Communicating the results of pcap data analysis through common metadata format

Youki Kadobayashi
NICT (National Inst of Comm Tech)
/ NAIST (Nara Inst of Sci & Tech)
/ WIDE
youki-k <at> is.naist.jp

Anomaly detection algorithms: the problem

- We are still in the dark ages
  - Incompatible datasets
  - Incomparable results
- No technical method to accurately communicate the result of anomaly detection, even if we share the common dataset
- Inability to benchmark their performance

Metadata format and associated tools for anomaly detection algorithms

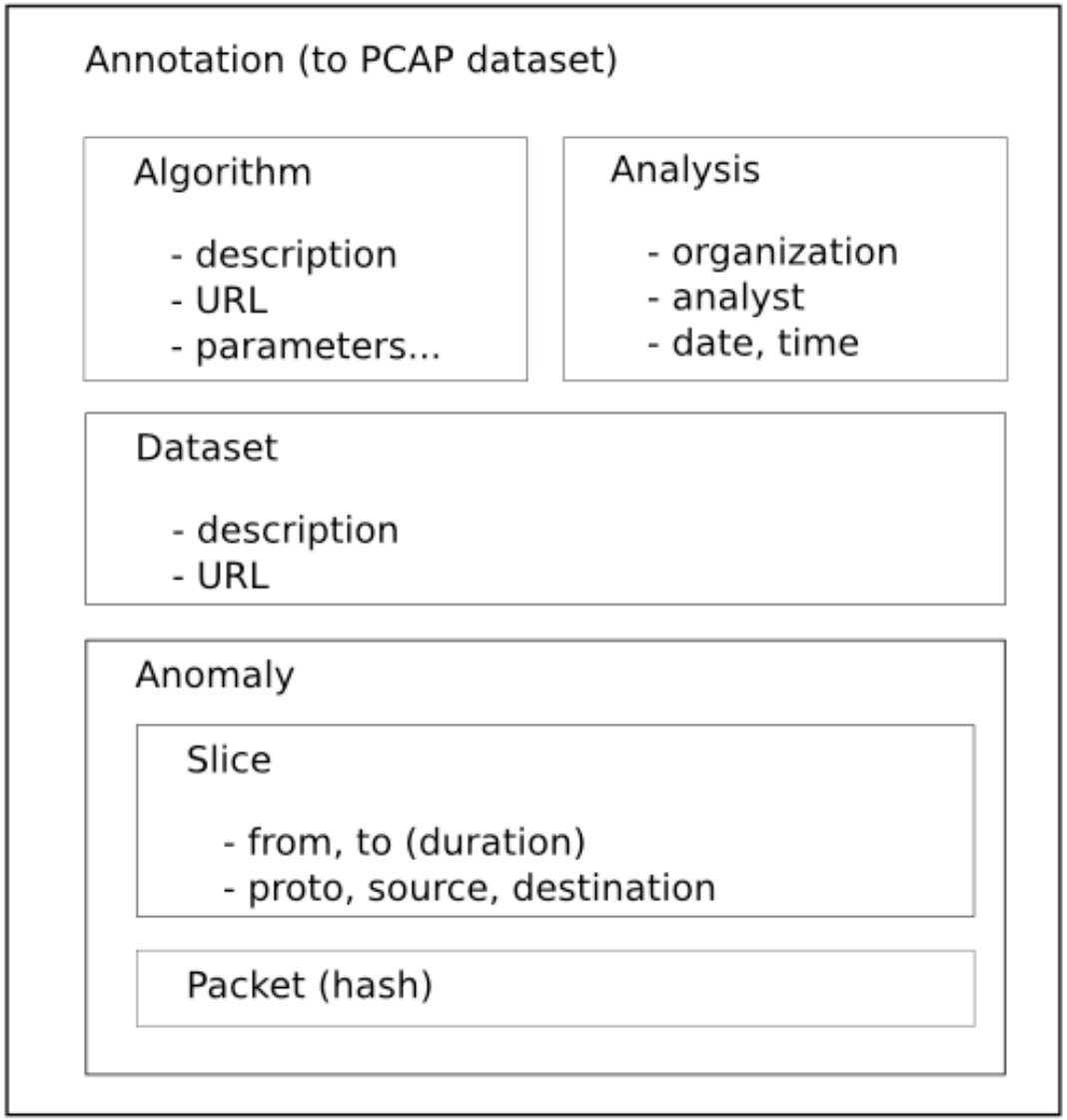
- Separate file for each algorithm
  - Idea: multiple results against single pcap dataset
- XML-based
  - header, {record1, record2, …}

- Envelope information: rely on DatCat tools

- C API, C++ API to annotate dataset
- Tools to slice, merge, or cross-validate dataset

Header

- Algorithm name
- Algorithm version
- Algorithm URL
- Algorithm parameters
- Date of analysis
- Analyst name
- Analyst organization
- Target dataset
- DatCat dataset name

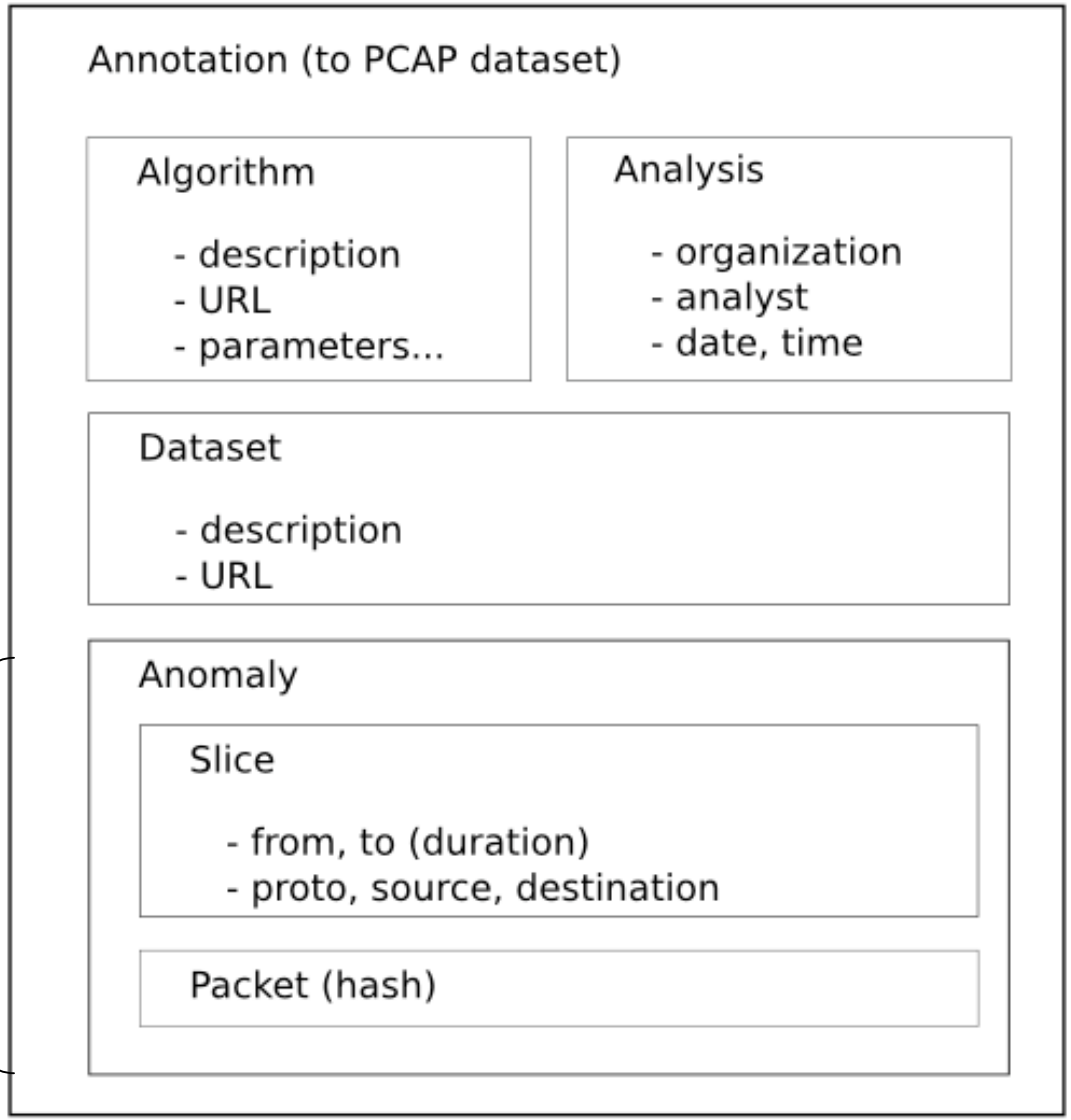


Header example

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<admd:annotation xmlns:admd="http://www.nict.go.jp/admd" xmlns:xsi="http://www.w
3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.nict.go.jp/admd ad
md.xsd">

 <algorithm name="algo_name" version="0.0.0">
  <description>algorithm detail</description>
  <url>http://example.com/example</url>
  <parameter>parameter for algorithm</parameter>
 </algorithm>

 <analysis>
  <description>analysis detail</description>
  <datetime>2000-01-01T12:00:00</datetime>
  <analyst>person name of analyst</analyst>
  <organization>organaization name of analyst</organization>
 </analysis>

 <dataset>
  <description>analysis detail</description>
  <url>http://example.com/example</url>
 </dataset>
```

Record

- Each record consists of:
  - src, dst,
    start_time, end_time,
    anomaly_type,
    anomaly_value
- Arbitrary # of records
- Either src or dst can be wildcard



Record example

```
<anomaly type="type" value="value">
  <description>first detected anomaly</description>
  <slice>
    <filter proto="udp" src_ip="163.221.8.12" src_port="53"/>
    <filter dst_ip="163.221.8.12" dst_port="53" proto="udp"/>
    <from sec="1" usec="1"/>
    <to sec="2" usec="2"/>
  </slice>
  <packet>RkZGRkZGRkZGRkZG</packet>
</anomaly>
```

C API

```
#include "admd/c_wrapper.h"

ANNOTATION_HANDLER annotation = admd_open_annotation();

admd_set_algorithm(annotation, "algo_name", "0.0.0", "algorithm detail",
                           "http://example.com/example", "parameter for algorithm");
admd_set_analysis(annotation, "analysis detail", "2000-01-01T12:00:00.000",
                           "person name of analyst", "organaization name of analyst");
admd_set_dataset(annotation, "analysis detail", "http://example.com/example");

ANOMALY_HANDLER first_anomaly = admd_add_anomaly(annotation,
                                                 "type", "value", "first detected anomaly");
SLICE_HANDLER slice = admd_add_slice(first_anomaly);
admd_add_filter(slice, "tcp", "ip1", 1, "", 0);  // proto, src_ip, src_port, dst_ip, dst_port
admd_add_filter(slice, "tcp", "", 0, "ip1", 1);
admd_set_from(slice, 1, 1);                      // sec, usec
admd_set_to(slice, 2, 2);

admd_print(annotation);
admd_close_annotation(annotation);
```

C++ API

```
#include "admd/admd.hpp"

admd::annotation_t annotation;

annotation.set_algorithm("algo_name", "0.0.0", "algorithm detail",
                                 "http://example.com/example", "parameter for algorithm")
    .set_analysis("analysis detail", "2000-01-01T12:00:00.000",
                      "person name of analyst", "organaization name of analyst")
    .set_dataset("analysis detail", "http://example.com/example");

admd::anomaly_t& first_anomaly = annotation.add_anomaly("type", "value",
                                          "first detected anomaly");   // type, value, description
first_anomaly.add_slice()
    .add_filter("tcp", "ip1", 1, "", 0)  // proto, src_ip, src_port, dst_ip, dst_port
    .add_filter("tcp", "", 0, "ip1", 1)
    .set_from(1, 1)                      // sec, usec
    .set_to(2, 2);

annotation.write(std::cout);
```

Slicing

- Slice anomalous segments of pcap data
  - Based on anomaly_type, anomaly_value
- Slice pcap data according to start_time, end_time
- Useful for generating synthetic dataset
- admd_slice [annotation xml file] [input pcap] [output pcap]

Merging

- Insert pcap slice B into pcap slice A
  - At particular time offset

- Useful for benchmarking anomaly detection algorithms with synthetic dataset

- admd_merge [annotation xml file] [base pcap] [merge pcap] [output pcap] [sec delay] [usec delay]

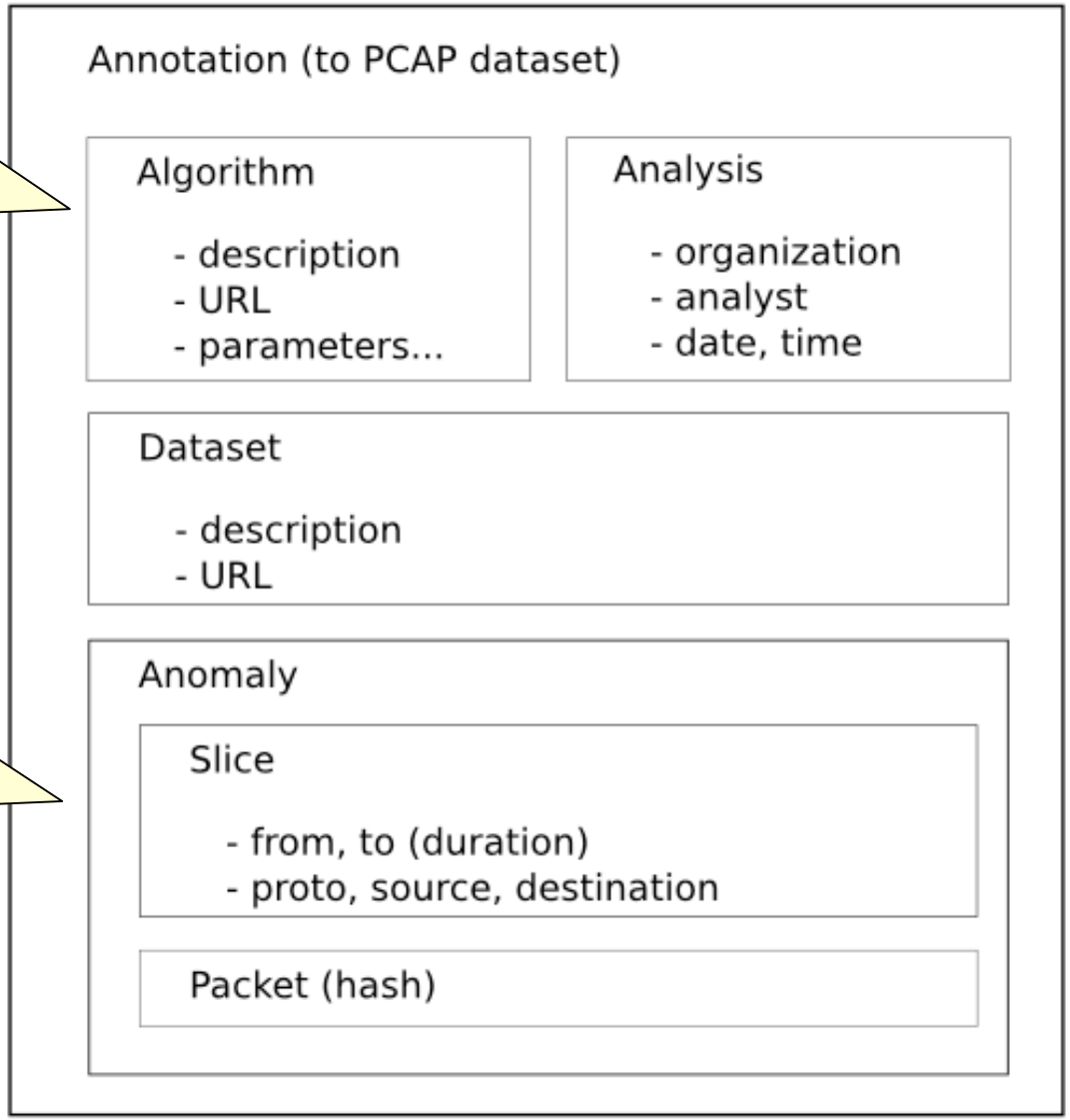
Comparison

- Visualize the spotted anomalies along timeline
- Compute TP/TN/FP/FN, tabulate results
- (no fancy html output yet)
- admd_validate [input dump file] [result1.xml result2.xml result3.xml …]
- admd_compare ...

Beyond XML schema: from xs:string to classification codes

- Algorithm is only described in string
- Canonical classification of algorithm?
- Canonical set of parameters?

- What kind of anomaly?
- Ontological representation?



Work has just begun

Current status

- Implementation available at
  - http://admd.sourceforge.net/
- Your feedbacks are welcomed
- youki-k <at> is.naist.jp