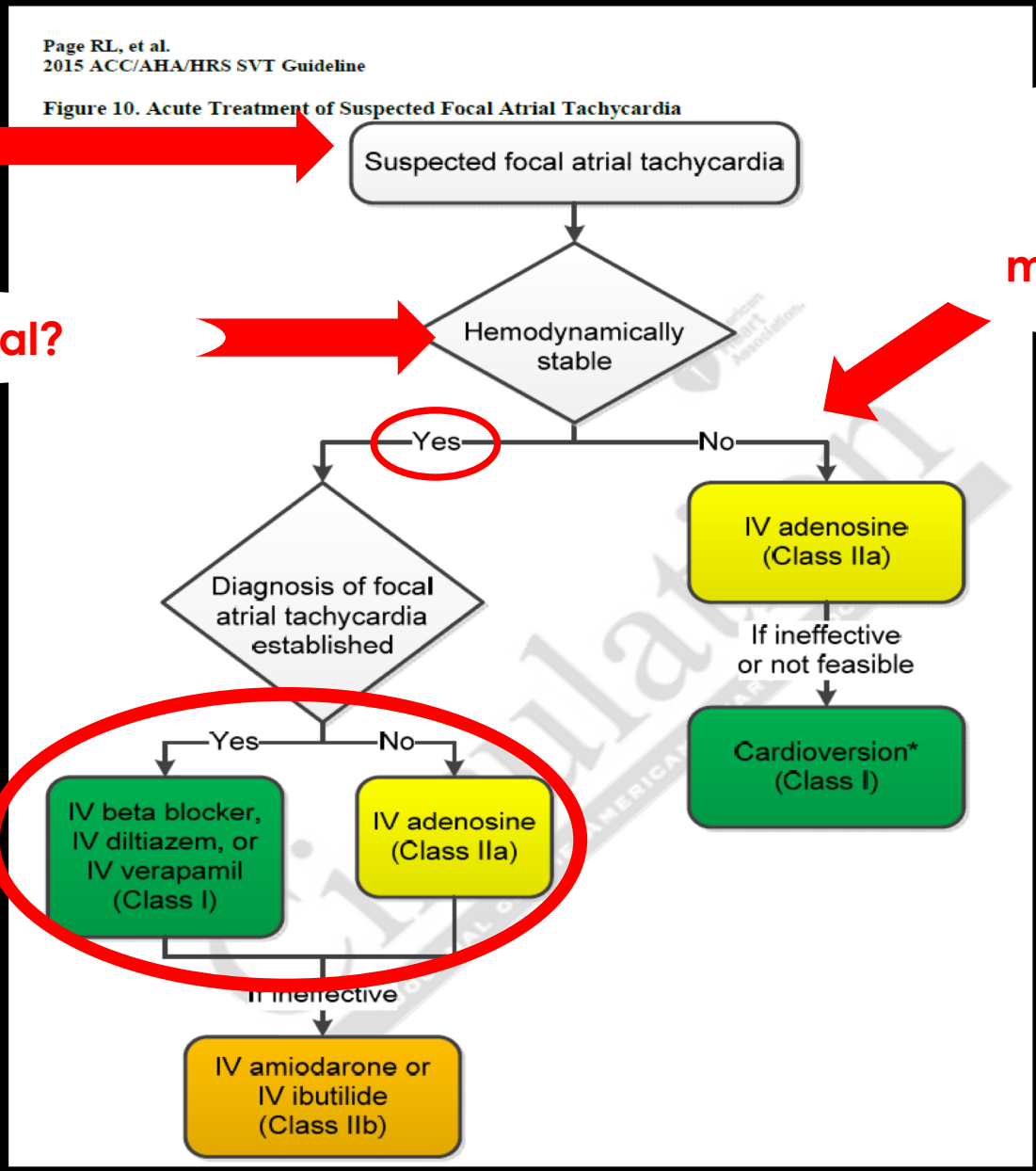
AMER. HEART ASSOC. & AMER. COLLEGE OF CARDIOLOGY – ARRHYTHMIA MANAGEMENT GUIDELINES

Fictitious intermittent cardiac rhythms

Blood Pressure Normal?

Medications Administered

If PM was modified



MITIGATIONS

- Encryption
- Authentication/Authorization
- Isolation
  - Network access controls
  - Physical access control

OTHER CONSIDERATIONS

- Attacker would require full knowledge of all aspects of the protocol and must create a full consistent set of parameters to avoid detection
- Local monitor continues to show the true state of the patient and cannot be altered by this attacker
  - Monitors do not require a network to function properly

SPECIAL THANKS

- Shaun Nordeck, MD
- Charles McFarland
- McAfee Production team
- ATR

QUESTIONS?