Alcatel·Lucent

Improving Anomaly Detection for Text-based Protocols by Exploiting Message Structures

"Security in NGNs and the Future Internet" Workshop, Berlin

Martin Güthle, Jochen Kögel, Stefan Wahl (Bell Labs),
Christian Müller, Matthias Kaschub

mguethle@xunit.de

September 20th, 2010

Universität Stuttgart
Institute of Communication Networks
and Computer Engineering (IKR)
Prof. Dr.-Ing. Andreas Kirstädter

Outline

Motivation

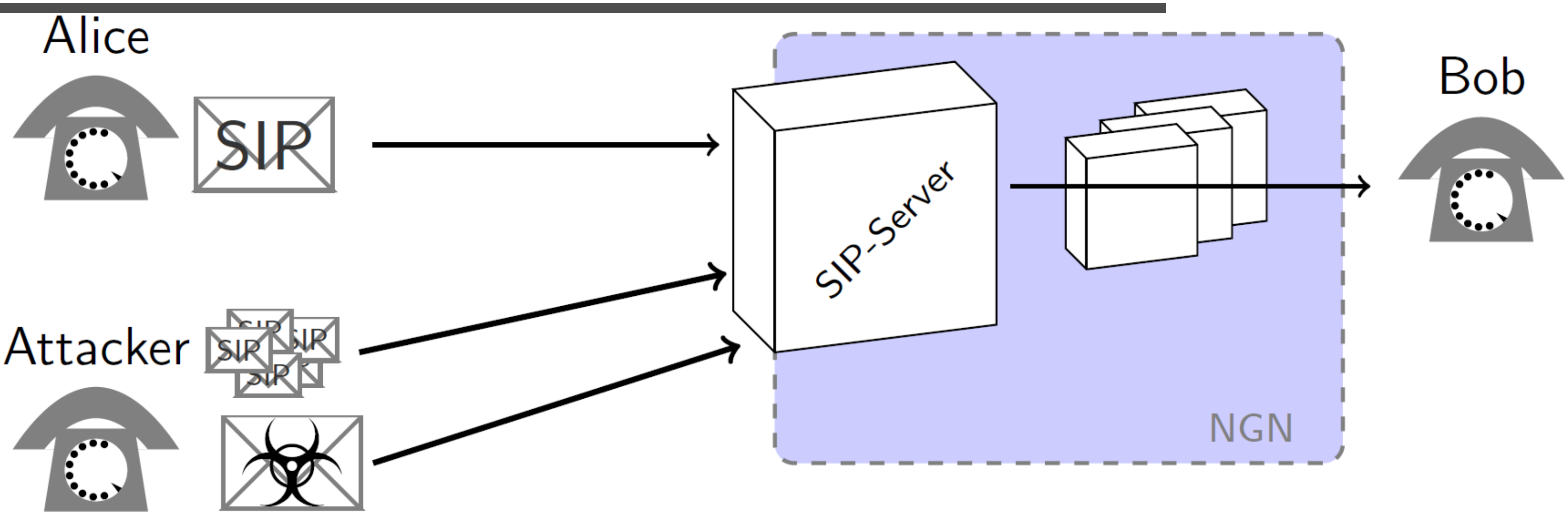
Approach

Improvement

- Extension for better detection
- Extension for higher throughput

Conclusion and Outlook

Motivation



Threat: Attacks on server

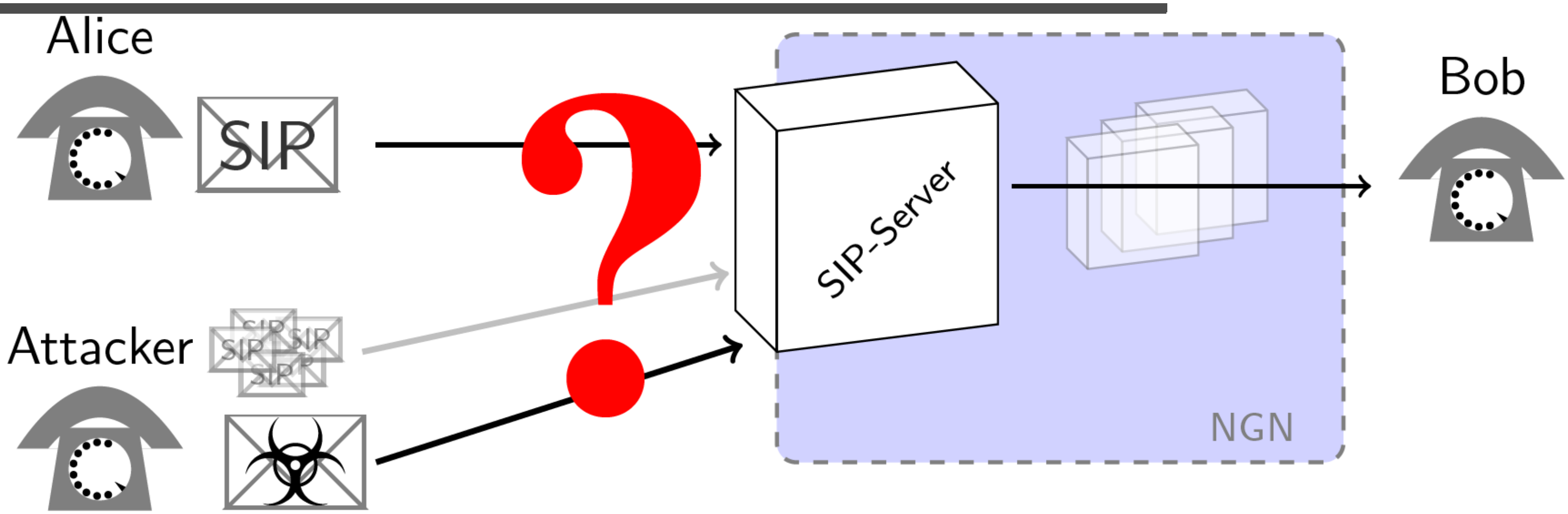
SIP: High susceptibility to vulnerabilities

- SIP server open to the outside: UNI of NGN
- SIP is complex and extensible
  - static filtering impossible
  - high probability of implementation weaknesses

Type of attacks against SIP servers

- Denial of Service
- Server integrity (e.g. gain root access) → effects thousand of millions customers

Motivation



Threat: Attacks on server

SIP: High susceptibility to vulnerabilities

- SIP server open to the outside: UNI of NGN
- SIP is complex and extensible
  - static filtering impossible
  - high probability of implementation weaknesses

Type of attacks against SIP servers

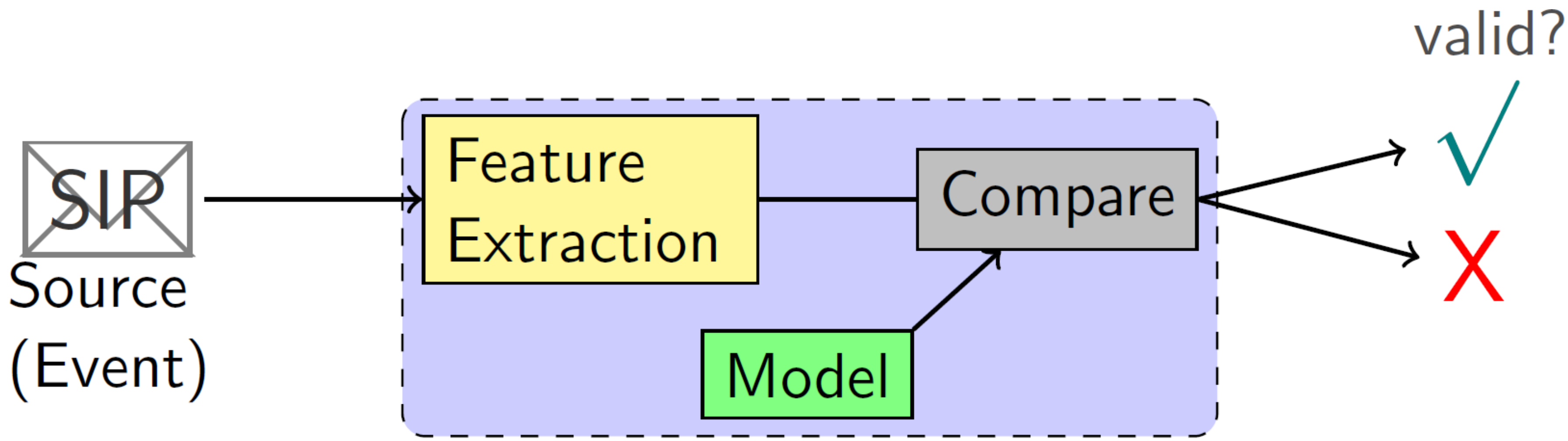
- Denial of Service
- **Server integrity**

→At border of the NGN (Firewall)

→Stateless

Approach

Overview



Intrusion detection by anomaly detection

- Compare against model: classification
- Predefined model based on a training set

Requirements

1. Good detection rate
   - ~100% true positive
   - <0.1% false positive
2. High throughput

Approach

Feature Extraction (n-grams)

Converting text into features with numerical values

- Header fields can occur in any order
- Leverage previous work [1]
  - N-grams for feature generation
  - Dimension with good trade off between detection and performance is **4** ([1])

Principle of n-gram extraction

A sliding window is shifted over the text

INVITE sip:bob@exampleiNVITE.com SIP/2.0

extracted features: $\underbrace{\begin{bmatrix} INVI \\ \\ \\ \\ \end{bmatrix}}_{Feature} = \underbrace{\begin{bmatrix} 1 \\ \\ \\ \\ \end{bmatrix}}_{Value}$

[1] A self-learning system for detection of anomalous SIP messages IPTComm 2008

Approach

Feature Extraction (n-grams)

Converting text into features with numerical values

- Header fields can occur in any order
- Leverage previous work [1]
  - N-grams for feature generation
  - Dimension with good trade off between detection and performance is **4** ([1])

Principle of n-gram extraction

A sliding window is shifted over the text

INVITE sip:bob@exampleiNVITE.com SIP/2.0

extracted features: $$\underbrace{\begin{bmatrix} INVI \\ NVIT \\ \\ \\ \end{bmatrix}}_{Feature} = \underbrace{\begin{bmatrix} 1 \\ 1 \\ \\ \\ \end{bmatrix}}_{Value}$$

[1] A self-learning system for detection of anomalous SIP messages IPTComm 2008

Approach

Feature Extraction (n-grams)

Converting text into features with numerical values

- Header fields can occur in any order
- Leverage previous work [1]
  - N-grams for feature generation
  - Dimension with good trade off between detection and performance is **4** ([1])

Principle of n-gram extraction

A sliding window is shifted over the text

INVITE sip:bob@exampleiNVITE.com SIP/2.0

extracted features: $\underbrace{\begin{bmatrix} INVI \\ NVIT \\ \ldots \\ /2.0 \end{bmatrix}}_{Feature} = \underbrace{\begin{bmatrix} 1 \\ 2 \\ \ldots \\ 1 \end{bmatrix}}_{Value}$

[1] A self-learning system for detection of anomalous SIP messages IPTComm 2008

Approach

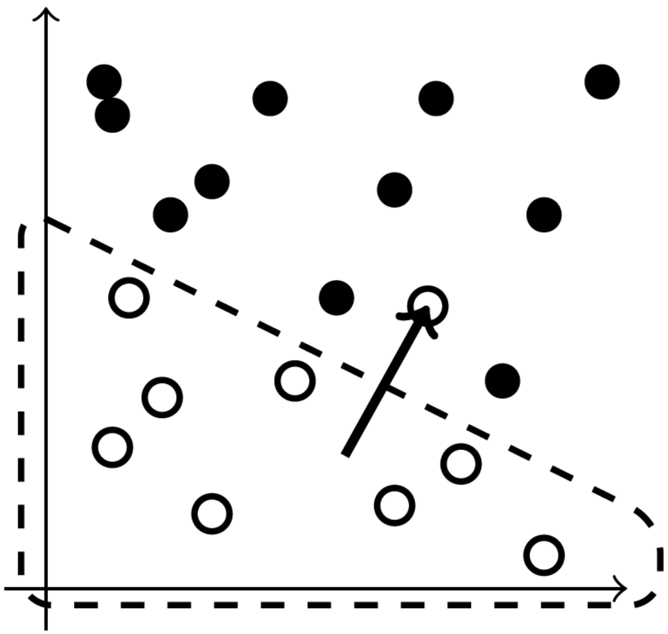
Model description and the compare unit

Classifier-based machine learning algorithm: *Support Vector Machine* (SVM)

- Cost factor defined with $\boxed{C \in [0;\infty)}$ (SVM extension [2])
- Additional extension: *one class classification*
- LibSVM implementation

Current limitations

- Labeled data set needed
- Training defines allowed features
- Retraining is not possible



Cost function allows outliers

[2] Support vector domain description Pattern Recognition Letters 20 (1999)

Basic results

Used data set

Three different training and test data sets

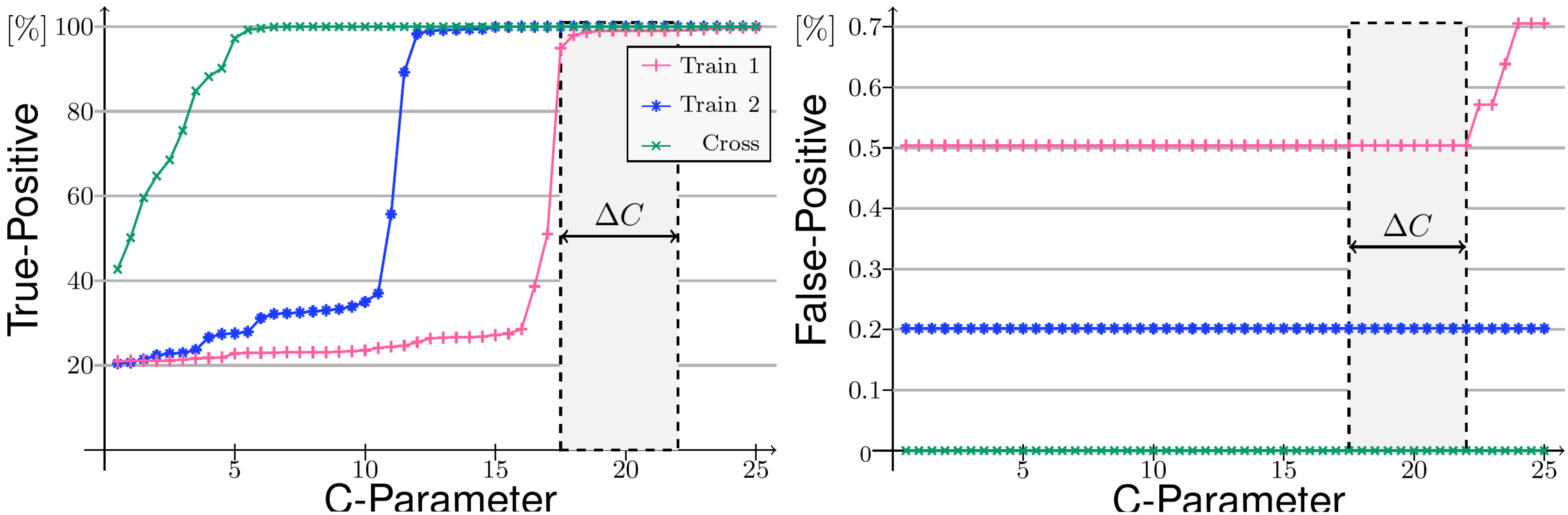
- Training and test data sets are labeled
- Data sets are automatically generated, based on Codenomicon

Overview of the used data sets

Name	# messages	# valids	# invalids	used for
Train 1	610	598	12	training only
Train 2	928	900	28	training only
Test 1	12,923	2,923	10,000	test only (Train 1 + 2)
Cross	12,586	11,579	1,007	10 fold cross validation

Basic results

Evaluation of cost factor (C)



Results

- High detection rate $\rightarrow$ approach works with these sets
- Remaining problem
  - Range ΔC very narrow
  - False-Positive rate still too high

$\rightarrow$ Improvement necessary

Basic results

What are reasons for the high False-Positive rate and narrow ΔC ?

- Different types of messages (Request / Response + INVITE / ACK ...)
- Optional header fields + different occurrence (e.g. multiple Via)
- Value of header fields may need session knowledge

```
SIP/2.0 180 Ringing
Via: SIP/2.0 ex.com;branch=abcd;
From: Alice <sip:alice@ex.com>
To: Bob <sip:bob@example.com>
CSeq: 1 INVITE
Content-Length: 0
```

```
ACK sip:bob@example.com SIP/2.0
From: Alice <sip:alice@ex.com>
To: Bob <sip:bob@example.com>
CSeq: 4511 ACK
Content-Length: 0
```

Improvement

Keyword extension

```
SIP/2.0 180 Ringing
Via: SIP/2.0 ex.com;branch=abcd;
From: Alice <sip:alice@ex.com>
To: Bob <sip:bob@example.com>
CSeq: 1 INVITE
Content-Length: 0
```

```
ACK sip:bob@example.com SIP/2.0
From: Alice <sip:alice@ex.com>
To: Bob <sip:bob@example.com>
CSeq: 4511 ACK
Content-Length: 0
```

Consider the parts which identify these reasons → **Keywords**

- A header field (e.g. Via)
- Any token inside the message (e.g. branch)

Possible actions correspond to a keyword

1. Keyword as additional feature
2. Replacement of session specific information
3. Start additional further processing

Improvement

Usage of the keywords

$$\underbrace{\begin{bmatrix} \mathit{INVI} \\ \dots \\ /2.0 \\ \text{Via} \\ \text{Content-Length} \end{bmatrix}}_{\mathit{Feature}} = \underbrace{\begin{bmatrix} 1 \\ \dots \\ 1 \\ 1 \\ 0 \end{bmatrix}}_{\mathit{Value}}$$

1. Keyword as additional feature

Option 1: Occur of the keyword

Option 2: Value correspond to the keyword

2. Replace session specific information

```
SIP/2.0 180 Ringing
Via: SIP/2.0 ex.com;branch=~~abcd~~;
Content-Length: 0
```

→ Independent to the session state (comparable to noise)

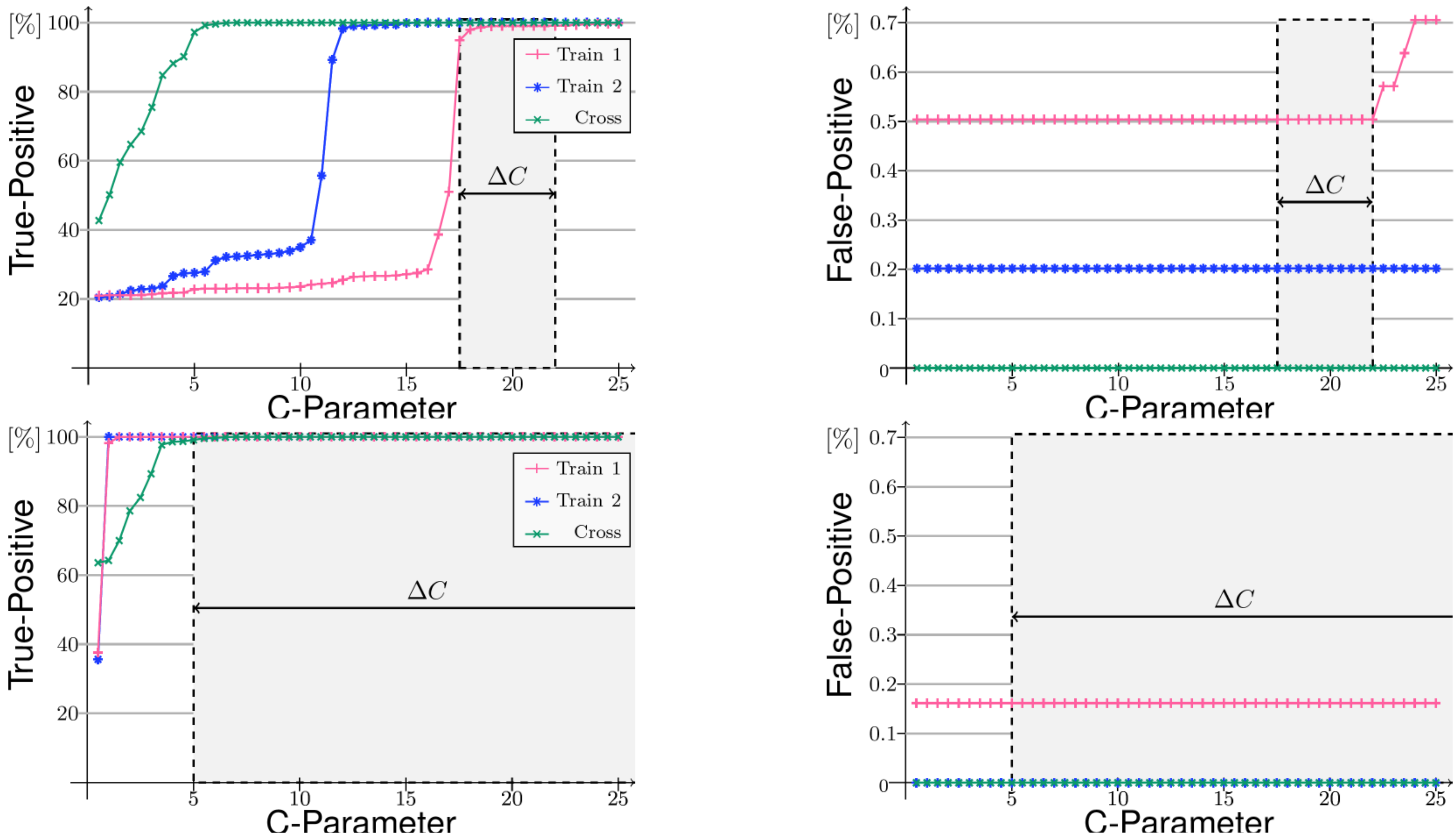
3. Start additional processing

Submodel 1		Submodel 2	Submodel 3
SIP/2.0 180 Ringing CSeq: 1 INVITE …	INVITE sip:bob@ex.comSIP/2.0 CSeq: 1 INVITE …	ACK sip:bob@ex.com SIP/2.0 CSeq: 4711 ACK …	OPTIONS sip:c@c.com SIP/2.0 CSeq: 63104 OPTIONS …

These keywords call additional code (e.g. using CSeq to generate submodels)

Improvement

Evaluation with Submodels *and* Remove of session information



→ **substantial improvement reached**

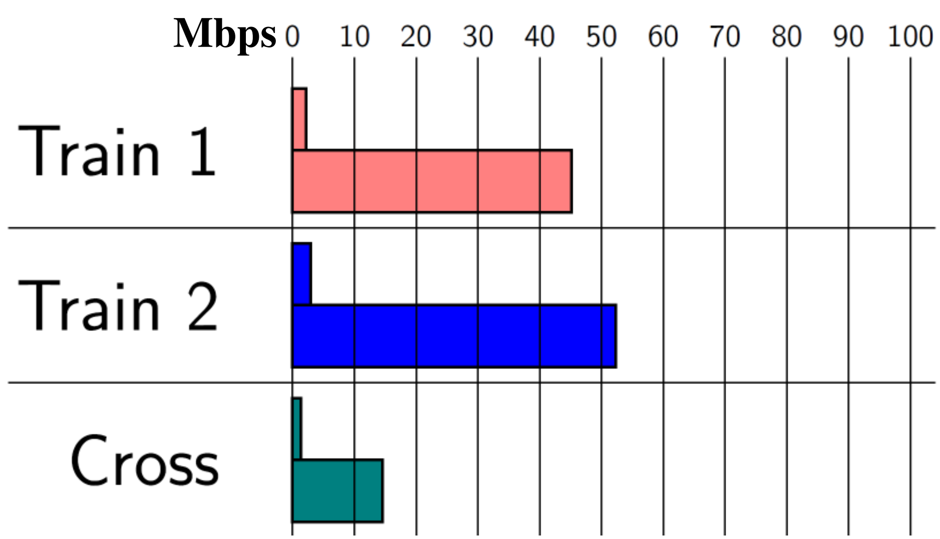
Improvement

Throughput optimization

Influence on the throughput

- Number of features (done)
- Number of support vectors (done)
- Data structures used inside the code (to-do)

Name	Before optimization	After optimization
Train1	2.2 *Mbps* 461 *msg/s*	45.1 *Mbps* 9 615 *msg/s*
Train 2	3.0 *Mbps* 633 *msg/s*	52.4 *Mbps* 11 162 *msg/s*
Cross	1.4 *Mbps* 374 *msg/s*	14.5 *Mbps* 3 904 *msg/s*



Conclusion and Outlooks

Conclusion

Anomaly detection for SIP messages based on

- Machine learning using SVM
- n-grams for feature extraction

Contribution: Significant improvement of sensitivity and detection

- Using keywords
  - As additional features
  - Removing of session information
  - Allow additional processing
- Introduction of multiple models

Throughput optimization

Outlook

- Definition of the training traces
- Simplify the expendability to any kind of SIP extensions
- Extend the detection method to other text based protocols