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

By Category of Websites

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

Symantec Reports, April 2016-17
Preventive approaches

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

Server 1
MySQL DB

Server 2
TomCat
Webserver
File system

BiD Web pages

Main functionalities

User

User login
Insert Reference
Generate BibTex
Get all bibTex
Quick search
Download article

Advanced search
Edit reference

BiD System

«include»

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

Actions: HTTP requests

Control flow: web navigation

BiD normative model

Labeled transitions: HTTP requests

Unlabeled transitions: web navigation
Steps 3&4: Getting event logs - 1

- Monitoring the application (Web server side)

### BiD: Apache Tomcat logs

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Event example</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Session ID (%S)</td>
<td>F81AE1A0E23438DE2B294D42ED34B74E</td>
</tr>
<tr>
<td>Host (%h)</td>
<td>10.0.2.2</td>
</tr>
<tr>
<td>User Auth (%u)</td>
<td>-</td>
</tr>
<tr>
<td>Date &amp; Time (%t)</td>
<td>07/03/2016 21:21:03</td>
</tr>
<tr>
<td>Request (%r)</td>
<td>&quot;GET /PUBLICATIONS/library-show-publications-bibtex.jsp?id=jws15-android HTTP/1.1&quot;</td>
</tr>
<tr>
<td>HTTPStatusCode (%s)</td>
<td>200</td>
</tr>
<tr>
<td>Bytes Sent (-headers) (%b)</td>
<td>1615</td>
</tr>
<tr>
<td>Request Method (%m)</td>
<td>GET</td>
</tr>
<tr>
<td>Query String (%q)</td>
<td>?id=jws15-android</td>
</tr>
<tr>
<td>Referer (header) %[Referer]</td>
<td>&quot;<a href="http://localhost:8080/PUBLICATIONS/library-show-publications.jsp?advanced=true&amp;keywords=Any&amp;entrytypes=Any&amp;otherKeywords=&amp;after=&amp;before=&amp;condition">http://localhost:8080/PUBLICATIONS/library-show-publications.jsp?advanced=true&amp;keywords=Any&amp;entrytypes=Any&amp;otherKeywords=&amp;after=&amp;before=&amp;condition</a> Authors=and&amp;authors=Eduardo&amp;Mena&amp;authors=Fernando&amp;Bobillo&amp;authors=&amp;otherAuthors=&amp;idBdi=idbdi+%3C%3E+%27NULL+%27+%not1=&amp;valueField1=&amp;fieldConstraint1=Any+Field&amp;joinConstraint1=AND&amp;not2=&amp;valueField2=&amp;fieldConstraint2=Any+Field&amp;joinConstraint2=AND&amp;not3=&amp;valueField3=&amp;fieldConstraint 3=Any+Field&amp;presentation=customized&amp;configuration=inverse+chronological+list&amp;order0=year&amp;asc0=DESC&amp;order1=month&amp;asc1=DESC&amp;order2=&amp;asc2=ASC&quot;</td>
</tr>
<tr>
<td>User Agent %[User]</td>
<td>&quot;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&quot;</td>
</tr>
</tbody>
</table>
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**

**Examples**

- **Apache Tomcat logs**
  - **Timestamp** → Date&Time
  - **Activity** → HTTP Request
  - **Case** → ?? . . . different possibilities
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 < 30 min
- 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:

<table>
<thead>
<tr>
<th>Events</th>
<th>Possible Tags</th>
<th>Final Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>1, 2</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>J</td>
<td>2, 3, 4</td>
<td>3</td>
</tr>
<tr>
<td>K</td>
<td>2, 3, 4</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>J</td>
<td>2, 3, 4</td>
<td>4</td>
</tr>
<tr>
<td>K</td>
<td>2, 3, 4</td>
<td>4</td>
</tr>
<tr>
<td>L</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>
Step 5: Detection of deviations

- Application of process mining

PM techniques

- 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

Van der Aalst, Process Mining, Springer 2016
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

- **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

<table>
<thead>
<tr>
<th>Events</th>
<th>Possible Tags</th>
<th>Final Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>1, 2</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>1, 2, 3</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>E</td>
<td>3, 2</td>
<td>3</td>
</tr>
<tr>
<td>F</td>
<td>3, 2</td>
<td>3</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>1, 2</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>1, 2, 3</td>
<td>1</td>
</tr>
</tbody>
</table>