---------- REVIEW 1 ----------
PAPER: 3
TITLE: Aspectizing JavaScript Security
AUTHORS: Florent Marchand de Kerchove, Jacques Noyé and Mario Südholt

---------- REVIEW ----------
Short Summary
****

This paper presents an overview an different approaches to secure web applications that are written in JavaScript. It also discusses opportunies for using aspect-oriented concepts in this domain.

Pros and Cons
****

The paper is a well-structured overview on approaches to secure JavaScript applications. Unfortunately, this is not my domain of expertise. Therefore, I can hardly tell whether the numerous cited papers are presented correctly.

The discussion on how aspects could be improve the existing approaches is quite vague. No concrete examples are given.

Typical phrases throughout the paper are "Aspects can be used to ..." and "Aspects are also well-suited to ...". This might be true, but, anyway, it is not evident what problem we have this existing (not aspect-based) approaches. A solid problem discuss would improve the readers motivation a lot.

The authors claim that due to their structured domain analysis, they found new applications areas for aspects. Therefore, it would be worth a discussion during the workshop.

Details
****

page 3: "that a access" -> "that access"

page 4: "the a system" -> "the system"

"Discussion": "our study ... has show taht aspects are generally useful ..." -> I would phrase that more carefully.

---------- REVIEW 2 ----------
PAPER: 3
TITLE: Aspectizing JavaScript Security
AUTHORS: Florent Marchand de Kerchove, Jacques Noyé and Mario Südholt

---------- REVIEW ----------
Summary:
The paper is about using AOP for securing JavaScript (JS) applications. Three security areas are examined: fine-grained access control, capabilities-based security, and information flow. The first area is already tackled by aspects, and the existing AO approaches are

surveyed. For the two other areas, AO solutions are proposed.

Evaluation:
The subject of the paper is interesting and and it opens new directions for AOP. However my main concern is that the actual AOP solutions that are proposed are too general. The recommendation is to accept the paper after revising the proposed solutions, adding more details and code examples.

Detailed:
- The connection between JS and AOP is very interesting. I would like to here more on existing AOP extensions for JS. How developed is the field?
- It is stated that "We provide evidence that aspects are useful...", what kind of such evidence is provided?
- "Our primary goal is the expression of security policies for web application using aspect-oriented programming". The paper should be revised so that more is said about the AOP policies themselves.
- There is not even a single code snippet showing how AOP make JS more secure. It is particularly interesting since usually the code examples are from Java and AspectJ.
- Section 2: Fine-grained access control
  - Which are the AOP technologies that ConScript and WebJail are based on?
  - It is mentioned that ZAC is based on AspectScript "a library that rewrites JavaScript code", what does it mean to rewrite JS code?.
- Section 3: Capabilities-based security
  - The nature of the capabilities mechanism is not clear from the intro of section 3. E.g., what does "isolation properties" mean?
  - It is explained how aspects may turn JS into a capability-safe language and the explanation is very lacking: it should be more detailed including some code examples.
  - It is mentioned that aspects are "well-suited to support security properties that are defined based on capabilities". Also here, the point is not clear and should be better clarified. To conclude, this section should be improved to (1) better explain the concept of capabilities, what does it mean for a language to be capability-safe? (2) explain in more detail how aspects can make JS a capability-safe language, what does it mean "to support security properties that are defined based on capabilities", and how aspects may help with that.
- Section 4: Information Flow
  - The introduction of section 4 has too many new terms, which are difficult to grasp for someone who is not familiar with this area. Some concrete examples are needed.

- A typo: "Capabilities are a language-level mechanism that cannot be forged and directly serve as a proxy and mediate any access to resources" => add s to serve and to mediate.

---------- REVIEW 3 ----------
PAPER: 3
TITLE: Aspectizing JavaScript Security
AUTHORS: Florent Marchand de Kerchove, Jacques Noyé and Mario Südholt

---------- REVIEW ----------
This work presents a review of major techniques for securing JavaScript applications. The author discuss the limitations of some

existing approaches take advantage of AOP, and show how other approaches may benefit from using AOP. The conclusions of this work are that no single approach is sufficient, but rather that a portfolio of techniques need to be considered, and aspects are a good candidate to be considered as a technique within of any comprehensive strategy for secure software systems.

The paper is well organized and well motivated, and the background support is suitably chosen and helps drive the argument. The scope of the work fits well within the goals of the MISS workshop and I believe that the review presented within will be of interest to the majority of workshop attendees.

Minor issue:
p.3, col.1, par.2: "a access" --> "access"

---------- REVIEW 4 ----------
PAPER: 3
TITLE: Aspectizing JavaScript Security
AUTHORS: Florent Marchand de Kerchove, Jacques Noyé and Mario Südholt

---------- REVIEW ----------
The paper discusses how AOP can be used to address security concerns in Javascript. For different categories of well-know security concerns, that
they start by describing, they discuss how AOP could be used. They also discuss previous work in the area and their limitations.

In general, the paper is well written and it is easy to follow, even for readers that are not experts in the area of Javascript security, as some background is provided in the paper (considering the size limits, it seems the
right amount of background).
The topic is also interesting, and pertinent, due to the proliferation of web applications that make extensive use of Javascript.

In the Discussion section, you briefly enumerate the characteristics of the
framework you envision. My main criticism to the paper is that few details
are provided there. I understand that this is a position paper, but I would
expect some prototype implementation, or something else that allowed me to
have a better idea of the challenges that implementing the solution will raise. I also think that describing the framework/language you expect to implement at the beginning, and during the description of the different categories of security concerns explain how your framework would be helpful,
would make the paper better.

Two of the challenges expected are discussed.
Regarding performance, I think that currently is widely accepted that portability is essential in web applications, therefore browser specific solutions does not seem to be an acceptable option.

However, I believe the main problem of using aspects will be to make easy to predict the effects of the aspects (that you also address in the last paragraph of Discussion).
From my experience with AOP, we often get unintended results from aspects,
and good tool support to allow us to visualize the results of weaving the aspects is essential. Do you think that, given the dynamic nature of Javascript, it will be possible to provide good tool support for AOP?
Moreover, we would be creating aspects that affect external libraries we usually do not know in detail.
I suspect this point is critical to allow a wide acceptance of aspects for
Javascript/security, and to make them more than just a proof of concept, therefore some additional discussion on the topic would be useful.

Despite my previous observations, I agree that it is worth to study the use of
AOP as a general approach to secure Javascript, and I think this paper provides
a good analysis of this topic.

---------- REVIEW 5 ----------
PAPER: 3
TITLE: Aspectizing JavaScript Security
AUTHORS: Florent Marchand de Kerchove, Jacques Noyé and Mario Südholt

---------- REVIEW ----------
The authors present a study about the implementation of security policies with aspect-oriented programming in JavaScript applications. Hereby, they discuss the topics "fine-grained access control", "capability-secure scripting" and "dynamic information flow". The paper concludes that AOP was well-suited for all three topics.

The paper is well-structured. Each of the three topics in discussed in a separate section, and in each section, the general introduction is followed by AOP-specifics.

Unfortunately, the paper gets often imprecise and sloppy. To name a few examples: "applications" do not have "cousins". AOP does not get "implanted". What are the "complex semantic rules regarding 'this'" exactly, and how are they relevant to AOP? Important examples, such as "the same-origin policy" and "Content Security Policy" lack proper references in the bibliography.

The kind of "fine-grained access control" that the authors envision remains unclear. In the beginning of the section, the authors state that a single policy may not be enough to adequately secure an application. Unfortunately, this point is not addressed in the following presentation. Instead, two approaches that target the browser and the application itself are provided, but the authors should really be more precise how AOP can help applications to provide better access control. In no case, the paper discusses what kind of thread models are considered. A concrete example would probably help a lot.

The section about capability-based systems is very confusing. The very

terse introduction (without references) is complemented with a weak example. A reader without specific knowledge about the design principles of capability-based systems is unlikely to follow the rest of the section. What follows is a very general discussion about implementing capabilities in Java, with the conclusion that JavaScript is a very problematic language for this goal. The authors suggest to overcome the limitations (which are only vaguely specified) with "aspects". The discussion remains very abstract, again, a concrete example would help a lot.

Section 4 starts with a discussion in what way static analysis is superior to dynamic analysis for ensuring information flow properties. The section continues with presenting existing approaches that implement both variants. The authors claim that aspects would be "well-suited" to implement dynamic analyses. Unfortunately, they this claim is not backed up at all: Section 4.2, paragraph 1 discusses the premises that must hold for the claim, but the claim itself remains unclear. Paragraphs 2 and 3 of this section also contain unbacked personal opinions of the authors that does not fit an objective survey.

The paper is something between a survey of AOP techniques to implement security policies, and a position paper. Unfortunately, the position of the authors is unconvincing, despite its repetition in various ways. The survey would be interesting to read if there were some noteworthy finding.

The abstract as well as the conclusion section of the paper are not really helpful to summarize the (sparse) findings and conclusions drawn from the paper.