



nodigt U graag uit op de openbare verdediging van het proefschrift van

## Maarten Vandercammen

ter behaling van de graad van Doctor in de Wetenschappen

Titel van het proefschrift:

**Inter-Process Concolic Testing of Full-stack JavaScript Web Applications**

Promotor: **Prof. dr. Coen De Roover**

De verdediging heeft plaats op

**Vrijdag 13 oktober 2023 om 16u in  
aula D.0.07**

De verdediging kan via een livestream gevolgd worden. Gelieve een e-mail te sturen naar [Maarten.Vandercammen@vub.be](mailto:Maarten.Vandercammen@vub.be) voor meer informatie.

### Samenstelling van de jury

Prof. dr. Viviane Jonckers (VUB, voorzitter)  
Prof. dr. Joeri De Koster (VUB, secretaris)  
Prof. dr. Beat Signer (VUB)  
Prof. dr. Kris Steenhaut (VUB)  
Prof. dr. Cristian Cadar (Imperial College  
Londen, UK)  
Prof. dr. Xavier Devroey (Université de Namur)

### Curriculum vitae

Maarten Vandercammen behaalde zijn Bachelordiploma in de Computerwetenschappen in 2013 en zijn Masterdiploma in de Computerwetenschappen in 2015 aan de Vrije Universiteit Brussel, beide summa cum laude. Nadien begon hij een doctoraat onder de begeleiding van Prof. Dr. Coen De Roover. Hij draagt momenteel bij aan het CyberSecurity ICON project "APAX". Zijn onderzoeksinteresses omvatten het automatisch testen van webapplicaties.

Maarten is coauteur van twee internationale journal publicaties (als eerste auteur), een internationale conferentie publicatie (als tweede auteur), en zes internationale workshop publicaties (waarvan vijf als eerste auteur). Zijn werk werd gepresenteerd op internationale conferenties en workshops. Hij begeleidde acht Bachelor theses en vier Master theses.

### Abstract van het doctoraatsonderzoek

Webapplicaties worden steeds populairder. Voorbeelden van dit soort applicaties zijn onder andere collaboratieve tekstverwerkingsprogramma's en tekenapplicaties. Wij definiëren full-stack JavaScript webapplicaties als webapplicaties waarvan zowel de client als de server geïmplementeerd werden in JavaScript.

Er bestaan verschillende geautomatiseerde testtechnieken om de correctheid van sequentiële, niet-gedistribueerde applicaties te verifiëren. Prominent hierin is concolic testing, welke systematisch alle executiepaden doorheen een programma test door symbolische constraints te verzamelen over diens programma-inputs.

We verplaatsen concolic testing naar het domein van full-stack JavaScript webapplicaties. Hierbij duiken verschillende uitdagingen op die uniek zijn aan dit soort systemen. Aangezien bijvoorbeeld zowel het client- als het server-proces event-driven zijn, moeten concolic testers van deze applicaties ingewikkelde sequenties van events aanmaken om bepaalde delen van het programma te testen. Bovendien kan, omwille van de sterk geconnecteerde aard van deze processen, de uitvoering van het ene proces de uitvoering van een ander op onverwachte manieren beïnvloeden.

We stellen een aanpak van concolic testing van deze applicaties voor die zich richt op deze uitdagingen. Deze aanpak maakt gebruik van inter-proces testing, waarbij alle instanties van client- en server-processen getest worden en hun onderlinge communicatie tegelijkertijd geobserveerd wordt. Inter-proces testing behoudt de informatiestroming tussen processen, waardoor precisie wordt verbeterd en vals-positieve fouten worden vermeden. Inter-proces testers onderscheiden zich van intra-proces testers, wiens executiepaden niet de grenzen van een proces overschrijden. We implementeren inter-proces testing in een nieuwe concolic tester genaamd StackFul.

StackFul houdt ook rekening met de event-driven aard van full-stack JavaScript webapplicaties. Event-driven code geeft aanleiding tot het probleem van state explosion, waarbij een exponentieel aantal staten wordt aangemaakt tijdens het testen van de applicatie. Om dit probleem op te lossen, introduceren we een nieuwe vorm van state merging voor concolic testing van event-driven applicaties, waarbij het aantal staten beperkt wordt door gelijkaardige staten samen te voegen.

We evalueren StackFul op acht full-stack JavaScript webapplicaties. We meten in hoeverre StackFul in staat is om i) hun executiepaden te verkennen, ii) fouten te vinden in de server van deze applicaties, en iii) hoog-prioritaire fouten te onderscheiden van laag-prioritaire serverfouten. Daarnaast evalueren we de impact van state merging. We tonen aan dat state merging bijna altijd minder testiteraties vereist om een groter deel van de applicatie te verkennen.