Network Protocol Reversing (Techniques)

Offensive Network Security
Florida State University
Spring 2014

Outline

- Examining bytes on a wire
- Use protocol
- Locate Patterns
- Cluster bytes together
- Bioinformatics

Byte Staring Time

- Look at individual bytes
- Does it contain any guessable data
  - ASCII
  - Known protocol headers, etc.
- Does it contain any patterns?
- Do we have more than one capture of data to compare?
  - Can we determine states of the protocol?
  - Possibly locate the fields depending on the data comparison

Byte Staring Time

```
0000   70 CA 9B DE DD 00 70 56  81 C3 27 13 08 00 45 00
0010   00 98 05 EF 40 00 40 06  87 AC C0 A8 48 69 C7 2C
0020   DC 86 E3 E4 01 BB EA D1  93 C2 61 BC F8 17 80 18
0030   20 10 9B 13 00 00 01 01  08 0A 94 6B 3F 6B 00 33
0040   F5 4F 16 03 01 00 5F 01  00 00 5B 03 01 53 45 76
0050   07 C4 EE A5 A5 37 61 9B  8D 01 DA 3C DF 47 18 CD
0060   FD FC DD 9D C6 EF CC 55  FC 9C 67 2E 3A 00 00 2E
0070   00 39 00 38 00 35 00 16  00 13 00 0A 00 33 00 32
0080   00 2F 00 9A 00 99 00 96  00 05 00 04 00 15 00 12
0090   00 09 00 14 00 11 00 08  00 06 00 03 00 FF 01 00
00a0   00 04 00 23 00 00
```

Byte Staring Time

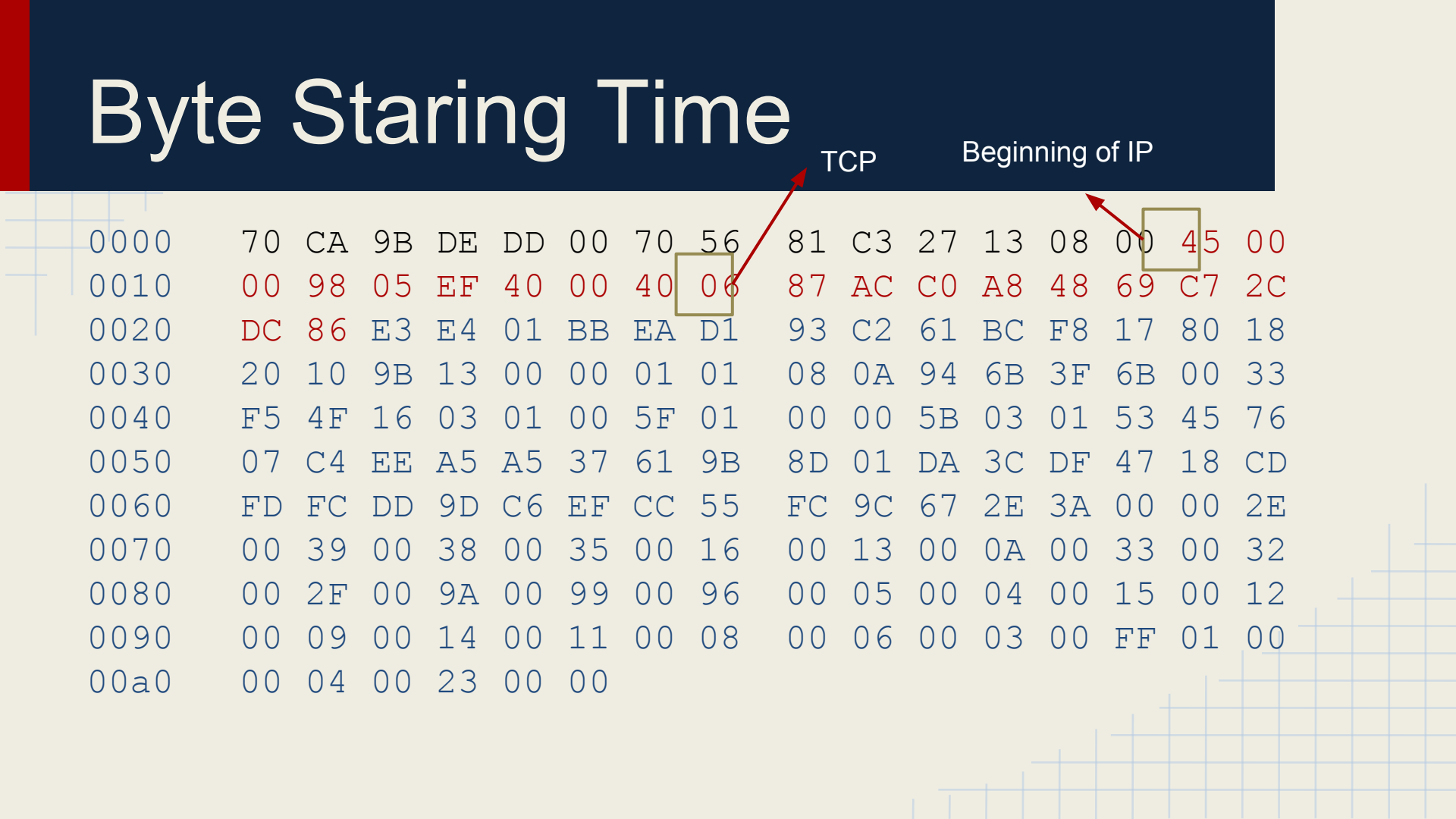
- Can you locate any patterns, familiar headers/fields?
- If given a set of bytes assume Ethernet frame is the most outer layer
- If Ethernet frame is not there assume IP packet
- If IP packet is not there assume TCP/UDP packet
- If none of the above our located then we might have an Application Layer protocol
- Can we replay the protocol?
- Else: it could belong to another data link layer (fun)
- Note: Most of the time it will be very obvious whether the bytes belong to Ethernet header, IP header, or TCP/UDP header
- Note: How hard is it to reverse a protocol / packet structure with only 1 packet and no further network communication?

Byte Staring Time

dst MAC | src MAC | IP

```
0000   70 CA 9B DE DD 00 70 56   81 C3 27 13 08 00 45 00
0010   00 98 05 EF 40 00 40 06   87 AC C0 A8 48 69 C7 2C
0020   DC 86 E3 E4 01 BB EA D1   93 C2 61 BC F8 17 80 18
0030   20 10 9B 13 00 00 01 01   08 0A 94 6B 3F 6B 00 33
0040   F5 4F 16 03 01 00 5F 01   00 00 5B 03 01 53 45 76
0050   07 C4 EE A5 A5 37 61 9B   8D 01 DA 3C DF 47 18 CD
0060   FD FC DD 9D C6 EF CC 55   FC 9C 67 2E 3A 00 00 2E
0070   00 39 00 38 00 35 00 16   00 13 00 0A 00 33 00 32
0080   00 2F 00 9A 00 99 00 96   00 05 00 04 00 15 00 12
0090   00 09 00 14 00 11 00 08   00 06 00 03 00 FF 01 00
00a0   00 04 00 23 00 00
```

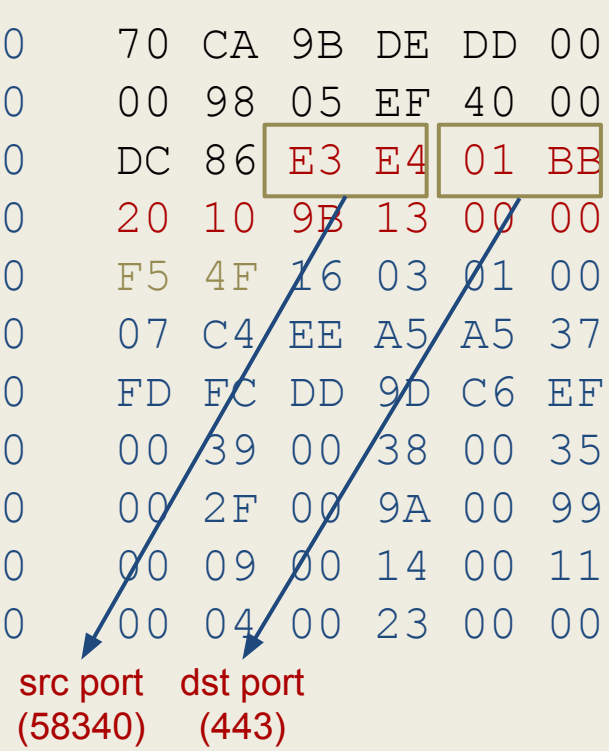
Byte Staring Time




```
0000    70 CA 9B DE DD 00 70 56   81 C3 27 13 08 00 45 00
0010    00 98 05 EF 40 00 40 06   87 AC C0 A8 48 69 C7 2C
0020    DC 86 E3 E4 01 BB EA D1   93 C2 61 BC F8 17 80 18
0030    20 10 9B 13 00 00 01 01   08 0A 94 6B 3F 6B 00 33
0040    F5 4F 16 03 01 00 5F 01   00 00 5B 03 01 53 45 76
0050    07 C4 EE A5 A5 37 61 9B   8D 01 DA 3C DF 47 18 CD
0060    FD FC DD 9D C6 EF CC 55   FC 9C 67 2E 3A 00 00 2E
0070    00 39 00 38 00 35 00 16   00 13 00 0A 00 33 00 32
0080    00 2F 00 9A 00 99 00 96   00 05 00 04 00 15 00 12
0090    00 09 00 14 00 11 00 08   00 06 00 03 00 FF 01 00
00a0    00 04 00 23 00 00
```

Byte Staring Time

```
0000   70 CA 9B DE DD 00 70 56   81 C3 27 13 08 00 45 00
0010   00 98 05 EF 40 00 40 06   87 AC C0 A8 48 69 C7 2C
0020   DC 86 E3 E4 01 BB EA D1   93 C2 61 BC F8 17 80 18
0030   20 10 9B 13 00 00 01 01   08 0A 94 6B 3F 6B 00 33
0040   F5 4F 16 03 01 00 5F 01   00 00 5B 03 01 53 45 76
0050   07 C4 EE A5 A5 37 61 9B   8D 01 DA 3C DF 47 18 CD
0060   FD FC DD 9D C6 EF CC 55   FC 9C 67 2E 3A 00 00 2E
0070   00 39 00 38 00 35 00 16   00 13 00 0A 00 33 00 32
0080   00 2F 00 9A 00 99 00 96   00 05 00 04 00 15 00 12
0090   00 09 00 14 00 11 00 08   00 06 00 03 00 FF 01 00
00a0   00 04 00 23 00 00
```



Byte Staring Time

```
0000   70 CA 9B DE DD 00 70 56  81 C3 27 13 08 00 45 00
0010   00 98 05 EF 40 00 40 06  87 AC C0 A8 48 69 C7 2C
0020   DC 86 E3 E4 01 BB EA D1  93 C2 61 BC F8 17 80 18
0030   20 10 9B 13 00 00 01 01  08 0A 94 6B 3F 6B 00 33
0040   F5 4F 16 03 01 00 5F 01  00 00 5B 03 01 53 45 76
0050   07 C4 EE A5 A5 37 61 9B  8D 01 DA 3C DF 47 18 CD
0060   FD FC DD 9D C6 EF CC 55  FC 9C 67 2E 3A 00 00 2E
0070   00 39 00 38 00 35 00 16  00 13 00 0A 00 33 00 32
0080   00 2F 00 9A 00 99 00 96  00 05 00 04 00 15 00 12
0090   00 09 00 14 00 11 00 08  00 06 00 03 00 FF 01 00
00a0   00 04 00 23 00 00
```

Byte Staring Time

- It appears we have an application layer protocol
- Does it look familiar?
- Let’s take a look at the response
- There does not appear to be any ASCII data
- Let’s take a look at the response

Byte Staring Time

```
. . . .
0040   3F 6B 60 7C BD 3D 94 72  BA 93 A0 75 D2 63 C6 B3   ?k`|.=.r...u.c..
0050   BE 1A 84 40 E3 9C BC 83  A5 59 1A DB EE 66 F4 B1   ...@.....Y...f..
0060   E5 CE C2 90 0F CA 9E 32  1E DA C8 99 BD B8 E8 58   .......2.......X
0070   D4 22 F2 47 8D F2 F3 83  93 AF 1A B4 1E C4 25 F3   ."".G..........%.
0080   4B A7 51 55 EB 7D E3 C4  91 9D 6A 17 43 70 CC AC   K.QU.}....j.Cp..
0090   3C C6 6A AF 5D 04 5D 63  3B E5 7E 62 C7 98 BA 19   <.j.].]c;.~b....
00a0   00 04 E6 30 82 04 E2 30  82 03 CA A0 03 02 01 02   ...0...0........
00b0   02 10 6E BA F0 8F 79 83  FA 9D E1 B2 6F 96 FC 6E   ..n...y.....o..n
00c0   98 BF 30 0D 06 09 2A 86  48 86 F7 0D 01 01 05 05   ..0...*.H.......
00d0   00 30 6F 31 0B 30 09 06  03 55 04 06 13 02 53 45   .0o1.0...U....SE
00e0   31 14 30 12 06 03 55 04  0A 13 0B 41 64 64 54 72   1.0...U....AddTr
00f0   75 73 74 20 41 42 31 26  30 24 06 03 55 04 0B 13   ust AB1&0$..U...
0100   1D 41 64 64 54 72 75 73  74 20 45 78 74 65 72 6E   .AddTrust Extern
0110   61 6C 20 54 54 50 20 4E  65 74 77 6F 72 6B 31 22   al TTP Network1"
0120   30 20 06 03 55 04 03 13  19 41 64 64 54 72 75 73   0 ..U....AddTrus
0130   74 20 45 78 74 65 72 6E  61 6C 20 43 41 20 52 6F   t External CA Ro
0140   6F 74 30 1E 17 0D 31 31  30 38 32 33 30 30 30 30   ot0...1108230000
0150   30 30 5A 17 0D 32 30 30  35 33 30 31 30 34 38 33   00Z..20053010483
0160   38 5A 30 70 31 0B 30 09  06 03 55 04 06 13 02 47   8Z0p1.0...U....G
0170   42 31 1B 30 19 06 03 55  04 08 13 12 47 72 65 61   B1.0...U....Grea
0180   74 65 72 20 4D 61 6E 63  68 65 73 74 65 72 31 10   ter Manchester1.
. . . . .
```

Byte Staring Time

- Response has ASCII characters that appear to be certficiates
- Using known information (dstport = 443, looking at TLS RFC)
- Protocol is TLS Client Handshake Request

Byte Staring Time

- Well that turned out to be a known protocol, what good is that?
- Get used to seeing patterns
- Request packet had TCP options:
  - how can you determine TCP options length?
  - Look at IP length field + TCP data offset field
- There has to be techniques better than staring at bytes and guessing
  - Certiainly!
- Reversing protocols can be a hastle
  - Data overload
  - Fields, where are the fields, strawberry fields?
  - How much time do you really have?
  - Help!?

Bioinformatics

- 2004, Marshall Beddoe brought us Bioinformatics for protocols
- Bioinformatics used to find patterns within strings
- Process large amounts of complex data
- Looks for patterns without manually staring at bytes
  - Programmers who Stare at Bytes
- Goal for protocols: locate specific protocol packet fields

Bioinformatics Algorithms

- Needleman-Wuncsh: Align protein sequences
  - Global Alignment
  - Dynamic programming
  - Similarity matrix + linear gap penalty
  - Not necessary to use gap penalties
- Smith-Waterman:
  - Local Alignment
  - Compares highly diverged sequences
  - Looks for most similar sequences, then pass to Global Alignment
  - Requires a gap penalty
  - Allows insertions and deletions

Bioinformatics Algorithms

- Phylogeny: the evolution of a genetically related group of organisms as distinguished from the development of the individual organism
  - Usually represented by a Binary tree
  - Shows mutations over time

- UPGMA: Unweighted Pair Group Method With Arithmetic Mean
  - Allows for Multiple Sequence Alignment
  - Catches evolutionary data
  - Create phylogenetic tree

Needleman-Wunsch

- Place first sequence in top row
- Place second sequence in left-most column
- For each cell,
  - Assign similarity values
  - Assess pathways (left, up, diagonal) in matrix and assign the maximum scoring pathway values using
  - $M_{i,j} = MAX(M_{i-1,j-1} + S_{i,j}, M_{i,j-1} + w, M_{i-1,j} + w)$
  - w = gap penalty (stays at 0), S is similarity weight
  - Construct path between highest scoring cell and beginning of matrix to obtain maximum global alignment
- Gap penalty reduces the number of gaps in final alignment
- Let's look at an example using SMTP

		E	H	L	O		m	.	w	e	b	.	c	o	m
	0														
E		1													
H			1												
L				1											
O					1										
						1									
m							1								1
x															
.								1				1			
a															
.								1				1			
t															
v															

Similar characters receive a score of 1

		E	H	L	O		m	.	w	e	b	.	c	o	m
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
H	0	1	2	2	2	2	2	2	2	2	2	2	2	2	2
L	0	1	2	3	3	3	3	3	3	3	3	3	3	3	3
O	0	1	2	3	4	4	4	4	4	4	4	4	4	4	4
	0	1	2	3	4	5	5	5	5	5	5	5	5	5	5
m	0	1	2	3	4	5	6	6	6	6	6	6	6	6	6
x	0	1	2	3	4	5	6	6	6	6	6	6	6	6	6
.	0	1	2	3	4	5	6	7	7	7	7	7	7	7	7
a	0	1	2	3	4	5	6	7	7	7	7	7	7	7	7
.	0	1	2	3	4	5	6	7	7	7	7	8	8	8	8
t	0	1	2	3	4	5	6	7	7	7	7	8	8	8	8
v	0	1	2	3	4	5	6	7	7	7	7	8	8	8	8

Start at 1,1 and execute the MAX() function

$$M_{i,j} = MAX(M_{i-1,j-1} + S_{i,j}, M_{i,j-1} + w, M_{i-1,j} + w)$$

		E	H	L	O		m	.	w	e	b	.	c	o	m
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
H	0	1	2	2	2	2	2	2	2	2	2	2	2	2	2
L	0	1	2	3	3	3	3	3	3	3	3	3	3	3	3
O	0	1	2	3	4	4	4	4	4	4	4	4	4	4	4
	0	1	2	3	4	5	5	5	5	5	5	5	5	5	5
m	0	1	2	3	4	5	6	6	6	6	6	6	6	6	6
x	0	1	2	3	4	5	6	6	6	6	6	6	6	6	6
.	0	1	2	3	4	5	6	7	7	7	7	7	7	7	7
a	0	1	2	3	4	5	6	7	7	7	7	7	7	7	7
.	0	1	2	3	4	5	6	7	7	7	7	8	8	8	8
t	0	1	2	3	4	5	6	7	7	7	7	8	8	8	8
v	0	1	2	3	4	5	6	7	7	7	7	8	8	8	8

Start at 1,1 and execute the MAX() function

$M_{i,j}$ =
MAX($M_{i-1,j-1} + S_{i,j}$,
$M_{i,j-1} + w$,
$M_{i-1,j} + w$)

Needleman-Wunsch

- Path determine now apply Needleman-Wuncsh Rules:
  - Travels left or up insert space (_) into a sequence
  - Up = space in Top Row Seqeunce
  - Left = space in Left-Most Row Sequence
- Notice how the fields are *somewhat* aligned
- `EHLO m` is considered a keyword, `x` is variable length, etc.
- Goal: See how data mutates (binary to binary, ASCII to ASCII)
- Not exact

```
EHLO m_.web.com
 ||||||  ||   | ||
EHLO mx.a__._tv
```

Similarity Matrix

- Special matrices with better similarity values for data (*S*)
  - PAM (Percent Accepted Mutation)
  - BLOSUM (Blocks Substitution Matrix)
- The problem with the previous example is every byte is weighted the same
- Alter similarity values / Matrix
- Set different weights (probabilities) for bytes
  - ASCII chracter set: 0.4
  - ASCII Printable: 0.4
  - Binary, 0.2
- Allows convergence and reduces number of incorrect gaps

Multiple Sequences

- Needleman-Wunsch compares two sequences at a time
- Can we compare multiple sequences at one time?
- Needleman-Wunsch will not be able to do it ($2^n \times L^n$)
  - How long will it take to calculate 500, 200 byte sequences?
- Use heuristic methods to align multiple sequences

UPGMA

- Protocols fields change which can be related to evolutionary change
- Algorithm:
  - Place each sequence into individual cluster, place each cluster in universal set
  - Calculate the distance between each cluster (UPGMA)
  - Look for two two clusters where $d_{i,j}$ is minimal
  - Form a new cluster, C_k, with C_i and C_j
  - Create tree node *k* with children *i* and *j*, remove C_i, C_j from tree

$$d_{ij} = \frac{1}{|C_i||C_j|} \sum_{p \in C_i, q \in C_j} D_{pq}$$

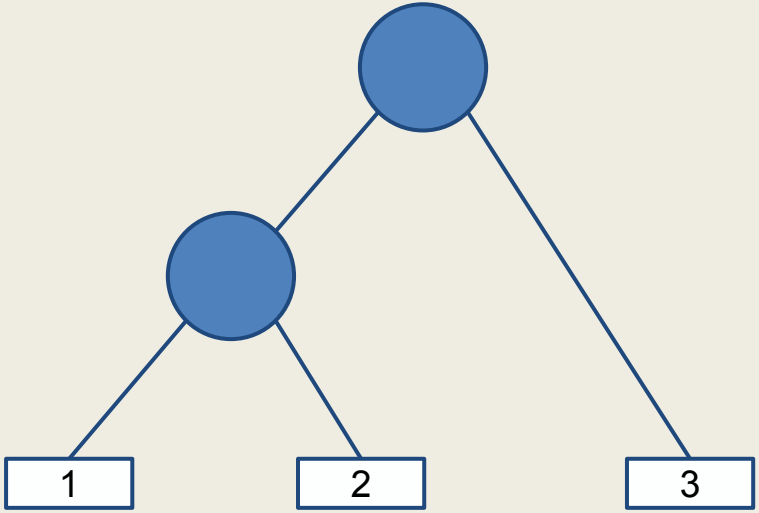
Eq.(2): Where dij is the distance between clusters Ci and Cj and Dpq is the Smith Waterman score.

(Equation from Beddoe's paper)

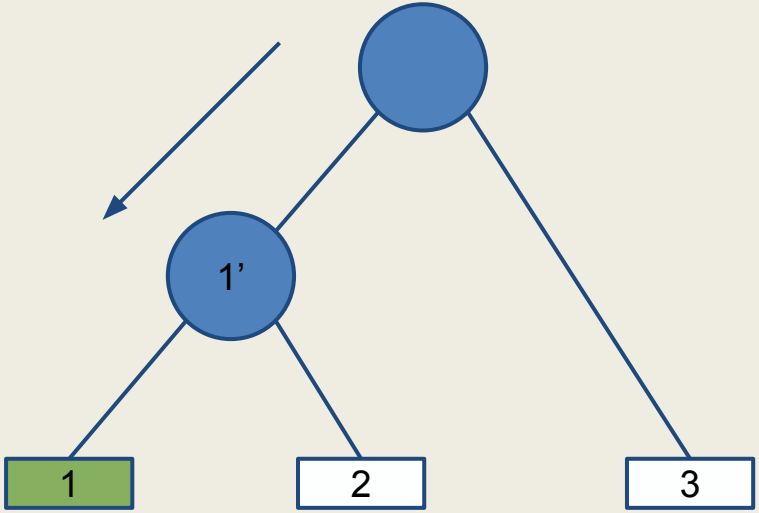
UPGMA

- Interpreting tree
  - if root == NULL: traverse left, then right
  - if left != NULL and right != NULL, align sequences
  - Choose the sequence with least gaps
    - Seq1: `EHLO m_.web.com`
    - Seq2: `EHLO mx.a__._tv`
    - Seq1 will be taken
    - Add new sequence to root
    - Keep track of changes in edge
- Help categorize/separate different message types
- Uses n comparisons on an n-height tree

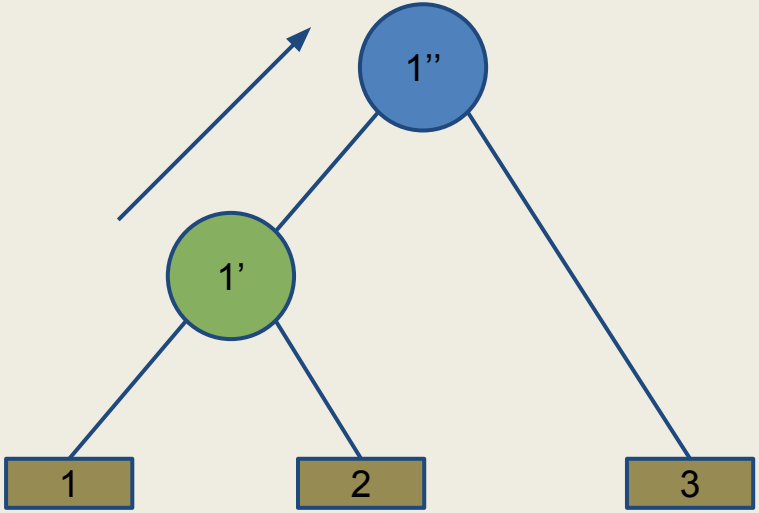
UPGMA



UPGMA



UPGMA



UPGMA

- Seq1: `EHLO m_.web.com`
- Seq1: `EHLO __.a__._tv`
- Seq1: `EHLO m_.a_b.edu`
- Results: `EHLO ??????????`

What’s Next

- Not perfect, step to *help* determining unknown protocol
- Provides better use-cases for fuzzers (we can see the fields)

References

- http://www.4tphi.net/~awalters/PI/PI_Toorcon.pdf
- http://www.southampton.ac.uk/~re1u06/teaching/upgma/