ON THE SEQUENTIAL PATTERN AND RULE MINING IN THE ANALYSIS OF CYBER SECURITY ALERTS

Thursday 31st August, 2017

Martin Husák
Jaroslav Kašpar
Elias Bou-Harb
Pavel Čeleda

CSIRT-MU

Motivation

Cyber Security Alerts

- Timely information about current security issues, e.g., events.
- Standardized outputs of intrusion detection.
- Important for information exchange.

Information Exchange

- Emerging topic of security research and practice.
- Collaborative security – alert sharing platforms.

CSIRT-MU

Motivation

Data Mining

- Current trend in cyber security (alongside machine learning).
- Can find concealed and indistinct patterns in the data.

Use Case

- Analysis of security alerts in the sharing platform.
- Discovery of common attack progression.
- Projection of attack continuation.

CSIRT-MU

Motivation

Sequence Mining

- Finds statistically relevant patterns between data where values are delivered in a **sequence**.
- Interesting choice for cyber security alert analysis
  - sequences of alerts correspond to **attack progression**.
- Sequential **pattern** mining finds frequent patterns only.
- Sequential **rule** mining finds also implications in sequences.

CSIRT-MU

Research Questions

Question I.

What are the use cases of sequence mining in the analysis of cyber security alerts?

Question II.

Which approaches are the most suitable and effective for mining sequences in security alerts?

Question III.

What are the effects of optimizations and data reductions?

Use Cases

Use Cases – Related Work

Alert correlation

- Frequent episode mining (4 papers),
- Association rule mining (4 papers),
- Sequential pattern mining (1 paper).

Attack prediction

- Association rule mining (3 papers),
- Continuous association rule mining (1 paper),
- Sequential pattern mining (1 paper).

Use Cases – Proposals

Related Work

- No consensus on which method to choose.
- Evaluation on data sets - a few experiments using real data.
- Association rule mining is the best–known approach.
- But is it actually suitable for cyber security use cases?

Alert Correlation

- Proposed approach – sequential **pattern** mining.

Attack Prediction

- Proposed approach – sequential **rule** mining.

CSIRT-MU

Experimental Evaluation

CSIRT-MU

Experiment Setup

Dataset

- 16 million alerts collected during 1 week.
- Collected in SABU alert sharing platform (mostly alerts from campus networks in Czech Republic).

Data mining methods

- 7 sequential pattern mining methods,
- 3 sequential rule mining methods (all implemented in SPMF library).

Example of an Alert


```
{
   "Format": "IDEA0",
   "ID": "3ad275e3-559a-45c0-8299-6807148ce157",
   "DetectTime": "2014-03-22T10:12:56Z",
   "Category": ["Recon.Scanning"],
   "ConnCount": 633,
   "Description": "Ping scan",
   "Source": [
      {
         "IP4": ["93.184.216.119"],
         "Proto": ["icmp"]
      }
   ],
   "Target": [
      {
         "Proto": ["icmp"],
         "IP4": ["93.184.216.0/24"],
         "Anonymised": true
      }
   ]
}
```


CSIRT-MU

Sequential Databases

Without port numbers

- Alerts with the same source and target (IP addresses),
- alerts with the same source (IP address),
- alerts with the same target (IP address).

With port numbers

- Alerts with the same source and target (IP addresses and ports),
- alerts with the same source (IP address and port),
- alerts with the same target (IP address and port).

CSIRT-MU

Method Selection

Approach	Algorithm(s)
Sequential pattern mining	CM-SPADE
Top-K sequential pattern mining	TKS
Closed sequential pattern mining	CM-ClaSP
Sequential generator pattern mining	VGEN
Maximal sequential pattern mining	VMSP
Compressing sequential pattern mining	GoKrimp
Sequential pattern mining with time constraints	HirateYamana
Closed sequential pattern mining with time constraints	Fournier08-Closed+time
Sequential rule mining	RuleGrowth
Sequential rule mining with window constraints	TRuleGrowth
Top-K sequential rule mining	TopKRules

Example Results

Frequent port combinations – sequential rules

```
Scan.1755  ==> Scan.1723 #SUP: 0.00025 #CONF: 0.69553
Scan.37777 ==> Scan.8000 #SUP: 0.00024 #CONF: 0.38748
Scan.1723  ==> Scan.1755 #SUP: 0.00023 #CONF: 0.35531
Scan.3392  ==> Scan.3391 #SUP: 0.00034 #CONF: 0.27006
Scan.3390  ==> Scan.3389 #SUP: 0.00024 #CONF: 0.10841
Scan.443   ==> Scan.80   #SUP: 0.00080 #CONF: 0.09309
Scan.80    ==> Scan.443  #SUP: 0.00066 #CONF: 0.02521
Scan.3389  ==> Scan.3390 #SUP: 0.00039 #CONF: 0.02226
Scan.2323  ==> Scan.23   #SUP: 0.00210 #CONF: 0.02031
Scan.23    ==> Scan.2323 #SUP: 0.00322 #CONF: 0.00461
```


CSIRT-MU

Result Samples

Scanned port groups

- Some groups of ports are typically scanned simultaneously.

```
(Scan.922, Scan.674) ==> Scan.930 #SUP: 0.02075 #CONF: 0.53690
(Scan.922, Scan.666) ==> Scan.930 #SUP: 0.02003 #CONF: 0.53096
```

Results

Method	Sources and Targets without ports	Sources and Targets with ports	Sources only without ports	Sources only with ports	Targets only without ports	Targets only with ports
Sequential pattern mining	16 min, 100 %	<1 min, 1 %	2 min, 100 %	<1 min, 5 %	✖	✖
Top-K sequential pattern mining	<1 min, 100 %	<1 min, 10 %	<1 min, 100 %	<1 min, 10 %	✖	✖
Closed seq. pattern mining	3 min, 100 %	2 min, 20 %	2 min, 100 %	2 min, 50 %	2 min, 5 %	✖
Seq. generator pattern mining	<1 min, 100 %	<1 min, 10 %	<1 min, 100 %	<1 min, 10 %	6 min, 60 %	✖
Maximal seq. pattern mining	<1 min, 100 %	<1 min, 10 %	<1 min, 100 %	<1 min, 10 %	4 min, 60 %	✖
Compressing seq. pattern mining	15 min, 100 %	3 min, 1 %	18 min, 10 %	4 min, 1 %	<1 min, 1 %	✖
Sequential pattern mining with time constraints	5 min, 100 %	6 min, 100 %	16 min, 100 %	11 min, 100 %	<1 min, 100 %	✖
Closed seq. pattern mining with time constraints	11 min, 100 %	11 min, 100 %	57 min, 100 %	34 min, 100 %	2 min, 100 %	✖
Sequential rule mining	1 min, 100 %	3 min, 100 %	<1 min, 100 %	<1 min, 100 %	<1 min, 100 %	✖
Sequential rule mining with window constraints	2 min, 100 %	4 min, 100 %	1 min, 100 %	1 min, 100 %	<1 min, 100 %	✖
Top-K sequential rule mining	1 min, 100 %	3 min, 100 %	<1 min, 100 %	<1 min, 100 %	<1 min, 100 %	✖

* Intel Xeon E5520, 8 threads, 16 GB RAM

CSIRT-MU

Lessons Learned

CSIRT-MU

Lessons Learned

Use cases

- Sequential **pattern** mining is suitable for **alert correlation**,
- more comprehensive results than association rule mining and frequent episode mining.
- Sequential **rule** mining is suitable for **attack prediction**,
- confidence value can be directly used for predictions.

CSIRT-MU

Lessons Learned

Performance

- Most methods show similar performance.
- Rule mining is faster than pattern mining.
- Feature selection makes the biggest difference.
- Beware of too long sequences.
- Positive impact of optimization on performance (also on soundness of results).

Lessons Learned

Soundness of the results

- **Source–target** interactions are interesting, but provide less patterns and rules than expected.
- Sequences with the same **source** are useful as they reflect attack progression.
- Sequences with the same **target** are hard to process and the results are not worth it.
- Including ports in the features is definitely useful.

Lessons Learned

Method extensions

- Item intervals provide valuable information about attack timing (for the cost of computation overhead).

Effects of optimizations

- Optimization influence performance as well as result soundness,
- maximal sequential pattern mining filters the results the most (pattern that are subsets of other patterns are discarded).

CSIRT-MU

Conclusion and Future Work

Conclusion

- 2 use cases considered – alert correlation and attack prediction,
- 11 sequence mining methods were evaluated in an experiment,
- lessons learned were gathered and summarized in the paper,
- source codes available at:
  `https://github.com/CSIRT-MU/SecAlertSeqMining`

Future Work

- Practical utilization of results – development of data mining component for SABU alert sharing platform.
- Detailed study of actual attack sequences from real world.

THANK YOU FOR YOUR ATTENTION!

csirt.muni.cz

@csirtmu

Martin Husák

husakm@ics.muni.cz