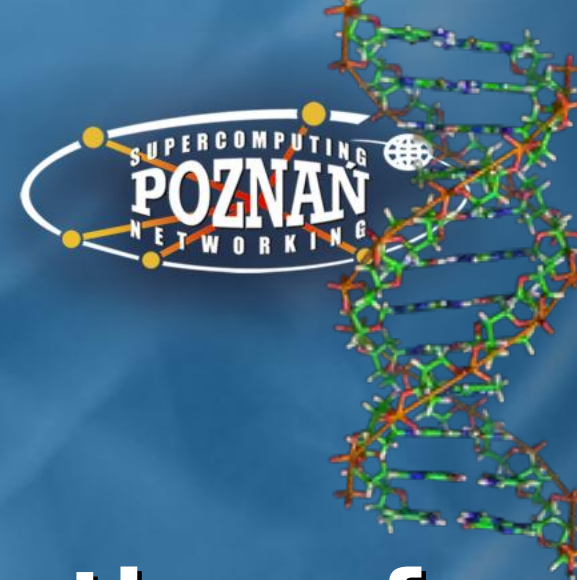


POZNAŃ
SUPERCOMPUTING AND
NETWORKING
CENTER



**Discovery and visualization of
network protocols with tools for
biological sequence alignment**

Mateusz Drygas, Tomasz Nowak

PSNC Security Team



Me and my team

- Programmer / software security
- Team created within Poznan Metropolitan Area Network
- Independent security research
 - Browsers security testing (2010)
- <http://security.psnc.pl>
- Part of Institute of Bioorganic Chemistry

Agenda



- Motivation
- Inspiration
- Exploitation
 - Packets as proteins
 - Clustering for “packet families”
 - Finding patterns
 - Fuzzing

Motivation (1)

- Integration with a commercial backup/recovery product
- Client software is expensive
 - vendor provides dynamic client library for free
- Proprietary protocol for Library \Leftrightarrow Server communication
- We test libraries used in our projects
- Fuzzing seems to be a good way

Motivation (2)

- Enabled core dumps
- Prepared a script to loop
 - zzuf for blind fuzzing (good parametrization)
 - proxyfuzz for redirection (good logging)
 - netcat & tee & bash for plumbing
- Too many rejected transactions (auth failures)
 - 0 knowledge fuzzing vs. challenge-response, nonces...
- How to avoid reverse engineering of the implementation?



Motivation (3)

- Just compare the packets and find patterns!

```
Client -----> server
'\x00\x04\xd\xa5'
Server -----> Client
'\x00\x1e\xa5f\x15\x07\xda\t\x17\x10\x1a\x01\x00\x00\x00\x07\x00\x07\x00\n\x00\x05\x00'
Client -----> server
'\x00r\x1a\xa5f\x00\x00\x07\x04\x02\x00\x07\x00\x0c\x00\x13\x00\x08\x00\x00\x1b\x00'
Server -----> Client
'\x00\x06\x13\xa5\x01\x00'
Client -----> server
'\x000:\xa5\x00\x00\x10\x00\x10\x00\x01\x02\x02\x01\x01\x00Linux86\x00\x00\x00'
Server -----> Client
'\x009a\x1f\xa5\x00\x00\x00}\x00\x00\x00<\x00\x00\x00\x01\x01\x02\x00\x00\x02\x00'
Client -----> server
''\x000:\xa5\x00\x00\x10\x00\x10\x00\x01\x02\x02\x01\x01\x00Linux86\x00\x00\x00'
Server -----> Client
'\x000:\xa5\x00\x00\x10\x00\x10\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
Client -----> server
'\x00(\xa5\x00\x00\x08\x00\x08\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
Server -----> Client
'\x004;\xa5\x07\xda\t\x17\x10\x1a\x01\x00\x00\x00\x00\x00\x00\x01\x02\x00\x00\x00'
Client -----> server
'\x00\x04\x18\xa5'
Server -----> Client
'\x00\x04\x18\xa5'
Client -----> server
''\x00\x14\xc4\xa5\x00\x00\x00\x00\x00\x00\x01\xc9\x00\x00\x00\x00\x00\x00\x00\x01'
Server -----> Client
'\x00\x06\x13\xa5\x02\x0b'
Client -----> server
''\x00\x14\xc4\xa5\x00\x00\x00\x00\x00\x00\x01\xc9\x00\x00\x00\x00\x00\x00\x00\x01'
Server -----> Client
'\x00\x06\x13\xa5\x02\x0b'
```

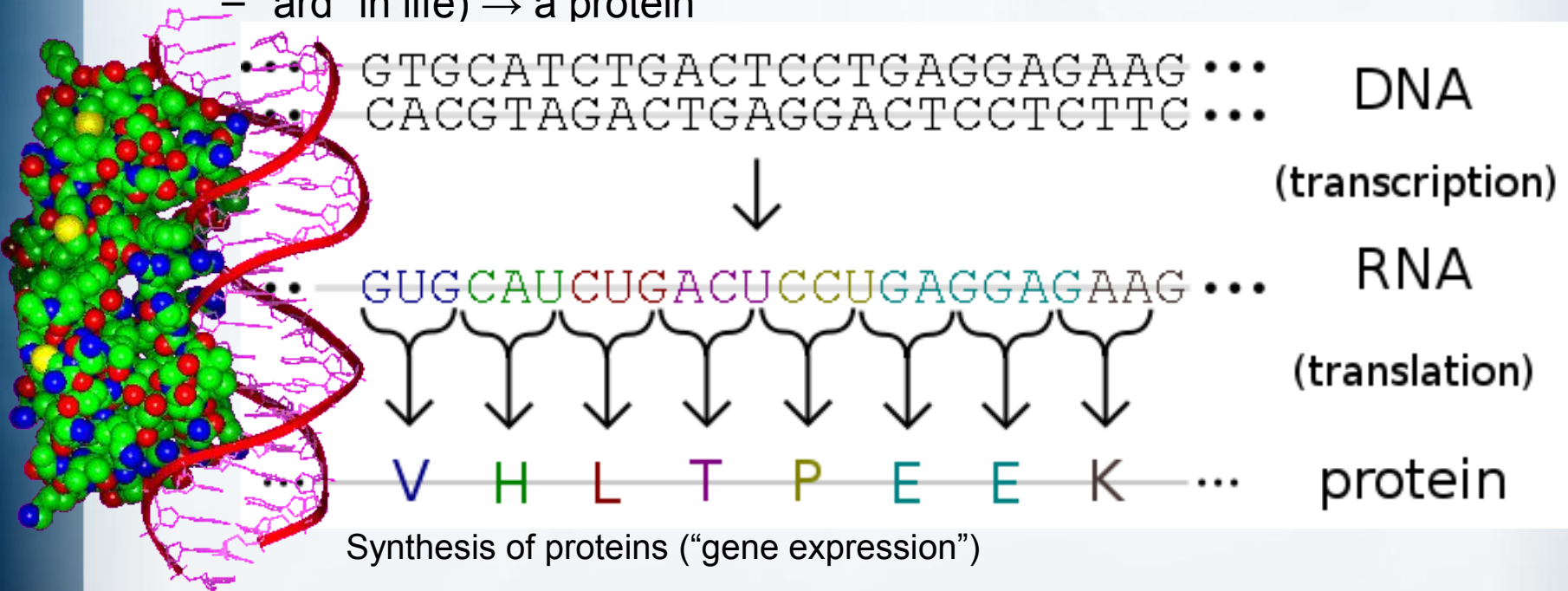
```
Client -----> server
'\x00\x04\xd\xa5'
Server -----> Client
'\x00>\x1e\xa5f\x15\x07\xda\t\x17\x10\x1a\x02\x00\x00\x00\x07\x00\x07\x00\n\x00\x05\x00'
Client -----> server
'\x00r\x1a\xa5f\x00\x00\x07\x04\x02\x00\x07\x00\x0c\x00\x13\x00\x08\x00\x00\x1b\x00'
Server -----> Client
'\x009a\x1f\xa5\x00\x00\x00}\x00\x00\x00<\x00\x00\x00\x02\x01\x02\x00\x00\x02\x00'
Client -----> server
'\x000:\xa5\x00\x00\x10\x00\x10\x00\x01\x02\x02\x01\x01\x00Linux86\x00\x00\x00'
Server -----> Client
'\x000:\xa5\x00\x00\x10\x00\x10\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
Client -----> server
'\x00(\xa5\x00\x00\x08\x00\x08\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
Server -----> Client
'\x004;\xa5\x07\xda\t\x17\x10\x1a\x01\x00\x00\x00\x00\x00\x00\x00\x01\x02\x00\x00\x00'
Client -----> server
'\x00\x04\x12\xa5\x00\x10\xa2\xa5\x07\xda\t\x17\x10\x1a\x01\x00\x00\x00\x00'
Server -----> Client
'\x00\x06\x13\xa5\x02\x02'
Client -----> server
'\x00\x04\x12\xa5\x00\x14\xa0\xa5\x00\x00\x00\x0cTSM4SECURITY'
Server -----> Client
'\x01x\xal\xa5\x00\x00\x01p\x00\x0c\x07\xdaSTANDARD\x00\x0c\x07\xe4STANDARD\x00\x0b\x07'
Client -----> server
'\x00\x04\x18\xa5'
Server -----> Client
'\x00\x04\x18\xa5'
Client -----> server
'\x00\x04\x12\xa5\x00\x10\xa2\xa5\x01\x00\x01p\x00\x0c\x07\xdaSTAN'
Server -----> Client
'\x00\x06\x13\xa5\x02\x02'
Client -----> server
'\x00\x04\x12\xa5\x00\x14\xa0\xa5\x00\x00\x00\x0cTSM4SECURITY'
Server -----> Client
'\x01x\xal\xa5\x00\x00\x01p\x00\x0c\x07\xdaSTANDARD\x00\x0c\x07\xe4STANDARD\x00\x0b\x07'
```

Motivation (4)

- Not far this way...
- How to embrace more byte sequences?
 - Available tools show differences between 2 or 3 files (diff, vimdiff, meld)
 - ... but we got hundreds of flows
- Bioinformatics deals with this problem for years!

A biological sequence

- Single, continuous molecule compound of
 - nucleotides (ACGT) → a nucleic acid (RNA, DNA etc.)
 - amino acids (22 “standards” in life) → a protein



- Stored in ASCII file formats like FASTA

FASTA format

>HSBGPG Human gene for bone gla protein (BGP)

```
GGCAGATTCCCCCTAGACCCGCCCGCACCATGGTCAGGCATGCCCTCCTCATCGCTGGGCACAGCCCAGAGGGT
ATAAACAGTGCTGGAGGCTGGCGGGGCAGGCCAGCTGAGTCCTGAGCAGCAGCCCAGCGCAGCCACCGAGACACC
ATGAGAGCCCTCACACTCCTCGCCCTATTGGCCCTGGCCGCACTTTGCATCGCTGGCCAGGCAGGTGAGTGCCCC
CACCTCCCCTCAGGCCGCATTGCAGTGGGGGCTGAGAGGAGGAAGCACCATGGCCCACCTCTTCTCACCCCTTTG
GCTGGCAGTCCCTTTGCAGTCTAACCACCTTGTTGCAGGCTCAATCCATTTGCCCCAGCTCTGCCCTTGCAGAGG
GAGAGGAGGGAAGAGCAAGCTGCCCGAGACGCAGGGGAAGGAGGATGAGGGCCCTGGGGATGAGCTGGGGTGAAC
CAGGCTCCCTTTTCCTTTGCAGGTGCGAAGCCAGCGGTGCAGAGTCCAGCAAAGGTGCAGGTATGAGGATGGACC
TGATGGGTTCTGGACCCTCCCCTCTCACCTGGTCCCTCAGTCTCATTTCCCCACTCCTGCCACCTCCTGTCTG
GCCATCAGGAAGGCCAGCCTGCTCCCCACCTGATCCTCCCAAACCCAGAGCCACCTGATGCCTGCCCTCTGCTC
CACAGCCTTTGTGTCCAAGCAGGAGGGCAGCGAGGTAGTGAAGAGACCCAGGCGCTACCTGTATCAATGGCTGGG
GTGAGAGAAAAGGCAGAGCTGGGCCAAGGCCCTGCCTCTCCGGGATGGTCTGTGGGGGAGCTGCAGCAGGGAGTG
GCCTCTCTGGGTTGTGGTGGGGGTACAGGCAGCCTGCCCTGGTGGGCACCCTGGAGCCCCATGTGTAGGGAGAGG
AGGGATGGGCATTTTGCACGGGGGCTGATGCCACCACGTCGGGTGTCTCAGAGCCCCAGTCCCCTACCCGGATCC
CCTGGAGCCCAGGAGGGAGGTGTGTGAGCTCAATCCGACTGTGACGAGTTGGCTGACCACATCGGCTTTCAGGA
GGCCTATCGGCGCTTCTACGGCCCGGTCTAGGGTGTGCTCTGCTGGCCTGGCCGGCAACCCAGTTCTGCTCCT
CTCCAGGCACCCTTCTTTCTCTTCCCCTTGCCCTTGCCCTGACCTCCCAGCCCTATGGATGTGGGGTCCCCATC
ATCCCAGCTGCTCCCAAATAAACTCCAGAAG
```

- Sequence elements
 - Nucleic acids: A, C, G, T/U
 - Amino acids: 24 letters

Visualizing similarity: Sequence alignment

- Calculating similarity of sequences
- Interweaving sequences with the knowledge
- Aligns same characters in one column
(by introducing dashes - “gaps”)
- Example (pairwise align.): ABFDEL, ABMNDE:

ABF-DEL
ABMNDE

- Many free tools available, e.g.
 - kalign
 - clustalw
 - muscle
 - blast
- Parameterizable heuristics, distributed algorithms

SEAVIEW – GUI for seq. alignment

random.aln.fasta

File Edit Align Props Sites Species Footers Search: Goto: Trees Help

sel=3 1 Seq:1 Pos:115|77 [pkt000] 144

```

pkt000 WWWWWWWWWWWAACFFDMKBLEB-----AMAKAAABAMAAAK---AEAAAACFFD---MKBLEB-RAPMAB---WTAAAAA-----AAAAAAR--APMABBDAAAAAAAAAAAA
pkt001 WWWWWWWWWWWAACFFDMKBLEB-----AMAKAAABAMAAAK---AEAAAACFFD---MKBLEB-RAPMAB---WTAAAAA-----AAAAAAR--APMABBDAAAAAAAAAAAA
pkt002 WWWWWWWWWWWAACFFDMKBLEB-----AMAKAAABAMAAAK---AEAAAACFFD---MKBLEB-RAPMAB---WTAAAAA-----AAAAAAR--APMABBDAAAAAAAAAAAA
pkt003 -----AACFFDMKBLEBAABQLEWREDSAMAAEFAAAAFBAAAAEAAEAB---BEABSRAPMABSQNK-WTPAWSNLAL---AEPPAADSTCSSDAEPCLCRRQKWREBTDMLDK--ADBMT
pkt004 AABQLEWREDSAACFFDMKBLEB-----AMAAEFAAAAFBAAAAEAAADN---BBELBSNKWTPAWSRAPMAB---SQAEPPN-----LALAADSPLFDAL--
pkt005 -----AACFFDMKBLEBAABQLEWREDSAMAAEFAAAAADRRDEAAAEAA---KNCNTRAPMABSQFOSKTSARQBTQB---WMP--QPSLEKRDDADLLRPFABMADTPBPNKAAAAAMA
pkt006 AABQLEWREDSAACFFDMKBLEB-----AMAAEFAAAAACMKLALAEAADLAKSB---KCFOSKTSAR-RAPMAB-----SQBWMFPBQTODDADLLRQPSLEKROFABAAWPFKTDAAAA
pkt007 -----AACFFDMKBLEBAABQLEWREDSAMAAEFAAAAEDEDWEAAAEAB---BLCKBRAPMAB---SORAPMAB---WTPFNMAADFAACWRC-----PEMKETAB--AAAAABAAAAA
pkt008 AABQLEWREDSAACFFDMKBLEB-----AMAAEFAAAABDKAAAAEAAEAB---BOEMSRAPMAB---WTRAPMABSQAADFPFNMAABCCAAER-----MKETMB--MAAABAAAAA
pkt009 -----AACFFDMKBLEBAABQLEWREDSAMAAEFAAAADRKLKMLEAAAEAA---KNMFCRAPMABSQSF---QENDPP---NACAAAFAKCLPDC---PFAAAAAAAPAACBKSANNTSA
pkt010 AABQLEWREDSAACFFDMKBLEB-----AMAAEFAAAADRRAAAEAAADFAKB---NSPSFQENDPPRAPMAB---SQAAFANACAEPELMPCKCLPDCPKPABCAQFAKDFQAAAA
pkt011 -----AACFFDMKBLEBAABQLEWREDSAMAAEFAAAADDELKMMEEAAEAA---KNMFCRAPMABSQSF---QENDPP---NACAAAFAKCLPDCPKPELMPDMABAAAF---RNSBAA
pkt012 -----AACFFDMKBLEBAABQLEWREDSAMAAEFAAAASELKMLEAAAEAA---KNMFCRAPMABSQSF---QENDPP---NACAAAFAKCLPDCPKPELMPDMABAAAF---RKPWFA
pkt013 -----AACFFDMKBLEBAABQLEWREDSAMAAEFAAAABLTKMPEAAAEAA---KNLASRAPMABSQSF---QENDPP---NACAAAFAKCLPDMEKEPELMPDMABMAAF---RSQCSA
pkt014 AABQLEWREDSAACFFDMKBLEB-----AMAAEFAAAADDEBCFAEAAADFAKA---LNCFSQENDPPRAPMAB---SQAAFANACAEPELMPDKCLPDMEKMABAAAAANLPSAAAA
pkt015 AABQLEWREDSAACFFDMKBLEB-----AMAAEFAAAADDEBCFBEAAADFAKA---LNBFSQENDPPRAPMAB---SQAAFANACAEPELMPDKCLPDNNAMABAAAAANNKFSAAAA
pkt016 AABQLEWREDSAACFFDMKBLEB-----AMAAEFAAACBMBCFCEAAADFAKA---FPRSFSQENDPPRAPMAB---SQAAFANACAEPELMPDKCLPDNNAMABMAAANSBNAAAA
pkt017 -----AACFFDMKBLEBAABQLEWREDSAMAAEFAAAADDELKMQEAAAEAA---KNMFCRAPMABSQSF---QENDPP---NACAAAFAKCLPDNNAEPELMLMABAAA--RNDWQA
pkt018 AABQLEWREDSAACFFDMKBLEB-----AMAAEFAAAADDEBCFDEAAADFAKA---LMSFSQENDPPRAPMAB---SQAAFANACAEPELMLKCLPDNNAMABBAANNELMAAAA
pkt019 -----AACFFDMKBLEBAABQLEWREDSAMAAEFAAAADDELKMRFAAEAA---KNMFCRAPMABSQSF---QENDPP---NACAAAFAKCLPDNNAEPELPMMBABBAK--RNDWMA
pkt020 AABQLEWREDSAACFFDMKBLEB-----AMAAEFAAAADDEBCFEEAAADFAKA---LMTFSQENDPPRAPMAB---SQAAFANACAEPELPMKCLPDNNBMBABAAAAANNEFEAAAA
pkt021 WWWWWWWWWWWAACFFDMKBLEB-----AMAKAAABAMAAAK---AEAAAACFFD---MKBLEB-RAPMAB---WTAAAAA-----AAAAAAR--APMABBDAAAAAAAAAAAA
pkt022 WWWWWWWWWWWAACFFDMKBLEB-----AMAKAAABAMAAAK---AEAAAACFFD---MKBLEB-RAPMAB---WTAAAAA-----AAAAAAR--APMABBDAAAAAAAAAAAA
pkt023 WWWWWWWWWWWAACFFDMKBLEB-----AMAKAAABAMAAAK---AEAAAACFFD---MKBLEB-RAPMAB---WTAAAAA-----AAAAAAR--APMABBDAAAAAAAAAAAA
pkt024 -----AACFFDMKBLEBAABQLEWREDSAMAAEFAAAAFBAAAAEAAEAB---BEABSRAPMABSQNK-WTPAWSNLAL---AEPPAADSCFLFAS-CTTQMECMWBKSENWT--ADDCQ
pkt025 AABQLEWREDSAACFFDMKBLEB-----AMAAEFAAAAFBAAAAEAAADN---BBELBSNKWTPAWSRAPMAB---SQAEPPN-----LALAADSBSCBDAPLD
pkt026 ABAAFTAAAAABAACFFDMKBLEB-----AMAAEKRAACEAAAAEAAAABACEB---KRRAPMAB---WTTAAAAAAB---NEAEAAAABB---KETR-BTAAAAAAAACL
pkt027 ABAAFTAAAAABAACFFDMKBLEB-----AMAAEKRAACMAAAAAEAAAABACEB---FDRAPMAB---WTTAAAAABK---NEAEAAAACC---AATR-ADAAAAAABACA
pkt028 AABQLEWREDSAACFFDMKBLEB-----AMAAEFAAAAFBAAAAEAAADN---BBELBSNKWTPAWSRAPMAB---SQAEPPN-----LALAADSLTARDAKB-
pkt029 -----AACFFDMKBLEBAABQLEWREDSAMAAEFAAAAFBAAAAEAAEAB---BEABSRAPMABSQNK-WTPAWSNLAL---AEPPAADSCWQWDAK---RRBENSE-RDNLFTRWWDET
pkt030 WWWWWWWWWWWAACFFDMKBLEB-----AMAKAAABAMAAAK---AEAAAACFFD---MKBLEB-RAPMAB---WTAAAAA-----AAAAAAR--APMABBDAAAAAAAAAAAA
pkt031 WWWWWWWWWWWAACFFDMKBLEB-----AMAKAAABAMAAAK---AEAAAACFFD---MKBLEB-RAPMAB---WTAAAAA-----AAAAAAR--APMABBDAAAAAAAAAAAA
pkt032 WWWWWWWWWWWAACFFDMKBLEB-----AMAKAAABAMAAAK---AEAAAACFFD---MKBLEB-RAPMAB---WTAAAAA-----AAAAAAR--APMABBDAAAAAAAAAAAA
    
```

]]><-+ <<< >>> >>>>

Multiple sequence alignment

- Discloses evolutionary events
 - point mutations
 - insertions / deletions (“indels”)
- Further analysis shows relationships between sequences
 - common ancestors



Transformation (1)

- Representing 256 states with some letters
- Ideas
 - patching all the software to support 256 values
 - introducing incompatibilities, time consuming
 - uuencode
 - 32 values, not so many amino-acids
 - base 2 – binary format, just two letters
 - too many alignments across byte boundary
 - using base 24; 2 digits are enough
 - the first one is hardly changing
 - hexadecimal!
 - use A-F and translate 0-9 to free amino-acid codes
- Correct FASTA file: header line with “>”, file extension

Transformation (2)

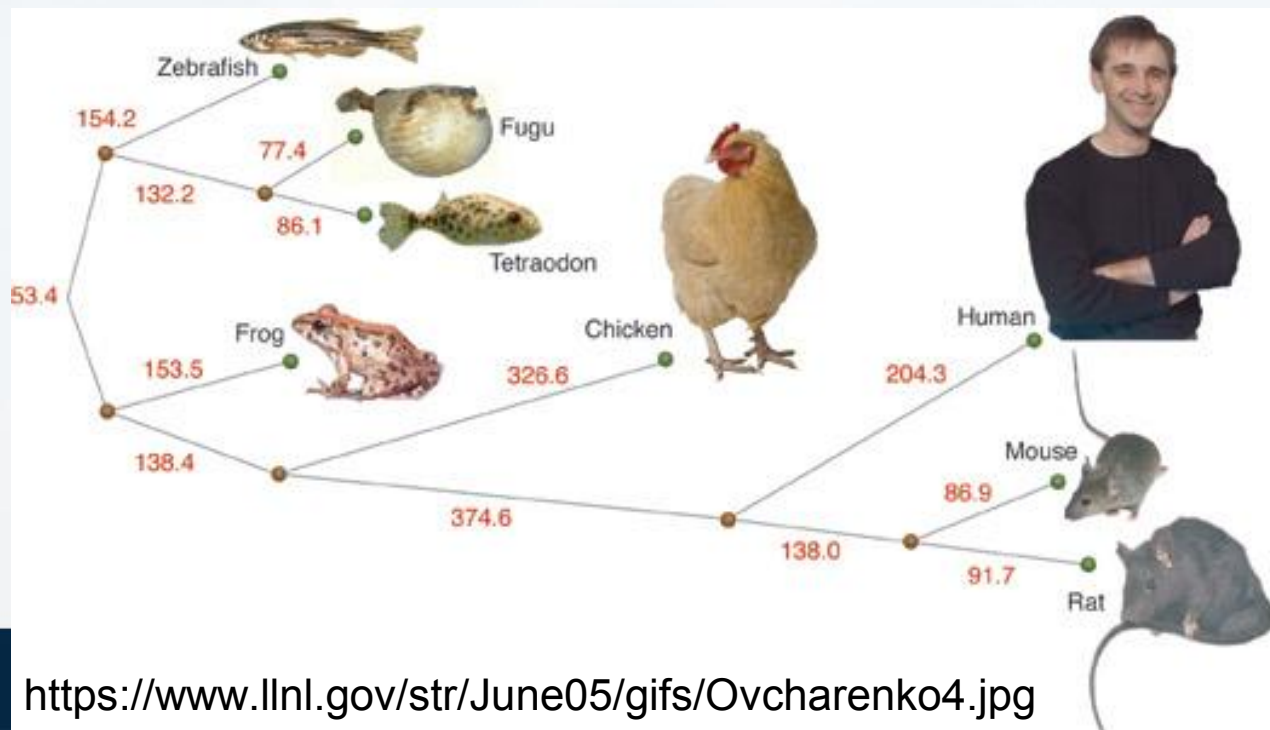
- Our tools transcode byte streams to amino-acid sequences
 - from proxyfuzz logs
 - from pcap dumps (using libpcap)
 - Ethernet, TCP and IP headers are preserved

```
$ ./pcap2fasta interesting.pcap
>pkt003 95 bytes
AACFFDMKBLEBAABQLLEWREDSAMAAEFAAAAFBAAAAEAAAEABBEABSRAPMABSQNKWTPAWSNLALAEPPA
ADSTCSSDAEPCLCRRQKWREBTDMLDKADBMTQMALEBENLEWFFLACKENEWADSNBLBRAKCFWFSQENEMBRF
DPDWNTKDBAQQLLDCBSLFMDAEPKQKSFSLADF
>pkt004 95 bytes
AABQLLEWREDSAACFFDMKBLEBAMAAEFAAAAFBAAAAEAAADNBBELBSNKWTPAWSRAPMABSQAEPPNLALA
ADSPFLADALFKAMSLKMWRNFBTTMCAFBDNRNNKKKWS PCCDNCQFDQLEKDKPTFFFTSPRSBRCNMQAPEFWF
RPQPANDBALQBSPPMBWNALMQFWNMBCAMLFLDN
>pkt005 62 bytes
AACFFDMKBLEBAABQLLEWREDSAMAAEFAAAADARRDEAAAEAAKWCNTRAPMABSQFQSKTSARQBTQBWMPQ
PSLEKRDDADLLRPFABMADTPBPNKAAAAAMAAAAA
```

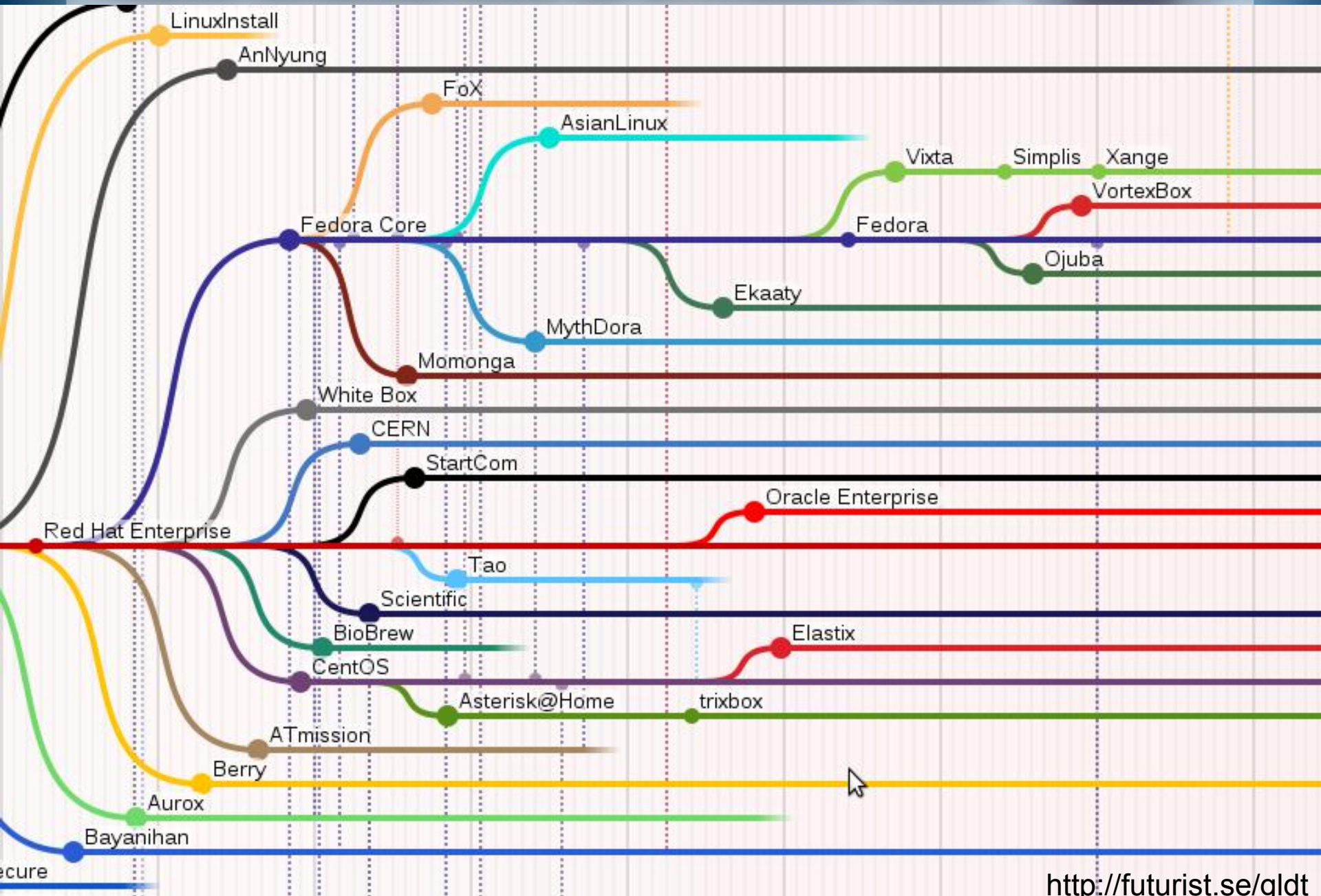
Transformation example (ping) DEMO

Phylogeny (1)

- Studies on evolution of life forms
- Tree of life forms and their similarity (“phylogenetic tree” or “cladogram”) based on:
 - phenotype (observable traits): pterodactyl ~ stork
 - genotype (DNA sequence): human ~ pig
- Similar forms are supposed to have a common ancestor
- Length of edges may have a meaning (time, difference)



POZNAŃ SUPERCOMPUTING AND NETWORKING CENTER

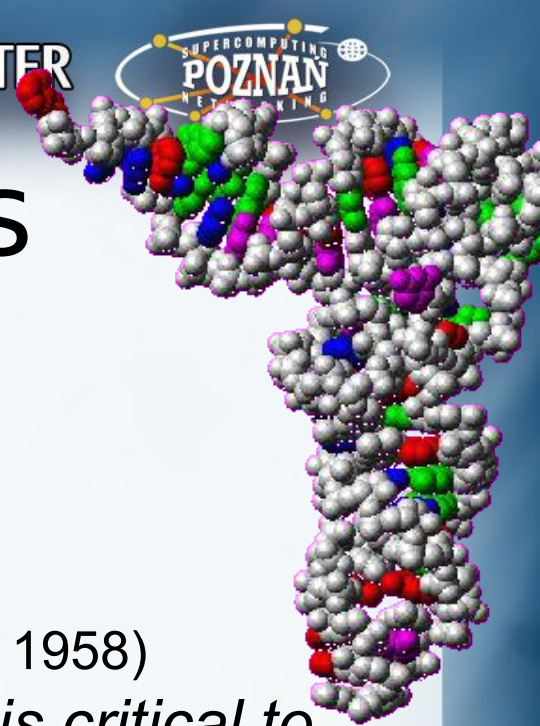


Analogies

- Similar packets have a common “ancestor”
 - generated by a specific function
 - a template
- Discovering the “template” helps in reverse engineering
 - nonces/cookies, encrypted data
 - headers, parameters
- (And then fuzzing only non-random data)

Protein families

- Descend from a common ancestor
- Similar functions, 3D structure, sequence
- Example: globins
 - Hemoglobin
 - Myoglobin (first protein sequenced to atoms, 1958)
- *“Reliable identification of protein families is critical to*
 - *phylogenetic analysis*
 - *functional annotation*
 - *the exploration of diversity of protein function in a given phylogenetic branch”* - reverse engineering!
(wikipedia)



Clustering packets (finding “families”)

- The famous original BLAST package (Basic Local Alignment Search Tool) from NCBI
- Tool: **blastclust**, input: FASTA packets
 - finds pairs of significantly similar sequences
 - clusters them in a lists

```
$ cat *.fasta | blastclust -S 1 -L 0.5
Nov 22, 2010 9:46 PM Start clustering of 33 queries
pkt000 pkt001 pkt002 pkt021 pkt022 pkt023 pkt030 pkt031 pkt032
pkt010 pkt014 pkt015 pkt018 pkt020
pkt009 pkt011 pkt017 pkt019
pkt004 pkt025 pkt028
pkt024 pkt029
pkt027 pkt026
pkt012
pkt016
pkt013
pkt008
```

Extracting family traits

- Creating a profile for each cluster
 - Take sequences from the cluster
 - Aligning cluster members once more
 - Extracting constant “sites” for families
 - Clustalx:
 - Quality → Save Column Scores to File
 - Copy columns with 100 (%) score
 - Discard other – insert a gap
 - Join lines & fold

```

A A A A A A A A A A      100
F F F F F F F F F F      100
E E E E E E E E E E      100
B D N A K R M S R C       17
Q R N C C D D T E E       16
M B P E E D F P K E       12
.....
    
```

```

sed 's:\(.\) .*100$:\1:' |
sed 's:.. \+:-:' |
tr --delete '\r\n' |
fold --width 70
    
```

```

AFE---QLLEWREDSAACFFDMKBLEBAMAAEFAAAAFE-
EAAAWLAB----SEESKEKFRAPMABSQAAAA----ASRA
AAF---TBWAER----ASAAAMANAPAQARASATAWBABB
BCBDBEBFBKBLBMBNBPBQBRBSBTBWCACBCCCDCECF
    
```

Applying to packets or files

- Intercept a byte sequence
- Find best matching profile & align with them
- Calculate byte ranges corresponding to constant sites (our tool)
 - Sample output: 2,3,4,14,15,16,34,
- Use ranges as a parameter for fuzzer
 - `zzuf --bytes=ranges`
- Forward the fuzzed data to application / network

Classifier

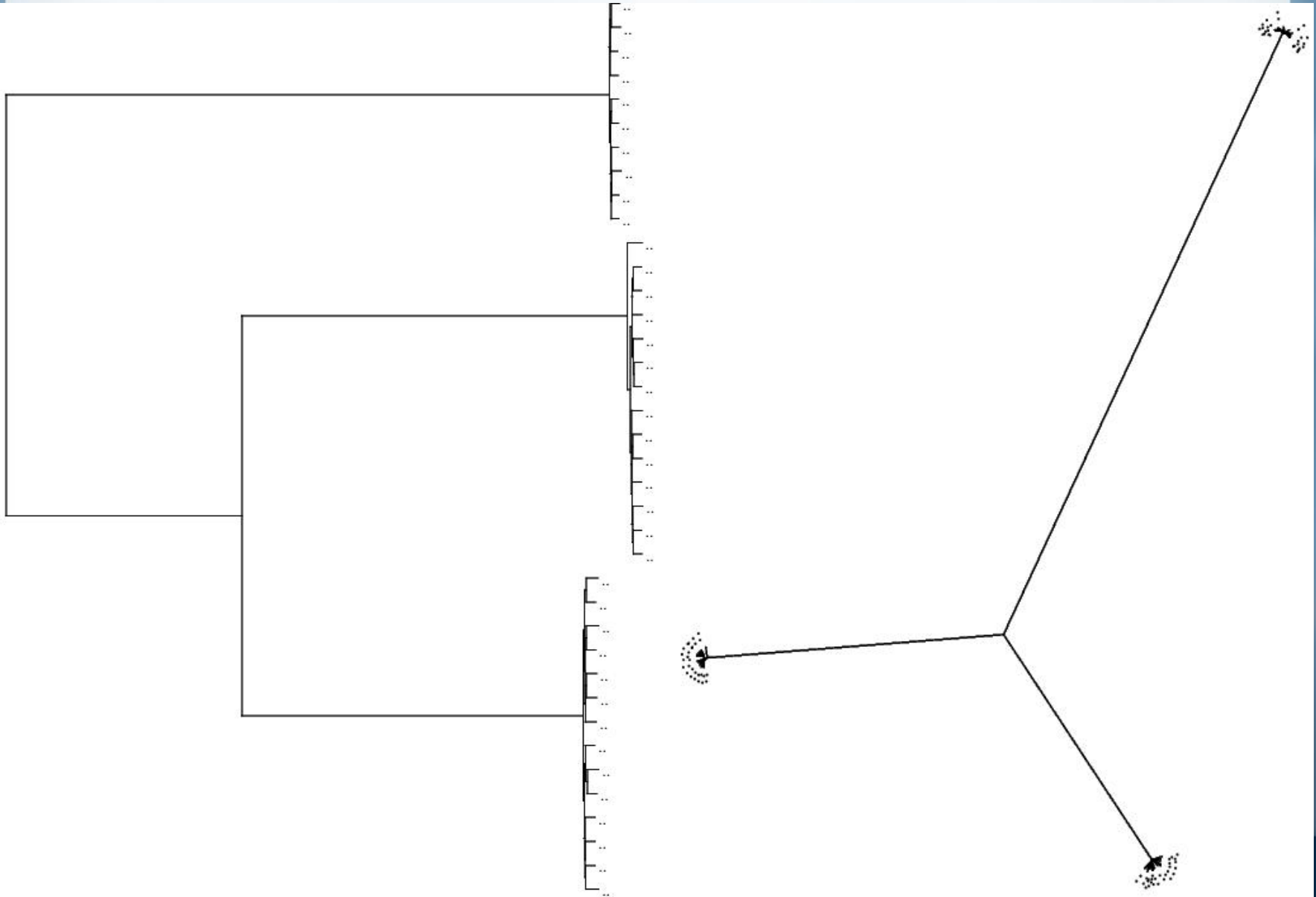
- Convert to regexp
- Drop byte halves (-X or X-)
- Decode “aminal” base encoding back to 256 values
- Use in ngrep or anywhere

Other surprises

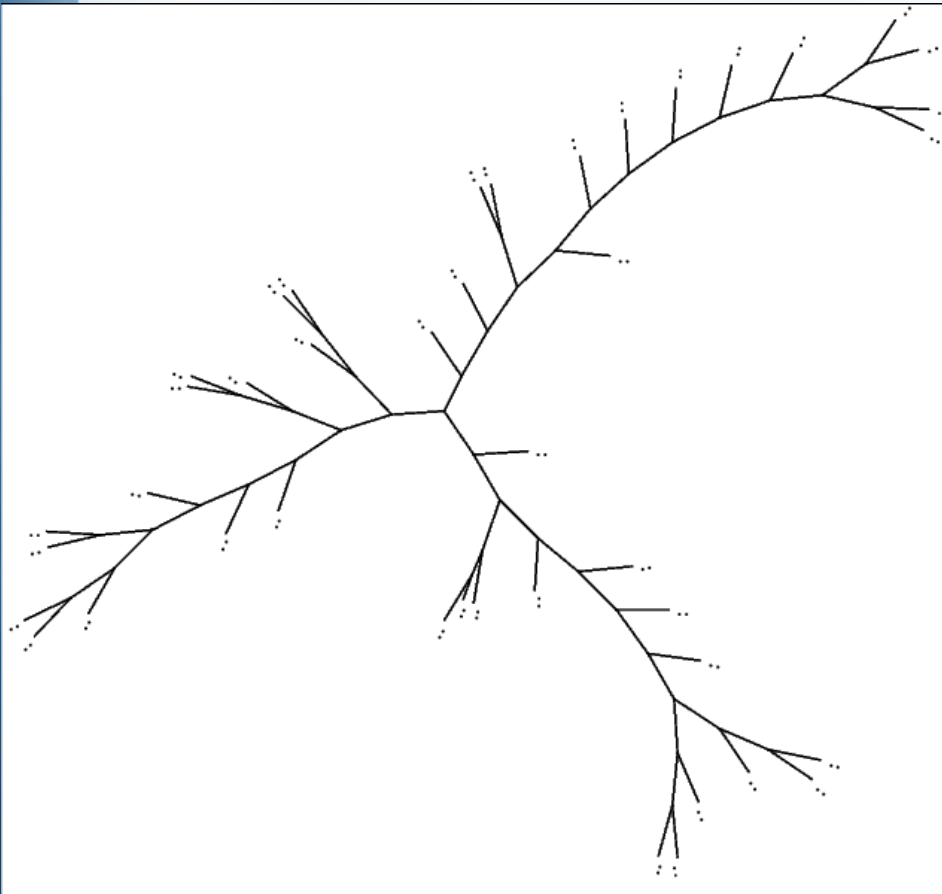


- Amino-acid distance matrix in alignment
 - Specifies probability of amino-acid replacement
 - In our case must be set to “identity”
- Some tools crashing when incorrectly used :)

Backup solution transmissions tree



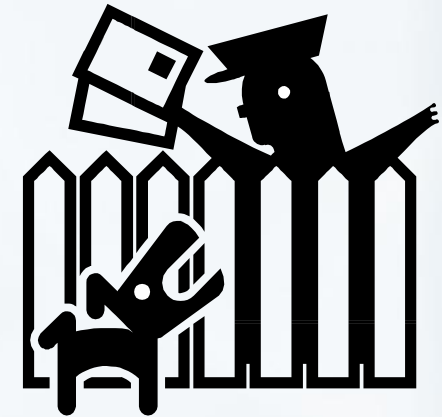
Backup solution transmission tree



- Each transaction split into 3 conversations
- Different
 - sizes
 - header parameters
 - positions of auth info
- Specialized fuzzing that doesn't break auth

Contact details

- Tomasz
 - tomasz.nowak@man.poznan.pl
- PSNC Security Team
 - <http://security.psnc.pl>
 - security@man.poznan.pl



Thank you for watching!



Questions, discussion...?