Process Mining to enhance security of Web information systems

Simona Bernardi, Raúl Piracés Alastuey, and Raquel Trillo Lado

Paris, 29th April 2017

Outline

- Context

Outline

- Context
- Method overview

Outline

- Context
- Method overview
- Model-driven approach to get a normative model

Outline

- Context
- Method overview
- Model-driven approach to get a normative model
- Monitoring & log pre-processing to get *event logs*

Outline

- Context
- Method overview
- Model-driven approach to get a normative model
- Monitoring & log pre-processing to get *event logs*
- Process mining to detect deviations

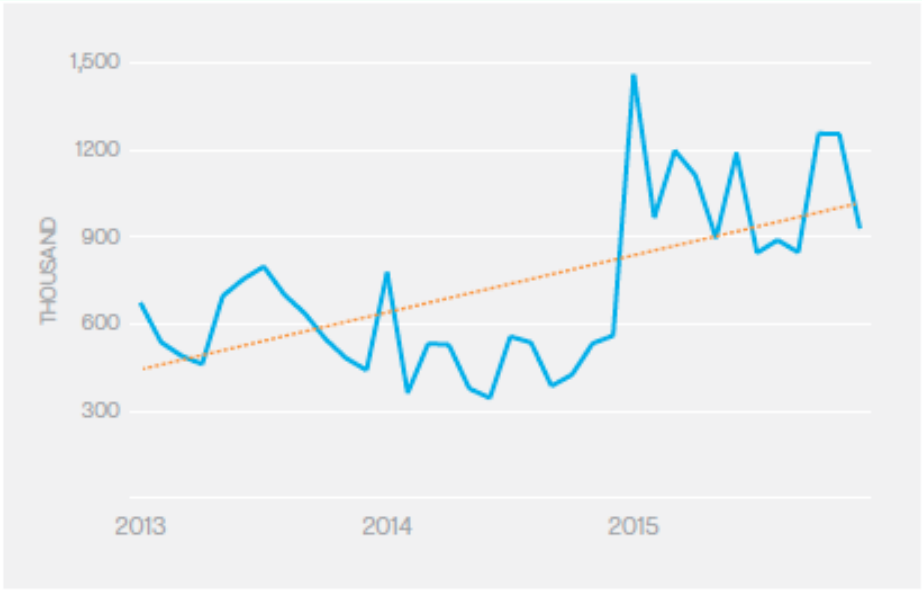
Outline

- Context
- Method overview
- Model-driven approach to get a normative model
- Monitoring & log pre-processing to get *event logs*
- Process mining to detect deviations
- Conclusion

Context: Web threats

- Wide use of Web applications in the last decade
- The number of Web attacks doubled in 2015

Attacks blocked per day



Symantec Reports, April 2016-17

By Category of Websites

Rank	Domain Categories	2015 (%)	2016 (%)	Percentage Point Difference
1	Technology	23.2	20.7	-2.5
2	Business	8.1	11.3	3.2
3	Blogging	7.0	8.6	1.6
4	Hosting	0.6	7.2	6.6
5	Health	1.9	5.7	3.8
6	Shopping	2.4	4.2	1.8
7	Educational	4.0	4.1	< 0.1
8	Entertainment	2.6	4.0	1.4
9	Travel	1.5	3.6	2.1
10	Gambling	0.6	2.8	2.2

Context: Intrusion detection approaches

- Preventive approaches
  - Establish rules to reduce or remove the success conditions of an attack
  - Detection based on known signature of attack

Context: Intrusion detection approaches

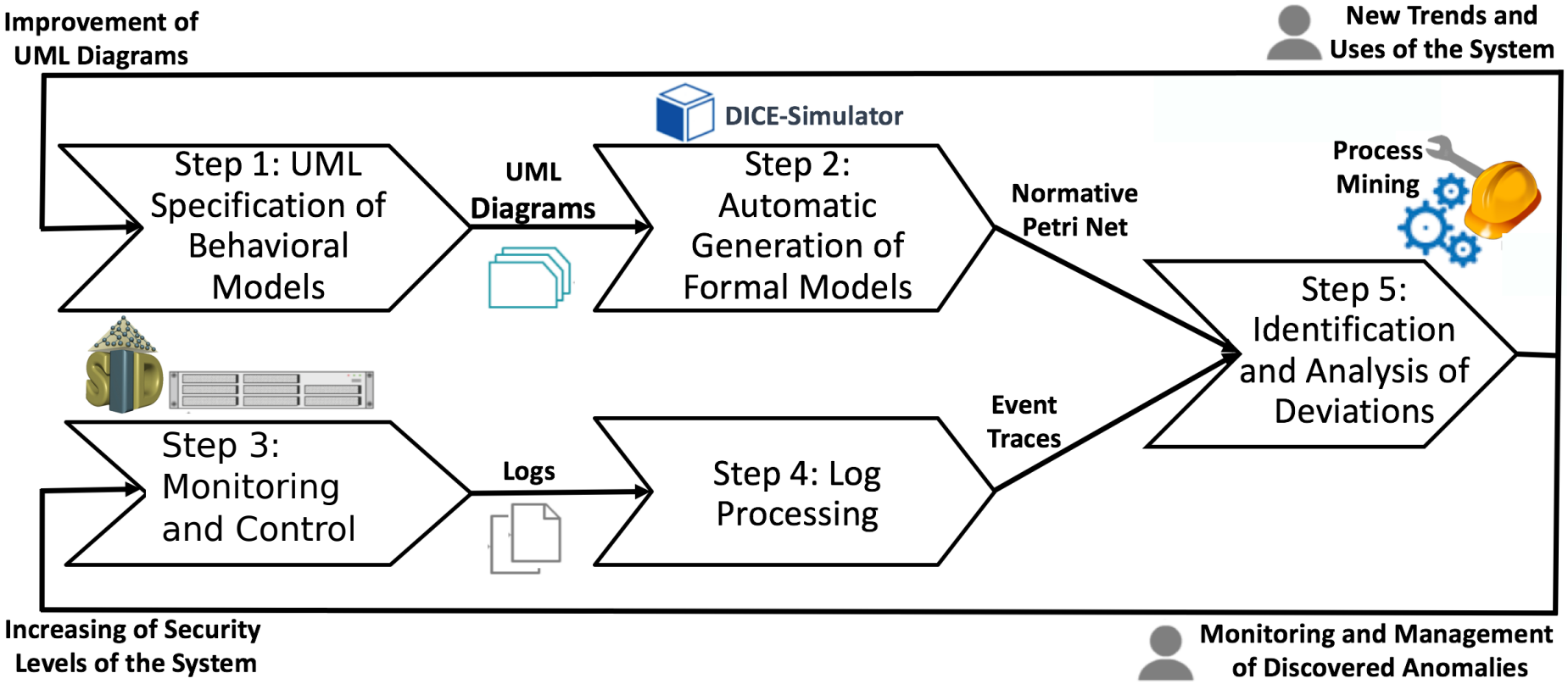
- Preventive approaches
  - Establish rules to reduce or remove the success conditions of an attack
  - Detection based on known signature of attack
- Reactive approaches
  - Based on anomaly detection
  - More suitable than the former to detect zero-days attacks
  - Generation of false positive/negative

Context: Intrusion detection approaches

- Preventive approaches
  - Establish rules to reduce or remove the success conditions of an attack
  - Detection based on known signature of attack
- Reactive approaches
  - Based on anomaly detection
  - More suitable than the former to detect zero-days attacks
  - Generation of false positive/negative
- Mixed preventive-reactive

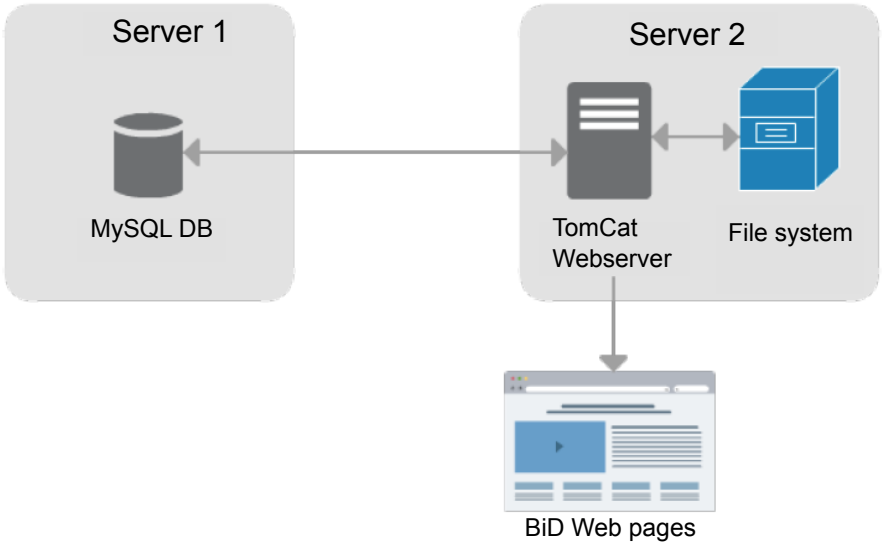
Method overview

- Mixed intrusion detection approach
- Use of model-driven and process mining techniques

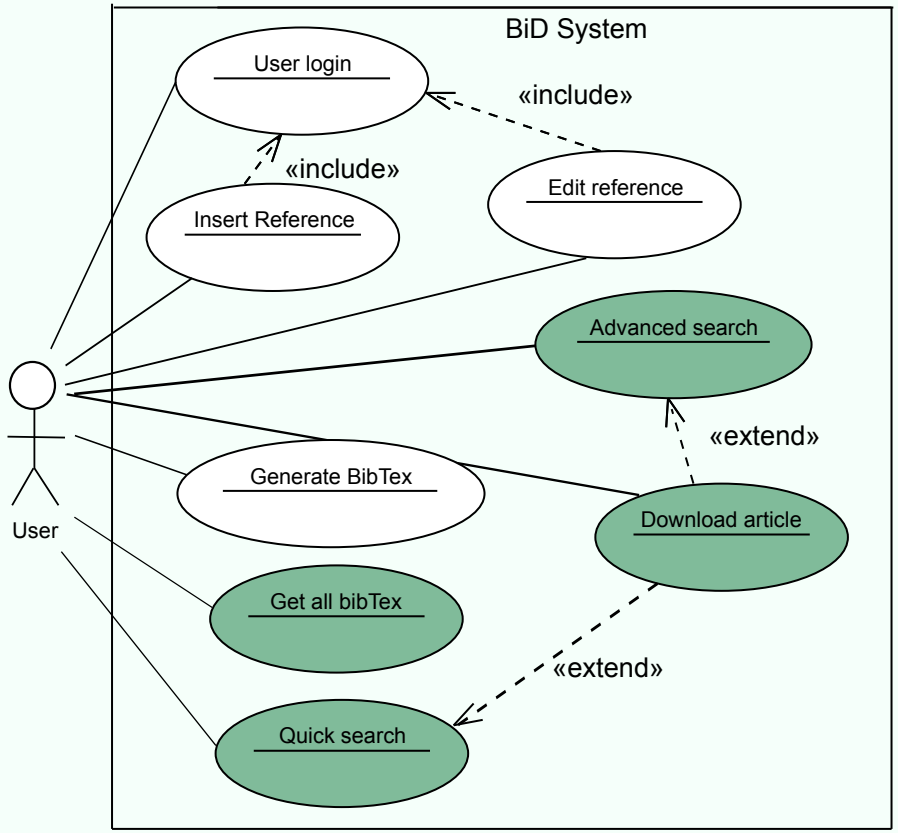


Method overview: The BiD case study

Architecture



Main functionalities



Steps 1&2: Getting a normative model - I

- Model-driven approach to get a Petri net model from UML-based specification

Steps 1&2: Getting a normative model - I

- Model-driven approach to get a Petri net model from UML-based specification
- UML behavioral diagrams
  - Activity/sequence that model (some) use case scenarios
  - Represent a system partial view
  - Obtained from the design or reverse engineering

Steps 1&2: Getting a normative model - I

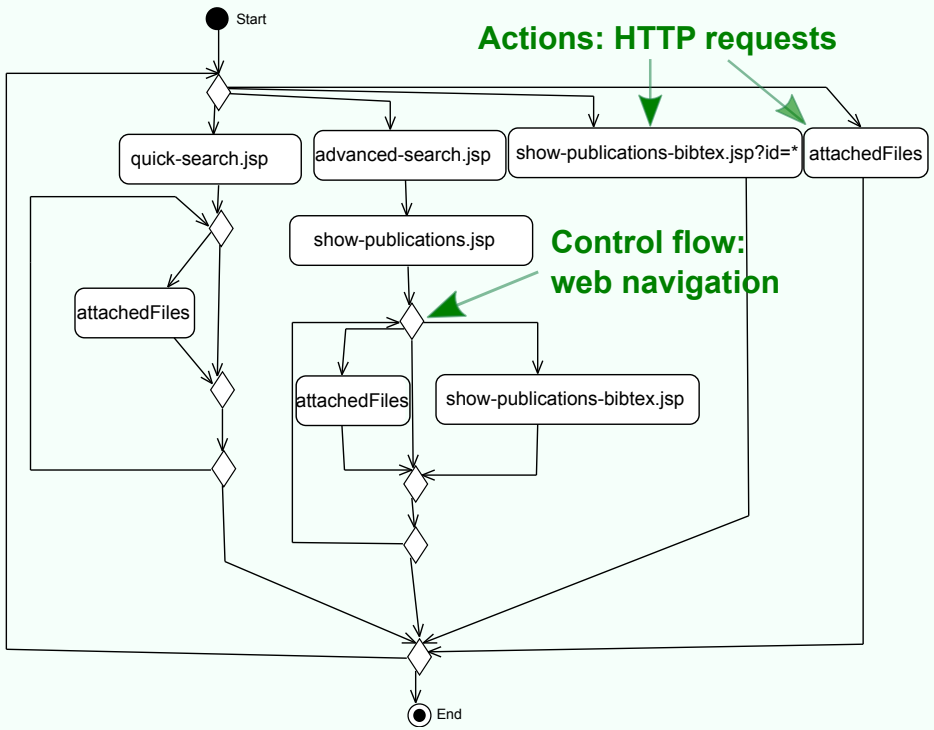
- Model-driven approach to get a Petri net model from UML-based specification
- UML behavioral diagrams
  - Activity/sequence that model (some) use case scenarios
  - Represent a system partial view
  - Obtained from the design or reverse engineering
- Model-to-model transformation (from literature)

Steps 1&2: Getting a normative model - I

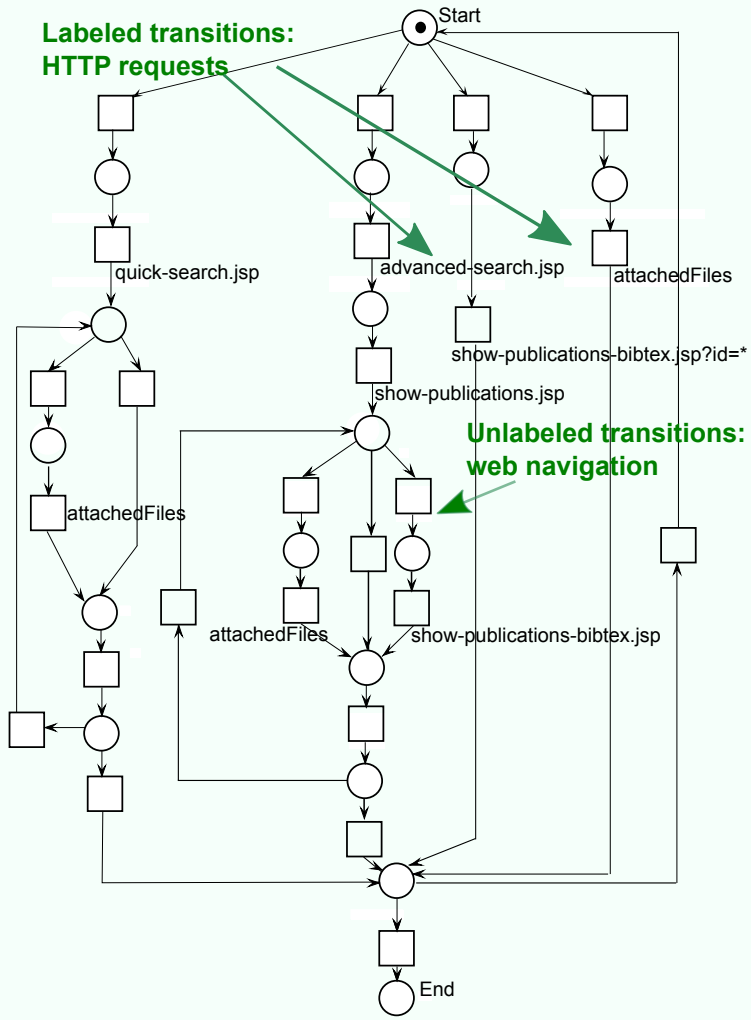
- Model-driven approach to get a Petri net model from UML-based specification
- UML behavioral diagrams
  - Activity/sequence that model (some) use case scenarios
  - Represent a system partial view
  - Obtained from the design or reverse engineering
- Model-to-model transformation (from literature)
- Generated Petri net model
  - Untimed
  - With **labeled** transitions that correspond to actions/events of the source UML model
  - **Normative** model

Steps 1&2: Getting a normative model - II

BiD behavioral view



BiD normative model



Steps 3&4: Getting event logs - I

- Monitoring the application (Web server side)

BiD: Apache Tomcat logs

Pattern	Event example
User Session ID (%S)	F81AE1A0E23438DE2B294D42ED34B74E
Host (%h)	10.0.2.2
User Auth (%u)	-
Date & Time (%t)	07/03/2016 21:21:03
Request (%r)	"GET /PUBLICATIONS/library-show-publications-bibtex.jsp?id=jws15-android HTTP/1.1"
HTTPStatusCode (%s)	200
Bytes Sent (-headers) (%b)	1615
Request Method (%m)	GET
Query String (%q)	?id=jws15-android
Referer (header) %{Referer}i	"http://localhost:8080/PUBLICATIONS/library-show-publications.jsp?advanced=true&keywords=Any&entrytypes=Any&otherKeywords=&after=&before=&conditionAuthors=and&authors=Eduardo+Mena&authors=Fernando+Bobillo&authors=&otherAuthors=&idBdi=idbdi+%3C%3E+%27NULL%27+&not1=&valueField1=&fieldConstraint1=Any+Field&joinConstraint1=AND&not2=&valueField2=&fieldConstraint2=Any+Field&joinConstraint2=AND&not3=&valueField3=&fieldConstraint3=Any+Field&presentation=customized&configuration=inverse+chronological+list&order0=year&asc0=DESC&order1=month&asc1=DESC&order2=&asc2=ASC"
User Agent %{User}i	"Mozilla/5.0 (Macintosh Intel Mac OS X 10_11_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/48.0.2564.116 Safari/537.36"

Steps 3&4: Getting event logs - II

- Log pre-processing

Event logs (in process mining)

- Each event is characterized by at least a case, an activity and a timestamp
- A case is a trace of related events/activities
- Events are ordered according to their timestamps

Steps 3&4: Getting event logs - II

- Log pre-processing

Event logs (in process mining)

- Each event is characterized by at least a case, an activity and a timestamp
- A case is a trace of related events/activities
- Events are ordered according to their timestamps

Apache Tomcat logs

- Timestamp $\rightarrow$ Date&Time,
- Activity $\rightarrow$ HTTP Request
- Case $\rightarrow$?? . . . different possibilities

Steps 3&4: Getting event logs - III

- How to get case identifiers ??

Steps 3&4: Getting event logs - III

- How to get case identifiers ??
- We aim at observing the behaviour of the user
  - in a work session, and
  - by considering which use cases are executed

Steps 3&4: Getting event logs - III

- How to get case identifiers ??
- We aim at observing the behaviour of the user
  - in a work session, and
  - by considering which use cases are executed
- Use of heuristics to identify
  - The user session (if the session ID field is not available)
  - The use cases executed within a session

Steps 3&4: Getting event logs - III

- How to get case identifiers ??
- We aim at observing the behaviour of the user
  - in a work session, and
  - by considering which use cases are executed
- Use of heuristics to identify
  - The user session (if the session ID field is not available)
  - The use cases executed within a session

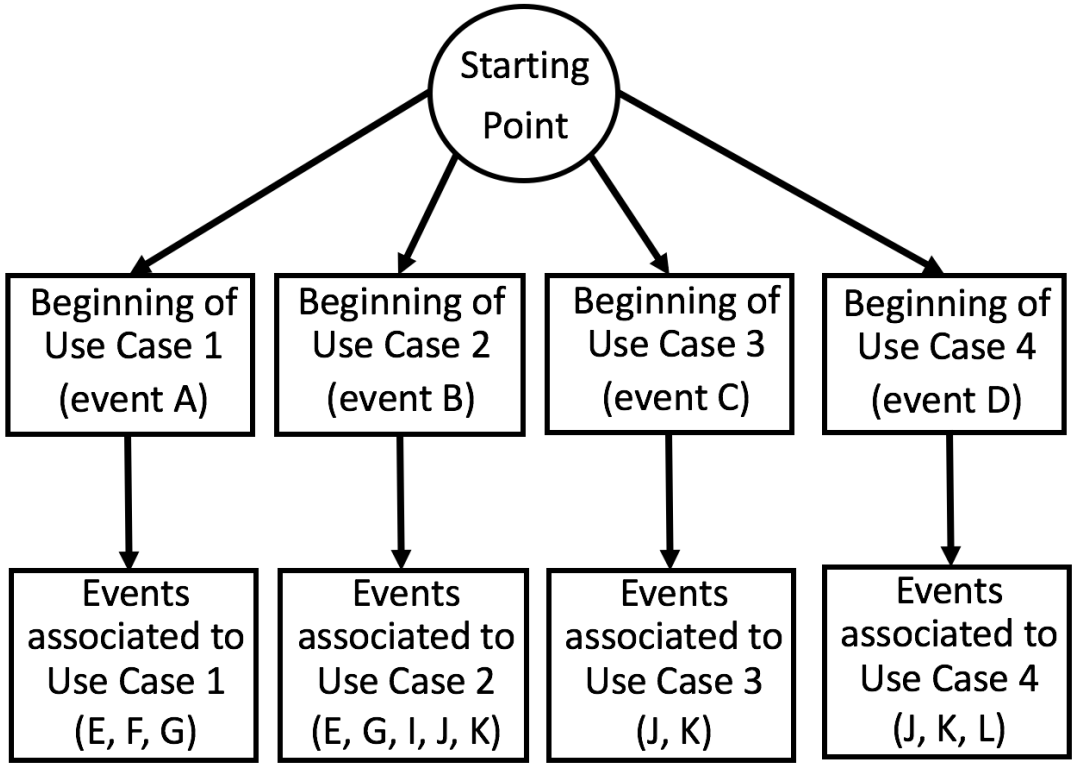
Case ID: user session

- Same IP address
- Elapsed time between two entries $< 30min$
- Events related to the same user and same browser

Steps 3&4: Getting event logs - IV

- Case ID: user session + use case executed

H1: Entry point



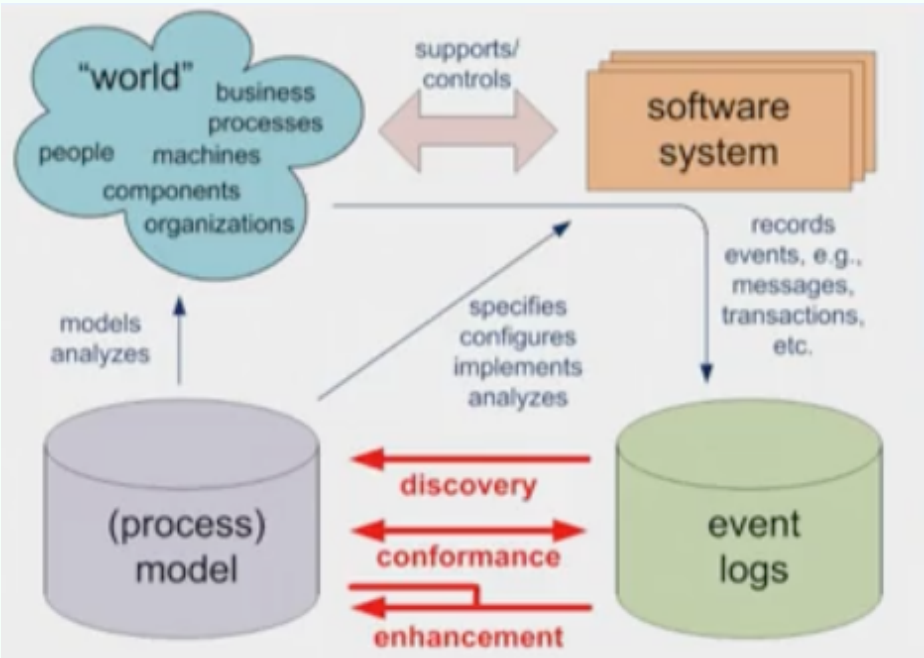
Event Log of a user session

Events	Possible Tags	Final Tags
A	1	1
E	1, 2	1
F	1	1
C	3	3
J	2, 3, 4	3
K	2, 3, 4	3
D	4	4
J	2, 3, 4	4
K	2, 3, 4	4
L	4	4

Step 5: Detection of deviations

- Application of process mining

PM techniques



Van der Aalst, Process Mining, Springer 2016

- To identify deviations with respect to the known behaviour
  - Replay event logs on the normative Petri Net model
- To detect attacks and discover new usage scenarios
  - Discovery fuzzy nets from event logs

Step 5: Identification of deviations

Replay event logs on the normative PN model

- Based on the optimal alignment
- Mapping of events and labeled transitions
- Cost assignment to asynchronous movements

Step 5: Identification of deviations

Replay event logs on the normative PN model

- Based on the optimal alignment
- Mapping of events and labeled transitions
- Cost assignment to asynchronous movements

BiD results (April 2015, 2423 traces, 4805 events

- Fitness: $\approx 58\%$
- New behavior
  - 2 traces (14 uploadify.swf events)
  - 4 traces (6-13 FCKEditor events)
- Fully aligned with the PN but suspicious
  - 4 traces (the longest with 123 attachedFile events)

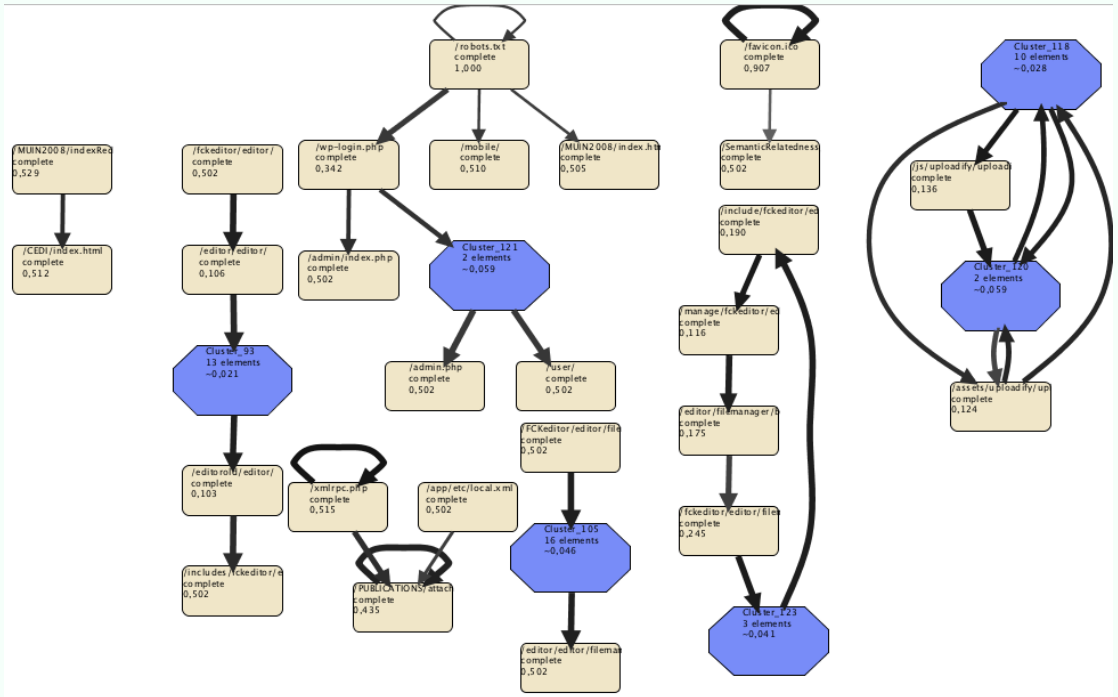
Step 5: Attack detection I

Discovery of fuzzy nets from filtered event logs

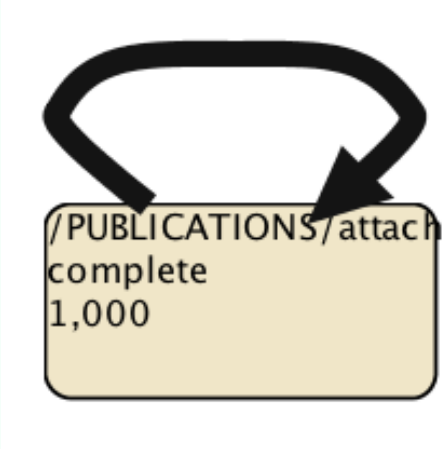
- Filtering logs by HTTP status code
- Fuzzy nets: Node $\Leftrightarrow$ Event, Arc $\Leftrightarrow$ Cause-effect
- Associated metrics:
  - importance:
    - frequencies (nodes/arcs)
  - correlation (arcs):
    - time proximity

Step 5: Attack detection II

Not found resource 404



Partial delivery 206



- 404: Attempt to find vulnerabilities
- 206: Denial of Service attack attempts

Conclusion

- Several attack patterns identified in the experiments
  - Cross-site scripting (XSS) attempts
  - Brute force attacks to get access passwords
  - Attempt to edit publications by unauthorized users

Conclusion

- Several attack patterns identified in the experiments
  - Cross-site scripting (XSS) attempts
  - Brute force attacks to get access passwords
  - Attempt to edit publications by unauthorized users
- . . . also new behavioral patterns (from logs filtered by 200 code).

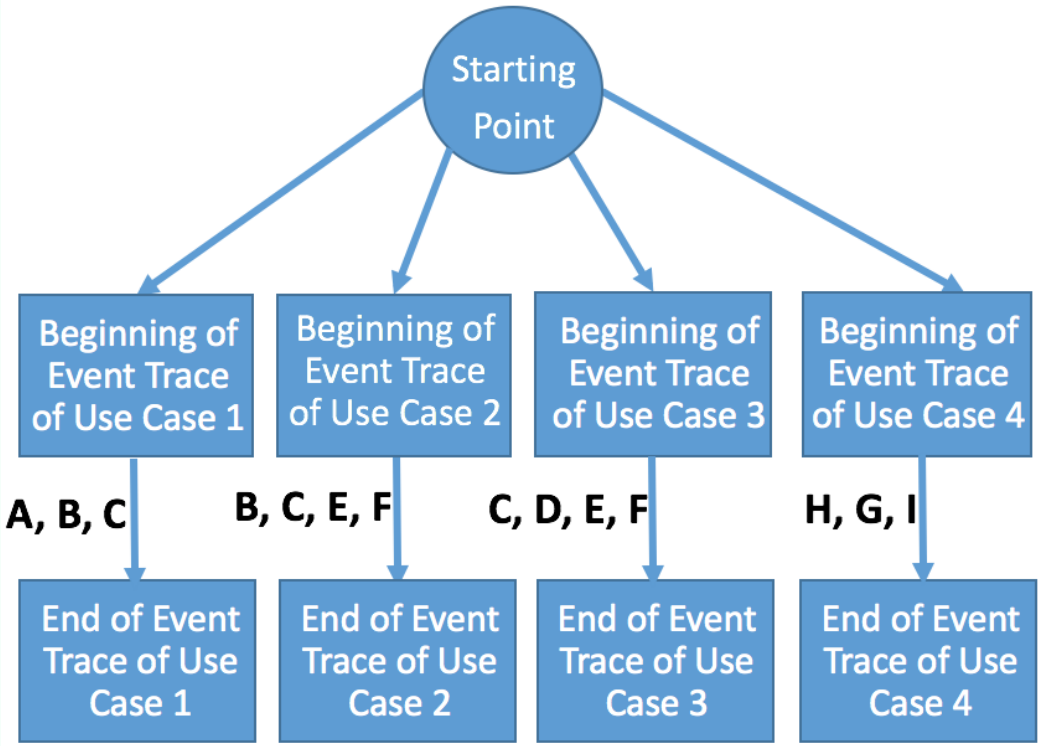
Conclusion

- Several attack patterns identified in the experiments
  - Cross-site scripting (XSS) attempts
  - Brute force attacks to get access passwords
  - Attempt to edit publications by unauthorized users
- . . . also new behavioral patterns (from logs filtered by 200 code).
- Open issues
  - Full automation of the method to be used on-line
  - Inclusion of new rules in a SIEM (such as Zabbix) that detect & mitigate the discovered attack patterns.
  - Application of the method to other case studies

Steps 3&4: Getting event logs - V

- Case ID: user session + use case executed

H2: Longest path of consecutive events



Event Log of a user session

Events	Possible Tags	Final Tags
A	1	1
B	1, 2	1
C	1, 2, 3	3
D	3	3
E	3, 2	3
F	3, 2	3
A	1	1
B	1, 2	1
C	1, 2, 3	1