Stephan Kleber, Henning Kopp, Frank Kargl
Institute of Distributed Systems

August 13, 2018

NEMESYS

Network Message Syntax Reverse Engineering by Analysis of the Intrinsic Structure of Individual Messages

Static Traffic Analysis

```
⊟ Data (48 bytes)
    Data: d9000afa000000000102900000000000000000000000
    [Length: 48]

0000  00 0c 41 82 b2 53 00 d0  59 6c 40 4e 08 00 45 00
0010  00 4c 0a 4f 00 00 80 11  cc 40 c0 a8 32 32 42 6f
0020  2e c8 00 7b 00 7b 00 38  be d5 d9 00 0a fa 00 00
0030  00 00 00 01 02 90 00 00  00 00 00 00 00 00 00 00
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0050  00 00 c5 02 04 ec ee d3  3c 52
```

Static Traffic Analysis

Observable in Transit

Requires no access to program of entities

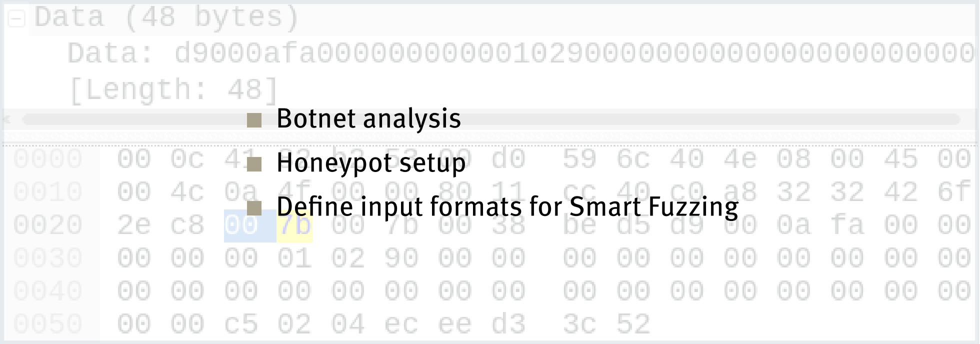
```
Data (48 bytes)
    Data: d9000afa000000000102900000000000000000000000000
    [Length: 48]

0000  00 0c 41 82 b2 53 00 d0  59 6c 40 4e 08 00 45 00
0010  00 4c 0a 4f 00 00 80 11  cc 40 c0 a8 32 32 42 6f
0020  2e c8 00 7b 00 7b 00 38  be d5 d9 00 0a fa 00 00
0030  00 00 00 01 02 90 00 00  00 00 00 00 00 00 00 00
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0050  00 00 c5 02 04 ec ee d3  3c 52
```

Static Traffic Analysis

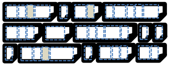
Observable in Transit

Requires no access to program of entities

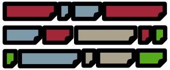


- Botnet analysis
- Honeypot setup
- Define input formats for Smart Fuzzing

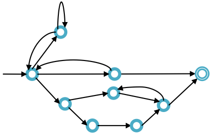
A Protocol Specification



Message Formats

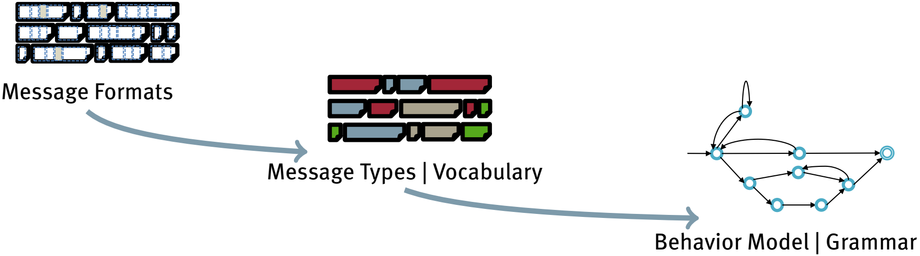


Message Types | Vocabulary



Behavior Model | Grammar

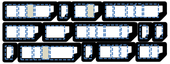
A Protocol Specification



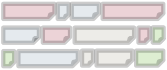
Reverse Engineering of

- **Vocabulary:** find message type by **format** similarities
- **Grammar:** generalize to sequence of **messages types**

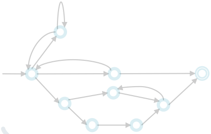
A Protocol Specification



Message Formats



Message Types | Vocabulary



Behavior Model | Grammar

Reverse Engineering of

- **Vocabulary:** find message type by **format** similarities
- **Grammar:** generalize to sequence of **messages types**

Format Inference

Determine field boundaries

Format Inference

Determine field boundaries:

Textual protocol (SMTP):

`RCPT TO: <twanda@blue6.ex>`

Binary protocol (DHCP):

`63825363350105360 4ac140301330400000e10`

Format Inference

Determine field boundaries:

Textual protocol (SMTP):

RCPT TO: <twanda@blue6.ex>

Binary protocol (DHCP):

638253633501053604ac140301330400000e10

Key:

Keyword

Separator

Value

Format Inference

Determine field boundaries:

Textual protocol (SMTP):

RCPT		TO:		<twanda@blue6.ex>	

Binary protocol (DHCP):

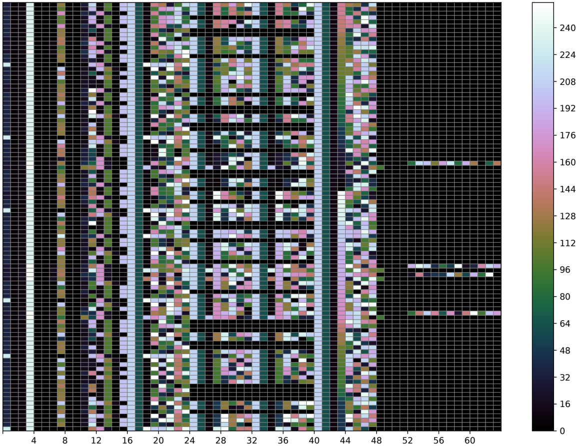
63825363	3501	053604ac140301	330400000e10

Key:

- Keyword
- Separator
- Value

A New Kind of Feature

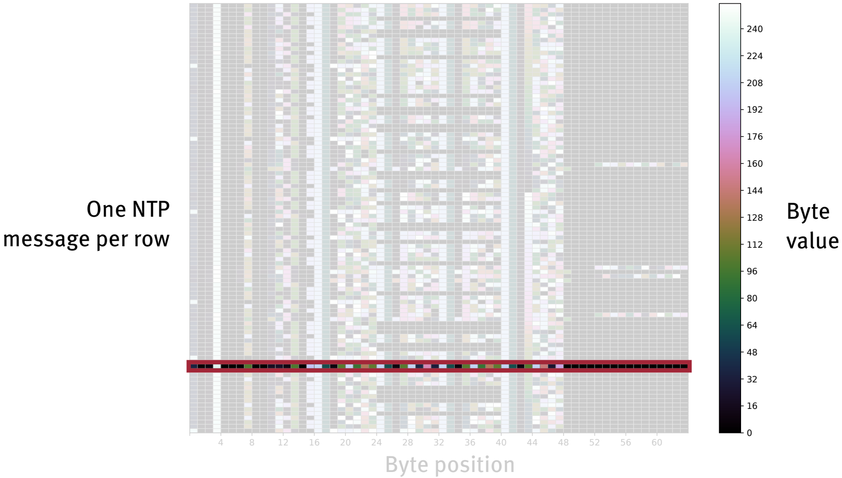
One NTP message per row



Byte position

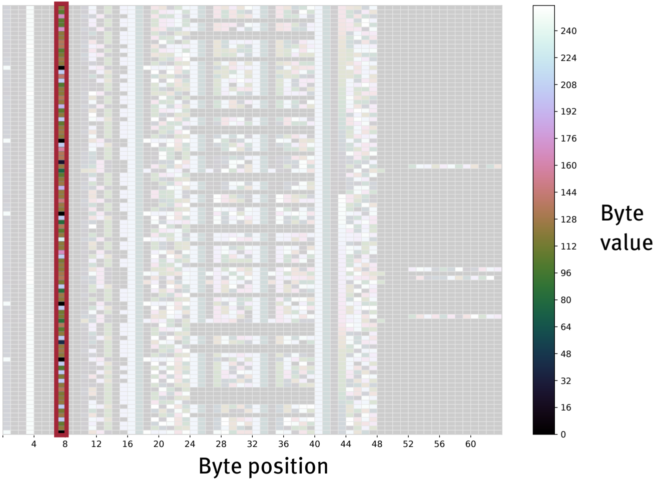
Byte value

A New Kind of Feature



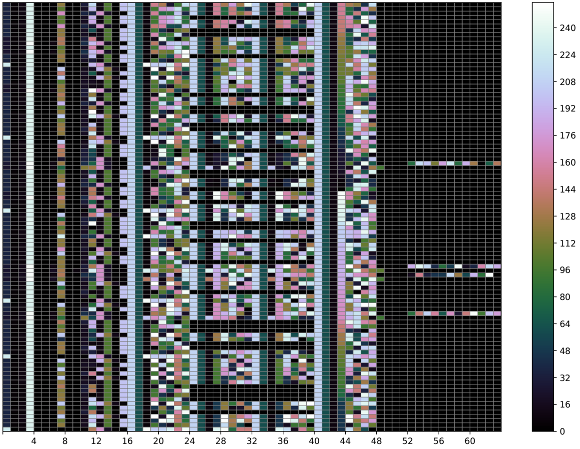
A New Kind of Feature

One NTP
message per row



A New Kind of Feature

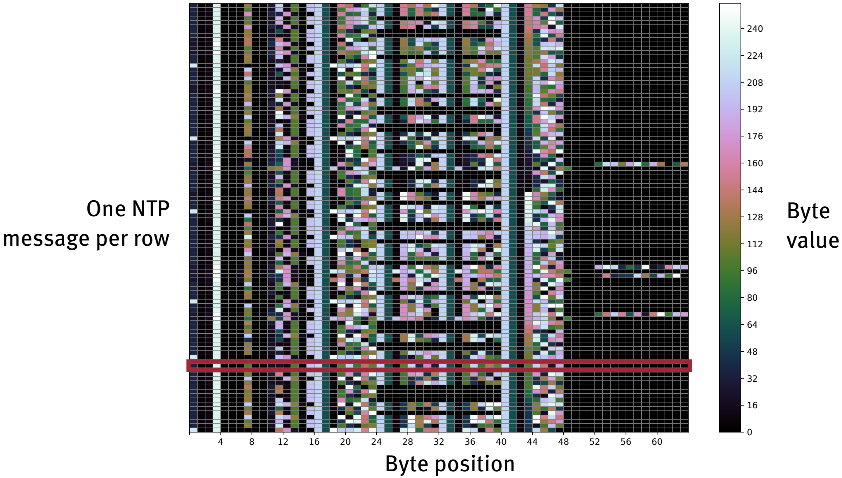
One NTP
message per row



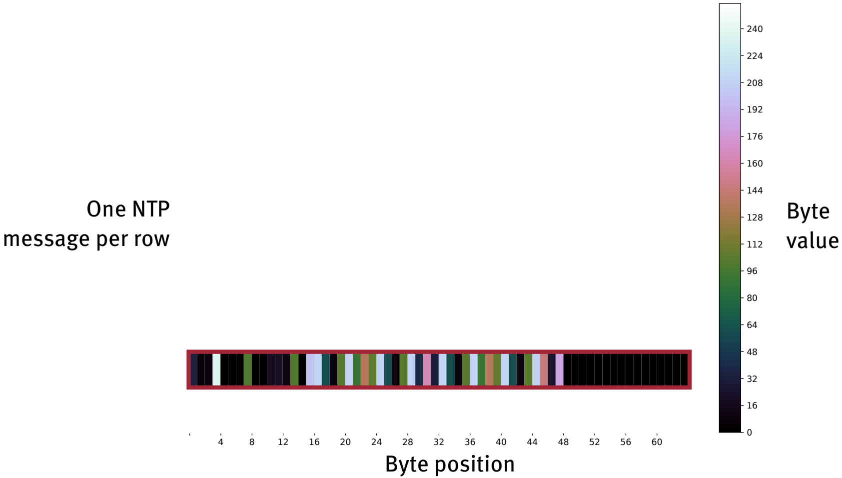
Byte position

Byte
value

A New Kind of Feature



A New Kind of Feature



Observations

- Intrinsic structure observable
- Messages designed for efficient parsing:
  → fixed-length data types
    → fields not uniformly filled
      `00 0a f8 fe`
- Counted numbers have specific variance distribution

Novel Feature to Pinpoint Boundaries

Change in the agreement of bits in consecutive bytes throughout ONE single message!

Deltas of Bit Congruence

Deltas of Bit Congruence

Bit Congruence:

based on similarity measure for bit strings
by Sokal and Michener (1958)

Deltas of Bit Congruence

Bit Congruence:

$$\mathrm{BC}(b, \overline{b}) = \frac{c_{\mathrm{agree}}(b, \overline{b})}{8}$$

$c_{\mathrm{agree}}(b, \overline{b})$: number of congruent bits for bytes b and $\overline{b}$

Deltas of Bit Congruence

$$\Delta\mathrm{BC} = \big(\mathrm{BC}(m_k, m_{k+1}) - \mathrm{BC}(m_{k-1}, m_k)\big)_{0<k<n}$$

with

$$\mathrm{BC}(b, \overline{b}) = \frac{c_{\mathrm{agree}}(b, \overline{b})}{8}$$

m_k: Message m's byte at position k, m has length $n + 1$

$c_{\mathrm{agree}}(b, \overline{b})$: number of congruent bits for bytes b and $\overline{b}$

Deltas of Bit Congruence

$$\Delta\text{BC} = \big(\text{BC}(m_k, m_{k+1}) - \text{BC}(m_{k-1}, m_k)\big)_{0<k<n}$$

with

$$\text{BC}(b, \overline{b}) = \frac{c_{\text{agree}}(b, \overline{b})}{8}$$

m_k: Message m's byte at position k, m has length $n + 1$

$c_{\text{agree}}(b, \overline{b})$: number of congruent bits for bytes b and $\overline{b}$

Deltas of Bit Congruence

$k = 1$

$$\Delta\mathrm{BC} = \big(\mathrm{BC}(m_k, m_{k+1}) - \mathrm{BC}(m_{k-1}, m_k)\big)_{0<k<n}$$

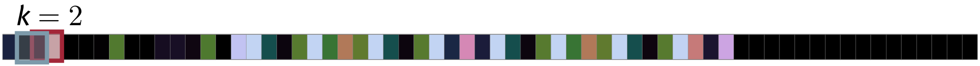
with

$$\mathrm{BC}(b, \overline{b}) = \frac{c_{\mathrm{agree}}(b, \overline{b})}{8}$$

m_k: Message m's byte at position k, m has length $n+1$

$c_{\mathrm{agree}}(b, \overline{b})$: number of congruent bits for bytes b and $\overline{b}$

Deltas of Bit Congruence



$$\Delta\text{BC} = \left(\text{BC}(m_k, m_{k+1}) - \text{BC}(m_{k-1}, m_k)\right)_{0<k<n}$$

with

$$\text{BC}(b, \overline{b}) = \frac{c_{\text{agree}}(b, \overline{b})}{8}$$

m_k: Message m's byte at position k, m has length $n+1$

$c_{\text{agree}}(b, \overline{b})$: number of congruent bits for bytes b and $\overline{b}$

Deltas of Bit Congruence

$k = 3$

$$\Delta\mathrm{BC} = \left(\mathrm{BC}(m_k, m_{k+1}) - \mathrm{BC}(m_{k-1}, m_k)\right)_{0<k<n}$$

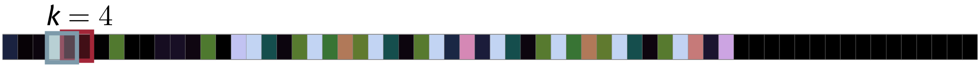
with

$$\mathrm{BC}(b, \overline{b}) = \frac{c_{\mathrm{agree}}(b, \overline{b})}{8}$$

m_k: Message m's byte at position k, m has length $n + 1$

$c_{\mathrm{agree}}(b, \overline{b})$: number of congruent bits for bytes b and $\overline{b}$

Deltas of Bit Congruence



$$\Delta\text{BC} = \left(\text{BC}(m_k, m_{k+1}) - \text{BC}(m_{k-1}, m_k)\right)_{0<k<n}$$

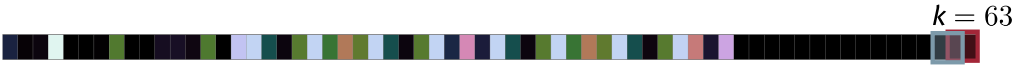
with

$$\text{BC}(b, \overline{b}) = \frac{c_{\text{agree}}(b, \overline{b})}{8}$$

m_k: Message m's byte at position k, m has length $n + 1$

$c_{\text{agree}}(b, \overline{b})$: number of congruent bits for bytes b and $\overline{b}$

Deltas of Bit Congruence



$$\Delta \mathrm{BC} = \left(\mathrm{BC}(m_k, m_{k+1}) - \mathrm{BC}(m_{k-1}, m_k)\right)_{0<k<n}$$

with

$$\mathrm{BC}(b, \overline{b}) = \frac{c_{\mathrm{agree}}(b, \overline{b})}{8}$$

m_k: Message m's byte at position k, m has length $n + 1$

$c_{\mathrm{agree}}(b, \overline{b})$: number of congruent bits for bytes b and $\overline{b}$

Applying the Feature

Feature ΔBC:
distinctive distribution for binary numbers:

- At field transition: low ΔBC
- Towards field end: high ΔBC

Applying the Feature

Feature ΔBC:
distinctive distribution for binary numbers:

- At field transition: low ΔBC
- Towards field end: high ΔBC
- Gaussian filter $g_\sigma(\cdot)$ to reduce noise

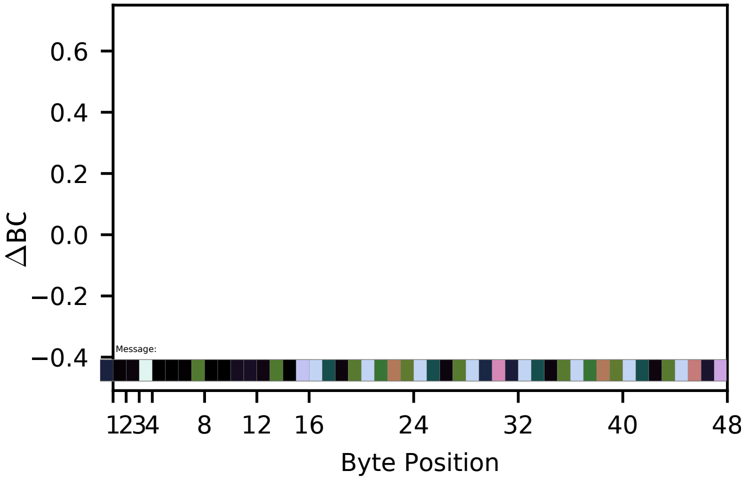
Applying the Feature

Feature ΔBC:
distinctive distribution for binary numbers:

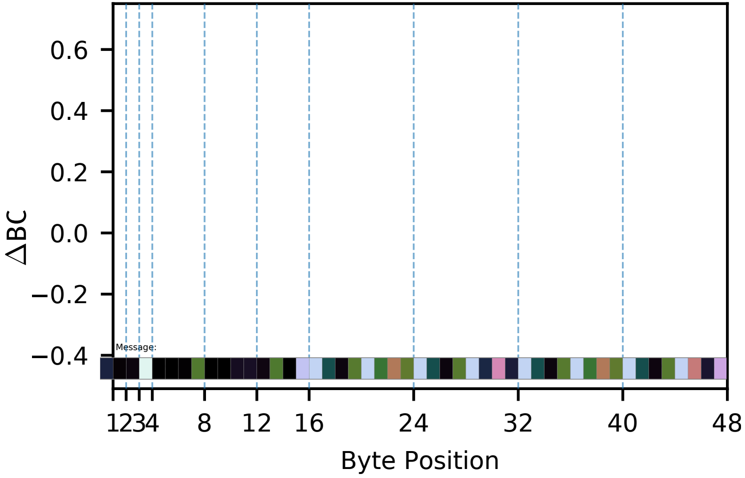
- At field transition: low ΔBC
- Towards field end: high ΔBC
- Gaussian filter $g_\sigma(\cdot)$ to reduce noise

Inflection points of rising edges of $g_\sigma(\Delta\text{BC})$

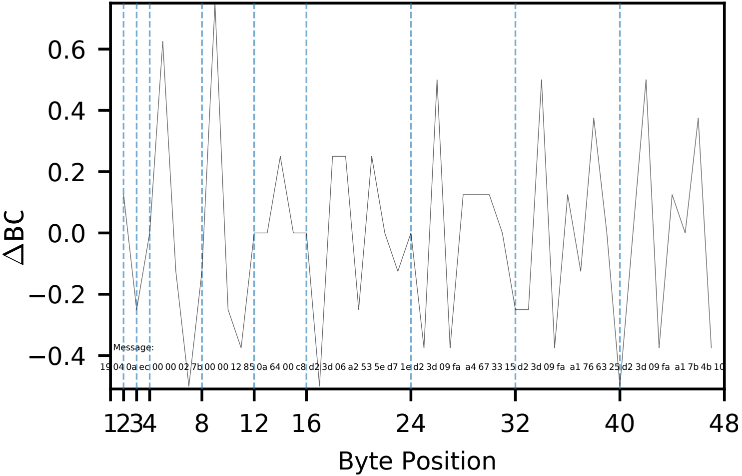
Value Pattern: Feature of one NTP message



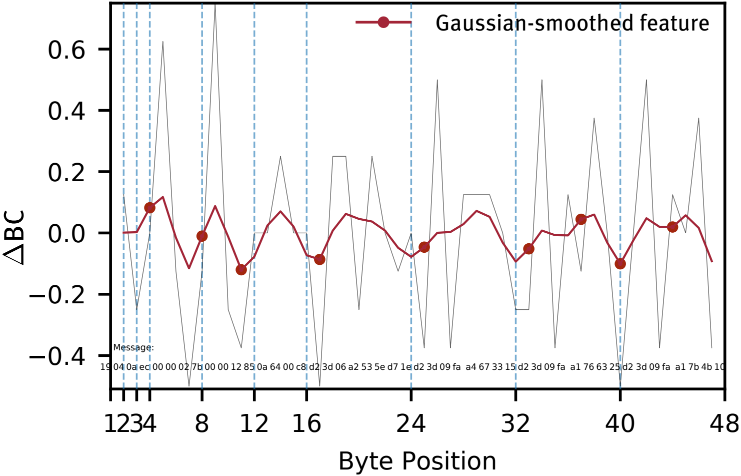
Value Pattern: Feature of one NTP message



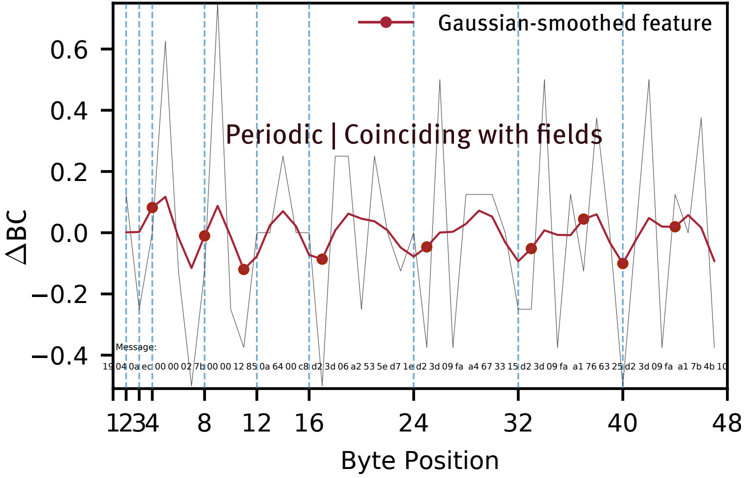
Value Pattern: Feature of one NTP message



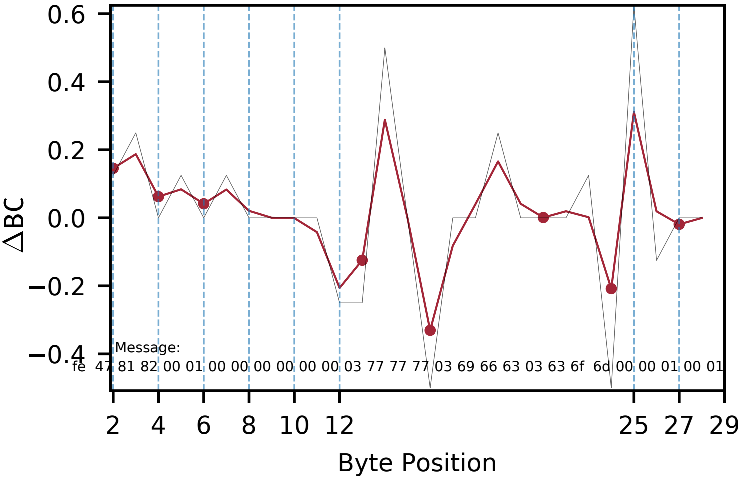
Value Pattern: Feature of one NTP message



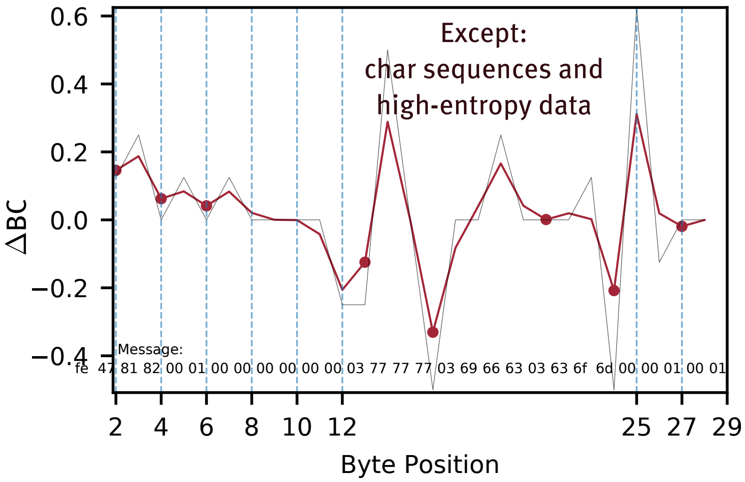
Value Pattern: Feature of one NTP message



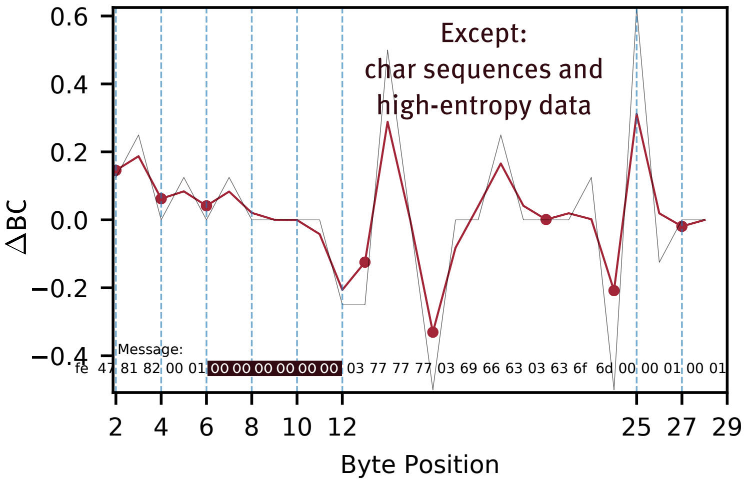
Value Pattern: Feature of one DNS message



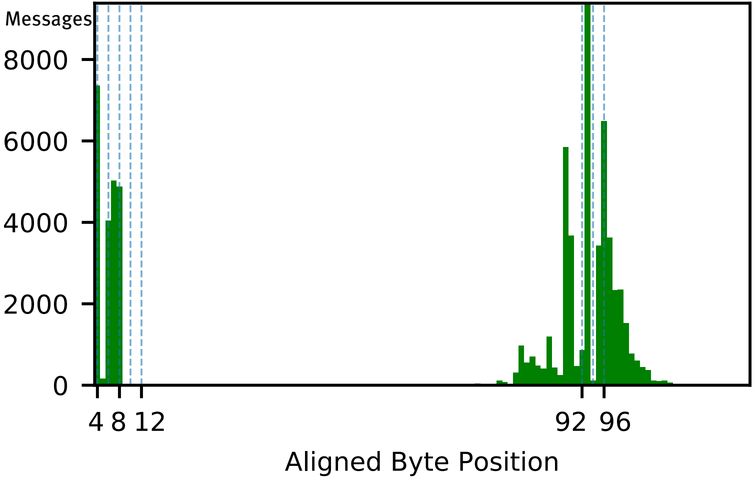
Value Pattern: Feature of one DNS message



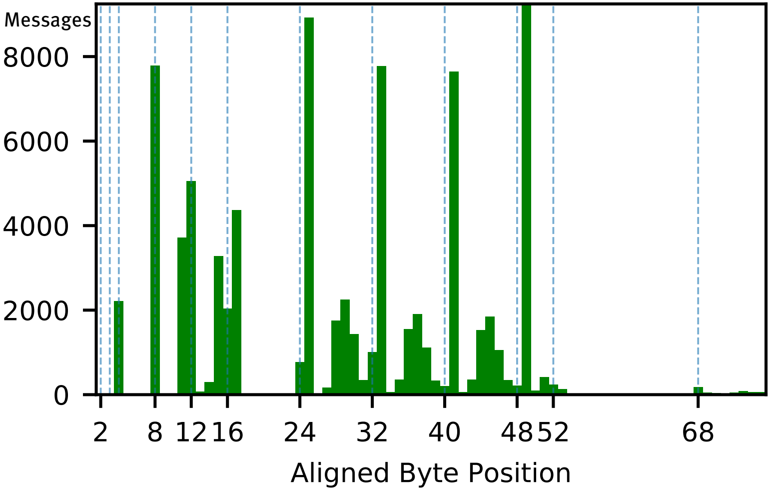
Value Pattern: Feature of one DNS message



Pinpoint Field Boundaries: DNS



Pinpoint Field Boundaries: NTP



Quantify Format Inference Quality

Validate format inference method:

Measure correctness by benchmarking with a known protocol

Format Match Score

$$\mathrm{FMS} = \underbrace{\exp\left(-\left(\frac{|R|-|I|}{|R|}\right)^2\right)}_{\text{Specificity penalty}} \cdot \underbrace{\frac{1}{|R|}\sum_{r\in R}\exp\left(-\left(\frac{\delta_r}{\gamma}\right)^2\right)}_{\text{Match gain}}$$

Format Match Score

$$\text{FMS} = \underbrace{\exp\left(-\left(\frac{|R| - |I|}{|R|}\right)^2\right)}_{\text{Specificity penalty}} \cdot \underbrace{\frac{1}{|R|}\sum_{r \in R} \exp\left(-\left(\frac{\delta_r}{\gamma}\right)^2\right)}_{\text{Match gain}}$$

Quality aspects:

- $|R|$ Number of real field boundaries
- $|I|$ Number of inferred field boundaries

Format Match Score

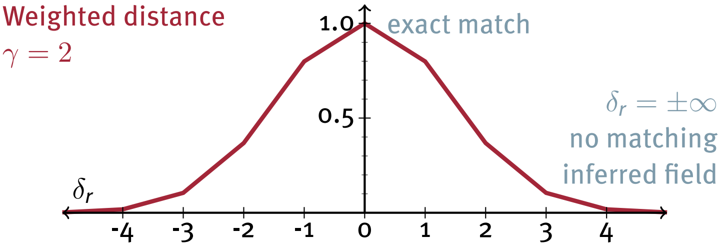
$$\mathrm{FMS} = \underbrace{\exp\left(-\left(\frac{|R| - |I|}{|R|}\right)^2\right)}_{\text{Specificity penalty}} \cdot \underbrace{\frac{1}{|R|} \sum_{r \in R} \exp\left(-\left(\frac{\delta_r}{\gamma}\right)^2\right)}_{\text{Match gain}}$$

Quality aspects:

- $|R|$ Number of real field boundaries
- $|I|$ Number of inferred field boundaries
- δ_r Distance of real boundary r from next inferred one
- γ Required accuracy

Format Match Score

$$\text{FMS} = \underbrace{\exp\left(-\left(\frac{|R|-|I|}{|R|}\right)^2\right)}_{\text{Specificity penalty}} \cdot \underbrace{\frac{1}{|R|}\sum_{r\in R} \exp\left(-\left(\frac{\delta_r}{\gamma}\right)^2\right)}_{\text{Match gain}}$$



Format Match Score

$$\mathrm{FMS} = \underbrace{\exp\left(-\left(\frac{|R| - |I|}{|R|}\right)^2\right)}_{\text{Specificity penalty}} \cdot \underbrace{\frac{1}{|R|}\sum_{r \in R} \exp\left(-\left(\frac{\delta_r}{\gamma}\right)^2\right)}_{\text{Match gain}}$$

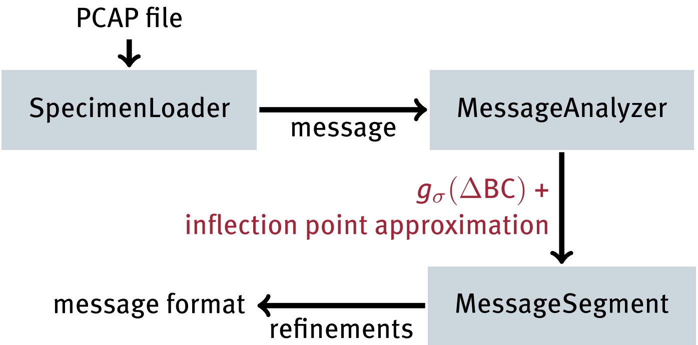
Quantify format correctness

Implementation

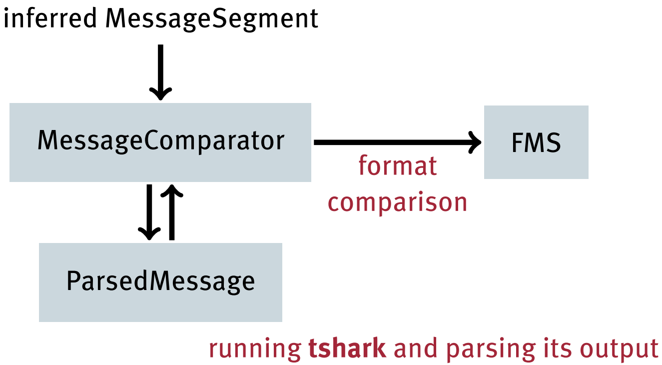
NEMESYS

NEtwork MEssage SYntax analysiS

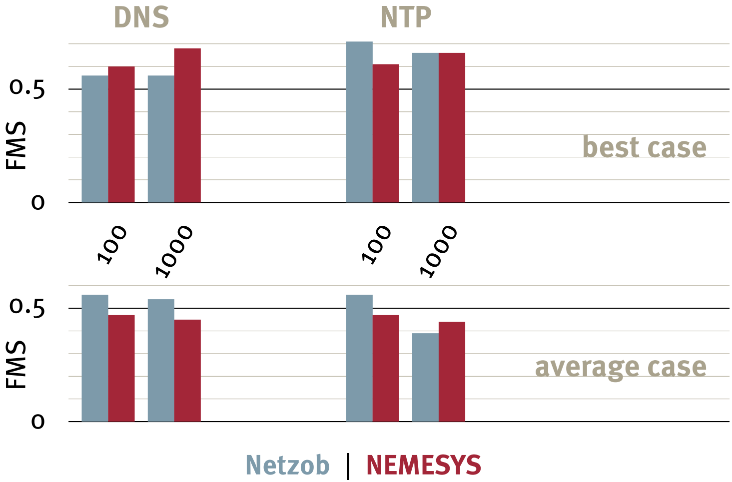
NEMESYS Architecture



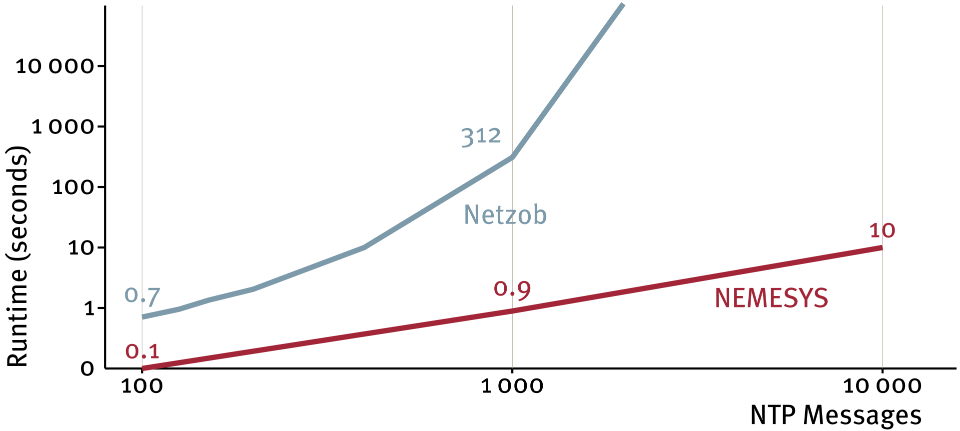
Evaluation Process



Evaluation Results: Quality



Evaluation Results: Performance



Reduces runtime from exponential to linear

Future Work

Use characteristic features to recognize field data types

- Integer: 00 0a f8 fe
- String: 69 44 53 00
- Padding: 57 b0 00 00
- Find more data-type-specific patterns:
  flags | addresses | signed numbers | floats | enumerations

Message Type Identification:

Cluster messages on patterns of segment data types
(based on Cui et al., 2007; FieldHunter, 2016)

Conclusion

NEMESYS:

Novel method for format inference

- Intrinsic message structure
- Binary protocols
- Abstracting from concrete byte values
- Linear time complexity

Conclusion

NEMESYS:

Novel method for format inference

- Intrinsic message structure
- Binary protocols
- Abstracting from concrete byte values
- Linear time complexity

Format Match Score:

Quality assessment of format inference methods

Thank you!

Questions?

web uulm.de?kleber
mail stephan.kleber@uni-ulm.de

Institute of Distributed Systems, Ulm University

web uulm.de/in/vs
github github.com/vs-uulm